

Content-based Clustering in Social Corpora

A New Method for Knowledge Identification based on Text Mining and Cluster Analysis

vorgelegt von

Dipl.-Ing.
Annette Bobrik
aus Berlin

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr. Ing. -

genehmigte Dissertation

Promotionsausschuß:

Vorsitzender: Prof. Dr. O. Kao
Technische Universität Berlin

Gutachter: Prof. Dr. H. Krallmann
Technische Universität Berlin

Gutachter: Prof. Dr. J. Mendling
Wirtschaftsuniversität Wien

Tag der wissenschaftlichen Aussprache: 9. November 2012

Berlin 2013
D 83

Abstract

Understanding the workforce and skill-set of an enterprise can be seen as the key to understand the capabilities of an organization. In today's large organizations it has become increasingly difficult to find people that have specific skills or knowledge or to explore the overall picture of an organization's portfolio of knowledge demand and supply. Bringing together human actors with similar interests, skills or knowledge is a major challenge in community-based knowledge management. In business processes communicational activities are an important means of collaboration which increases with the amount of knowledge generation and reuse. Electronic communication media provide a wide range of possibilities for decentralized interaction and collaboration that blend contents and communication. The collection of all electronic traces of interrelated communication relationships forms a network structure which can be defined as a social corpus. One of the primary methods for studying the resulting electronic communication, collaboration and interaction and their inherent communities is social network analysis (SNA).

There are estimates that about 85% of business-relevant information originates in unstructured form ranging from short text messages to exchanging or even simultaneously editing large documents. Although the content information can often be directly linked with the relational data forming the social network the domain of content analysis with means of text mining and information retrieval has not yet been sufficiently accommodated in the methodological framework of SNA. Text data that can be retrieved from social media are often analyzed separately from the structure of the network. However, neglecting the content of text data during network analysis can limit the understanding of the network structure, its evolution over time and the multiple roles an actor can obtain in this context. Therefore, conventional SNA will not be able to detect groups of similar knowledge that evolve within a multi-contextual network where each node can obtain multiple roles.

To meet the described shortcomings of SNA research, this study concentrates on a new approach by integrating both levels of analysis: content analysis and network analysis. First, the Social Network Intelligence (SNI) framework is introduced which extends conventional SNA providing novel insights into network structure, content, behavior and context. The new method of content-based clustering for knowledge identification in social corpora is based on the SNI framework. It is designed as a static content analysis with the main focus being on group level analysis including also elements of structural analysis and network dynamics. Therefore, a general overview of social network analysis, text mining and cluster analysis is given as theoretical foundations including definitions, basic concepts, as well as selected research to provide an understanding of the context of this work. Afterwards, this work provides a detailed research guideline and the description of algorithms and metrics for the new method of content-based clustering for knowledge identification in social corpora to structure and guide the analysis. Based on its requirements a prototype is implemented to allow an IT-supported analysis integrating relevant methods of text mining, cluster analysis and social network analysis. Within a case study the method is applied to a corporate e-mail data set using the prototype.

Zusammenfassung

Das Verständnis der Fähigkeiten und Erfahrungen der Mitarbeiter eines Unternehmens kann als Schlüssel im Geschäftsprozessmanagement angesehen werden. In den heutigen großen, global-agierenden Unternehmen ist es zunehmend schwieriger geworden, Mitarbeiter mit bestimmten Fähigkeiten oder Kenntnissen zu finden oder das Gesamtbild des Unternehmens-Portfolios für Wissensangebot und -nachfrage zu erforschen und zu verstehen. Das Zusammenführen von Akteuren mit ähnlichen Interessen, Fähigkeiten oder Wissen ist eine große Herausforderung im Community-orientierten Wissensmanagement. In modernen Geschäftsprozessen sind kommunikative Aktivitäten ein wichtiges Mittel der Zusammenarbeit, deren Menge sich mit zunehmender Wissensgenerierung und -wiederverwendung erhöht.

Elektronische Kommunikationsmedien bieten eine breite Palette von Möglichkeiten für die dezentrale Interaktion und Zusammenarbeit, die Inhalte und Kommunikation miteinander verbinden. Die Menge aller elektronischen Spuren von wechselseitigen Kommunikationsbeziehungen bildet eine Netzstruktur, die als sozialer Korpus definiert werden kann. Eine der wichtigsten Methoden für die Untersuchung dieser elektronischen Kommunikation, Zusammenarbeit und Interaktion und ihrer inhärenten Communities ist die Soziale Netzwerkanalyse (SNA).

Es gibt Schätzungen, dass etwa 85% der geschäftsrelevanten Informationen in unstrukturierter Form vorliegt. Obwohl diese inhaltlichen Informationen häufig direkt mit den relationalen Daten verknüpft werden können, die das soziale Netzwerk bilden, ist das Gebiet der Inhaltsanalyse mit Hilfe von Text Mining und Information Retrieval noch nicht ausreichend in den methodischen Rahmen der SNA eingebunden. Textdaten, die aus sozialen Medien abgerufen werden können, werden oft getrennt von der Struktur des Netzwerkes analysiert. Diese Vernachlässigung von Kommunikationsinhalten während der Netzwerkanalyse kann das Verständnis der Netzwerkstruktur, ihrer zeitlichen Entwicklung und den vielfältigen Rollen der Akteure im Netzwerk begrenzen. Daher ist es her den Methoden der herkömmlichen SNA nicht möglich, Gruppen mit ähnlichem Wissen und die unterschiedlichen Rollen der Beteiligten zu entdecken, die sich in einem multi-kontextuellen Netzwerk entwickeln.

Um den beschriebenen Defiziten der SNA Forschung zu begegnen, wurde im Rahmen dieser Arbeit ein neuer Ansatz entwickelt, der die Integration von Inhaltsanalyse und Netzwerkanalyse ermöglicht. Zunächst wird das Social Network Intelligence (SNI) Framework eingeführt, das die herkömmliche SNA um Inhalts-, Verhaltens- und Kontextkomponenten erweitert. Die neue Methode der inhaltsbasierten Clusteranalyse zur Wissensidentifikation in sozialen Korpora stellt ein methodisches Werkzeug innerhalb des SNI Frameworks dar. Die Methode ist als statische Inhaltsanalyse konzipiert mit dem Hauptaugenmerk auf der Analyse von Gruppen, prominenter Akteure und deren Beziehungen einschließlich Elementen der Strukturanalyse und der dynamischen Analyse. Zum Verständnis dieser Arbeit wird daher als theoretische Grundlage eine allgemeine Einführung in die Analyse sozialer Netzwerke, das Text Mining und die Clusteranalyse gegeben. Dies

beinhaltet neben Definitionen, Grundbegriffen und Methoden auch ausgewählte Forschungsergebnisse. Anschließend werden die entwickelten Algorithmen und Metriken der inhaltsbasierten Clusteranalyse zur Wissensidentifikation in sozialen Korpora detailliert beschrieben sowie eine Richtlinie für ein strukturiertes Vorgehen vorgestellt. Basierend auf diesen Anforderungen, wird ein Prototyp vorgestellt, der die einschlägigen Methoden des Text-Minings, der Clusteranalyse und der Sozialen Netzwerkanalyse beinhaltet, um eine IT-gestützte Analyse zu ermöglichen. Innerhalb einer Fallstudie wird das Verfahren auf einem Firmen-E-Mail-Datensatz mit Hilfe des Prototyps angewendet.

Style Guide

If neutral or plural pronouns are not appropriate in a specific context man and he are used as generic masculine in this study, i.e. the words include women as well as men.

Table of Contents

- Style Guide v
- Figures xi
- Tables xix
- Abbreviations xxv
- 1 Introduction 1
 - 1.1 Motivation 1
 - 1.2 Problem Statement of the Study and Research Questions 6
 - 1.3 Research Procedure and Methodology 7
 - 1.4 Guide to Content 10
- 2 Social Network Intelligence Framework 13
 - 2.1 Overview of Social Network Analysis 13
 - 2.1.1 Definitions 13
 - 2.1.2 Basic Concepts 14
 - 2.1.3 Overview SNA Research 18
 - 2.1.4 Metrics 28
 - 2.2 SNI Framework 38
 - 2.2.1 SNI Data Model 39
 - 2.2.2 SNI Dimensions 47
 - 2.2.3 SNI Process 51
 - 2.3 IT Support 68
 - 2.3.1 Overview of SNA Software 68
 - 2.3.2 Commetrix 70
- 3 Text Mining 73
 - 3.1 Introduction to Text Mining 73
 - 3.1.1 Definition 73
 - 3.1.2 Data Source 74
 - 3.1.3 Data Collection 75
 - 3.1.4 Knowledge Discovery Process for Text Data Mining 76
 - 3.1.5 Application Areas 77
 - 3.1.6 Challenges 79

Table of Contents

3.2	Linguistic Preprocessing.....	80
3.2.1	Word Sense Disambiguation.....	80
3.2.2	Part-of-Speech Tagging.....	84
3.2.3	Parsing.....	85
3.2.4	Chunking.....	89
3.3	Text Transformation (Feature Generation).....	90
3.3.1	Tokenization.....	91
3.3.2	Vector Space Model.....	94
3.4	Feature Selection.....	96
3.4.1	Motivation: Performance Measures.....	97
3.4.2	Eliminating Features.....	99
3.4.3	Weighting Features.....	100
3.4.4	Normalizing Features.....	103
3.4.5	Evaluation of Feature Selection Methods.....	108
3.5	Applications of Text Mining in Social Corpora.....	110
4	Cluster Analysis.....	115
4.1	Introduction to Cluster Analysis.....	115
4.1.1	Definition.....	115
4.1.2	Categorization of Clustering Methods.....	116
4.1.3	Clustering Methodology.....	118
4.1.4	Application Areas.....	120
4.1.5	Challenges.....	121
4.2	Preparation of the Data Set and Initial Screening.....	121
4.2.1	Representation.....	122
4.2.2	Data Measurement.....	125
4.2.3	Dealing with Noise.....	129
4.3	Proximity Measures.....	131
4.3.1	Proximity Measures as Distance Metrics.....	132
4.3.2	Correlation Coefficients.....	132
4.3.3	Distance Measures.....	135
4.3.4	Association Coefficients.....	137
4.3.5	Ties in the data.....	138
4.4	Cluster Strategy: Hierarchical Clustering.....	139

4.4.1	Agglomerative Hierarchical Clustering.....	139
4.4.2	Divisive Hierarchical Clustering.....	147
4.4.3	Mathematical Properties of Hierarchical Clustering Methods.....	148
4.5	Cluster Strategy: Partitioning Clustering.....	149
4.5.1	General Objectives of Partitioning Clustering Procedures.....	150
4.5.2	Hard Clustering.....	151
4.5.3	Fuzzy Clustering.....	158
4.6	Clustering Tendency and Cluster Validation.....	160
4.6.1	Introduction to Cluster Validation.....	160
4.6.2	Cluster Validation with Hypothesis Testing.....	161
4.6.3	Validity of Hierarchies.....	162
4.6.4	Validity of Partitional Structures.....	164
4.6.5	Validity of Internal Clusters.....	171
4.6.6	Clustering Tendency.....	174
4.7	Applications of Cluster Analysis in Social Corpora.....	175
4.7.1	Motivation: Detecting Community Structures.....	176
4.7.2	Graph-based Community Detection by Girvan and Newman.....	177
4.7.3	Applications.....	180
5	Content-based Clustering for Knowledge Identification.....	185
5.1	Method.....	185
5.1.1	Data Preparation.....	187
5.1.2	Initial Screening.....	190
5.1.3	Network Analysis.....	192
5.1.4	Cluster Analysis.....	195
5.1.5	Categorization & Comparison.....	200
5.2	Prototype.....	209
5.2.1	Architecture.....	209
5.2.2	Data Access and Exchange.....	212
5.2.3	Data View.....	215
5.2.4	Initial Screening.....	216
5.2.5	Temperature View.....	218
5.2.6	Cluster Strategy.....	222
5.2.7	Clustering View.....	223

Table of Contents

5.3	Case Study “ Corporate E-Mail Exchange”	238
5.3.1	Data Set	238
5.3.2	Data Preparation	239
5.3.3	Initial Screening	242
5.3.4	Network Analysis	242
5.3.5	Cluster Analysis	248
5.3.6	Summary	270
6	Conclusions.....	273
6.1	Summary of the Results.....	273
6.2	Future Research	277
	Appendix	281
A	Text Mining	283
B	Prototype	285
B.1	Data Format	285
B.2	Cluster Analysis.....	286
B.2.1	Content-based Clustering on Nodes	286
B.2.2	Content-based Clustering on Linkevents	289
C	Case Study	291
C.1	Data Preparation	291
C.2	Text Mining	293
C.3	Network Analysis	294
C.3.1	SNA Metrics on Node Level.....	294
C.3.2	Node Role Categorization	295
C.3.3	Temperature View	297
C.4	Cluster Analysis.....	299
C.4.1	Structural Clustering	299
C.4.2	Content-based Clustering on Nodes	302
C.4.3	Content-based Clustering on Linkevents	306
C.4.4	Comparison of Clustering Results.....	330
	Literature	343

Figures

Figure 1-1: Information systems research framework. Source: Hevner et al. (2004: 80)	8
Figure 1-2: Content-based clustering for knowledge identification in social corpora – guide to content	10
Figure 2-1: The network as a directed graph. Node color represents gender: white: female; gray: male. Source: Hanneman und Riddle (2005).....	15
Figure 2-2: Different types of network structures: (a) star; (b) line; (c) circle; (d) combination. Based on: Wasserman and Faust (1994: 171)	16
Figure 2-3: Participation of the southern women in events represented as a bipartite graph. Source: Borgatti (2009).....	19
Figure 2-4: Random networks versus scale-free networks. Left: random network resembling U.S. highway system; right: scale-free network resembling the U.S. airline system. Source: Barabási and Bonabeau (2003: 53).....	23
Figure 2-5: Overview of social theories. Source: Monge and Contractor (2003)	24
Figure 2-6: Social capital, in metaphor and network structure. Source: Burt (2002).....	24
Figure 2-7: Measuring social capital: brokerage roles. Node colors represent different organizational departments. Based on: Gould and Fernandez (1989).....	25
Figure 2-8: Network with core/periphery structure. Source: Krebs and Holley (2007).....	29
Figure 2-9: Routes through the network: (a) walk; (b) trail; (c) path; (d) geodesic	30
Figure 2-10: Concepts of connectivity: (a) cutpoint; (b) bridge.....	31
Figure 2-11: Comparison of centrality metrics. Graph showing nodes with highest degree (D), closeness (C), betweenness (B), and eigenvector (E) centrality. Source: Borgatti (2002)	34
Figure 2-12: Comparison of centrality metrics: Betweenness centrality versus brokering activity	35
Figure 2-13: Structural holes and related metrics.....	37
Figure 2-14: The sixteen possible triads in a directed graph. Source: Wasserman and Faust (1994: 244)	38
Figure 2-15: The SNI data model. Based on: Trier (2008)	40
Figure 2-16: Graphical representation of the SNI data model as a communiagraph. Enron data set, subsample January – December 2000. Node label: node index; node color: degree (yellow: large values; blue small values); node size: number of linkevents; link color: reciprocity (dark gray: reciprocal; light gray: non-reciprocal); link strength: number of linkevents	41
Figure 2-17: SNI data model. Sender-recipient communication. Left: interaction events in an e-mail data set. Right: visualization as communiagraph, link labels correspond to number of events	42
Figure 2-18: SNI data model. Thread-based communication in a newsgroup. Left: action events and reaction events in a newsgroup. Right: visualization as communiagraph, link labels correspond to number of events.....	42

Figure 2-19: SNI data model. Citation network with co-authorship. Left: action events (single author), interaction events (co-authorship), and reaction events (citation; dotted arrow). Right: visualization as communiagraph, link labels correspond to number of events	43
Figure 2-20: Dynamic network analysis: (a) cumulative approach; (b) sliding time window approach	45
Figure 2-21: The impact of time window size. Interaction data from McFarland's classroom observations viewed at various levels of time aggregation from 35 minutes (one entire class period) to 1 minute (two-three turns of interaction). Source: Bender-deMoll and McFarland (2006: 5).....	46
Figure 2-22: Event-driven dynamic network analysis with Commetrix: (a) cumulative approach; (b) sliding time window approach.....	47
Figure 2-23: SNI dimensions: level of investigation; level of detail; temporal dimension.....	48
Figure 2-24: Different levels of detail in a network: ego, group and network analysis	49
Figure 2-25: Different levels of investigation: structure versus content analysis	50
Figure 2-26: SNI Process. Extending conventional SNA. Based on: Trier and Bobrik (2007c).....	52
Figure 2-27: KM entity model. Source: Trier and Müller (2004: 243)	58
Figure 2-28: Procedural guide for capturing knowledge-intensive business processes and its relation to the KM entity model. Source: Trier and Müller (2004: 245).....	59
Figure 2-29: Commetrix: IT support from original data to network insights. Source: IKM Research group (2011).....	71
Figure 2-30: CMXAnalyzer 2.0. Screenshot. Node label: index; node size: linkevents sent; node color: direct contacts; link length: number of contacts; link color: number of contacts	72
Figure 3-1: Knowledge discovery process. Source: Fayyad et al. (1996).....	76
Figure 3-2: Text mining process	77
Figure 3-3: Example of Context-free Grammar. Source: Salton and McGill (1984: 90).....	86
Figure 3-4: Top-down versus bottom-up parsing. Based on: Carstensen et al. (1986: 305-306).....	87
Figure 3-5: Example of syntactic ambiguity: coordination ambiguity. Source: Bird et al. (Bird et al. 2007: 165).....	88
Figure 3-6: Chunking versus parsing. (a) tree representation of chunking; (b) tree representation of parsing. Source: Bird et al. (2007: 117).....	90
Figure 3-7: Example of bag-of-words representation after tokenization.....	91
Figure 3-8: Replacement filter. Example on Enron e-mail corpus.....	92
Figure 3-9: Vector space representation and illustration of the cosine similarity $\text{sim}(d_1, d_2) = \cos\theta$	95
Figure 3-10: Text transformation for feature generation. Overview	95
Figure 3-11: Resolving power of significant medium-frequency words. Based on: Salton and McGill (1984: 62) and Luhn (1958).....	96
Figure 3-12: Feature selection methods. Overview.....	96
Figure 3-13: Precision and recall. Example	97
Figure 3-14: Term characterization in frequency spectrum. Based on: Salton and McGill (1984: 87)	98

Figure 3-15: A stop word list of 25 semantically nonselective words that are common in Reuters-RCV1. Source: Manning et al. (2008: 25) 99

Figure 3-16: Stop word removal on Enron data set. Eliminated stop words are indicated as red dots . 99

Figure 3-17: High-frequency and low-frequency filter on Enron data set. Eliminated stop words are indicated as red dots. Term frequency (tf) and document frequency (df) thresholds are indicated as dashed lines. 100

Figure 3-18: Semantic graph built from the news article "Apple to Make iTunes More Accessible For the Blind". Source: Grineva et al. (2009: 665)..... 111

Figure 4-1: Clusters with internal cohesion and/or external isolation. Source: Gordon (1980)..... 116

Figure 4-2: Tree of classification types. Based on: Lance and Williams (1967) and Jain and Dubes (1988: 56)..... 116

Figure 4-3: Clustering methodology. Source: Jain and Dubes (1988: 135) 118

Figure 4-4: Complete linkage agglomerative cluster analysis on investment strategies of 59 Barclay Managed Futures (CTA) fund managers based on their investment strategies January 2004 - December 2007. Cluster labels assigned after clustering. Source: Noma and Shtrapeina (2010) 120

Figure 4-5: Taxonomy of data representation: formats, types, and scales of data. Source: Jain and Dubes (1988: 13) 122

Figure 4-6: Graphical representation of bivariate data as scatterplot or histogram: Birth and death rates for 69 countries. Based on: Everitt et al. (2001: 14)..... 123

Figure 4-7: Illustration of matrix shading: (a) proximity matrix based on Iris data set by Merz et al. (1997) using Euclidean distance; (b) randomly ordered shaded proximity matrix; (c) reordered shaded distance matrix using a seriation algorithm. Source: Wang et al. (2002)..... 124

Figure 4-8: Cluster representatives: medoid versus centroid 125

Figure 4-9: Illustration of hierarchical agglomerative clustering methods: (a) single linkage; (b) complete linkage; (c) average linkage; (d) centroid linkage..... 139

Figure 4-10: Comparison of different agglomerative clustering methods. (a) six three-dimensional pattern vectors; (b) proximity matrix using squared Euclidean distance; (c) dendrograms. Source: Jain and Dubes (1988: 82) 145

Figure 4-11: Illustration of K-means clustering algorithm: 1) initial randomized centroids and some data objects; 2) objects are associated with the nearest centroid; 3) centroids are moved to the center of their respective cluster; 4) step 2 & 3 are repeated until a suitable level of convergence has been reached; 5) final result 156

Figure 4-12: Examples of cluster structures: (a) well-separated clusters; (b) overlapping clusters. Source/Following: Jain and Dubes (1988: 130)..... 158

Figure 4-13: Different types of dendrograms. (a) chaining; (b) binary; (c) arbitrary. Source: Jain and Dubes (1988: 171) 163

Figure 4-14: Illustration of the gap statistic: (a) original data set; (b) within sum of squares function W_k ; (c) observed and expected $\log(W_k)$; (d) gap curve. Source: Tibshirani et al. (2001: 416) ... 170

Figure 4-15: Cluster profiles for a priori cluster. Source: Jain and Dubes (1988: 193) 172

Figure 4-16: A schematic representation of a network with community structure. Communities denoted by dashed circles. Source: Newman and Girvan (2004d)	176
Figure 4-17: Community identification according to the GN algorithm: (a) a simple network with two communities; (b) binary tree generated by the GN algorithm. Source: Guimerá et al. (2003)...	179
Figure 4-18: Comparison of community structures identified by applying the GN algorithm to the URV e-mail network represented as a binary tree: (a) complex: URV e-mail network, clustering coefficient $c=0.254$ and average shortest path length $d=3.606$; (b) trivial: randomly generated network, $c=0.028$ and average shortest path $d=3.317$. Binary trees without leaf nodes. Coloring due to Horton-Strahler index. Guimerá et al. (2003: 2)	181
Figure 4-19: Contextualizing Tags in Collaborative Tagging System: Subgraph of internal bookmarking service, Labbies, used by a group of researchers at HPLabs (136 users, 95,155 bookmarks, 8,012 tags, 61,453 co-occurrences); root="mit"; cluster distance <2 ; minimum tag use >2 ; NCO >0.0 . Source: Simpson (2008: 223)	181
Figure 4-20: Multiplex social network graph generation. Link types are defined by the social network transaction types. Weights are provided by the content extraction algorithm as a confidence measure of the observation. For each link type a subgraph can be retrieved and analyzed. Source: Weinstein et al. (2009)	182
Figure 4-21: Community dynamics with changing members. History view. The revealed communities are displayed in different colors. Data source: Online international student community in the University of Magdeburg (1,000 members; 250,000 guestbook entries over a period of 18 months). Source: Falkowski et al. (2006)	184
Figure 5-1: Content-based clustering for knowledge identification. Categorization using the SNI dimensions	185
Figure 5-2: Content-based clustering for knowledge identification. Research guideline.....	186
Figure 5-3: Network and content extraction from collaborative content creation processes	187
Figure 5-4: Components of the ContentMiner text mining software	188
Figure 5-5: ContentMiner. Example of the manual data inspection by the ModelViewer component	188
Figure 5-6: Representation of nodes in the vector space. Example: (a) extracting topics from collaborative content objects; (b) assigning content objects to nodes and retrieving topic vectors; (c) representing nodes in a three-dimensional vector space	189
Figure 5-7: Content profile. Example.....	194
Figure 5-8: Visual representation of content profiles and temperature graphview: (a) overview of content profiles; (b) temperature graphviews for three selected nodes	195
Figure 5-9: Comparison of clustering strategies: (a) structural clustering; (b) content-based clustering on nodes; (c) content-based clustering on linkevents	196
Figure 5-10: Node roles in different clustering solutions (focal node is marked with a red ring): (a) structural clustering; (b) content-based clustering on nodes; (c) content-based clustering on linkevents.....	200
Figure 5-11: Architecture of the prototype. Conceptual design	211
Figure 5-12: Architecture of the prototype. Class diagram	212
Figure 5-13: Prototype data format. CLAN XSD schema. Network elements partly expanded.....	213

Figure 5-14: Initial data view. Node view. Selected node 1252.....	215
Figure 5-15: Temperature overview. Content profiles of active nodes	218
Figure 5-16: Temperature graphview. Structural distribution of content similarities of a selected node. Comparison of different node weighting schemes: (a) node weighting scheme: sum; (b) node weighting scheme: average	221
Figure 5-17: Structural clustering view	224
Figure 5-18: Content-based clustering on nodes. Graphview of the entire network and representative view. Hierarchical clustering	226
Figure 5-19: Content-based clustering on nodes. Graphview perspectives on selected cluster: (a) cluster graphview; (b) temperature view (content profiles); (c) temperature view (graphview of selected node)	227
Figure 5-20: Content-based clustering on nodes. Clustering overview. Dendrogram of hierarchical clustering solutions	227
Figure 5-21: Content-based clustering on nodes. Level validation. Hierarchy overview with plots of fusion coefficient and validation index versus level.....	228
Figure 5-22: Content-based clustering on nodes. Charts for validation of hierarchical clustering solutions. Plot of average content dissimilarity	228
Figure 5-23: Content-based clustering on nodes. Partitioning clustering with random initial partition (level 0, left) and optimal solution (level 2, right): (a) clustering overview; (b) network graphview; (c) representative graphview	229
Figure 5-24: Content-based clustering on nodes. Network activity view: (a) inter-cluster activity; (b) intra-cluster activity	230
Figure 5-25: Content-based clustering on nodes. Cluster details	231
Figure 5-26: Content-based clustering on nodes. Node details	232
Figure 5-27: Content-based clustering on linkevents. Graphview: (a) network, (b) cluster and (c) representative	234
Figure 5-28: Content-based clustering on linkevents. Different cluster graphviews for selected node 1194	235
Figure 5-29: Cluster comparison view	237
Figure 5-30: Case study. Results from automated feature selection. Axes: x-axis: log(term frequency); y-axis: log(document frequency). Topics plotted as blue dots; topics removed marked as red dot; topics added marked as green dot; centroid of all topics marked by a red circle	241
Figure 5-31: Case study. Network activity plot.....	243
Figure 5-32: Case study. Graphview. Structural key players highlighted with red ring. Node color indicates type of interaction: green=only linkevents sent; blue=only linkevents received; orange=linkevents sent and received	244
Figure 5-33: Case study. Mapping content to network structure. Parts of network with similar graphviews. Exemplary temperature graphviews	247
Figure 5-34: Case study. Structural clustering. Optimal solution: (a) graphview; (b) representative view (of main component). Structural key players (top 9) highlighted with red ring	249

Figures

Figure 5-35: Case study. Structural clustering. Network activity view: (a) intra-cluster activity; (b) inter-cluster activity	250
Figure 5-36: Case study. Content-based clustering on nodes. Hierarchical clustering overview (selected level marked with blue vertical line): (a) dendrogram (clusters marked with green horizontal lines); (b) plot of validation index versus level (excerpt); (c) plot of content dissimilarity versus level (excerpt)	252
Figure 5-37: Case study. Content-based clustering on nodes. Optimal solution: (a) graphview; (b) representative view (main component). Structural key players (top 9) highlighted with red ring	253
Figure 5-38: Case study. Content-based clustering on nodes. Network activity view: (a) intra-cluster activity; (b) inter-cluster activity	254
Figure 5-39: Case study. Content-based clustering on nodes. Level overview. Graphviews of selected clusters. Node color indicates type of interaction: green=only linkevents sent; blue=only linkevents received; orange=linkevents sent and received. Structural key players highlighted with thick red ring (top 9) or thin red ring and lighter node color (top 21)	256
Figure 5-40: Case study. Content-based clustering on linkevents. Hierarchical clustering. Plot of validation index (excerpt); selected level marked with blue vertical line.....	258
Figure 5-41: Case study. Content-based clustering on linkevents. Example of cluster size. Largest clusters by number of (a) linkevents (cluster 19) and (b) nodes (cluster 63).....	259
Figure 5-42: Case study. Content-based clustering on linkevents. Examples of local clusters: (a) cluster 46; (b) cluster 84	260
Figure 5-43: Case study. Content-based clustering on linkevents. Examples of distributed clusters: (a) cluster 3; (b) cluster 44	261
Figure 5-44: Case study. Content-based clustering on linkevents. Examples of different types of knowledge: (a) cluster 57; (b) cluster 59; (c) cluster 73; (d) cluster 74.....	261
Figure 5-45: Case study. Content-based clustering on linkevents. Node role categorization. Cluster 6: (a) graphview; (b) knowledge domain categorization; (c) node roles (slice size indicates number of linkevents)	264
Figure 5-46: Case study. Content-based clustering on linkevents. Portfolio of knowledge domains. Node color indicates activity type (low, medium, high); node size indicates activity value (number of linkevents).....	265
Figure 5-47: Case study. Content-based clustering on linkevents. Portfolio of knowledge profiles. Node color indicates activity type (low, medium, high); node size indicates activity value (number of linkevents).....	266
Figure 6-1: Gartner’s hype cycle for emerging technologies (2011). Source: Gartner (2011)	277
Figure B-1: Prototype data format. XSD schema. Element “xclusterstrategy” partly expanded	285
Figure C-1: Case study. Temperature view. Overview of content profiles. Similarity values ranging from 0% (white) to 100% (red). Structural key players marked with red solid line (top 9) or dashed line (top 21)	297
Figure C-2: Case study. Temperature view. Comparison of content profiles. Nodes ordered by profile similarity. Groups of similar profiles separated by red dashed lines. Similarity values ranging	

from 0% (white) to 100% (red). Structural key players marked with red solid line (top 9) or
dashed line (top 21) 297

Tables

Table 2-1: Different types of relations. Overview.....	15
Table 2-2: The network as an asymmetric adjacency matrix. Based on: Hanneman und Riddle (2005)	17
Table 2-3: Participation of the southern women in events. Source: Davis et al. (1941)	18
Table 2-4: Identified groups and core/periphery assignment. Overview of 11 studies. Level of core/periphery is indicated by brackets and decreases from left to right. Based on: Freeman (2003: 62/63)	20
Table 2-5: Examples of scale-free networks. Source: Barabási and Bonabeau (2003: 54).....	23
Table 2-6: Overview of centrality metrics.....	35
Table 2-7: SNI data model. Overview of event types and their application.....	44
Table 2-8: Survey to evaluate the particular needs of an organization. Source: Krebs (1996)	55
Table 2-9: Strategies and recommendations for actions to improve organizational networks. Based on: Anklam (2006).....	67
Table 2-10: Overview of selected programs for social network analysis: objective, data format (type, input format, missing values), functionality (visualization techniques, analysis methods) and support (availability of the program, manual, online help). Source: Huisman and van Duijn (2005).....	69
Table 2-11: Scores for selected SNA software from Table 2-10. Source: Huisman and van Duijn (2005).....	70
Table 3-1: An example of an automatically generated thesaurus employing Latent Semantic Indexing. Based on: Manning et al. (2008: 176).....	82
Table 3-2: Accuracy and applicability of the one-sense-per-discourse hypothesis. Source: Yarowsky (1995: 189).....	83
Table 3-3: Part-of-speech tagging. Example on Enron data set using the Brill tagger	84
Table 3-4: Example of syntactic categories. Source: Bird et al. (2007: 175).....	86
Table 3-5: SMART notation for tf.idf variant. Source. Manning et al. (2008: 118)	102
Table 3-6: A comparison of the stemming algorithms by Lovins, Porter and Paice on a sample text. Source: Manning et al. (2008: 32)	107
Table 3-7: Comparison of the effectiveness of advanced feature selection methods applied to European languages measured in change in mean average precision. Source: Hollink et al. (2004)	109
Table 3-8: The effect of feature selection on the number of terms, non-positional postings and tokens for Reuter-RCV1. Source: Manning et al. (2008: 80)	110
Table 4-1: Correlation coefficients. Overview	133
Table 4-2: Distance measures. Overview	136
Table 4-3: Association table for comparison of binary data	137
Table 4-4: Association coefficients. Overview	137

Tables

Table 4-5: Standard agglomerative hierarchical clustering methods. Overview. Following: Everitt (2001: 62).....	144
Table 4-6: General matrix updating algorithm. Based on: Johnson (1967) and Lance and Williams (1967).....	146
Table 4-7: Lance-Williams parameters for SAHN Matrix Updating Algorithms. Source: Jain and Dubes (1988: 80).	147
Table 4-8: Admissibility parameters for agglomerative hierarchical clustering methods. Source: Everitt (2001: 63).....	149
Table 4-9: General algorithm for iterative partitioning clustering by Anderberg. Source: Jain and Dubes (1988: 96)	151
Table 4-10: Contingency table for indicator functions. Source: Jain and Dubes (1988: 173)	164
Table 4-11: External indices of partitional adequacy. Based on: Jain and Dubes (1988: 174).....	165
Table 4-12: Relative indices of partitional adequacy	166
Table 4-13: Internal indices for internal cluster validation. Based on: Everitt et al. (2001: 91)	173
Table 4-14: Divisive clustering algorithm for finding community structures (GN algorithm). Source: Newman and Girvan (2004: 4)	178
Table 4-15: Edge weighting algorithm. Source: Newman and Girvan (2004: 5).....	178
Table 4-16: Edge weighting algorithm. Source: Newman and Girvan (2004: 5).....	179
Table 5-1: Pairwise content similarities using the cosine similarity	193
Table 5-2: Temperature values. Numeric expression of content profiles.....	194
Table 5-3: General algorithm of content-based clustering on nodes. Based on: Bobrik and Trier (2009)	197
Table 5-4: General algorithm of content-based clustering on linkevents.....	199
Table 5-5: Node role categorization on degree and betweenness centrality	201
Table 5-6: General algorithm of homogeneity categorization of knowledge domains	202
Table 5-7: General algorithm of frequency categorization of knowledge domains	203
Table 5-8: General algorithm of activity type categorization of knowledge domains	204
Table 5-9: General algorithm of homogeneity categorization of knowledge profiles.....	205
Table 5-10: General algorithm of diversification categorization of knowledge profiles	206
Table 5-11: General algorithm of activity type categorization of knowledge profiles.....	206
Table 5-12: Comparison of content similarity profiles. The effect of initial screening and cluster strategy parameters on node similarities.....	220
Table 5-13: Case study. Term reduction by different text mining methods.....	241
Table 5-14: Case study. SNA metrics. Network level.....	244
Table 5-15: Case study. Content-based clustering on nodes. Overview of knowledge domain & node role categorization	255
Table A-1: Comparison of different tag sets. CLAWS5, Brown and Penn Treebank tag set. Source: Manning and Schütze (1999: 141-142).....	283

Table A-2: Comparison of different tag sets (Ctd.). Source: Manning and Schütze (1999: 142)	284
Table B-1: Content-based clustering on nodes. Level overview	286
Table B-2: Content-based clustering on nodes. Level details	287
Table B-3: Content-based clustering on nodes. Node overview	288
Table B-4: Content-based clustering on linkevents. Level overview	289
Table C-1: List of Enron employees. Nodes 1125-1191	291
Table C-2: List of Enron employees. Nodes 1192-1275	292
Table C-3: Case study. Text mining methods. Configurations	293
Table C-4: Case study. Network analysis. Node level. SNA Metrics. Structural key players: top 9 marked bold, top 21 underlined	294
Table C-5: Case study. Network analysis. Node level. Node role categorization. Nodes 1125-1195	295
Table C-6: Case study. Network analysis. Node level. Node role categorization. Nodes 1196-1275	296
Table C-7: Case study. Groups of similar content profiles. Structural key players marked bold (top 9) or underlined (top 21)	298
Table C-8: Case study. Structural clustering. Overview of clustering solution (main component). Cluster representatives are marked with asterix, structural key players marked bold (top 9) or underlined (top 21)	299
Table C-9: Case study. Structural clustering. Node role categorization. Clusters 0-4	300
Table C-10: Case study. Structural clustering. Node role categorization. Clusters 5-7	301
Table C-11: Case study. Content-based clustering on nodes. Overview of clustering solution. Cluster representatives are marked with asterix, structural key players marked bold (top 9) or underlined (top 21).....	302
Table C-12: Case study. Content-based clustering on nodes. Node role categorization. Clusters 0-2	303
Table C-13: Case study. Content-based clustering on nodes. Node role categorization. Cluster 2 (ctd.)- 9	304
Table C-14: Case study. Content-based clustering on nodes. Knowledge domain categorization	305
Table C-15: Case study. Content-based clustering on linkevents. Level overview. Part 1. Clusters 0-39	306
Table C-16: Case study. Content-based clustering on linkevents. Level overview. Part 1. Clusters 40- 69	307
Table C-17: Case study. Content-based clustering on linkevents. Level overview. Part 1. Cluster 70-86	308
Table C-18: Case study. Content-based clustering on linkevents. Level overview. Part 2. Cluster 0-59	309
Table C-19: Case study. Content-based clustering on linkevents. Level overview. Part 2. Clusters 60- 86	310
Table C-20: Case study. Content-based clustering on linkevents. Level overview. Part 3. Clusters 0-59	311

Tables

Table C-21: Case study. Content-based clustering on linkevents. Level overview. Part 3. Clusters 60-86	312
Table C-22: Case study. Content-based clustering on linkevents. Node membership in clusters. Overview. Nodes 1125-1175	313
Table C-23: Case study. Content-based clustering on linkevents. Node membership in clusters. Overview. Nodes 1177-1229	314
Table C-24: Case study. Content-based clustering on linkevents. Node membership in clusters. Overview. Nodes 1232-1274	315
Table C-25: Case study. Content-based clustering on linkevents. Node membership in clusters. Metrics. Nodes 1125-1192.....	316
Table C-26: Case study. Content-based clustering on linkevents. Node membership in clusters. Metrics. Nodes 1193-1274.....	317
Table C-27: Case study. Content-based clustering on linkevents. Share of structural key players in cluster. Overview	318
Table C-28: Case study. Content-based clustering on linkevents. Cluster memberships of structural key players	319
Table C-29: Case study. Content-based clustering on linkevents. Node role categorization. Node overview. Nodes 1125-1188	320
Table C-30: Case study. Content-based clustering on linkevents. Node role categorization. Node overview. Nodes 1191-1275	321
Table C-31: Case study. Content-based clustering on linkevents. Node role categorization. Cluster overview. Clusters 0-40	322
Table C-32: Case study. Content-based clustering on linkevents. Node role categorization. Cluster overview. Clusters 0-40	323
Table C-33: Case study. Content-based clustering on linkevents. Knowledge domain categorization. Clusters 0-49	324
Table C-34: Case study. Content-based clustering on linkevents. Knowledge domain categorization. Clusters 50-86	325
Table C-35: Case study. Content-based clustering on linkevents. Overview of knowledge domain categorization.....	326
Table C-36: Case study. Content-based clustering on linkevents. Knowledge profile categorization. Nodes 1125-1192.....	327
Table C-37: Case study. Content-based clustering on linkevents. Knowledge profile categorization. Nodes 1193-1275	328
Table C-38: Case study. Content-based clustering on linkevents. Overview of knowledge profile categorization.....	329
Table C-39: Case study. Cluster comparison. Structural versus content-based clustering on nodes. Cluster-centric group membership stability values.....	330
Table C-40: Case study. Cluster comparison. Structural versus content-based clustering on nodes. Actor-centric group membership stability values	331

Table C-41: Case study. Cluster comparison. Structural clustering versus linkevents. Cluster-centric group membership stability values. Part I.....	333
Table C-42: Case study. Cluster comparison. Structural clustering versus linkevents. Cluster-centric group membership stability values. Part II	334
Table C-43: Case study. Cluster comparison. Structural clustering versus content-based clustering on linkevents. Actor-centric group membership stability values. Node 1174.....	335
Table C-44: Case study. Cluster comparison. Structural clustering versus content-based clustering on linkevents. Actor-centric group membership stability values. Node 1198.....	336
Table C-45: Case study. Cluster comparison. Structural clustering versus content-based clustering on linkevents. Actor-centric group membership stability values. Node 1264.....	336
Table C-46: Case study. Cluster comparison. Content-based clustering on nodes versus linkevents. Cluster-centric group membership stability values. Part I.....	338
Table C-47: Case study. Cluster comparison. Content-based clustering on nodes versus linkevents. Cluster-centric group membership stability values. Part II.....	339
Table C-48: Case study. Cluster comparison. Content-based clustering on nodes versus linkevents. Actor-centric group membership stability values. Node 1174	340
Table C-49: Case study. Cluster comparison. Content-based clustering on nodes versus linkevents. Actor-centric group membership stability values. Node 1264	340
Table C-50: Case study. Cluster comparison. Content-based clustering on nodes versus linkevents. Actor-centric group membership stability values. Node 1198	341

Abbreviations

AT&T	American Telephone & Telegraph Corporation
BC	Betweenness Centrality
CAD	Computer-aided Design
CALO	A Cognitive Assistant that Learns and Organizes
CC	Closeness Centrality
Cf	Collection Frequency
CFG	Context-free Grammar
CLAWS	Constituent-Likelihood Automatic Word-tagging System
CLEF	Cross Language Evaluation Forum
CM	Clustering Method
CPCC	Cophenetic Correlation Coefficient
CRISP-DM	Cross Industry Standard Process of Data Mining
CRM	Customer Relationship Management
CSG	Context-sensitive Grammar
CSV	Comma-separated Values
DC	Degree Centrality
Df	Document Frequency
DOM	Document Object Model
Dpa	Deutsche Presse-Agentur
DSS	Decision Support System
FOAF	Friend-of-a-Friend
FR	Fruchterman-Reingold
GN	Girvan-Newman
GUI	Graphical User Interface
HMM	Hidden Markov Model
Idf	Inverted Document Frequency
IKM	Information & Knowledge Management
IR	Information Retrieval
IRIS	Information Systems Research in Scandinavia
IT	Information Technology
JUNG	Java Universal Network/Graph Framework
KDD	Knowledge Discovery in Databases
KDT	Knowledge Discovery from Text Knowledge Discovery in Textual Databases
KM	Knowledge Management

Abbreviations

KPI	Key Performance Indicator
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
NLP	Natural Language Processing
NP	Noun Phrase
PCFG	Probabilistic Context-free Grammar
PHC	Public Health Communications
POS	Part-of-Speech Tagging
QAP	Quadratic Assignment Procedure
SAHN	Sequential, Agglomerative, Hierarchical, Non-overlapping
SAX	Simple API for XML
SMS	Short Message Service
SNA	Social Network Analysis
SNI	Social Network Intelligence
SNIBR	Social Network Intelligence-based Routing
SQL	Structured Query Language
SVD	Singular value decomposition
SWT	Standard Widget Toolkit
Tf	Term Frequency
TIC	Truly Informed Consent
TREC	Text REtrieval Conference
UMLS	Unified Medical Language System
VP	Verb Phrase
WSD	Word Sense Disambiguation
WSM	Web and Social Media
WWW	World Wide Web
XML	Extensible Markup Language
XSD	XML Schema Definition

1 Introduction

This thesis will introduce a novel approach for identifying groups of similar knowledge in social corpora. Therefore, the theoretical foundations of this work cover social network analysis, text mining and cluster analysis. Based on these three fields of research the proposed method contributes to the analytical means of the Social Network Intelligence framework proposed by the IKM Research group¹ for information and knowledge management (IKM) founded at the Department for Systems Analysis and IT at TU Berlin, Germany. The method and its prototypical implementation are applied to a corporate e-mail corpus to evaluate its properties and benefits on a real-world data set.

In the first section of this chapter the basic motivation of this study is given to allow determining its relevance in the context of knowledge management and information systems research. Section 1.2 then presents the research objectives of this work and in section 1.3 the research methodology is explained which is chosen to ensure the quality of the results regarding relevant business needs and rigorous methods from the applicable knowledge base. Finally, the chapter overview in section 1.4 will serve as a guide to the content of this work.

1.1 Motivation

According to Nonaka and Takeuchi (1995) knowledge can be *tacit* as well as *explicit*. Tacit means that the knowledge is locked in the individual's mind whereas explicit knowledge is already accessible to others. There are four different mechanisms to convert these two types of knowledge: *socialization* (tacit to tacit), *externalization* (tacit to explicit), *internalization* (explicit to tacit) and *combination* (explicit to explicit). Their consecutive succession can be viewed as a knowledge spiral of a continuous learning process. This process can be physical, virtual, mental or based on relationships. To make knowledge available to others knowledge management (KM) aims at identifying, acquiring, developing, distributing and exchanging, storing as well as evaluating knowledge (Probst et al. 1998; Thielscher 1999). A stricter definition of tacit knowledge requires that it cannot be codified, but can only be transmitted via training or gained through personal experience (Polanyi 1966). According to e.g. Granovetter (1973), Allen (1977) or Burt (1992) social relations are of major importance for acquisition of information and the exchange of knowledge. Therefore, the creation of knowledge can be defined as a social process (Mead 1934; Wittgenstein 1953; Berger and Luckman 1966). Thus, an effective transfer of tacit knowledge generally requires extensive personal contact and trust as it involves habits and culture.

The basic motivation for this study is how the analysis of knowledge networks can enhance workflow modeling and business process analysis. In this context, van der Aalst et al. (2005) combine concepts from both workflow management and social network analysis. They retrieve social networks from mining highly structured event-logs, e.g. information of which tasks is handed over from one performer to the next. However, some business units cannot be

¹ <http://www.ikmresearch.de/>

designed and organized by predefined process models that – once established – never or rarely ever change. There only exist more or less detailed parameters how and when certain activities have to be performed. These activities are not described by a (complex) series of functions and events but by the knowledge and experience required to achieve a certain goal. A collection of these activities can be subsumed as a knowledge-intensive business process which is characterized by the individuals that are involved, their knowledge, and their interactions that establish a network among them. According to Trier and Bobrik (2007b; 2007c) the analysis of knowledge-intensive business processes is part of the discipline of process-oriented knowledge management. In recent years this approach has become quite popular. Knowledge-intensive business processes can be identified by a high degree of complexity, a low degree of structuredness, a high degree of communication, a low degree of predictability as being hard to schedule, a high degree of exceptions from predefined business rules, a high degree of work autonomy and ongoing high degree of information need (see e.g. Davenport et al. 1996; Abecker et al. 2002; Heisig 2002). Knowledge workers transform their knowledge into productivity during their daily work. Therefore, analyzing these processes can help to identify sources and demands of knowledge. As part of a systems analysis project process-oriented knowledge management can help to derive suitable recommendations for actions and improve the workflow. Thus, managing these processes requires focusing on the individuals' knowledge and how it can be identified and made accessible. Understanding the workforce and skill-set of an enterprise can be seen as the key to understand the capabilities of an organization. In today's large organizations it has become increasingly difficult to find people that have specific skills or knowledge or to explore the overall picture of an organization's portfolio of topic expertise (Brunnert et al. 2007). Bringing together human actors with similar interests, skills or knowledge is a major challenge in community-based knowledge management. According to Belbin (1993) there are different roles an individual can obtain in a team which can be defined as the way an individual behaves in the team, contributes to its purpose and goals and interacts with others. He distinguishes nine complementary team roles: plant, resource investigator, coordinator, shaper, monitor evaluator, team worker, implementer, completer finisher, and specialist. These team roles can be utilized in formation of new teams as well as in evaluation of existing teams. However, as a prerequisite it is essential to understand the characteristics and capabilities of each individual that are developed during formal as well as informal collaboration processes within the organization.

In business processes communicational activities are an important means of collaboration which increases with the amount of knowledge generation and reuse. Electronic communication media provide a wide range of possibilities for decentralized communication and collaboration. They have become the most widespread means of interacting in a company. In a study by Fallows (2002), 98% of all employees with internet access collaborate via e-mail at their workplace, most of them several times a day. 75% of the questioned employees used it to improve their teamwork and to increase the number of people they actively communicate with (Fallows 2002). E-Mail is thus a strong force for connecting people at the workplace. It has been shown to complement formal work networks and provide more diverse, participative and less formally aligned relations (Bikson and Eveland 1990). Such effects are not limited to the domain of e-mail communication: The internet sports new

combined forms of interaction that blend contents and communication (Trier and Bobrik 2008b). All these interactions of people captured over time form a social network structure via communication (Krackhardt 1992). More recently, in the context of internet 2.0 or web 2.0 they are often called social software applications as they allow socializing via the internet. According to McAfee (2006) web 2.0 technologies have the unique ability to capture tacit knowledge, best practices and relevant experiences of a company and make them readily available to more users. These benefits are crucial for successful knowledge management in the enterprise. Integrating web 2.0 concepts and applications like wikis, blogs, or messaging software (so called “*social software*”) for groups and individuals into the corporate intranet is therefore termed *enterprise 2.0* (McAfee 2006). It is a system of web-based technologies that provide rapid and agile collaboration, information sharing, emergence and integration capabilities in the extended enterprise. Through the adoption of web 2.0 techniques, information workers are able to control their own user experiences with less guidance from IT, and thus create for themselves a more intuitive and efficient work environment. Enhanced by these technologies the intranet can develop into a constantly changing structure built by distributed, autonomous peers. This results into a spontaneous, knowledge-based collaborative platform. However, there is still a debate whether enterprise 2.0 will last and contribute to the discipline of knowledge management (Lakhani and McAfee 2007; Davenport 2008). Apart from case studies of enterprise 2.0 approaches only little in-depth research has been made in this area of knowledge management. Most literature on this topic can be found in blog postings, e.g. by Dion Hinchcliffe². Social software applications that can be reasonably adapted for enterprise 2.0 include e-mail as a basic means of communication, wikis for collaborative writing, blogs for storytelling and best practice transfer, social bookmarking for tagging and building organizational folksonomies, RSS for information distribution, collaborative planning software for peer-based project planning and management idea banks for idea generation, social networking tools for human relations, mashups for visualization and prediction markets for forecasting and identifying risks.

According to McAfee (2006) these social software applications must fulfill the following functionalities to be able to enhance the corporate intranet (McAfee 2006): search, links, authoring, tags, extensions and signals. These functionalities are usually abbreviated as SLATES. Additional functions have been proposed by Dion Hinchcliffe (Hinchcliffe 2007):

- (1) *Freeform function*: No barriers to authorship (e.g. no learning curve or restrictions).
- (2) *Network-oriented function*: Requiring web-addressable content in all cases.
- (3) *Social function*: Stressing transparency (to access), diversity (in content and community members) and openness (to structure).
- (4) *Emergence function*: Requiring the provision of approaches that detect and leverage the collective wisdom of the community.

McAfee (2006) recommends installing easy-to-use software which does not impose any rigid structure on users. He envisages an informal roll-out on a common platform to enable future collaboration between areas and recommends strong and visible managerial support to achieve this. The resulting organizational communication patterns can lead to highly

² <http://blogs.zdnet.com/Hinchcliffe/>

productive and highly collaborative environments by making both the practices of knowledge work and its outputs more visible. Based on case studies and survey data McAfee (2006) presents four rules for implementing the new technologies:

- (1) *Culture*: A receptive culture must be encouraged to prepare the way for new practices.
- (2) *Technology*: A common platform must be created to allow for a collaboration infrastructure.
- (3) *Strategy*: An informal rollout of the technologies may be preferred to a more formal procedural change.
- (4) *Management*: Managerial support and leadership is crucial.

However, there are fundamental differences between the internet and intranet, e.g. privacy and security of information, or work and leisure time context which have to be regarded (Stenmark 2005). Nevertheless, properly understood and deployed, web 2.0 technologies, methods and patterns can be used in the enterprise to improve organizational productivity and efficiency

One of the primary methods for studying the resulting electronic communication, collaboration and interaction and their inherent communities is social network analysis (SNA) (Garton et al. 1997). In Bobrik and Trier (2009) the collection of all electronic traces of interrelated communication relationships is defined as a *social corpus*. In the broad field of SNA research there are a number of publications concerning search and identification of important key players in social corpora. Some of them only consider the content objects, some of them additionally exploit network structures to enhance the analysis. In the context of enterprise 2.0 these media have become popular means and social networking an important but yet unsolved issue. This study concentrates on social networks evolving on different kinds of collaborative content creation in contrast to the conventional structural definition of social networks.

Research in such social corpora is directed at understanding the structures and properties of these complex systems to identify structural patterns (Kossinets and Watts 2006). One important interest is the location of groups in such networks. For this objective SNA offers a series of clustering and grouping algorithms that find widespread application in practice. For example, Tyler et al. (2003) applied a graph-theoretic betweenness centrality algorithm to automatically identify communities in a research department of 400 employees at Hewlett-Packard solely based on studying e-mail interactions. They detected 66 distinct groups reaching from 2 up to 57 members. A subsequent manual validation has shown that about 60% of the persons found the automatically identified groups to be completely correct. This example shows that networks simultaneously allow flows of knowledge and provide connectivity but are also often structured in clusters of actors (Levine and Kurzban 2006). Actors tend to cluster in homogenous groups that are only loosely connected to other groups (Ravasz and Barabási 2003). However such research and the current algorithmic implementations in common SNA tools like Pajek or UCInet only provide structural analysis of community groups.

According to Borgatti and Cross (2003) SNA research mainly focuses on the analysis of on structural properties of networks to identify declarative (“know-what”) and procedural (“know-how”) knowledge but neglect the mechanisms of social relationships (“know-who”)

for acquiring information. Therefore, the authors propose three relational characteristics that are predictive of the behavior of information seeking and knowledge exchange: (1) *knowing* what that person knows; (2) *valuing* what that person knows; (3) being able to gain timely *access* to that person's thinking. Furthermore, knowing and access will mediate the relationship between physical proximity and information seeking. Here, Borgatti and Cross (2003) emphasize the importance of face-to-face interactions during the course of a given project but also propose alternative means for virtual work environments to promote interaction among employees and support social interventions to develop knowledge and access relations as relational conditions of knowing and access (e.g., skill profiling systems, developmental staffing practices, or action-learning techniques). However, this will require a detailed understanding of the enterprise's supply and demand of knowledge as well as its structure and characteristics.

There are estimates that about 85% of business-relevant information originates in unstructured form (Grimes 2010). In this study the term collaborative content is invented to subsume the various kinds of text data that are available from social media which can range from short text messages to exchanging or even simultaneously editing large documents. Although the content information can be directly linked with the relational data forming the social network the domain of content analysis with means of text mining and information retrieval has not yet been sufficiently accommodated in the methodological framework of SNA. Related research in content analysis and topic mining is focused on algorithms for generating keyword and topic descriptors in semi-formal text (Castellanos 2003). In the few studies of contents of social corpora, the focus is on expert profiling and on subsequent assessment of search strategies to search for persons with appropriate knowledge in online networks (Zhang and Ackerman 2005). The authors apply conventional text mining to index all the messages of a person and create a keyword vector, in which a keyword is weighted by its term frequency-inverted document frequency. A precursor to the work in this paper is the content-based analysis and exploration concept called social search as discussed in Trier and Bobrik (2008b). It combines network visualization and the ability to create topic oriented sub-networks of an online discourse. Despite such initial approaches, these and similar studies mostly use content analysis to profile and search for individuals in social corpora. A comprehensive method of applying content-based analysis to study the group level of social networks is missing. Consequently, there is not much known about how topical profiles are spread in social corpora, about how virtual groups reflect topics, or if topical analysis would yield divergent group segmentation. Text data that can be retrieved from social media are often analyzed separately from the structure of the network (see chapter 3, section 3.5). Few approaches exist where both types of data are combined (see e.g. Bobrik and Trier 2009; Diesner and Carley 2010). Neglecting the content of text data during network analysis can limit the understanding of the network structure, its evolution over time and the multiple roles an actor can obtain in this context. Therefore, conventional SNA will not be able to detect groups of similar knowledge that evolve within a multi-contextual network where each node can obtain multiple roles. Such communities are not only a structural phenomenon but have been defined by seminal literature as *networks about something* (Wenger et al. 2002). To meet the described shortcomings of SNA research, this study concentrates on a new approach to

approximate the context of the network by integrating both levels of analysis: content analysis and network analysis.

1.2 Problem Statement of the Study and Research Questions

The basic motivation of this study is how the knowledge lifecycle can be managed (organizational KM approach) and supported (techno-centric KM approach). As knowledge is often locked in the individual's mind (i.e. tacit knowledge, see section 1.1) this question can be specified by which benefits are provided by social networking principles (organizational perspective) and applications (techno-centric perspective) to identify and exchange knowledge. This includes integrating web 2.0 application and principles in the corporate intranet (enterprise 2.0) and evaluating its benefits, e.g. by identifying and grouping people by knowledge requirements. Social networking involves collaboratively creating, distributing and maintaining verbal as well as non-verbal content. Evaluating, controlling and benchmarking the benefits of these social networking activities in the enterprise involve SNA approaches and techniques. Based on the recognition of the above shortcomings, this thesis focuses on extending conventional structural social network analysis, i.e. how people connect, with a new perspectives on the dynamics of collaborative content. Here, content is meant to approximate the context of network, i.e. why people connect. Therefore, this thesis will introduce an approach for content-based community detection in social corpora as part of the Social Network Intelligence framework. With this method, current structural insights about groups in social corpora can be extended with a topical perspective. Identified groups of shared experience can be related to structural patterns in order to identify topic communities which are inefficiently spread across the interaction structure or which are isolated from other people. A contribution to SNA research can be made on better algorithms to identify group-level properties of electronic networks. This will enable sophisticated automated awareness services, i.e. to answer questions like who is related to my context, or who is a relevant contact, the support of large overlapping global teams or other collaborative community services. In summary, in the course of this work a novel method for analyzing collaborative content and identifying the different groups of interest, knowledge, or skills is designed and implemented as a prototype. To proof its significance, the nowadays popular graph-based community detection methods will be compared with this method in a case study.

The research activities of this work are driven by the following research questions:

- (1) How can social network analysis, text mining and cluster analysis be combined to identify and analyze groups of similar knowledge in social corpora based on the content of interaction?
- (2) Does the method of content-based clustering help to gain insights on the network, e.g. shared knowledge that cannot sufficiently be explained by the state-of-the-art structural clustering methods available? To what extend do groups identified by existing methods of structural clustering correlate with content-based communities?
- (3) Examining the results of the different clustering procedures, do actors obtain different roles on different levels of details (i.e. group versus network level) and different levels of investigation (i.e. structure versus content level)?

- (4) Can the proposed method of content-based clustering for knowledge identification help to categorize and generalize knowledge profiles (actor level) and fields of knowledge (group level)?
- (5) Can the identification of fields of knowledge in social networks provide benefits to improve the acquisition, maintenance and exchange of knowledge in the network? If the data is retrieved from business applications, do these insights help to understand and improve informal work relations, e.g. in knowledge-intensive business processes, as part of the Social Network Intelligence framework?

The first question motivates the new method of content-based clustering for knowledge identification based on the combination of techniques and methods from different scientific disciplines. The second question investigates in how far additional insights can be gained using the proposed method compared to existing approaches. The third question refers to the identification of context-specific roles of an actor. The fourth question deals with the categorization of knowledge to establish knowledge profiles (actor level) and to identify different fields of knowledge which allow deriving generalized characteristics (group level).. Finally, the fifth question focuses on the benefits of the new approach for knowledge identification for social network analysis in general and business applications in particular.

1.3 Research Procedure and Methodology

According to Hevner et al. (2004) the discipline of information systems (IS) research deals with the interrelation of people, organizations and technology: “IS research must address the interplay among business strategy, IT strategy, organizational infrastructure, and IS infrastructure” (Hevner et al. 2004: 78). According to Kalakota and Robinson (2001) and Orlikowski and Barley (2001) this interplay is crucial in modern business applications as information technology has become the enabler of business strategies and organizational infrastructure. Thus, innovative information systems strongly influence the success of business strategies (Drucker 1988; Drucker 1991; Orlikowski 2000).

The main purpose of this work is to develop a new technology-based method for knowledge identification to support managerial and organizational purposes (Zmud 1997; Hevner et al. 2004: 76). To achieve this goal Hevner et al (2004) recommend to combine the two complementary paradigms of behavioral science and design science. As rooted in natural science research the primary goal of *behavioral science* is to develop and verify *theories* to explain and predict the interactions among people, technology, and organizations which have to be considered if an information systems is intended to improve the effectiveness and efficiency of an organization. These theories are strongly interrelated with decisions about the design and implementation of the information system. Here, *design science* is a problem-solving paradigm rooted in engineering science and artificial intelligence (Simon 1996; Hevner et al. 2004: 76). It aims at creating new and innovative *artifacts* to expand the boundaries of human and organizational capabilities. These artifacts may be ideas, practices, technical capabilities, and products (Denning 1997; Tsichritzis 1997) to support the analysis, design, implementation, management and use of information systems. Creating these artifacts depends on natural laws or existing behavioral theories that are applied and verified in practice (Walls et al. 1992; Markus et al. 2002). As technology and behavior are inseparable in general (2000), they are also inseparable in IS research (Hevner et al. 2004: 77). Therefore,

combining behavioral science with design science paradigms helps to address the fundamental problem of productive application of information technology (March and Smith 1995). Design science in IS research includes design processes and design artifacts (Walls et al. 1992; March and Smith 1995). A design process in IS research is a set of activities, i.e. *build* and *evaluate*. A design artifact is a product, i.e. *constructs* (vocabularies and symbols), *models* (abstractions and representations), *methods* (algorithms and practices) and *instantiations* (implemented and prototype systems).

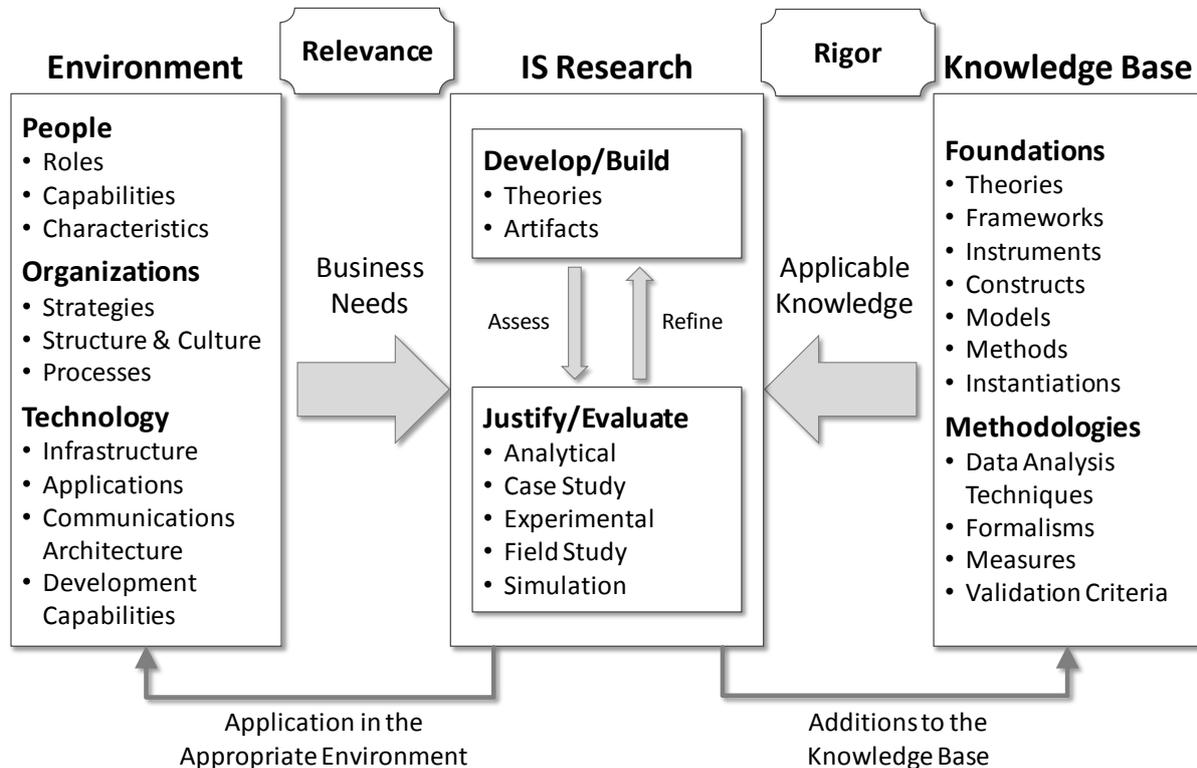


Figure 1-1: Information systems research framework. Source: Hevner et al. (2004: 80)

In order to enhance the understanding, execution, and evaluation of IS research combining behavioral-science and design-science paradigms Hevner et al. (2004) propose the conceptual framework illustrated in Figure 1-1 focusing on technology-based design. There are numerous studies based on this framework for high-quality design-science research in business context, e.g. Mendling (2007), Recker (2008), or Offermann (2009). According to Simon (1996) the *environment* defines the problem space. *Relevance* is achieved by conducting IS research with respect to identified *business needs* that determine the problem space. In IS research it consists of *people*, *organizations* and *technologies* (Silver et al. 1995). With respect to the business needs IS research covers the two complementary phases “develop/build” and “justify/evaluate” that are influenced by the behavioral-science and design-science paradigms. Behavioral science is conducted during *development* and *justification* of theories explaining or predicting phenomena related to the identified business needs to obtain *truth*. Design science is conducted during *building* and *evaluation* of artifacts designed to meet the business needs to obtain *utility*. In the context of IS research truth and utility are inseparable and interrelated (Hevner et al. 2004: 80). The *knowledge base* provides the necessary means to accomplish IS research including *foundations* that are prior IS research and results from reference disciplines used in the “develop/build” phase as well as *methodologies* that are guidelines used in the

“justify/evaluate” phase. *Rigor* is achieved by appropriately applying this knowledge base. In behavioral science, these methodologies cover data collection and empirical analysis. In design science, computational and mathematical methods are used to evaluate the artifacts. In contrast to *routine design* design-science research is based on the clear identification of a contribution to the knowledge base (Hevner et al. 2004: 81).

The IS research framework helps to assess the contributions of behavioral science and design science in IS research. High quality and usefulness of the results can be achieved if the contributions to IS research are “applied to the business need in an appropriate environment and [...] add to the content of knowledge base for further research and practice” (Hevner et al. 2004: 81).

This study aims at contributing to the discipline of information systems research in order to improve the effectiveness and efficiency of the knowledge lifecycle within an organization. The research methodology is related to the IS research framework proposed by Hevner et al. (2004) as depicted in Figure 1-1. It includes the analysis of data retrieved from social systems within the organizations (“technology”) to improve structures and processes that strongly rely on knowledge acquisition, maintenance and exchange (“organizations”). Identifying heterogeneous roles in different fields of knowledge and deriving categories of knowledge profiles to support the individual capabilities to participate in the knowledge lifecycle will meet important requirements to support knowledge work within the organization (“people”). These business needs determine the relevance of this study (see also section 1.1). A design science research methodology is employed as the focus is less on empirical insights about factor relationships but on the design and evaluation of a method, i.e. a design artifact (Vaishnavi and Kuechler 2004). As cluster analysis is an exploratory data analysis it remains a tool for discovery and experience. Therefore, the needs and special circumstances encountered in individual problems influence the methodology (Jain and Dubes 1988). Applicable knowledge includes rigorous and well-examined foundations and methodology of social network analysis, text mining and cluster analysis that also provide measures and validation criteria to evaluate the results. Here, the Social Network Intelligence framework is presented that extends conventional SNA and provides the basic foundation of this study to support researchers as well as practitioners. As a result an IT artifact is developed including a set of algorithms and a research guideline (method) and their implementation within a prototype (instantiation) to identify groups of knowledge to improve the knowledge lifecycle within an organization (model). A detailed case study will allow evaluating the benefits of the IT applying the method to a corporate e-mail data set using the prototype. Existing empirical and qualitative methods of the applicable knowledge base are used to retrieve and refine the data, conduct the analysis, validate the results and determine their quality to support researchers as well as practitioners. Furthermore, applied to an e-mail corpus it tries to explain existing phenomena in SNA in an organizational context and tries to provide a categorization of knowledge profiles and fields of knowledge. It therefore includes IS behavioral-science research as well.

1.4 Guide to Content

Retrieving all kind of documents (“content”) from social software applications will form the *knowledge base* to provide the data source for the content-based cluster analysis prototype which will involve information retrieval as well as clustering techniques. The results from the prototype will be analyzed using the Social Network Intelligence framework developed by the IKM Research group. Altogether a new method for analyzing collaborative content in social corpora using content-based clustering will be developed in this study. As illustrated in Figure 1-2 this work is organized into six chapters.

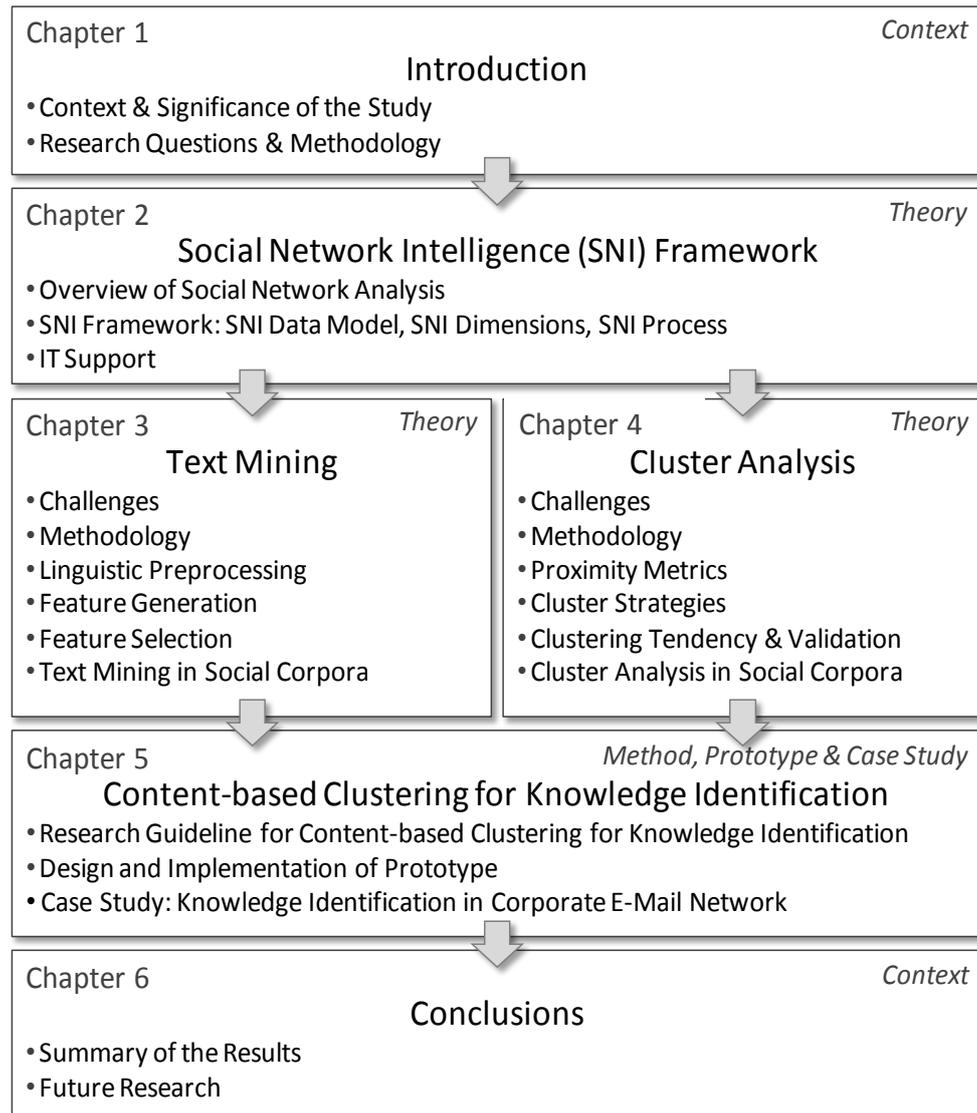


Figure 1-2: Content-based clustering for knowledge identification in social corpora – guide to content

In chapter 1 the basic motivation of the study is given. It provides an overview of the context of the study within knowledge management research and discusses the significance of the study in this context. Furthermore, five research questions are proposed and the research methodology is explained which will guide the theoretical foundation of this work as well as the design, implementation and evaluation of the new method of content-based clustering for knowledge identification in social corpora.

In chapter 2 the Social Network Intelligence framework is introduced which extends conventional social network analysis providing novel insights into network structure, content, behavior and context. Therefore, a general overview of social network analysis is given including its definition, the basic concepts, as well as selected research and network metrics to provide an understanding of the context of this work in general and the Social Network Intelligence framework in particular. Based on this theoretical foundation the SNI framework consists of three components: the SNI data model, the SNI dimensions and the SNI process. The last section gives an overview of available IT support for network analysis. As a result, the SNI framework is designed to enable IT-supported, network-oriented knowledge management on multiple integrated levels of analysis.

The new method of content-based clustering for knowledge identification in social corpora is based on the SNI framework. It is designed as a static content analysis on ego level (i.e. key players), group level and network level whereas the main focus is on group level including also elements of structural analysis and network dynamics. Therefore, chapter 3 and chapter 4 present text mining and cluster analysis as theoretical foundations.

Chapter 3 provides an overview of concepts and methods for automated mining of text samples in general and their application on social corpora as part of a SNA study in particular. First, a brief introduction to text mining is given covering its definition and related research areas as well as application areas and current challenges. Afterwards, text preprocessing techniques to generate the knowledge base are explained. Finally, an overview of text mining applications in social corpora is given.

Cluster analysis provides methods and techniques to identify group memberships in a data set. Chapter 4 is therefore organized as follows: First, a general introduction to cluster analysis is given and the consecutive steps of a clustering methodology are explained. The following sections discuss the methods and techniques that are involved in the clustering methodology steps including the initial data collection, the clustering process itself as well as some final tests for cluster tendency and validity. Finally, this chapter presents applications of cluster analysis in social corpora with special focus on the popular graph-based edge-betweenness clustering algorithm.

In chapter 5 the research guideline for the new method of content-based clustering for knowledge identification in social corpora is proposed. Being part of the SNI framework presented in chapter 1 the method contributes to its analytical toolset for analyzing social networks that can be characterized by the three SNI dimensions. Based on its requirements a prototype is implemented to allow an IT-supported analysis integrating relevant methods of text mining, cluster analysis and social network analysis. This chapter concludes with a case study which applies the method to a corporate e-mail data set using the prototype.

Finally, in chapter 6 a conclusion of the entire work is given relating the results of the case study to the research questions.

2 Social Network Intelligence Framework

The first section of this chapter provides a general introduction to social network analysis including its definition, the basic concepts as well as selected research and network metrics to provide an understanding of the context of this work in general and the Social Network Intelligence (SNI) framework in particular (section 2.1). Based on this theoretical foundation the three elements of the Social Network Intelligence framework are discussed in detail (section 2.2): the SNI data model, the SNI dimensions and the SNI process. The last section gives an overview of available IT support for network analysis with special focus on the Commetrix software (section 2.3).

2.1 Overview of Social Network Analysis

As social networks can be derived from the different types of electronic communication and collaboration media like social software applications, social network analysis (SNA) provides suitable means to measure, evaluate, improve and maintain these networks. Assuming that communities are resting on an underlying social network enables the systematic examination of such computer-networked communities (Wellman et al. 1996). The relationships between them are of value for both the community and the organization. For this objective of analysis, the rapid and regular advance in social networks research provides a vast body of measurements and methodologies (Wasserman and Faust 1994).

After the definitions of the terms social network and social network analysis is defined in section 2.1.1, section 2.1.2 explains the basic concepts of SNA whereas section 2.1.3 gives a brief overview of selected SNA research. The overview of SNA research concludes with a summary of popular SNA metrics (section 2.1.4).

2.1.1 Definitions

The term *social network* was first introduced J. A. Barnes in the 1950s. It then described an association of people drawn together by friendship, kinship, or work (Barnes 1954). Wasserman and Faust (1994) more generally define a social network as a set of *social entities* and the structured relationships (e.g. knowledge or prestige) or interdependencies (e.g. financial exchange) between them. The term *social entity* is not limited to human individuals but can also be extended to e.g. organizations or animals. Relationships can be political, economic, interactional, or affective. Social networks have been used to examine how organizations interact with each other, characterizing the many informal connections that link executives together as well as associations and connections between individual employees at different organizations. For example, power within organizations often comes less from his actual job title but more from the degree to which an individual within a network is at the center of many relationships (“central hub”). Social networks also play a key role in employee hiring, in business success, and in job performance. These networks provide ways for companies to gather information, deter competition, and collude in setting prices or policies (Tichy et al. 1979; Wasserman and Faust 1994). An example of such data is provided in the case study in chapter 5, section 5.3.

More theoretically, a social network is a social relational system with the elements *actors* and *relations* (relational ties). As described in section 2.2.2.1 actors can also be analyzed on different levels of detail, like dyads (pairs of actors in two states, adjacent or not adjacent), triads (three nodes subgraph with four states: 0, 1, 2, 3 connections), subgroups, or the overall network (Wasserman and Faust 1994). Using a graph theoretic approach, the actor is also being termed *vertex* or *node* and the relation is being termed *edge* or *link*. In the context of electronic communication media social networks are virtually present whenever a group of people interacts electronically (Wellman 1997).

Social network analysis (SNA) is related to network theory and has emerged as a key technique in modern sociology. It has also gained a significant following in anthropology, biology, communication studies, economics, geography, information science, organizational studies, social psychology, and sociolinguistics where it has become a popular topic of speculation and study. A summary of the progress of social networks and social network analysis can be found in Freeman (2006).

SNA is a theory-driven methodology, not a theory by itself. It hardly fits a single school of thought but is a multi-theoretical and multi-level way of analysis, as Contractor et al. (2006) as well as Monge and Contractor (2003) argue. Most authors have a structural approach to social networks analysis: For example, Wasserman and Faust (1994) subsume the analysis of structured social relationships within social networks under this term. Here, social network analysis concentrates on the patterns and implications of these relationships. SNA provides both a visual and a mathematical analysis of human relationships.

The following analytical tendencies distinguish social network analysis (Freeman 2004): First, there is no assumption that groups are the building blocks of society. This approach is open to studying less-bounded social systems, from nonlocal communities to links among websites. Rather than treating individuals (persons, organizations, or states) as discrete units of analysis, it focuses on how the structure of ties affects individuals and their relationships. In contrast to analyses that assume that socialization into norms determines behavior, network analysis looks to see the extent to which the structure and composition of ties affect norms.

2.1.2 Basic Concepts

In general, there are two formal methods how social relations among a set of actors can be represented: graph theory (see section 2.1.2.1) or matrix algebra (see section 2.1.2.2). Using such a formal representation concepts of social structure can be established and precisely defined using mathematics (Hanneman and Riddle 2005). Two-mode networks are a special type of networks that provide interesting properties. This concept is explained in section 2.1.2.3.

2.1.2.1 The Data as a Graph

In graph theory the elements of the network are displayed as a graph representing each actor in the network as node (or vertex) and each relation between two actors as link (or edge, or tie). In sociology the representation of a social network as a graph is called a *sociogram* (Moreno 1934). Coloring, shading, or different shapes and sizes are often used to represent

attributes of the individual nodes and links (see Bertin 1967/1983; Card et al. 1999). See section 2.2.3.4 for more details.

Each relation may be *directed* (i.e. originates with a source actor and reaches a target actor), or it may be *undirected* representing e.g. co-occurrence or co-presence between the pair of actors (Hanneman and Riddle 2005). Directed relations are represented with arrows. Reciprocated directed relations can be represented either as two links or one link with a double-headed arrow. Figure 2-1 shows the representation of a network as a directed graph.

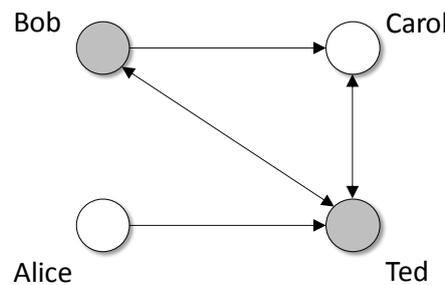


Figure 2-1: The network as a directed graph. Node color represents gender: white: female; gray: male. Source: Hanneman und Riddle (2005)

The strength of relations among actors in a graph may be *nominal* or *binary* (representing presence or absence of a relation); *signed* (-1 = negative, 0 = neutral, i.e. no relation, $+1$ = positive); *ordinal* (representing whether the relation is the strongest, next strongest, etc.); or *valued* or *weighted* (measured on an interval or ratio level).

Table 2-1: Different types of relations. Overview

Name	Description/Examples
Power	Relations of formal roles and hierarchy
Kinship	Family ties such as marriage and descent
Sentiment	Relations of positive or negative affection of one person for another, e.g. friendship, respect, love, or hatred
Interaction	Relations of physical co-location and interactions, e.g. in service deliveries
Instrumental	Relations to secure valuable goods, services or information, e.g. finding a job or seeking political advice
Collaboration	Relations of organizational co-operative work
Information & Communication	Transmission of messages and immaterial resources, especially information and knowledge
Transaction	Transfer of material resources through buying, selling, lending, borrowing or gift-making
Movement	Either physical movement (migration) or social movement (between occupation or status)
Affiliation	Relations of membership in clubs and organizations or attendance of events

Social actors are often connected by more than one kind of relationship. Based on the examples given by Knoke and Kuklinski (1983: 15-16) and Wasserman and Faust (1994: 37)

a number of not mutually exclusive types of relations can be derived, e.g. power, kinship, or sentiment (see Table 2-1).

A *simplex* graph contains only one type of relation, whereas *multiplex* graphs combine the information about multiple relations into a single link in the graph or represent them with different symbols, colors, shapes, line widths, etc. (Hanneman and Riddle 2005). There are different kinds of multiplex graphs for each network depending on which types of relation are combined into a single graph.

When analyzing large complex networks the data usually reveals different sub-structures whose vertex sets are subsets of the entire network. These complex structures are comprised of basic relational structures. According to Wasserman and Faust (1994: 171) the three basic structures are *star*, *line* and *circle* (see Figure 2-2). Some metrics presented in section 2.1.4 are able to capture the difference in network structure and allow to express it numerically, e.g. centrality measures.

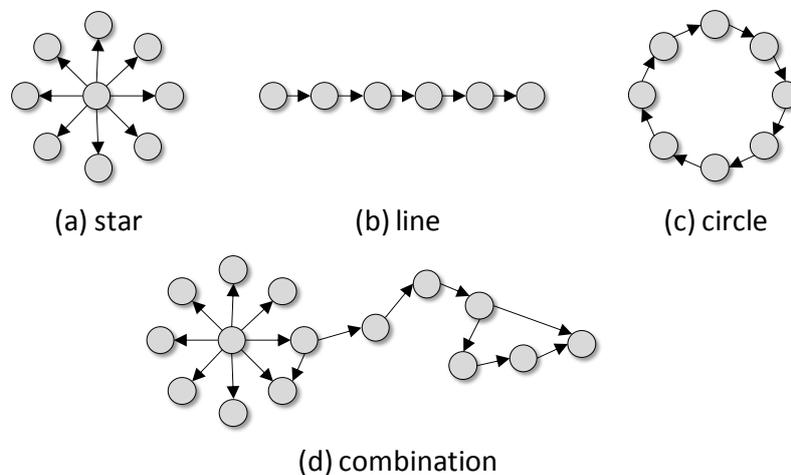


Figure 2-2: Different types of network structures: (a) star; (b) line; (c) circle; (d) combination. Based on: Wasserman and Faust (1994: 171)

Ego-centered networks are a common means to extract subsamples from large, complex networks or collect network data by surveys (see section 2.2.3.3). An *ego-centered network* consists of a focal actor, called *ego*, and a set of actors, called *alteri*, who are directly related to the ego (Wasserman and Faust 1994: 41). The network only includes relations between the ego and the alteri or among the set of alteri. Such data is often referred to as *personal data*. Thus, ego-centered networks are often used to explain the social support of personal relationships (e. g. Hammer 1983; Cohen and Syme 1985).

2.1.2.2 The Data as a Matrix

Although graphs are very useful for presenting information about social networks, especially with large networks (i.e. many actors and/or many relations) it can become very difficult to indentify patterns. Alternatively, the data can be represented as a matrix. Mathematical operations can then be performed to summarize the information in the graph, e.g. vector operations, blocking and partitioning, and matrix mathematics (inverses, transposes, addition, subtraction, multiplication and Boolean multiplication), to calculate elaborate SNA metrics (see section 2.1.4).

Matrices are often used in network analysis to represent the adjacency between any pair of actors in the network. An *adjacency matrix*, or *sociomatrix*, is a square actor-by-actor matrix where the presence of pairwise relations is recorded as elements (Wasserman and Faust 1994: 70). The main diagonal represents the self-relation of an actor to himself. It is often ignored in network analysis (Hanneman and Riddle 2005). Given a binary matrix only the presence (value 1) and absence (value 0) is recorded. Signed graphs are represented in matrix form with -1 (negative relation), 0 (neutral relation), and $+1$ (positive relation). This is actually a specialized version of an ordinal relation.

If directed relations are available an asymmetric matrix can be used where the rows represent the source nodes and the columns the target nodes. In contrast to a symmetric matrix, the element (i, j) does not necessarily equal the element (j, i) . When relations are measured at the ordinal or interval level, their numeric magnitude is entered as the element of the matrix.

Table 2-2: The network as an asymmetric adjacency matrix. Based on: Hanneman und Riddle (2005)

		Actor			
		Bob	Carol	Ted	Alice
Actor	Bob	-	1	1	0
	Carol	0	-	1	0
	Ted	1	1	-	1
	Alice	0	0	1	-

Table 2-2 shows an asymmetric adjacency matrix corresponding to the example given in Figure 2-1. The matrix contains binary values indicating the affection between four actors.

2.1.2.3 Two-Mode Networks

Usually, social network data will contain only one type of nodes, but it is also possible to include different sets of actors indicated by the *mode* of the network (Wasserman and Faust 1994: 29). A *two-mode network* consists of two disjoint sets of nodes and the relations that connect the two sets. A common example of a two-mode network is the attendance of people to social events (Davis et al. 1941; Breiger 1974). This type of a two-mode network is also called *affiliation network* (Wasserman and Faust 1994: 291), *membership network* (Breiger 1974; Breiger 1990), *bipartite networks* (Borgatti and Everett 1997), or *hypernetwork* (McPherson 1982). Other examples of two-mode networks include voting networks with politicians voting for suggestions (e.g. Doreian and Fujimoto 2003), company board networks whose board members are linked to the companies they lead (e. g. Battiston and Catanzaro 2004; Robins and Alexander 2004), (co-)citation networks with authors citing the papers (e.g. Garfield et al. 1964; Hummon and Doreian 1989; Gingras 2007), and (co-)authorship networks whose authors are linked to their papers (e.g. Newman 2001a; Newman 2001b; Newman 2004b; Molka-Danielsen et al. 2007).

A two-mode network can be represented as a *bipartite graph* where links connect only nodes from different sets (Wasserman and Faust 1994: 120). Inside these sets there are no connections. Using a matrix representation the columns correspond to one set of nodes and the rows to the other set of nodes (Wasserman and Faust 1994: 154).

Most of the techniques to analyze ordinary one-mode networks cannot be directly applied to two-mode networks. There are two possibilities how to analyze two-mode network: either by

using adapted or special techniques which allow simultaneously examining the duality of the data, e.g. using two-mode block modeling (Doreian et al. 1994; Batagelj et al. 1998), or by changing two-mode network into two separated one-mode networks.

2.1.3 Overview SNA Research

In order to provide an understanding of the context of this work and to demonstrate the diversity of current methods and research objectives in network analysis this chapter gives a brief overview of selected SNA research.

2.1.3.1 Finding Social Groups – The Southern Women Data Set by Davis et al.

Davis et al. (1941) provide a famous social anthropological study to which extent social relations tend to occur in social classes in the American south at the beginning of the 1940s.

Table 2-3: Participation of the southern women in events. Source: Davis et al. (1941)

Names of Participants of Group I		Code Numbers and Dates of Social Events Reported in <i>Old City Herald</i>														
		(1) 6/27	(2) 3/2	(3) 4/12	(4) 9/26	(5) 2/25	(6) 5/19	(7) 8/15	(8) 9/16	(9) 4/8	(10) 6/10	(11) 2/23	(12) 4/7	(13) 11/21	(14) 8/3	
1	Mrs. Evelyn Jefferson	+	+	+	+	+	+		+	+						
2	Miss Laura Mandeville	+	+	+		+	+	+	+							
3	Miss Theresa Anderson		+	+	+	+	+	+	+	+						
4	Miss Brenda Rogers	+		+	+	+	+	+	+							
5	Miss Charlotte McDowd			+	+	+		+								
6	Miss Frances Anderson			+		+	+		+							
7	Miss Eleanor Nye					+	+	+	+							
8	Miss Pearl Oglethorpe						+		+	+						
9	Miss Ruth DeSand					+		+	+	+						
10	Miss Verne Sanderson							+	+	+				+		
11	Miss Myra Liddell								+	+	+			+		
12	Miss Katherine Rogers								+	+	+			+	+	+
13	Mrs. Sylvia Avondale							+	+	+	+			+	+	+
14	Mrs. Nora Fayette						+	+		+	+	+	+	+	+	+
15	Mrs. Helen Lloyd							+	+		+	+	+			
16	Mrs. Dorothy Murchison								+	+						
17	Mrs. Olivia Carleton									+		+				
18	Mrs. Flora Price									+		+				

Over a period of 9 months, the participation of 18 women to 14 events (e.g. a meeting of a social club, a church event, or a party) of the social season in a community had been collected. The original data is given in Table 2-3. The data has been subject to a number of analyses on social affiliation and emergence of groups. For example, Breiger (1974) proposes the *duality of persons and groups*: there is a dual focus of social network analysis on how individuals create social structures while, at the same time, social structures develop an institutionalized reality that constrains and shapes the behavior of the individuals embedded in them.

The Davis data is an example of a two-mode affiliation network (see section 2.1.2.3). It can be represented as a bipartite graph (see Figure 2-3).

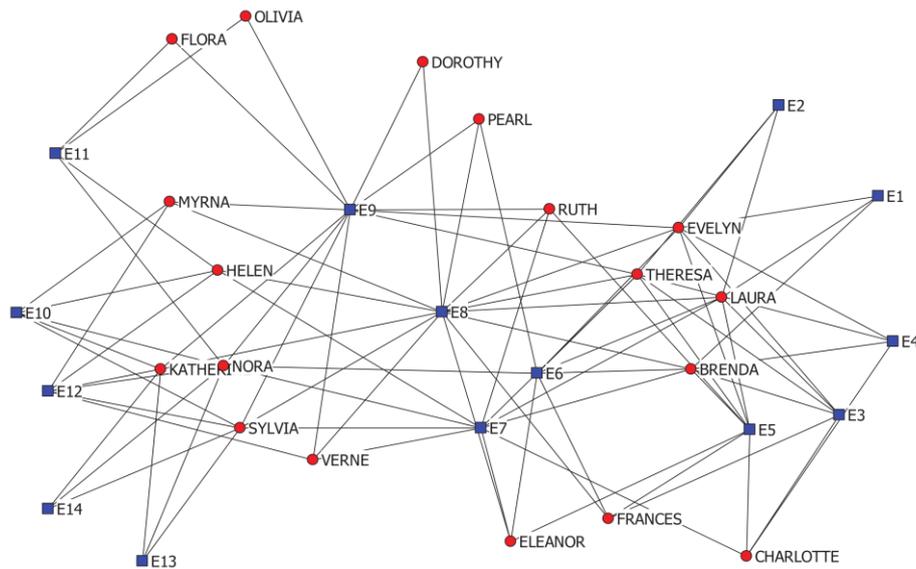


Figure 2-3: Participation of the southern women in events represented as a bipartite graph. Source: Borgatti (2009)

Davis et al. (1941) examine patterns of which women are present (or absent) at which events to infer an underlying pattern of social ties, factions, and groupings among the women. At the same time, by examining which women were present at the 14 events, it is possible to infer underlying patterns in the similarity of the events. They pursue two major research objectives: (1) identification of social groups due to co-attendance of events and (2) identification of the position of each woman in these groups (core versus periphery). Therefore, they define three levels of core/periphery participation (Davis et al. 1941: 150): (1) *core members* participate together most often in the group, *primary members* participate frequently with core members but do not form a group themselves, and *secondary members* participate only infrequently. Davis et al. (1941: 150) do not use any systematic analytic approach (see Freeman 2003: 44). Their intuition based approach relies on general ethnographic knowledge and intuition (see Table 2-4).

The original findings and even the data itself are not undisputed. There exist a large number of subsequent analyses of the data using different analytic approaches and resulting into differing implications. A thorough review and comparison of 21 approaches and their results can be found in Freeman (2003). Table 2-4 provides an overview of the group identification and core/periphery assignment of 11 comparable studies together with the analytic approach applied. Here it is most notable that the majority of them take only a subsample of the data into account.

Table 2-4: Identified groups and core/periphery assignment. Overview of 11 studies. Level of core/periphery is indicated by brackets and decreases from left to right. Based on: Freeman (2003: 62/63)

Source	Group 1	Group 2	#UA	OG	Analytic Approach
Davis et al. (1941)	[1,2,3,4][5,6,7][8,9]	[13,14,15][11,12][9,10,16,17,18]	-	+	ethnography & intuition
Homan 1950	[1,2,3,4,5,6,7][8]	[11,12,13,14,15][8,17,18]	2	+	intuition
Bonacich (1978)	[5][1,2,3,4,6]	[14][10,11,12,13,15]	6	-	Boolean algebra
Doreian (1979)	[1,3][2,4][5,6,7,9]	[12,13,14][10,11,15]	4	-	algebraic topology
Bonacich (1991)	[5][4][2][1,6][3][7][9][8]	[17,18][12][13,14][1][15][10][16]	1	+	SVD
Freeman and White (1993)	[1,2,3,4][5,6,7,8,9][16]	[13,14,15][10,11,12,17,18][16]	-	+	Galois lattice
Freeman and White (1993)	[1][2,3,4][5][6][7,9]	[14][12,13,15][11,17,18][10]	2	-	Galois sub-lattice
Borgatti and Everett (1997)	[3,4][2][1][7][6][9][5]	[12,13][11][14][10][15]	4	-	bi-cliques
Skvoretz and Faust (1999)	[1,3][2][4][5][6][7][9][8]	[12,13][14][15][11][10][17,18]	1	-	p* model
Roberts (2000)	[1][2][4][3][5][6][7][9][8]	[12][13][14][11][15][10][16][17,18]	-	-	SVD with normalization
Newman (2001c)	[1,2][3][4][6][5][7,9]	[13,14][12][11][15][10][17,18][8,16]	-	-	weighted co-attendance

#UA: number of unassigned nodes

OG: overlapping groups

2.1.3.2 Dunbar's Number

Derived from cross-cultural studies in sociology, the anthropologist R. I. M. Dunbar has suggested that the size of an ego-centered network is limited due to possible limits in the capacity of the human communication channel ("*Dunbar's number*") (Dunbar 1993): "This limit is a direct function of relative neocortex size, and that this in turn limits group size [...] the limit imposed by neocortical processing capacity is simply on the number of individuals with whom a stable inter-personal relationship can be maintained." As explained in section 2.1.2.1 an ego-centered network is the network solely consisting of the direct contacts ("alteri") of the actor under consideration ("ego"). Dunbar's number is then a theoretical cognitive limit to the number of people with whom one can maintain stable social relationships. Usually it is approximated by about 150 contacts.

2.1.3.3 Granovetter's Strength of Weak Ties

Granovetter's study provides evidence about the *strength of weak ties*, i.e. while seeking new information and insights many but weak ties, or links, are favorable (1973). Within well established groups ("cliques") their members tend to have homogeneous opinions which make them attractive towards each other. However, being similar each member of the clique would also know more or less what the other members knew. One therefore has to look beyond the clique to other, less intimate friends and acquaintances. According to Granovetter (1973: 1461) the strength of a link is "a (probably linear) combination of the amount of time, the emotional intensity, the intimacy, and the reciprocal series which characterize the tie". In practice there exist quite different ways to measure this strength, e.g. strong ties as

reciprocated links and weak ties as unreciprocated links (Friedkin 1980), recency of contact (Lin et al. 1978), or simply by using the frequency of interaction (1973: 1371).

After a decade of research, Granovetter (1983: 113) revises his hypothesis: On the one hand, weak ties provide access to information and resources beyond those available in one's own social circle. On the other hand, strong ties have greater motivation to be of assistance and more easily available. People in insecure position are more likely to resort to the development of strong ties for protection and uncertainty reduction (Pool 1980). Additionally, according to Krackhardt (1992) strong ties are important in cases of severe change as they constitute a base of trust that helps to reduce resistance and provides comfort. The pattern of friendship ties within an organization are an important means to facilitate changes and are also critical when dealing with crises (Krackhardt and Stern 1988).

2.1.3.4 Milgram's Small World Phenomenon

Another popular concept in SNA research is the so called *small world phenomenon*: The chain of social acquaintances required to connect one arbitrary person to another arbitrary person anywhere in the world is generally short. The psychologist S. Milgram performed the original small world experiment. Here, a sample of US individuals was asked to reach a particular target person by passing a message along a chain of acquaintances (Milgram 1967; Travers and Milgram 1969). The average length of successful chains turned out to be about five intermediaries or six separation steps ("*six degrees of separation*"). However, this study had low chain completion rates. Milgram's subsequent study of acquaintance networks between racial groups also reveals not only a low rate of chain completion but also the importance of social divides (Korte and Milgram 1970). Further research that replicates Milgram's findings indicates that the degrees of connection needed could be higher (Kleinfeld 2002). More research based on Milgram's experiment can be found for example in Guiot (1976), Barabási and Albert (1999), Gladwell (1999) and Barabási (2003). Watts and Strogatz (1998) published the first mathematical graph theoretic model on the small world phenomenon. This model accounts for Granovetter's observation of the strength of weak ties (see section 2.1.3.3). It has been generalized by Kleinberg (2000b) to an infinite family of random networks. An electronic small world experiment at Columbia University ascertains that about five to seven degrees of separation are sufficient for connecting any two people through e-mail (Watts 2003; Watts 2004).

2.1.3.5 Identity and Search in Social Networks

Based on the experiment on the small world phenomenon described in the previous section Watts et al. (2002) provide an interesting study about identity and search in social networks. The ability of ordinary people to be capable of directing messages through their network of acquaintances in order to reach a specific but distant target person in only a few steps is an important property of social networks. It is termed *searchability* by Watts et al (2002). According to the authors there has been much research on what types of networks allow searchability (e.g. Barabási and Albert 1999; Kleinberg 2000a; Adamic et al. 2001; Kim et al. 2002) but it does not provide a satisfactory model of society. To overcome this shortage they present a model to explain this phenomenon that is derived from six contentions about social networks:

- (1) Individuals in social networks have not only network ties but also social identities (White 1992). Here, *social identity* is defined as a set of characteristics which an individual attributes to himself and others.
- (2) Individuals break down, or cluster, the world hierarchically into a series of *layers*. The top layer accounts for the entire world and each successively deeper layer represents a cognitive division into a greater number of increasingly specific groups.
- (3) Besides the own set of characteristics *group membership* are employed to define and distinguish individual identity. Social identity is a primary basis for social interaction (see e.g. Nadel 1957; Breiger 1974), and therefore acquaintanceship.
- (4) Individuals hierarchically cluster the social world in more than one way, e.g. work relations, friendships etc. These categories or *dimensions* of their social world are independent, i.e. proximity in one does not imply proximity in another.
- (5) Based on their perceived proximity to other nodes, individuals construct a measure of *social distance* as the minimum ultrametric distance over all dimensions.
- (6) Individuals forward a message given only *local information* about the network.

In order to find a target person nodes in searchable networks therefore involve the shortest distances from all types of acquaintanceships, e.g. direct or indirect contacts in different social dimensions.

2.1.3.6 Scale-Free Networks

Based on the work of Erdős and Rényi (1959; 1960; 1961) social networks have been regarded as *random networks* where each link is placed randomly between nodes. Interestingly, as a result most nodes have approximately the same number of links and the nodes follow a Poisson distribution with a bell-shape. When studying citation networks Price (1965) found out that there was no equal distribution of links between nodes. Instead, the number of citations per paper followed a Pareto or power law distribution. Later, Price (1976) proposed the mechanism to explain the occurrence of such degree distributions which he called *cumulative advantage*. This mechanism has also been termed *preferential attachment* in later publications: New nodes that are added to the network attach preferentially to already well-connected nodes (Barabási and Albert 1999: 509). As a result, those networks consist of many nodes with only few links and few nodes with many links (“*hubs*”). They are called *scale-free networks*. Figure 2-4 shows the different properties of scale-free networks compared to random networks.

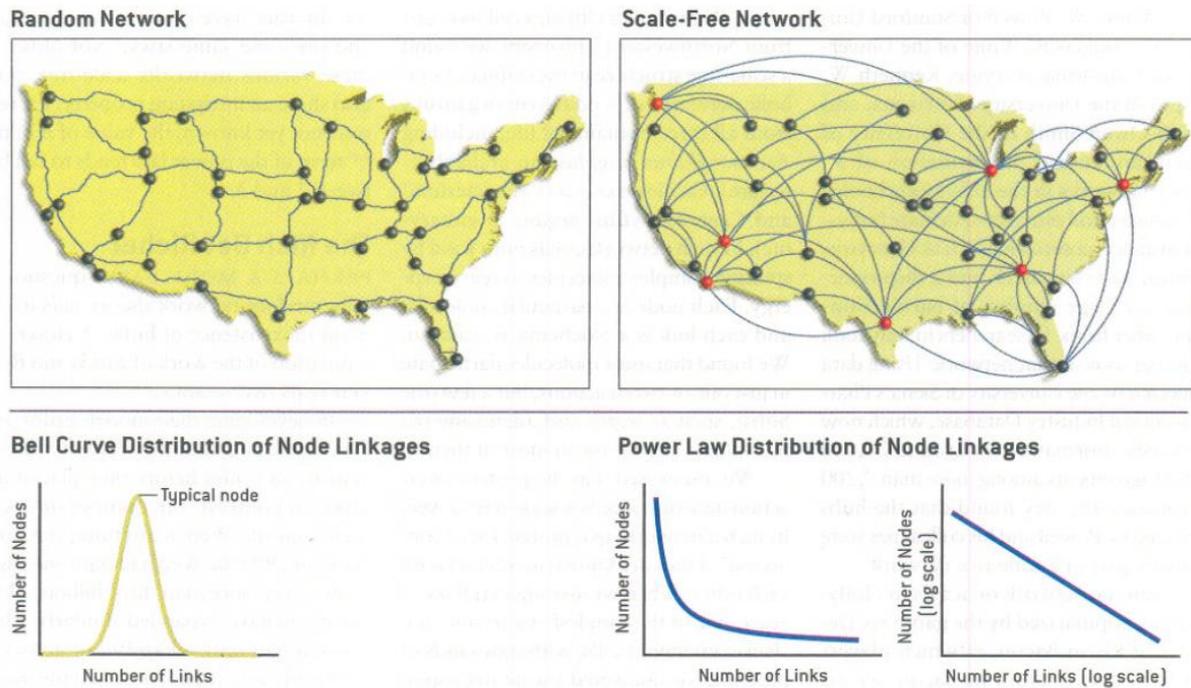


Figure 2-4: Random networks versus scale-free networks. Left: random network resembling U.S. highway system; right: scale-free network resembling the U.S. airline system. Source: Barabási and Bonabeau (2003: 53)

Due to the large number of weakly connected nodes scale-free networks are extremely robust against accidental failure (Barabási and Bonabeau 2003: 56). A random removal will mainly take out nodes with small degree and miss the hubs. Thus, in contrast to random networks where each node has the same share on the degree distribution the network topology is not significantly affected. However, scale-free networks are extremely vulnerable to directed attacks. Concentrating on the hubs efficiently destroys the network structure. Besides social networks there are quite a number of networks showing the same behavior. Some examples are given in Table 2-5.

Table 2-5: Examples of scale-free networks. Source: Barabási and Bonabeau (2003: 54)

Network	Nodes	Links
Cellular metabolism	Molecules involved in burning food for energy	Participation in the same biochemical reaction
Hollywood	Actors	Appearance in the same movie
Internet	Routers	Optical and other physical connections
Protein regulatory network	Proteins that help to regulate a cell's activities	Interactions among proteins
Research collaborations	Scientists	Co-authorship of papers
Sexual relationships	People	Sexual contact
World Wide Web	Web pages	URLs

2.1.3.7 Social Capital and Structural Holes

Monge and Contractor (2003) provide a review of social theories why actors create, maintain, dissolve, and reconstitute relations. An overview of these theories is given in Figure 2-5.

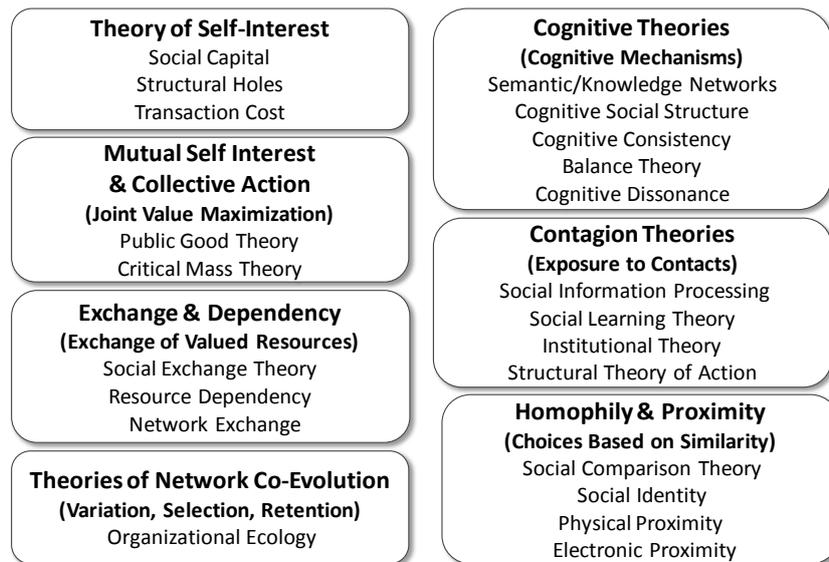


Figure 2-5: Overview of social theories. Source: Monge and Contractor (2003)

The concepts of social capital and structural holes belong to the theory of self-interest. Burt (1992) compares different definitions of *social capital*, e.g. by Coleman (1990), Bourdieu and Wacquant (1992), Burt (1992), and Putman (1993). For instance, according to Putnam (1993: 167) social capital refers to features of organizations, such as trust, norms and networks, that can improve efficiency of society by facilitating coordinated actions. Although the definitions differ in origin and style they provide some general agreement: Generally speaking, social capital is a metaphor about *competitive advantage* (Burt 1992: 8, 45). As a result, social structure is a kind of capital that can create for certain individuals or groups competitive advantage in pursuing their interests. Consequently, better connected people enjoy higher returns.

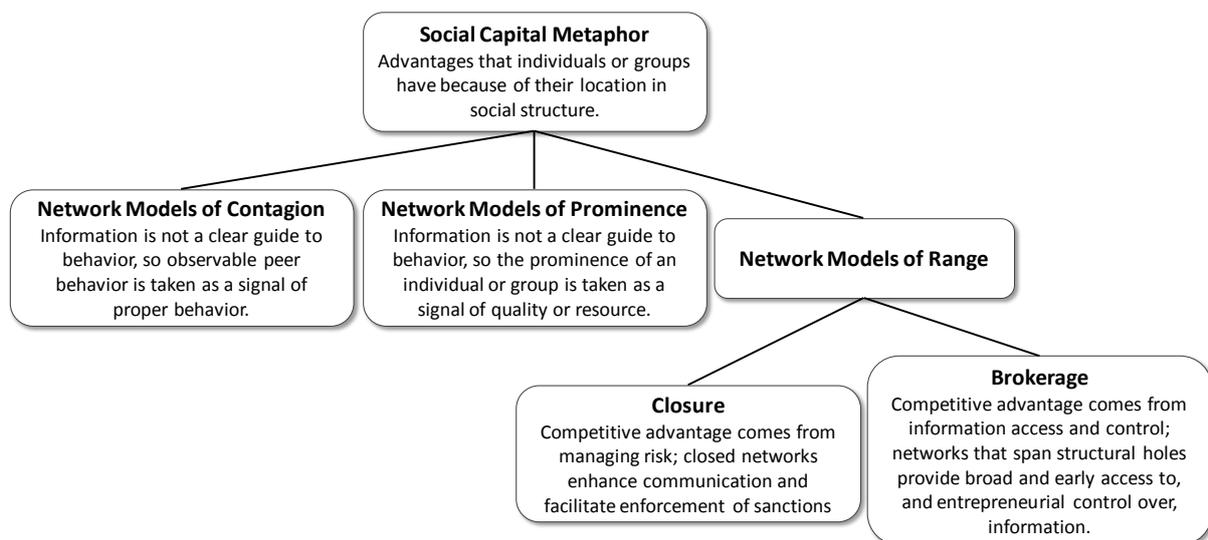


Figure 2-6: Social capital, in metaphor and network structure. Source: Burt (2002)

Social capital facilitates co-operation and mutually supportive relations in communities. This results into increased personal access to information and skill sets and enhanced power. According to Burt (2002) there are four network mechanisms that define social capital: contagion, prominence, closure and brokerage across structural holes. Structural holes are

missing links between nodes (see section 2.1.4.2, Figure 2-13). The amount and distribution of these holes can be used to explain positional advantages or disadvantages of individuals stemming from their embeddedness in their network neighborhoods (Hanneman and Riddle 2005). The idea is that an actor whose alteri lack ties among each other (i.e. where structural holes exists between the alteri) benefits from higher levels of autonomy, control, and information. Structural holes are one important approach in the theory of social capital.

There are three categories of empirical evidence for structural holes (Burt 2002):

- Evidence of reward and achievement associated with brokerage.
- Evidence of creativity and learning associated with brokerage.
- Evidence on the process of bridging structural holes.

There are several approaches how to measure social capital as the value of an actor within its social context. Three very popular measures can be found in Freeman (1977): degree centrality, closeness centrality and betweenness centrality (see also section 2.1.4.2 for more details). *Degree centrality* is a measure of activity (out-degree) and popularity (in-degree). *Closeness centrality* is a measure of efficiency to access information and resources and independence from others to gain a resource or information. *Betweenness centrality* is a measure of influence by controlling the information flow and network resilience, i.e. how ineffective the network gets when the node is removed. Similarly, individuals can exercise influence or act as *brokers* within their social networks by bridging two networks that are not directly linked (called filling *structural holes*) (Scott 1991). Gould and Fernandez (1989) investigate on the different *brokerage roles* of a node. Nodes that lie on paths between other nodes can act as broker in various ways: as *coordinators*, *gatekeepers*, *representatives*, *itinerant brokers* (also called *consultants* (Hanneman and Riddle 2005)), or *liaisons*. One person can occupy more than just one of these brokerage roles at the same time (Aalbers et al. 2004). Also, different actors can concurrently assume the same role at varying degrees.

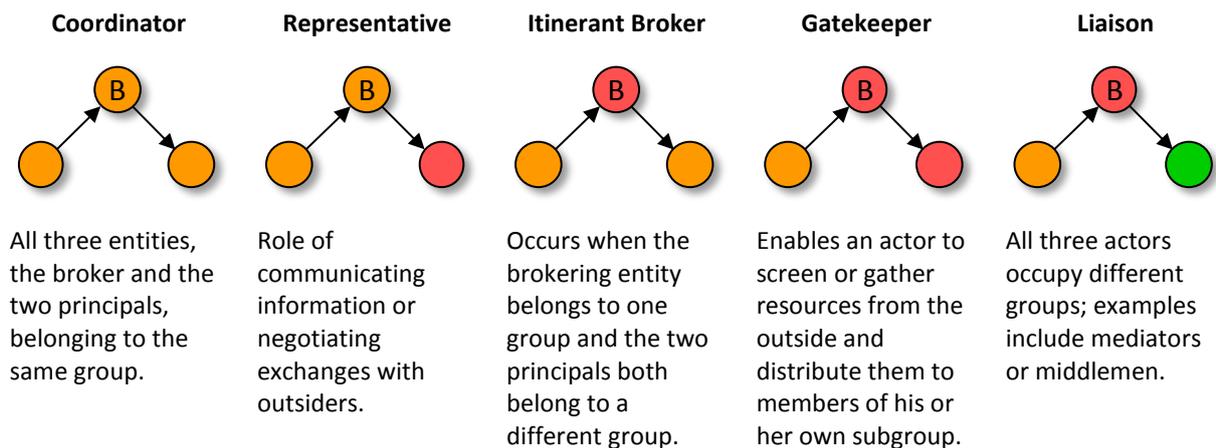


Figure 2-7: Measuring social capital: brokerage roles. Node colors represent different organizational departments. Based on: Gould and Fernandez (1989)

Täube (2004) more generally considers brokers to be actors that enable or accelerate the flow of resources among actors that are otherwise unconnected or only weakly connected. They also gain advantages from their strategic position in the network. Trier and Bobrik (2007a) provide another study of brokerage based on network resilience inventing the *brokering activity* measure. Some of these concepts are further described in section 2.1.4.2.

In summary, one actor can assume multiple of these roles at the same time, depending on the perspective and the scope of the network (e.g. whole network versus group analysis). However, all these node roles are based on a structural or positional analysis and do not take the content of the relationships into account.

2.1.3.8 Dynamic Network Analysis and Visualization

Social network analysis using graph representations and graph theoretic analysis has a long history. The social network graph was first introduced by Moreno (1934). Using this sociogram actors are represented as nodes and their relationships as links between the nodes (see section 2.1.2.1). However, conventional SNA provides a wide range of methods and metrics for comparative network analysis but do no sufficient means for advanced analysis and visualization of network dynamics. Today, there are several approaches to dynamic network analysis that try to extend conventional SNA, e.g. by comparing discrete „waves“ of network data (e.g. Freeman 2000), time-window analysis (e.g. Moody et al. 2005; Trier et al. 2007; Trier 2008), theoretical mathematical models to explain and find properties at the general network level (e.g. Watts and Strogatz 1998; Albert and Barabási 2002), stochastic actor-driven models to predict the explanatory power of known behavioral properties like preferential attachment or the small world phenomenon (e.g. Snijders 2001; Snijders 2004), statistical simulation models to estimate complex network development based on rules of changes (e.g. Carley 2003), or agent-based simulation of networking decisions (e.g. Carley 2002).

In the domain of dynamic network analysis, the representation of change in the graph is a fundamental issue. It requires algorithms for handling transitions between incremental network states in order to represent structural change. According to Trier (2008) this major aspect of dynamic visualization can be termed *transition problem*. There are three contemporary approaches that deal with this problem and are therefore related to the event-driven approach for dynamic network analysis presented in section 2.2.1.3. Perer and Shneiderman (2006) introduced an approach that includes some functions to trace changes in network data by hiding links outside a selected moveable time window. Nodes maintain a fixed position based on the final network configuration. This approach is a static technique that reveals how a network structure unfolds over time based on interactions. It has been termed *flipbook* by Moody et al. (2005: 1234). However, the lack of dynamic repositioning of nodes yields interim networks with uninformative layouts. For example, nodes with an early but weak relationship would eventually be placed far apart. However, they would better be positioned near each other early in the sequence of subnetworks and then move apart to slowly give room for later but stronger relationships among other nodes. It is hence less suitable for recognizing cluster formation or sudden changes in actor's network positions.

Two more advanced approaches of dynamic network visualization that try to handle this problem can be found in Moody et al. (2005) and Gloor et al. (2004). Both try to represent structural change as motions. Their work is based on a *sliding time window* that is moved through the overall sample period (see Figure 2-20 in section 2.2.1.3). For each of these time windows, or time frames, a new subnetwork is retrieved from the data containing all nodes and links that are active within the selected period of time. Structural changes can then be detected by comparing the nodes' positions in two subsequent subnetworks. To create a

consistent transition, the authors render interim frames. The visualization is then “gradually adjusting node coordinates and adding or deleting nodes and arcs“ (Moody et al. 2005) or “the animation of the changing layout is interpolated between [...] keyframes” (Gloor et al. 2004). That means that positional transitions are computed between subsequent visualization frames to provide visual consistency for the changing node locations. Thus, both approaches calculate subnetworks at different states (e.g. per day) and then linearly move nodes from their position in the network of the first time window to their position in the subsequent time window. In order to describe the observed dynamic motion Moody et al. (2005) introduce new process terms, e.g. *ritual*, *dance*, *pulse*, *repeated ritual behavior*, or moments of *order* and *chaos*. Both approaches have the disadvantage of rendering transition frames and the relocation of nodes disturbs the impression of organic evolution of network structures. Nodes cross other nodes, swap their position without need, or move at unintuitive changing speeds or in quickly changing directions across the screen. As a result, the user gains only a suboptimal impression of transitions between separate layouts instead of observing network behavior based on single events and their impact on the remaining structure. The SNI data model and the event-driven dynamic network analysis try to overcome this problem (see section 2.2.1).

Since the beginning of social network analysis using graph representations and graph theoretic analysis, powerful software tools for semi-automated analysis and visualization of large network structures have been developed (see Freeman 2000). Until recently, these visualizations simply compared static graphs of cumulative networks states at different times. Examples of current analytical software packages are Ucinet (Borgatti et al. 1992) or Pajek (Batagelj and Mrvar 1998). They usually import formatted data files from external sources and provide features for sophisticated statistical analysis and graph visualization (see section 2.3.1). However, these tools use the aggregated data as a single static network and do not automatically capture, evaluate or animate dynamic data and events from communication sources. Pajek recently introduced means to compute partial networks within certain time frames but this approach is still limited and does only extend the analysis to several smaller static subnetworks without providing means to analyze their interrelation.

A related strand of research, not directly focused on the quantitative analysis of relationship structures, developed rich and animated representations of online social spaces of electronic communication. They suggest various intuitive metaphors to represent online social activity, e.g. graphical tree-like hierarchies of postings (e.g. Smith and Fiore 2001), a garden with flower petals, or a tree with leaves to convey the ‘health’ of the electronic group (e.g. Girgensohn et al. 2003). This has also resulted in the formulation of the concept of *social translucency*, as “an approach to designing digital systems that emphasizes making social information visible within the system” (Erickson and Kellogg 2000). This family of approaches was the first to employ motion for insightful and “living” virtual representations of changes in the conversation. However, these concepts are mainly designed to aid the user in visually navigating online spaces. They do not explicitly focus on relationships or allow for a quantitative analysis of the displayed dynamic structures.

Software libraries for dynamic graph drawing have been recently introduced. One example is Graphviz of AT&T Labs Research (Ellson et al. 2004). It is an open source graph visualization package for viewing and manipulating abstract graphs in the field of software engineering, networking, databases, knowledge representation, and bioinformatics. The focus

is on interactive editors for general graph drawing with applicable technical layout concepts and the Dynagraph software libraries to dynamically update a graph view. However, the libraries include no network analytical approach or perspective and are not focused on social networks.

In section 2.3.2 the SNA software Commetrix is presented which allows enhanced visualization and analysis of network dynamics.

2.1.4 Metrics

This section presents a selection of the most important indices and general network attributes in social network analysis. It is not limited to calculable measures, but it also contains common terms and labels, network roles, and typical network configurations. They can all be considered as metrics in the broader sense and are adequate for network analysis and characterization. They are classified on network elements into three categories: metrics on network level (section 2.1.4.1), node level (section 2.1.4.2), and link level (section 2.1.4.3). However, some concepts or metrics refer to more than one category, e.g. the concept of core and periphery describes a phenomenon on network level as well as on node level. As the node metrics derived from it are based on the overall network topology, it is included in section 2.1.4.1.

All network metrics can be considered on group level including only group members as nodes and only those links between group members (intra-cluster links). Additionally, node metrics and link metrics can also be calculated on a reduced data set and will provide additional insight into this network structure.

2.1.4.1 Network Metrics

The set of network metrics presented in this section are basic and more advanced metrics that describe the complete structure of a network. They are the basis of other metrics, e.g. node centrality metrics based on the definition of the shortest path.

The *size* of a network refers to the number of actors it contains. Size is sometimes also called *volume*. However, this is an ambiguous term since volume can also refer to the total number of links or to the overall number of messages, events or resources in the network that aggregate to links between nodes (Trier 2005a: 169).

Besides the size of the network, the *sum*, *average* as well as *minimum* and *maximum* values of network properties (e.g. number of nodes, number of links, first and last participation, etc.) and metrics (e.g. average number of link per node, average degree centrality, average path length between all nodes, etc.) can yield important insights of their distribution in the network and help to describe the general network structure. To interpret these metrics depends strongly on the data, the purpose of the analysis and the experience of the researcher.

The *density* of a network is one of the most commonly used metrics in social network analysis. It is defined as the quotient of the existing links and the maximum number of possible links in a network (Wasserman and Faust 1994: 164). If n denotes the number of nodes, the maximum number of links is given by $[n * (n - 1)]/2$ in an undirected graph and by $n * (n - 1)$ in a directed graph. Density is an index for *structural compactness* and *cohesion*: The denser a network, the more links exist between the nodes and thus it is more

likely that the nodes have a direct connection to each other. In a directed graph, density and cohesion have a slightly different meaning, as *cohesion* only takes symmetric links into account (Knoke and Kuklinski 1983: 50-51).

Usually, the links in the network are not equally distributed among all nodes (see section 2.1.3.6). Often a well-connected *core* (sometimes also called *nucleus*) and a sparsely linked *periphery* can be distinguished. The identification and examination of the core and periphery of a network will often yield important insights and potential points of reference for further investigations. In Figure 2-8 an illustration of such a core/periphery structure can be found.

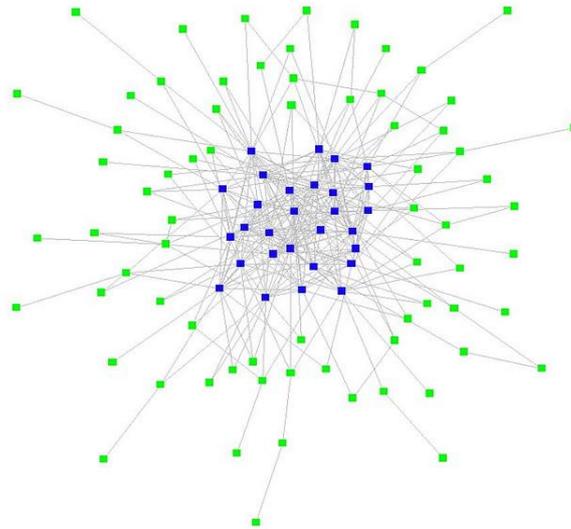


Figure 2-8: Network with core/periphery structure. Source: Krebs and Holley (2007)

Core and periphery are both not well defined (see section 2.1.3.8). The core can be identified by examining the level of the activity, e.g. by measuring the number of messages sent or resources shared. Trier (2005a: 167) defines the core as the group of actors who are responsible for at least 80% of this network “traffic”. According to Scott (1991: 92-97), the core is the center of the network, which comprises the actors with the highest values of degree centrality (see section 2.1.4.2). In Everett and Borgatti (1999) the concept of a cohesive subgroup is defined as the core of a network. The authors consider the periphery simply to be the group of nodes that are not part of the core. In contrast, Scott (1991: 94) defines the periphery as the set of network members that have considerably lower centrality measures as the margin of the core whose members are noticeably less central than the core itself.

Davis et al. (1941: 150) define three levels of core/periphery participation: *core members* participate together most often in the group, *primary members* participate frequently with core members but do not form a group themselves, and *secondary members* participate only infrequently. Similarly, Everett and Borgatti (1999: 406) distinguish between the core, the periphery of the core, which is still part of it, and the rest of the nodes belonging to the network. The distance of a node from the core is called *coreness* which is the ratio of already present links to the number of links that are needed to make a node member of the core (Borgatti and Everett 1999; Everett and Borgatti 1999).

On network level, the *core group size* (the number of nodes in the core) and *core group share* (the size of the core in relation to the overall number of nodes) are two valuable indices for the analysis of a network structure.

There are a number of network metrics as well as node metrics based on different concepts of routes through the network (see Figure 2-9). In general, routes start and end at nodes and connect intermediate nodes with links (1994: 105-107). When examining these concepts it is essential to distinguish between directed and undirected graphs. In a directed graph one node may be reachable for another node, but not vice versa (see section 2.1.2.1).

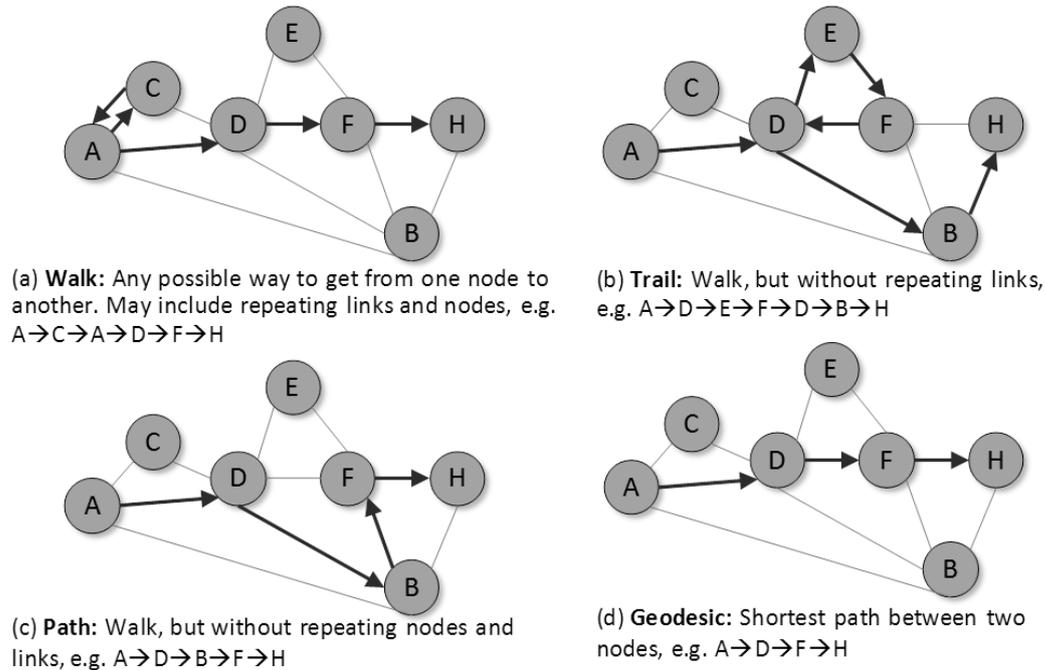


Figure 2-9: Routes through the network: (a) walk; (b) trail; (c) path; (d) geodesic

A *walk* is a sequence of incident nodes and links with no further constraints (see Figure 2-9 (a)). Nodes and links may be used several times. A *trail* is a walk in which all links are distinct (see Figure 2-9 (b)). Nodes may be used several times. A *path* is a walk in which all links and all nodes are distinct (see Figure 2-9 (c)). Hence, each path is a trail, and each trail is a walk. The *length* of a walk, trail, or path is the number of included links. A *geodesic* is the shortest path between two nodes (see Figure 2-9 (d)). The length of a geodesic is called the *distance*, or *geodesic distance*. The *diameter* specifies the length of the shortest path between the two furthestmost nodes in the graph. It can be identified by determining the node with the largest *eccentricity*, also referred to as *association number* (Wasserman and Faust 1994: 111-112): A node's eccentricity is the largest geodesic distance between that and any other node in the network. According to Milgram (1967) the concept of eccentricity is also called *degrees of separation* (see section 2.1.3.4). A small value indicate that the resources are transmitted in a timely and accurately way through the network (Cross et al. 2000). If the network is not fully connected but consist of two or more unconnected components only connected pairs of nodes should be taken into account. A modified version of this metric is called *average distance-weighted reach*, or simply *reach*, which is based on reciprocal geodesics (Schilling and Phelps 2007). *Reachability* is a mutual property of two nodes and a dichotomous index: either a node is reachable by another one or it is not.

Connectivity can be calculated on node and network level. On node level, one can distinguish node-connectivity and link-connectivity. *Node-connectivity* is calculated with regard to the number of nodes that need to be removed from the network in order to make two nodes

unreachable (Hanneman and Riddle 2005). Similarly, *link-connectivity* is defined on the number of links that have to be removed. Calculating the average connectivity of one node with all other nodes a high value indicates strong interconnectedness and independence from single links or paths to gain access to network resources. A network is *connected* if there is a path between every pair of nodes; otherwise, it is *disconnected* (Wasserman and Faust 1994: 109). Every graph with at least one isolated node is disconnected. A connected graph consists of only one *component*. A component is a subgraph in which all pairs of nodes are reachable and there is no path between a node of this component and a second node that is not a part of the component. A graph with more than one component is disconnected. The concept of connectedness of graphs can be extended to the minimum number of nodes or links that have to be removed to make it disconnected. A *k-node connected* graph would require the removal of k nodes. If only one node is necessary, this node is referred to as a *cutpoint*. For $k > 1$ the collection of nodes that have to be taken out is called a *cutset*. Similarly, a graph is *l-line connected* if l links need to be removed in order to increase the number of components in the graph. The equivalent of a cutpoint is called a *bridge*, and the corresponding concept to a cutset is an *l-line cut* (Wasserman and Faust 1994: 109-117). Figure 2-10 illustrates the concept of a cutpoint and a bridge. Here, the network will dissociate into two disconnected components if node D or the link between node D and node E is removed.

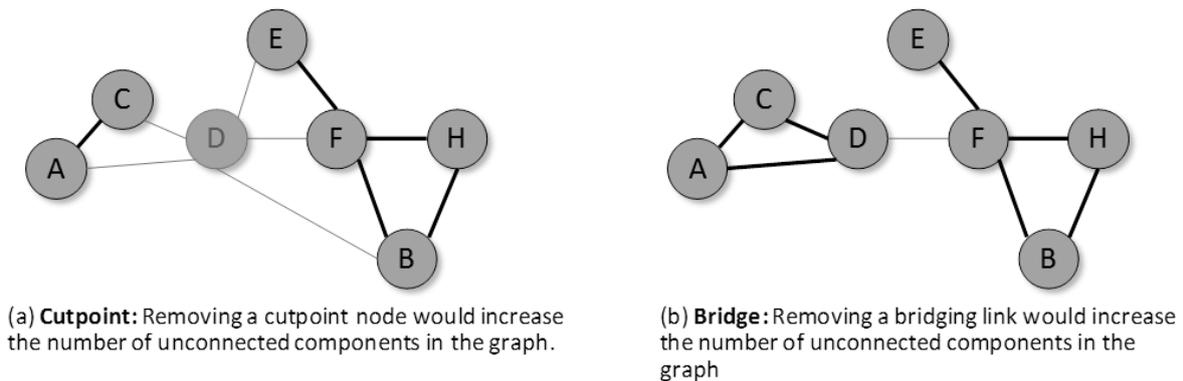


Figure 2-10: Concepts of connectivity: (a) cutpoint; (b) bridge

In section 2.1.4.2 centrality measures on node level are explained. In order to compare whole network structures with regard to their centrality the *centralization* can be calculated on network level. It is a measure of the heterogeneity of the individual node centrality values. It therefore measures the variability and dispersion of the network (Wasserman and Faust 1994: 176). If the index is high it is more likely that a single actor is very central and dominates all others. The maximum value of centralization is 1, which is the case for the star (see Figure 2-2 (a)). If the centralization index is zero all nodes have exactly the same centrality value (as in the circle in Figure 2-2 (c)). Freeman (1979) developed a general definition for centralization, which can be calculated from all centrality measures (see Table 2-6). Thus, there is not a single centralization index, but only a network-wide centralization measure based on one of the centrality metrics.

On network level, the *clustering coefficient* can be defined as the average of the local clustering coefficients of all the nodes in the network (Watts and Strogatz 1998). It is a measure of the degree to which nodes in a graph tend to cluster together (*triadic closure*, see section 2.1.4.2). A graph is considered a small world graph (see 2.1.3.4) if its *network*

average clustering coefficient is significantly higher than in a random graph with the same number of nodes and approximately the same average shortest path length. Based on the definition of the local clustering coefficient on node level (see section 2.1.4.2), the *global clustering coefficient* can also be directly measured as the relation of triangles and triples in the network. Formally, it is defined as

$$C = 3 \times \text{number of triangles} / \text{number of connected triples}.$$

This measure can be applied to both undirected and directed networks. In Wasserman and Faust (1994: 243) it is also called *transitivity*.

2.1.4.2 Node Metrics

There are several approaches how to measure the social capital (see section 2.1.3.7) of an actor expressing his value within his social context. These approaches try to describe quantitative node properties based on their position in the network. They help to identify *key players*, e.g. formal or informal leaders or important knowledge worker. The set of node metrics presented in this section are basic and more advanced metrics that describe the structural position of a node in the network. To interpret these metrics depends strongly on the data, the purpose of the analysis and the experience of the researcher.

The *degree* of a node is the number of adjacent nodes, or from an ego-centered perspective the number of alteri. It can be expressed as an absolute value or in relation to the maximum possible number of adjacent nodes in the network (see degree centrality). Degree is a measure for the *activity* and *visibility* of a node (Wasserman and Faust 1994: 179). Actors with high degrees are most likely to influence and be influenced directly. Actors with low levels of degree are often in the periphery of the network and therefore less influential. *Isolates* are unconnected nodes and thus have a zero degree.

In directed graphs, it is necessary to distinguish between in-degree and out-degree. *In-degree* is the number of actors that choose one node (links pointing towards it), and *out-degree* is the number of actors that are chosen by one node (links that point away from the actor). In-degree is a metric expressing an actor's *prestige*. *Status* is another common name for prestige (Wasserman and Faust 1994: 175). Another derived measure is that of *uniformity*: It is simply the variance of degrees in the network. Total uniformity exists if all actors have the same degree. In that case the network is considered to be *d-regular* (Wasserman and Faust 1994: 101).

On node level, centrality measures are among the most prominent and extensively employed metrics in social network analysis. They describe quantitative node properties based on their position in the network and help to identify key players due to the structure of the network. The concept of centrality can be interpreted in a number of ways, e.g. as a proxy for *brokerage*, *control*, *influence*, *power*, *stress*, *independence*, or *autonomy* (Borgatti and Everett 2006). There is a wide variety of centrality measures trying to formalize these notions, but they all seek to answer the question of who is the most important (or central) actor in a network (Newman 2008).

Importance, *power*, *prominence*, *prestige*, and *popularity* are all tightly connected to centrality measures and often mentioned in this context. None of them are accepted metrics

but rather interpretations and circumscriptions for other measures. Moreover, only the latter two terms are properly defined and connected to a particular index. *Prestige* and *popularity* are the level of *deference* towards a certain actor, i.e. the extent to which an actor is chosen by others (Wasserman and Faust 1994: 174-175). They are related to the in-degree of a node. Among the different notions of centrality degree centrality, betweenness centrality and closeness centrality by Freeman (1979) are best known and most frequently used in network studies. They can be calculated as absolute measures or relative measures ranging from 0 to 1.

Degree centrality indicates the *activity*, *visibility* and *popularity* of an actor in the network. It is calculated from the degree of an actor in relation to the maximum possible number of adjacent nodes in the network. Actors with high values are most likely to influence and be influenced directly. If calculated for a directed graph, degree centrality based on the in-degree is a measure of *popularity* and *prestige*, whereas based on the out-degree it is measure of *activity* and *integration*.

Closeness centrality is a measure of *efficiency* as actors with high values can reach more nodes spreading information and have relatively easy and fast access to network resources and information (Wasserman and Faust 1994: 183-184). Additionally, it is a measure of *independence* as actors with high values are more independent from others to gain a resource or information. According to Freeman (1979), closeness is calculated as the reciprocal of the sum of geodesic distances from one node to all other nodes. Therefore, it takes the number of direct and indirect contacts into account as well as the distance to these contacts. Closeness centrality is related to reach on network level. Unfortunately, it is not defined for unconnected graphs. Newman (2008) therefore suggests that closeness centrality should only consider distances to reachable nodes when applied to disconnected graphs.

Betweenness centrality takes a different approach to measure the prominence of nodes in the network. It is defined as the fraction of geodesics between a pair of nodes that go through the actor in question. If an actor lies on a lot of shortest paths between any two nodes, it has a high betweenness centrality. It is a measure of *influence*: Actors with high betweenness usually have great control over communication. They can broker and synthesize information. On the other hand, actors in such a central role also have a high level of *responsibility* or stress because almost all communication flow is dependent on them (Shimbel 1953). Betweenness centrality is also a measure of *network resilience* when asking how many geodesic paths will get longer when a node is removed from the network. However, this index has some weaknesses: First, all paths of the same length are considered to be equal, i.e. the type or strength of the relations as well as the properties of the nodes involved in the path are neglected. Yet it is more likely that a path with nodes of high degree is chosen over a path with less connected nodes (Wasserman and Faust 1994: 193). Furthermore, it is assumed that information and other resources that flow through the network always use the shortest path. This is likewise not realistic since it is possible that actors deliberately choose longer paths in order to circumvent certain actors and hide information from them (Stephenson and Zelen 1989).

In addition to Freeman's centrality measures Bonacich and Lloyd (2001) propose a fourth centrality measure called *eigenvector centrality* which is basically an advanced degree centrality metric. This centrality metric takes into account that nodes with the same degree are

not necessarily equally important. The concept of the eigenvector centrality and its comparison to degree, closeness and betweenness centrality is illustrated in Figure 2-11. Within the graph the four nodes with highest centrality values are highlighted with corresponding labels: (D) degree, (C) closeness, (B) betweenness, and (E) eigenvector centrality. Nodes *D* and *E* have both a degree of six. Yet, the ego-centered network of node *E* is better connected than that of node *D*. Eigenvector centrality takes these indirect relationships into account and recursively calculates the most central nodes based on the quality off their acquaintances.

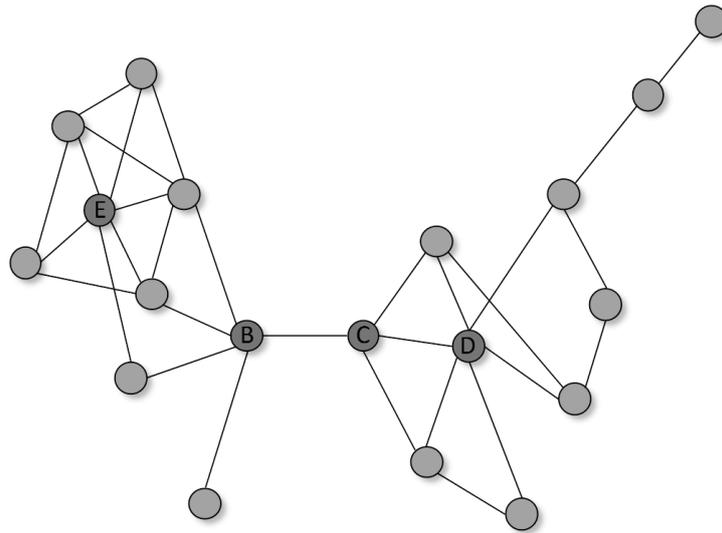


Figure 2-11: Comparison of centrality metrics. Graph showing nodes with highest degree (D), closeness (C), betweenness (B), and eigenvector (E) centrality. Source: Borgatti (2002)

Based on the concept of betweenness centrality Trier and Bobrik (Trier and Bobrik 2007a) propose a new centrality measure called *brokering activity*. It is especially designed for understanding network structures as well as dynamic community formation. In general, it is based on the idea of network resilience: By removing a node from the network it is calculated in how far the network dissociates, i.e. how many paths between pairs of nodes get longer or even destroyed. Based on geodesic distances it is related to Freeman's betweenness centrality. A comparison of betweenness centrality and brokering activity on a subsample of the Enron data set³ is provided in Figure 2-12. The five nodes with highest betweenness centrality and brokering activity values are highlighted. Obviously, there is some overlap but the nodes identified by brokering activity are more distributed through the network and integrate loosely connected parts of the network. Thus, brokering activity is more independent from the concept of degree centrality and adds an additional perspective to the analysis.

³ See chapter 5.3 Case Study “Corporate E-Mail Exchange”

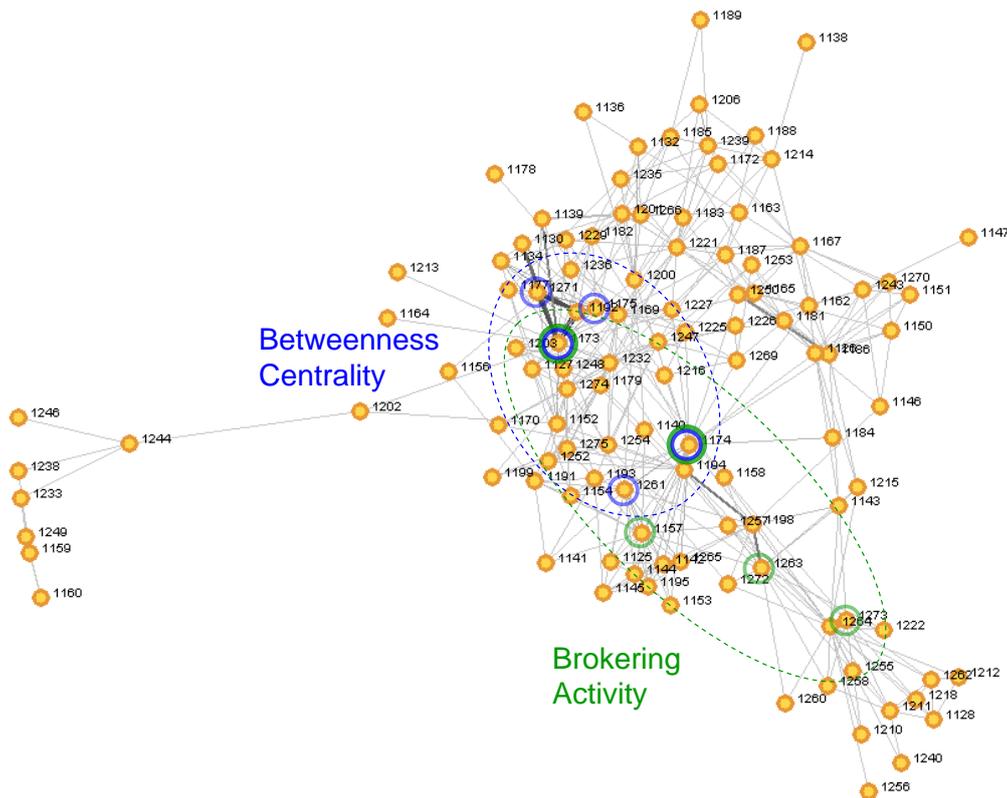


Figure 2-12: Comparison of centrality metrics: Betweenness centrality versus brokering activity

There are several other centrality metrics which can be related to either degree, betweenness, closeness or eigenvector centrality. A review of these metrics can be found in Wasserman and Faust (Wasserman and Faust 1994), Borgatti (2005) and Hanneman and Riddle (2005). Table 2-6 provides a brief overview.

Table 2-6: Overview of centrality metrics

Metric	Class	Proposed by
Influence/Status	Degree	Katz (1953), Hubbel (1965), Hoede (1978)
Total effects centrality	Degree	Friedkin (1991)
Immediate effects centrality	Closeness	Friedkin (1991)
Information centrality	Closeness	Stephenson and Zelen (1989)
Flow betweenness	Betweenness	Freeman et al. (1991)
Brokering activity	Betweenness	Trier and Bobrik (2007a)
Power	Eigenvector	Coleman (1973)
Prestige	Eigenvector	Burt (1982)
Alpha centrality	Eigenvector	Bonacich and Lloyd (2001)

The concept of centrality on network level, called centralization, is described in section 2.1.4.1.

Triadic closure is a concept in social network theory invented by Simmel (1908/1950). For instance, there are three nodes (*triad*), *A*, *B*, and *C*, and there is a link between *A* and *B* as

well as B and C (*triple* $A - B - C$). If triadic closure occurs there is also a link between A and C . Thus all three nodes form a *triangle*. Granovetter (1973) combines this concept with the theory of cognitive balance first introduced by Heider (1946). *Cognitive balance* refers to the propensity of two individuals to want to feel the same way about an object. Thus, if the triad of three individuals is not closed the person connected to both of the individuals (*focal node*) will seek to close this triad in order to achieve closure in his relationship network. A formalization of the concept of triadic closure is the clustering coefficient. On node level, the *clustering coefficient*, or *local clustering coefficient*, is the relation of all triples and triangles in a network where the focal node is involved. That is, this measure counts how many triples form triangles as well. Depending on the focal node each triangle represents three types of triples. Therefore it is counted three times. This metric can be considered as a measure of an ego's density and, as such, can reveal emergent groups or cliques. On network level, all triples and triangles in the network are taken into account and the ratio is normalized by the number of nodes with a degree > 1 (see global clustering coefficient in section 2.1.4.1).

The *E/I-Ratio*, or E-I-Index, measures the relative density of internal links within a group compared to its external links (Krackhardt and Stern 1988). It can only be computed if the network has already been divided into subgroups. Normalized with the overall number of links the value ranges from -1 (all links are internal links) to $+1$ (all links are external links). The E/I-Ratio can be calculated on the group level, but also for each individual or the whole network (Hanneman and Riddle 2005). The measure provides an insight into how open or closed the culture of one particular group or the average group is, or how embedded individuals are in their groups.

The *pulsetaker* measure was defined by Trier (2005a: 168) as the relation between an actor's indirect contacts to its direct contacts. Individuals with high pulsetaker values have the advantage of still having access to a lot of information even if they spend little time on maintaining their direct relationships. On the other hand, with rising pulsetaker values actors become more dependent on their direct contacts and more vulnerable to be separated from the network if their direct contacts fail.

Based on the concept of *social capital* (see Monge and Contractor (2003) and section 2.1.3.7), Burt (1992) proposes a set of indices to identify *structural holes*. His two primary measures for structural holes are effective size and constraint. *Effective size* is calculated for one node and specifies the extent of its actual reach into the network. It is the number of adjacent nodes minus a redundancy factor, which is higher the more links they share among themselves. Therefore, an actor who is embedded in a tightly connected subgroup has a small effective size because the other group members are well connected and therefore redundant to the ego. As a relative measure, the *efficiency* is the effective size divided by the actual size of the ego's neighborhood. High values indicate that the actor holds some kind of brokerage role and has contacts to different regions in the network, which increases the potential control and information benefits (Borgatti et al. 1998). *Constraint* is the extent to which all of an actor's relationships directly or indirectly involve a single contact (Borgatti et al. 1998). The higher the constraint value the fewer alternatives an actor has to reach other actors and take action. The measure varies with the size, density and hierarchy of the network (Burt 2001). *Hierarchy* is another measure proposed by Burt (2001) to characterize the network structure.

The value for constraint is high if the network is very dense or if all contacts are indirectly connected through one focal actor. The latter case is a hierarchical network in which the hierarchy measure assumes a high value. The constraint emphasizes that an actor maintaining a high number of relationships can actually lose freedom of action instead of gaining more power from the multitude of relationships (Hanneman and Riddle 2005). The following graph in Figure 2-13 illustrates some of the metrics of structural holes. The network highly depends on the central node *A* and therefore has a high hierarchy value. Consequently, most nodes are highly constraint; node *B* to a larger extent than node *C*. Node *C* has much more direct contacts to others that are not connected among them. Thus, its effective size is higher than that of node *B*.

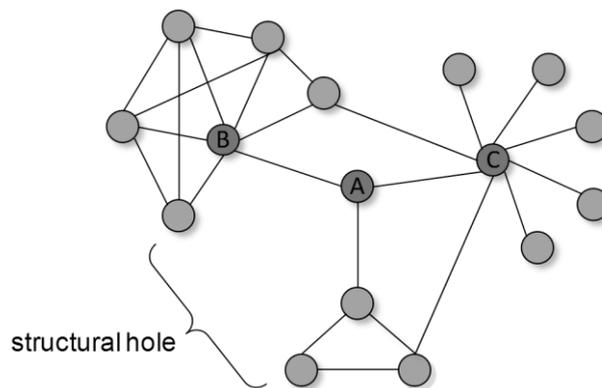


Figure 2-13: Structural holes and related metrics

The concept of cutpoints and bridges is related to the idea of structural holes. It is explained in section 2.1.4.1 together with network connectivity.

2.1.4.3 Link Metrics

Most network analyses concentrate on the nodes and their position in the network. Thus, besides some basic metrics, such as link strength in weighted networks, there only exist a small number of advanced metrics or concepts on link level.

The analysis of small substructures such as dyads (two nodes) and triads (three nodes) often involves the examination of *reciprocity*, *mutuality*, or *transitivity* of links. Although being definitions for special link configurations, these concepts can be used to construct measures, e.g. triadic closure (see section 2.1.4.2). *Reciprocity* and *mutuality* describe the same concept and can only be observed in directed networks. A mutual link between two nodes exists if both have a relationship to each other, i.e. if a link is reciprocated by the other node (Wasserman and Faust 1994: 510-511). Reciprocity can be an important measure of the extent of free-riding and consequentially of the health of a voluntary group (Cross et al. 2006: 53). *Transitivity* can only occur in a triad. If actor *A* has a connection to *B* and actor *B* has a connection to *C*, a relationship from *A* to *C* would result in a transitive triad (Hanneman and Riddle 2005). The links can thereby either be directed or undirected. According to the SNI data model (see section 2.2.1) reciprocity can be defined as the relation between the outgoing events of both actors that constitute the link. A similar definition can be found in Trier (2005a: 169).

Besides merely node-based metrics, current research also concentrates on the identification of significant patterns in complex social networks based on link analysis. Complex networks are studied across many fields of science. To uncover their structural design principles, Milo et al. (2002; 2004) define *network motifs* as patterns of interconnection that occur in real networks more frequently than in randomized networks with the same number of incoming and outgoing links.

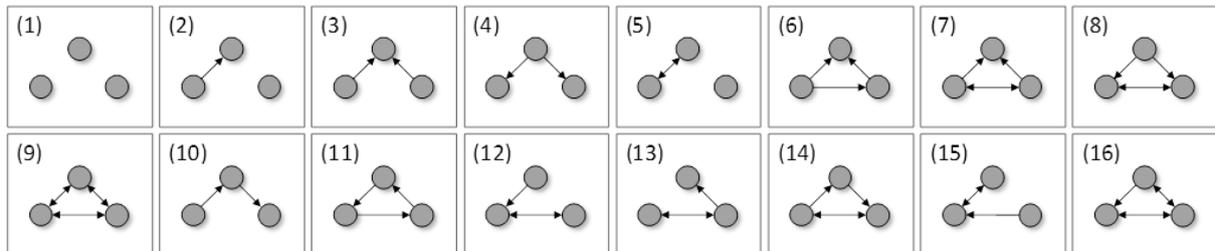


Figure 2-14: The sixteen possible triads in a directed graph. Source: Wasserman and Faust (1994: 244)

Network motifs may be based on two (dyad), three (triad), or more nodes. The triad census of a given network consists of 16 values for the number of appearances of each of the possible three-node subgraphs in directed networks (see Figure 2-14). The four-node census of a given non-directed networks consists of 199 values for the number of appearances of each of the possible four-node subgraphs. Thus, current research on network motifs mainly concentrates on triads. Milo et al. (2002; 2004) compare the appearances of motifs in different types of networks, e.g. networks from biochemistry, neurobiology, ecology, and engineering. For example, the motifs shared by ecological food webs are distinct from the motifs shared by genetic networks or from those found in the World Wide Web. However, similar motifs are found in networks that perform information processing. Motifs may thus define universal classes of networks. According to the authors, their approach allows to uncover the basic building blocks of most networks.

2.2 SNI Framework

The *Social Network Intelligence* (SNI) Framework has been developed by the IKM Research group⁴ for information and knowledge management (IKM) founded at the Department for Systems Analysis and IT at TU Berlin, Germany: The framework has been introduced in Trier and Bobrik (2007b; 2007c) and successfully applied in several studies (e.g. Molka-Danielsen et al. 2007; Trier and Bobrik 2007a; Trier et al. 2007; Trier 2008; Trier and Bobrik 2008b; Bobrik and Trier 2009; Trier et al. 2009). It is based on social network analysis, principles of network-oriented knowledge management (see Hansen et al. 1999; Stamps and Lipnack 2000; Thomas et al. 2001) as well as conventional process-oriented systems analysis (see Krallmann et al. 2007). The SNI frameworks aims at improving *social translucency* (see Erickson and Kellogg 2000) of implicit networking structures and generating a benefit for its members and moderators. Thus, among the different areas of applications are (see Trier and Bobrik 2007c: 388):

⁴ <http://www.ikmresearch.de/>

- Improving informal and formal collaboration by supporting actions or targeted selection and support of key players for informal knowledge exchange in the enterprise.
- Identifying important but isolated knowledge workers, or identifying sources of knowledge and improving knowledge distribution channels to reduce bottlenecks.
- Creating awareness and accelerating the information flow between departments and business units by initiating contacts between different teams.
- Identifying and promoting talents.
- Identifying groups and opinion leaders for staffing project teams.
- Identifying, supporting, moderating, and benchmarking topic-related communities of practice.
- Consolidating informal structures to support mergers and acquisitions.

The framework is designed to suit as a methodology for enhanced SNA in theory and practice. It therefore can be applied to research as well as business applications. Depending on the object of investigation, the application area and the actual problem different parts of the framework will be emphasized whereas other parts will be less important for the analysis.

In this section, the three basic parts of the SNI Framework are explained: the SNI data model (section 2.2.1), the SNI dimensions (section 2.2.2) and the SNI process (section 2.2.3).

2.2.1 SNI Data Model

The data visualization and analysis of the SNI framework is based on the method of *event-driven dynamic network analysis* as a novel approach for longitudinal social network analysis. In summary, this extended data model allows for more accurate dynamic graph visualization as well as measurement of network evolution in domains, where the links among actors are related to some underlying time-dependent variable, e.g. interaction frequency in online social networking. Furthermore, event properties like keywords and content objects can be used for content-oriented analysis or similarity-based grouping.

In the first section 2.2.1.1, the basic concept of the event-driven approach together with the corresponding data model and its visual representation as a communigraph are explained. In section 2.2.1.2 a more formal description with several examples is given. Finally, section 2.2.1.3 explains the two approaches of event-driven dynamic network analysis based on this data model.

2.2.1.1 Basic Concepts

The event-driven approach has been designed for symmetric as well as asymmetric graphs built from any kind of social activity. Conventional SNA data sets are based on a graph $G = (N, L)$ which consists of a finite set of nodes N and a finite set of links L that connect pairs of nodes (Wasserman and Faust 1994: 122). However, this approach only provides data analysis on an already aggregated level and does not allow investigating further into the dynamics that establish the structure and behavior of a network. Thus, relating to Doreian's and Stokman's (1996: 3) definition of a network process as a "series of events that create, sustain, and dissolve social structures", the underlying SNI data model disaggregates the relationships in the network into smaller units. Thus, links are not directly considered but their

constituting timed events are captured. For instance, in communication network analysis such relational events are created by exchanging messages with others. These events can then be aggregated as links. More general, in collaboration networks these relational events are the various information items (project report, e-mails, work descriptions etc.) on which the actors of the network interact and through which collaboration constitutes.

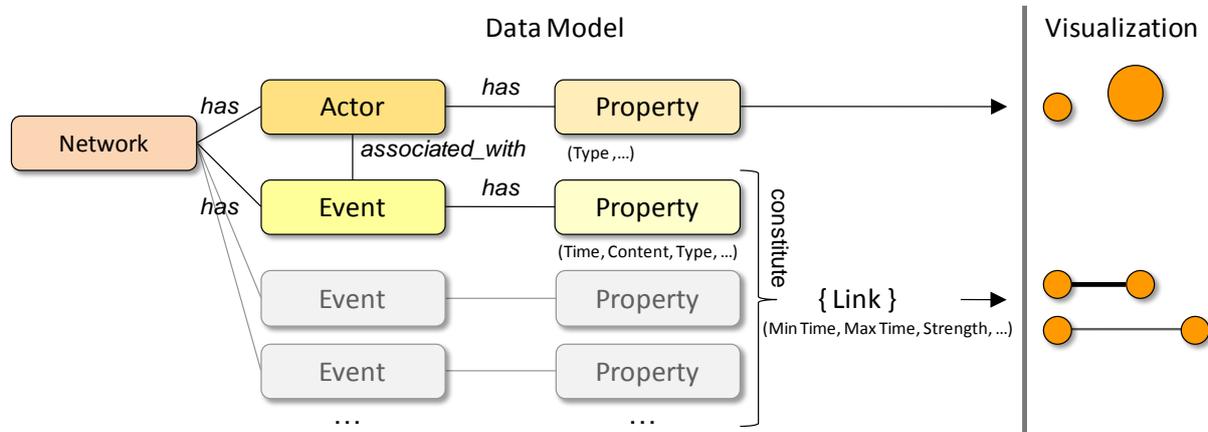


Figure 2-15: The SNI data model. Based on: Trier (2008)

Basically, the SNI data model as illustrated in Figure 2-15 consists of a collection of *actors* with actor properties (e.g. name, organization, organizational rank, or location) and a collection of timed *linkevents*, or simply *events*, with event properties (e.g. time, content, or communication type). *Links* are then time oriented aggregations of these basic linkevents. For example, given an e-mail data set the time stamp of each message event is recorded as a message property. Hence, the sequence of messages and the change in relationship structure or strength is represented as a series of relational events in the data model. This data model corresponds to the three types of data in a social network as described by Scott (1991): relational, attribute, and ideational data. *Relational data* is the representation of contacts, ties and connections between the members of a social network. They link the network's participants. *Attribute data* describes the participants in a network and can be e.g. gender, age or rank in an organization. *Ideational data* describes meanings, motives, definitions and typologies of network members.

As individual events are the smallest unit of computation, event-driven network analysis is utilized best with complete empirical network data (Trier 2008). This data can be retrieved from various social networks emerging from electronic communication, e.g. e-mails among employees. The comprehensive data allows for capturing complete sequences of networking history over time. Thus, the large number of events and their relatedness to a defined context help to ensure that the network structure corresponds with real interaction and social networks.

As in conventional sociograms (see section 2.1.2.1), the corresponding visualization of the SNI data model represents actors as nodes and the aggregated events among them as links between nodes. The sociogram is extended by different visual variables like size or color which can be used to encode various properties. Due to Jacques Bertin's semiology of graphics there are six visual variables of a graphical object that can capture information (Bertin 1967/1983: 42): texture, size, color, orientation (as the variation of pattern from vertical to horizontal), shape, value (as the variation of color), and plane (as the position of a

object in a two dimensional planar space). Trier (2008) and Trier and Bobrik (2008b) propose the use of the *communigraph* as a dynamic visual model which extends the static sociogram used in SNA (see section 2.1.2). A communigraph is the graphical representation of the SNI data model which incorporates Bertin's visual variables and the temporal variation as an additional variable on both nodes and links.

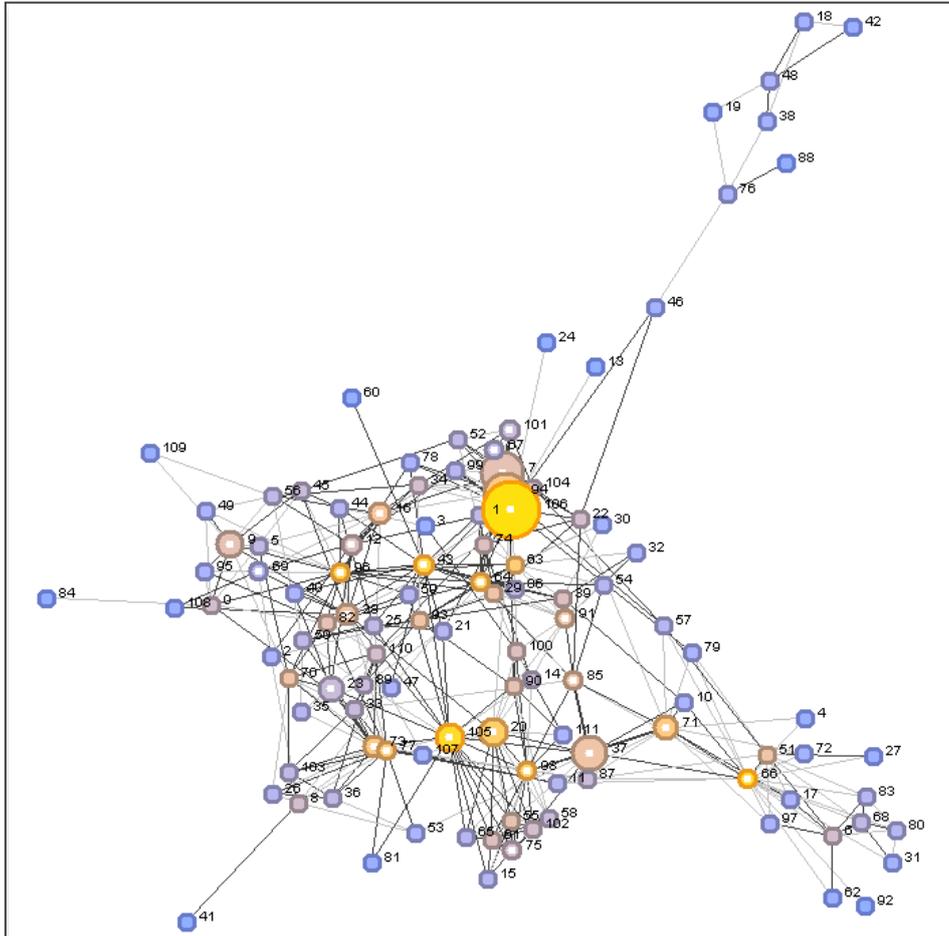


Figure 2-16: Graphical representation of the SNI data model as a communigraph. Enron⁵ data set, subsample January – December 2000. Node label: node index; node color: degree (yellow: large values; blue small values); node size: number of linkevents; link color: reciprocity (dark gray: reciprocal; light gray: non-reciprocal); link strength: number of linkevents

The potentials of the communigraph are shown in Figure 2-16: The subnetwork of the Enron e-mail data set from January to December 2000 ('Enron 2000') contains 112 nodes and 4534 linkevents aggregated on 394 links. The node labels represent the node index, the node color ranges from blue (small node degree values) over lilac to yellow (large node degree values). The node size encodes the number of linkevents sent or received by the node. The link color encodes the dichotomized property of reciprocity. Links in light gray are non-reciprocal; links in dark gray are reciprocal. The link strength represents the number of linkevents aggregated as a single link.

⁵ see chapter 5, section 5.3.1

2.2.1.2 Formalization

The SNI data model consists of a collection of actors A , a collection of (link-)events E , and a collection of links L that aggregate events. Events are all kind of network activity including communication flow, information exchange, collaboration, document sharing, or status change. On a more abstract level three types of events can be distinguished: timed interaction events, timed action events, and timed reaction events.

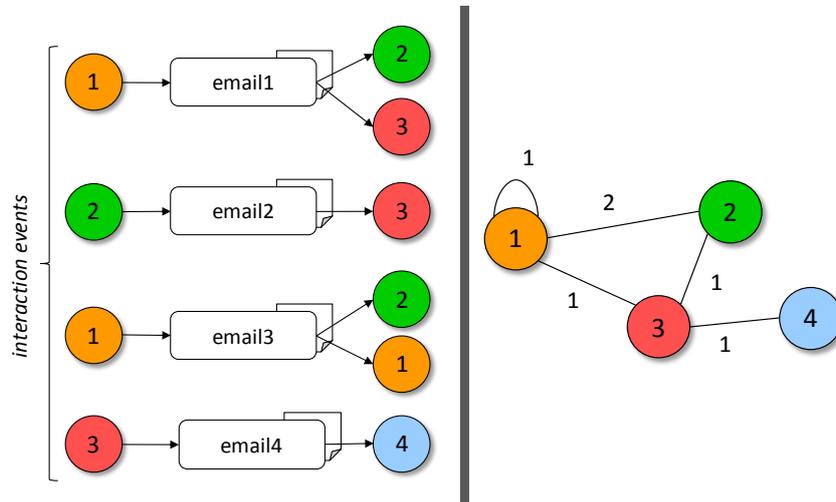


Figure 2-17: SNI data model. Sender-recipient communication. Left: interaction events in an e-mail data set. Right: visualization as communicagraph, link labels correspond to number of events

A *timed interaction event* $e_t(a_i, a_j)$ relates to a timed activity of two actors a_i and a_j , that is either performed together, as with working together on the same document, or performed by one actor with some effect on the other actor, as with writing an e-mail (sender-recipient communication, see Figure 2-17). A *timed action event* $e_t(a_i)$ is a timed activity performed by one actor without directly involving another actor. This relates to posting a message in a newsgroup with no reference to another posting creating a new discussion thread (thread-based communication, see Figure 2-18).

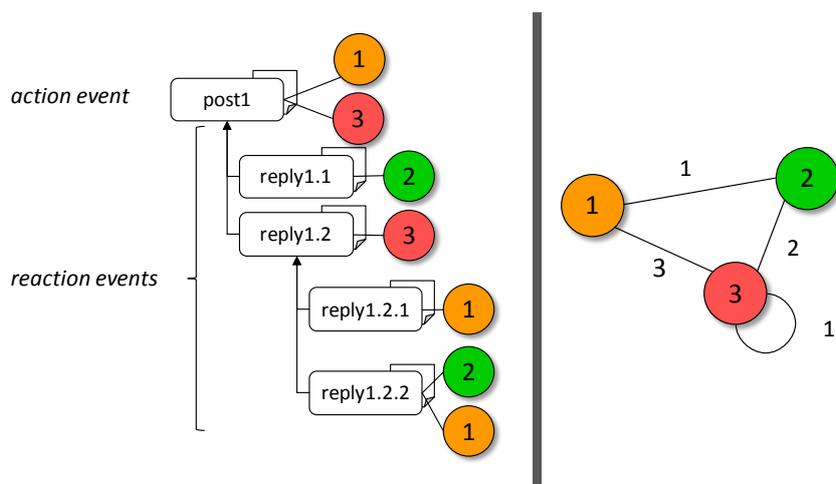


Figure 2-18: SNI data model. Thread-based communication in a newsgroup. Left: action events and reaction events in a newsgroup. Right: visualization as communicagraph, link labels correspond to number of events

A *timed reaction event* by actor a_i on actor a_j is then a timed activity related to some prior interaction event (e.g. citation of a co-authored paper, see Figure 2-19), denoted as

$e_{t_2}(a_i, e_{t_1}(a_j, a_k))$, or to some prior action event (e.g. responding to some posting in a newsgroup, see Figure 2-18), denoted as $e_{t_2}(a_i, e_{t_1}(a_j))$. The condition $t_1 < t_2$ demands that the referenced event has to be prior to the referencing event.

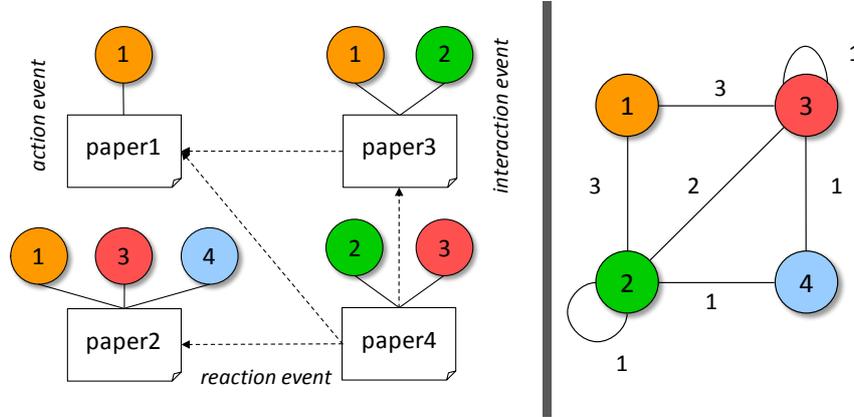


Figure 2-19: SNI data model. Citation network with co-authorship. Left: action events (single author), interaction events (co-authorship), and reaction events (citation; dotted arrow). Right: visualization as communigraph, link labels correspond to number of events

All events are either monadic (action events) or dyadic (interaction and reaction events). If it is necessary for the purpose of the analysis, several events can be aggregated to a group of events if they relate to the same context. For example, an e-mail sent to more than one recipient will be designed as several dyadic events $e_t^g(a_i, a_j)$, $e_t^g(a_i, a_k)$, ..., $e_t^g(a_i, a_n)$ that can be grouped together. This grouping of related events will be indicated by an index g . Interaction events with several inventors and reaction events referring to more than one action or interaction event and any combination of these examples can be designed as group events as well.

As time stamps can distinguish events by milliseconds it will rarely ever happen that two events involving the same actors occur exactly at the same time. However, if it is necessary one can number the events by an additional index.

Depending on the type of network the links between two actors will be established from either one type of events or a combination of them. *Timed links* include only linkevents from a fixed period of time. They can be formalized as:

$$l_{t,t'}(a_i, a_j) = \left\{ e_{t_1}, e_{t_2} \in E: e_{t_1}(a_i, a_j) \vee e_{t_2}(a_i, e_{t_1}(a_j)) \vee e_{r,t_2}(a_i, e_{t_2}(a_j, a_k)), \right. \\ \left. t \leq t_1 < t_2 \leq t' \right\}$$

Self-events with $i = j$ and therefore self-links are possible. With symmetric data, link $l_{t,t'}(a_i, a_j)$ and link $l_{t,t'}(a_j, a_i)$ with $i \neq j$ will not be distinguished in the communigraph and thus can be summarized to one single link.

Each actor and each event can have properties, e.g. types of messages or rank of actors. Links aggregate the properties of the linkevents by some function, such as the sum, the average, or the maximum of all values. The properties of an actor can depend on the actor himself, e.g. gender or name, or be related to the context to the network, e.g. position in the organizational hierarchy. Moreover, some properties can be derived from the position of the actor in the

network, e.g. measures of centrality. This holds for event properties and link properties as well.

If each property can be coded by a numerical value an actor a_i can then be described with a d -dimensional property vector $\bar{a}_i = (a_{i1}, a_{i2}, \dots, a_{id})^T$. The same holds with events and links. The following Table 2-7 illustrates how the different event types can be applied.

Table 2-7: SNI data model. Overview of event types and their application.

Network	Actor	Interaction Event	Action Event	Reaction Event	Example
E-mail Exchange	Any person, identified by e-mail address	E-mail to one or more recipients	-	-	Figure 2-17
Newsgroup	Any person, identified by user account	-	Posting new thread (unrelated message)	Posting new message in a thread (related to a prior posting)	Figure 2-18
Citation	Any person, identified by full name	Paper with several authors (co-authorship)	Paper with one author	Citation of any paper	Figure 2-19

2.2.1.3 Event-driven Dynamic Network Analysis

The event-driven dynamic network analysis based on the SNI data model is motivated by questions of network evolution and change, e.g.

- Has the actor a long steady or a short but quick growth in his degree?
- Did the actor react to external events?
- Which are the most important nodes?
- Is the network stable or eventually already decaying again?
- Which clusters do form and decay?
- Is a cluster stable or just an additive artifact?
- Can the lifecycle of the network be analyzed?
- How and where did the network change?

Therefore, examples for novel research applications based on event-driven dynamic network analysis include the visualization and analysis of group formation and stabilization over time, of actors' paths to central positions, or of process-oriented activity patterns with a structural impact on the network. Focusing on relational events is further able to capture the growth of relationships and the reaction of the network to external events. Generally, the method provides multiple integrated levels of analysis (see section 2.2.2) by linking actor properties, actors' activity patterns, and the resulting impact on general network structures (Trier 2008: 4). On a macro-perspective, general changes of the network can be tracked and traced back to networking events of individual actors. With this new method, networks become less a static phenomenon but can be perceived as a flexible and dynamic structure in constant change and motion.

Today, there are several approaches to dynamic network analysis that try to extend conventional SNA. An overview is given in section 2.1.3.8. The event-driven approach is an example of time window analysis and related to similar works by Moody et al. (2005) and

Gloor et al. (2004). In contrast to the static network analysis where the network is established from all linkevents e_{t_n} with $t \leq t_n \leq T$ covering the overall sample period $\Delta_{t,T}$, a dynamic network analysis partitions the data into a number of temporal subnetworks. Structural changes can then be detected by comparing the positions of the nodes in two subsequent subnetworks.

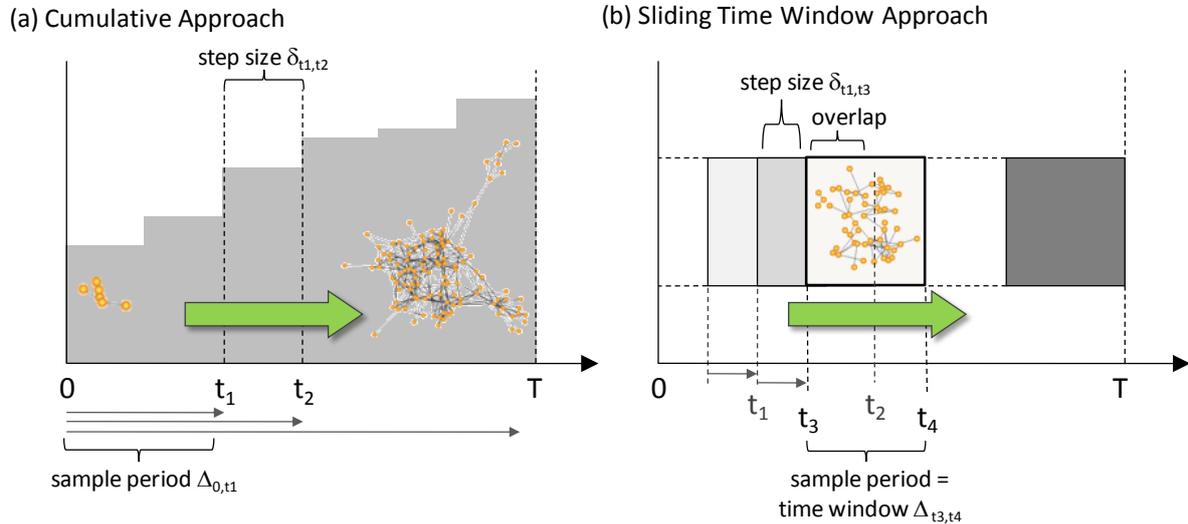


Figure 2-20: Dynamic network analysis: (a) cumulative approach; (b) sliding time window approach

In general, there are two approaches for dynamic network analysis: the cumulative approach and the sliding time window approach. Using the *cumulative approach*, the analysis starts with the first linkevent and successively adds new linkevents to the network until all data is present in the network (see Figure 2-20 a)). Each subnetwork is aggregated from the data that is active during its time period $\Delta_{0,t}$. Thus, the network gradually evolves from its very beginning to its final representation in time period $\Delta_{0,T}$. This final temporal subnetwork is equal to the entire network in the static network perspective. The number of linkevents that are added to the network depends on the step size δ_{t_1,t_2} , i.e. the time between the ends of two successive periods Δ_{0,t_1} and Δ_{0,t_2} with $0 < t_1 < t_2 \leq T$. However, this approach has the disadvantage that recent links have the same importance as old relationships, which in reality might have become less important or even not exist anymore (Falkowski and Bartelheimer 2008).

In contrast, using the *sliding time window approach* a time window is moved through the overall sample period (see Figure 2-20 b)). For each of these time windows, or time frames, a new subnetwork is retrieved from the data containing only those nodes and links that are active within the selected period of time due to some linkevent. Using the sliding time window approach, one has to decide if the windows are non-overlapping or overlapping (Moody et al. 2005). Using the non-overlapping mode, consecutive time windows have no data in common. Thus, one will analyze distinct, unrelated subnetworks. In the overlapping mode, the next time window partially overlaps with the prior window. This mode allows analyzing interrelated subnetworks. According to Falkowski and Bartelheimer (2008) it has the advantage that the fluctuations between two periods will be reduced. The size of the overlap is determined by the step size δ_{t_1,t_3} , i.e. the time between the start of two successive time windows Δ_{t_1,t_2} with $0 \leq t_1 < t_2 \leq T$ and Δ_{t_3,t_4} with $0 \leq t_3 < t_4 \leq T$. If the step size is

less than the time window size, consecutive time windows will likely intersect some of the same events (Bender-deMoll and McFarland 2006: 12).

In both approaches, the length of the time window Δ_{t_1, t_2} depends on the goal of the analysis, on the number of interactions within this period and the distribution of events (Trier and Bobrik 2007a; Falkowski and Bartelheimer 2008). Figure 2-21 illustrates the impact of the time window size on the graphical representation of the network. Smaller time windows yield smaller subnetworks with only a few connections between the nodes, whereas larger time windows yield larger, well-connected subnetworks. To better allow the comparison of these time windows isolated nodes are not removed from the visualization. However, one will usually exclude nodes without any activity in the time window employing an isolates filter.

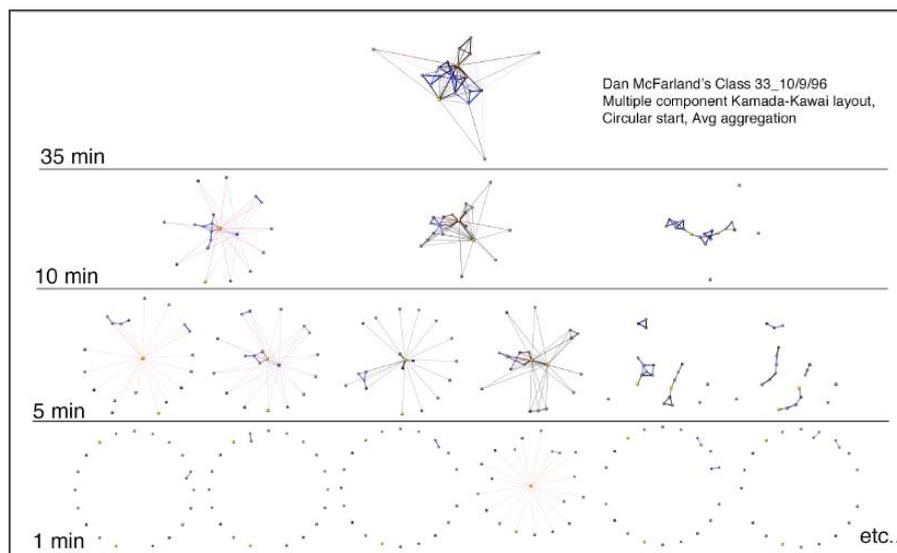


Figure 2-21: The impact of time window size. Interaction data from McFarland's classroom observations viewed at various levels of time aggregation from 35 minutes (one entire class period) to 1 minute (two-three turns of interaction). Source: Bender-deMoll and McFarland (2006: 5)

Both approaches of event-driven dynamic network analysis have been implemented into the Commetrix software for extended network analysis providing a novel technique of visualizing network evolution (see section 2.3.2). In Figure 2-22 examples of a) the cumulative approach and b) the sliding time window approach using the Commetrix software on an e-mail data set are given. In this example, there is no overlap between two successive time windows using the sliding time window approach.

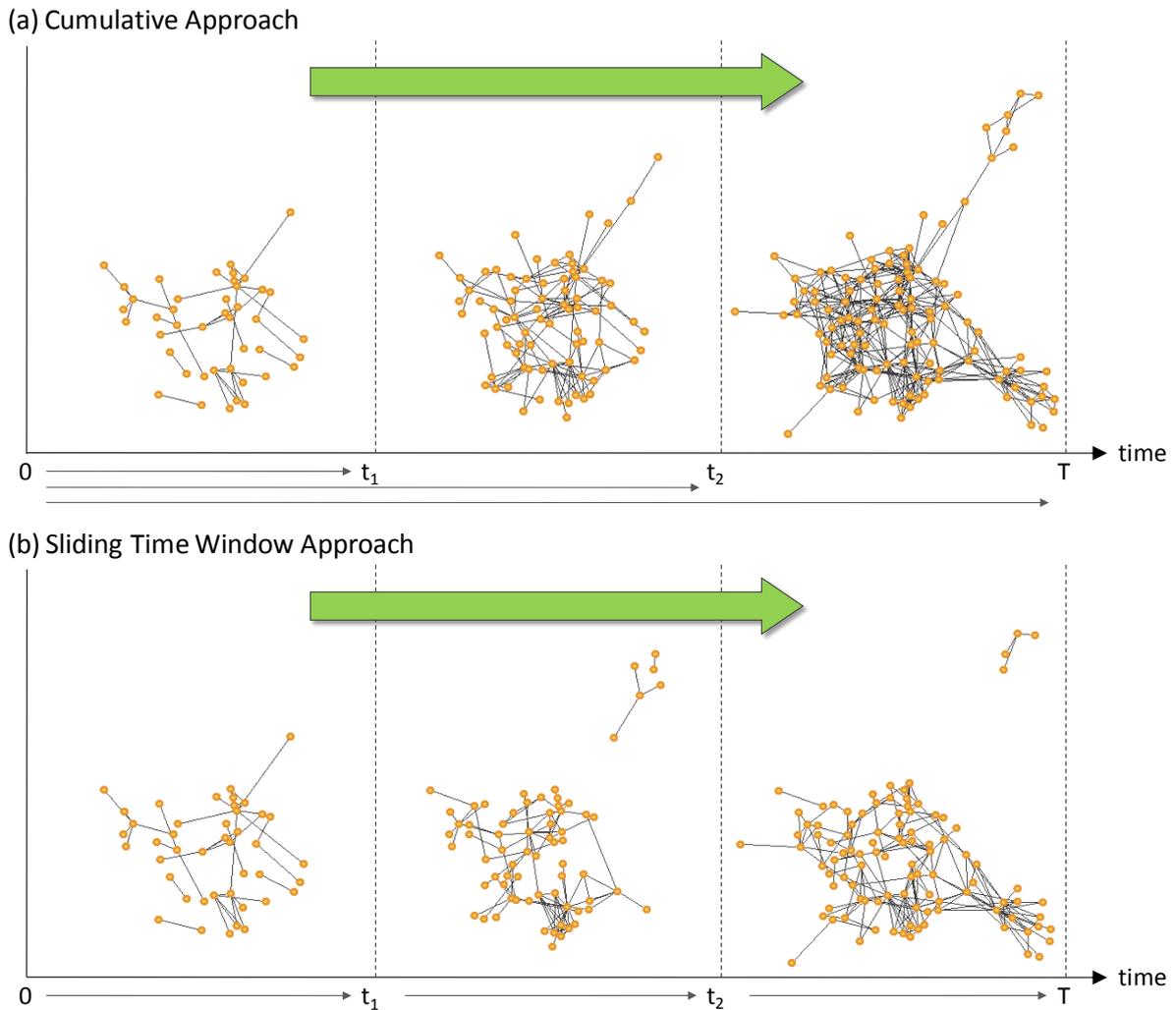


Figure 2-22: Event-driven dynamic network analysis with Commetrix: (a) cumulative approach; (b) sliding time window approach

2.2.2 SNI Dimensions

The event-driven approach of the SNI data model (see section 2.2.1) provides powerful means of multiple integrated levels of analysis belonging to the following three dimensions: the level of detail dimension (see section 2.2.2.1), the level of investigation dimension (see section 2.2.2.2) and the temporal dimension (see section 2.2.2.3). Conventional SNA metrics (see section 2.1.4) can be assigned to the different dimensions providing novel insights into structure, content, behavior and context of the network. Hence, the SNI dimensions provide suitable key performance indicators (KPI) for all kind of SNI applications. By linking different perspective on the network and its components, these three SNI dimensions are the core of the analytical toolbox within the Social Network Intelligence framework.

As illustrated in Figure 2-23 the different perspectives on network analysis can be understood as a three-dimensional dice. The dimensions and their values can be selected and combined to guide the analysis and help to select suitable metrics and measures. One can decide to perform the analysis on whole slices or single components of this dice. Thus, the SNI dimensions provide different options of network analysis. This approach therefore supports a well-structured analysis of the data.

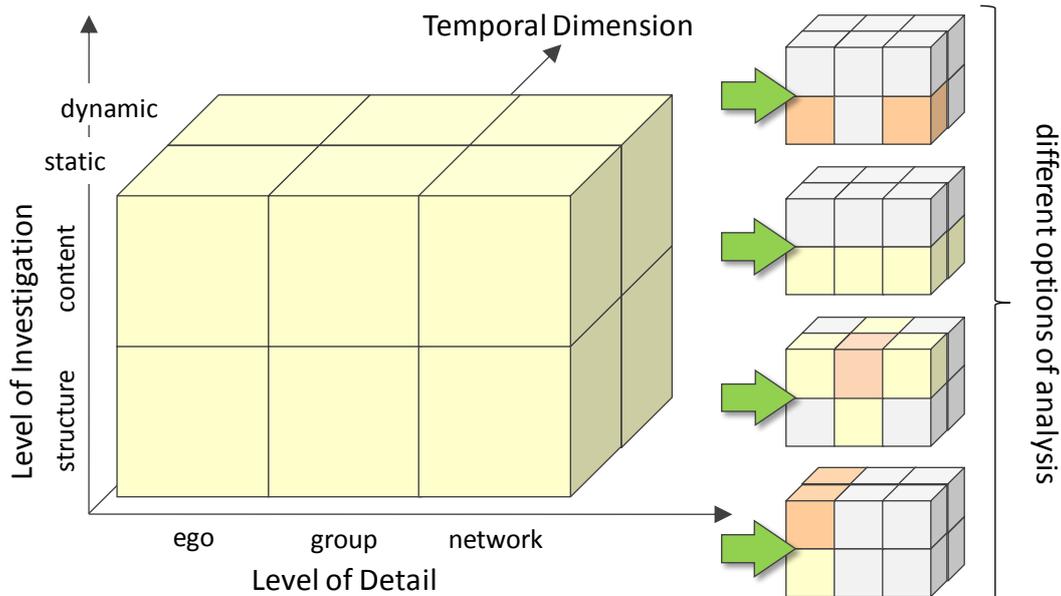


Figure 2-23: SNI dimensions: level of investigation; level of detail; temporal dimension

As explained in section 2.2.3.2, the SNI dimensions should be taken into account when defining goal and scope of the analysis during the SNI process as this can strongly influence early steps like data capturing and refinement.

2.2.2.1 Level of Detail Dimension: Ego, Group and Network Analysis

Besides analyzing the structural embeddedness of an actor in the whole network SNA also focuses on the formation of cohesive subgroups. Therefore, when considering the appropriate level of detail, the analysis can focus on single actors (egos) and their context, groups of actors and the entire network. Naturally, as shown in Figure 2-24 these three perspectives are not entirely separable but the finer levels of detail (e.g. ego, dyad, or triad) are embedded into a broader network context on the more aggregated levels (group or network). Combining the insights of the different levels of detail with the other two SNI dimensions one can gain a thorough understanding of network structure and content and its evolution depending on the focus of the analysis.

Figure 2-24 further illustrates how the different levels of detail determine the behavior of an actor in the context of the network. Starting from a micro-perspective one will concentrate on a focal node (ego) and his behavior determined by individual properties and then successively extend the analysis involving direct contacts, group relations, and network embeddedness. The focal node can be selected by inspecting some general KPIs like the centrality metrics (see section 2.1.4). Usually, one will identify more than one focal node. Therefore, the analysis will start at different parts of the networks and then join the insights obtained from the different levels of details to an overall picture about network structure and behavior. Starting on a macro-perspective, significant patterns like the formation of cliques and general changes of the network can be tracked and traced back to networking events of individual actors.

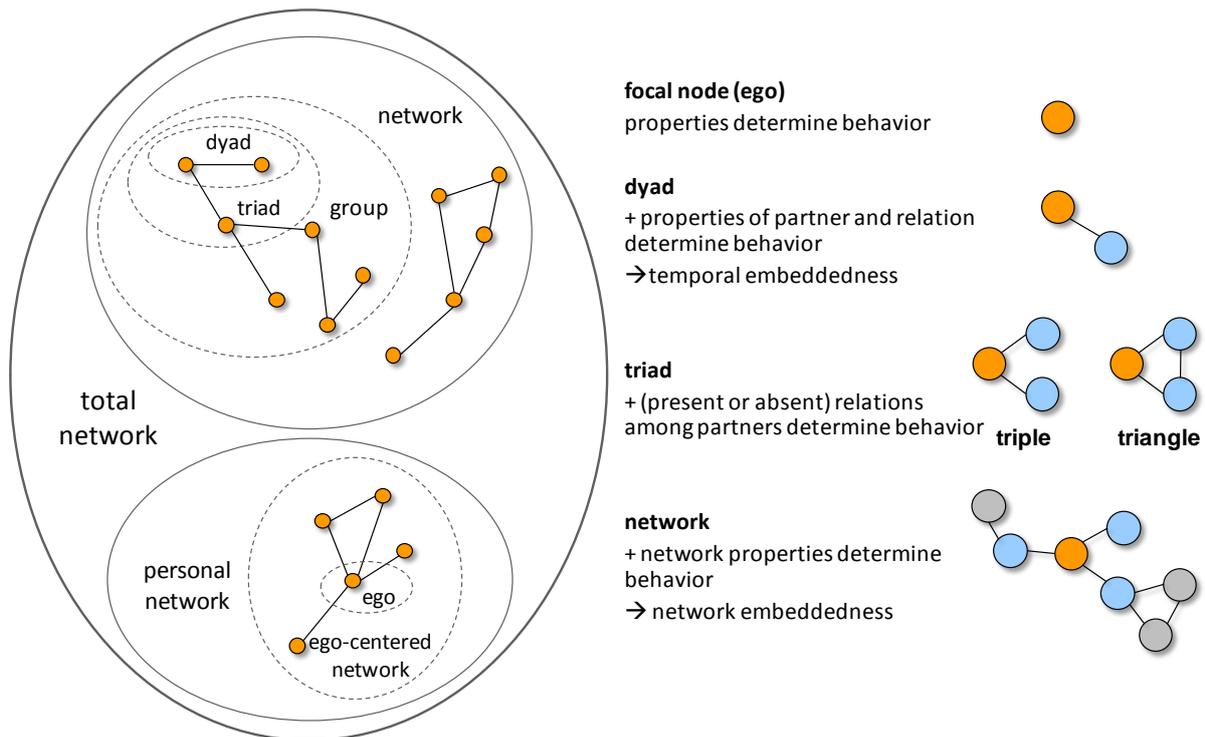


Figure 2-24: Different levels of detail in a network: ego, group and network analysis

Depending on the number of actors involved in the selected perspectives all metrics presented in section 2.1.4 can be calculated on different subsamples of the entire network. For example, one can calculate and compare the node metrics based on the entire network with those obtained when concentrating only on a group of actors. Those subsamples might not only be identified by network structure as illustrated in Figure 2-24 but also by the context of the network, e.g. by similarity of the content of the messages exchanged in an e-mail network.

A more detailed discussion on subgroup detection in general and with respect to network structure can be found in chapter 4 “Cluster Analysis”.

2.2.2.2 Level of Investigation Dimension: Structure versus Content Analysis

As illustrated in Figure 2-25 collaborative content creation is the underlying process of evolving networks. *Collaborative content creation* can be explained as any kind of interaction on or with content objects. This definition allows summarizing all kind of networking activity. It includes relational data by directly or indirectly linking actors with content objects (e.g. e-mail, postings and comments, ratings, work schedules, or documents) as well as a temporal dimension. Aggregating the data during a selected sample period one can retrieve a *content layer* containing all content objects and a *network layer* containing the relationship between the involved actors as a network structure.

The network layer allows analyzing the structural properties by means of static and dynamic measures and methods. It is the initial point of entry to gain an understanding of the data. However, a mere structural perspective fails to reveal the complex nature of collaborative content. Examining the content objects created, distributed and exchanged between the network participants provides a powerful instrument for a more detailed analysis. This type of analysis not only helps to verify the results from the structural analysis but also to derive new insights from the data.

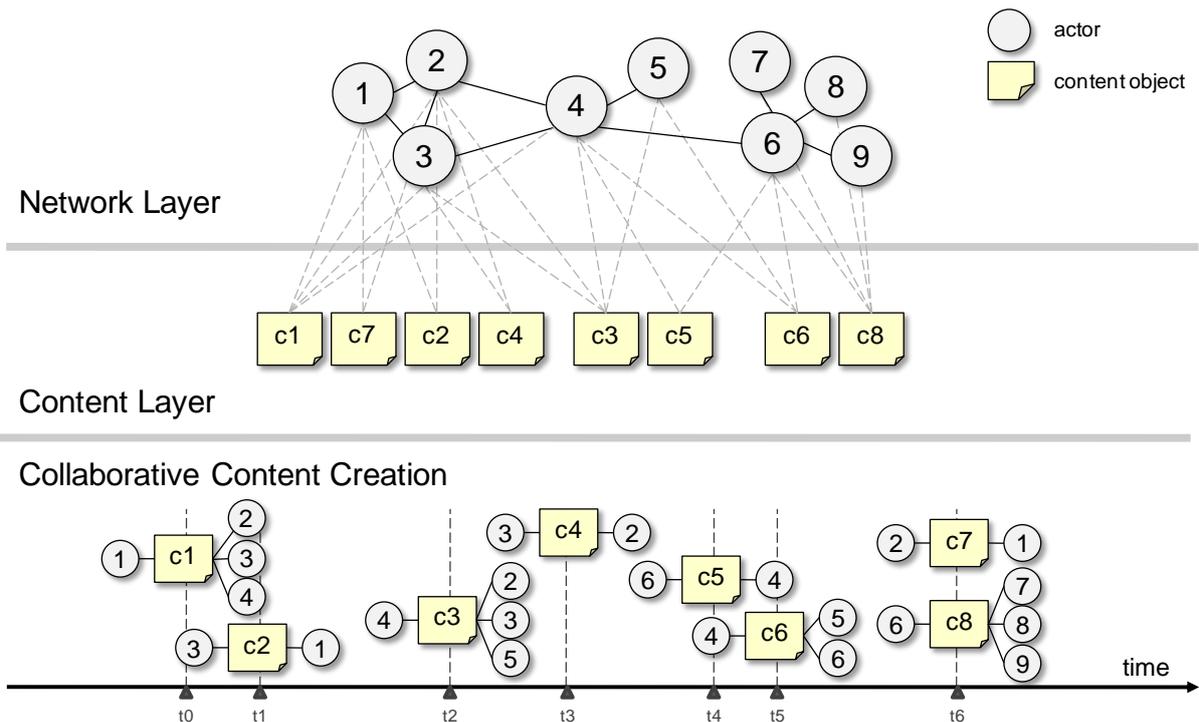


Figure 2-25: Different levels of investigation: structure versus content analysis

Combining text mining techniques with SNA offers the possibility to uncover hidden sources of information and knowledge domains and investigate on their relations to the actors of the network: To examine the affiliation of single actors or groups of actors to knowledge domains identified by their communicational and collaborative content can reveal important insights about the complex nature of social networks. Only the analysis of the content of the network allows to assess the context of the network and to identify triggering external events. For example, the content-based clustering method presented in chapter 5 groups together those actors from different parts of the network that with similar knowledge without regarding their structural relationships. All SNA metrics can then be calculated on these subsamples of actors and relations. This allows identifying import key players and knowledge workers from different knowledge domains.

2.2.2.3 Temporal Dimension: Static versus Dynamic Analysis

The static network analysis covers all data from the sample period and aggregates them into a single network neglecting the temporal dimension. Here, the research questions concern the overall structure of the network and what parts of the network are noticeable well or poorly connected or even isolated from the rest. Furthermore, on ego level the position of an actor in the network is analyzed to derive information about his status and influence. Metrics involved here are either merely structural (e.g. size of the network, density, diameter, number of sent and received events of an actor), or socio-structural (e.g. centrality measures, clustering coefficient, relationship strength between two actors) as they provide insight about the social status of an actor or a group of actors within the network (Trier and Bobrik 2007c: 392).

In contrast to the static analysis which assumes that node appearances and links between nodes are persistent during the entire sample period, the dynamic network analysis incorporates the temporal dimension as an additional strategy for analysis. The dynamic

analysis employs a time-related filter that allows observing, how the network changes over time and searching through temporal episodes of network development. There are two possible approaches: the cumulative approach and the sliding time window approach. Using the cumulative approach the analysis starts with the first network activity and gradually adds new data to the network. At the end, the dynamic network equals the static network. Using the sliding time window approach all information which is outside the time window is either not yet being included or has expired. The dynamic analysis allows a better understanding of how and why networks evolve and the impact of temporal events. A general overview of recent SNA research on dynamic network analysis is given in section 2.1.3.8 whereas section 2.2.1.3 explains the potentials of event-driven dynamic network analysis based on the SNI data model.

All metrics from section 2.1.4 can be calculated on the static network as well as each temporal subnetwork. This allows complementing the qualitative observations of network stability and change with a reliable quantitative analysis when comparing networks from time-related periods.

2.2.3 SNI Process

As illustrated in Figure 2-26 the SNI Process is the basic methodology of the SNI Framework. It can be used as a method of IT-supported, network-oriented knowledge management to analyze knowledge-intensive business processes (see chapter 1.1, section 1). The SNI process consists of six steps which are passed through iteratively and require the supervision of business and domain knowledge workers: 1) project initiation, 2) definition of scope, 3) data capturing & refinement, 4) visualization & analysis, 5) deriving concepts & action plan, 6) execution & implementation of actions. These steps correspond to the process model of process-oriented systems analysis (see Krallmann et al. 2007: 135): project initiation corresponds to step 1) and 2), as-is analysis corresponds to step 3) and 4), to-be analysis corresponds to step 5) and implementation corresponds to step 6). Similar to this process model the SNI process requires constant project management and user participation. Other process models which address the problem of process-oriented knowledge management are the Business Process-oriented Knowledge Management approach (BPO-KM, in German: GPO-WM, see Heisig 2005) or the Knowledge Modeling and Description Language (KMDL, see Gronau and Fröming 2006). However, they are not especially designed for identifying, analyzing and managing knowledge networks that evolve from knowledge-intensive business processes.

The essential part of this process is the visualization and analysis of the data (step 4). It is based on conventional SNA and contains of measures and recommendations for action on the three SNI dimensions: different levels of detail (ego, group, and network level), temporal dimension (static versus dynamic), and different levels of investigation (structure versus content).

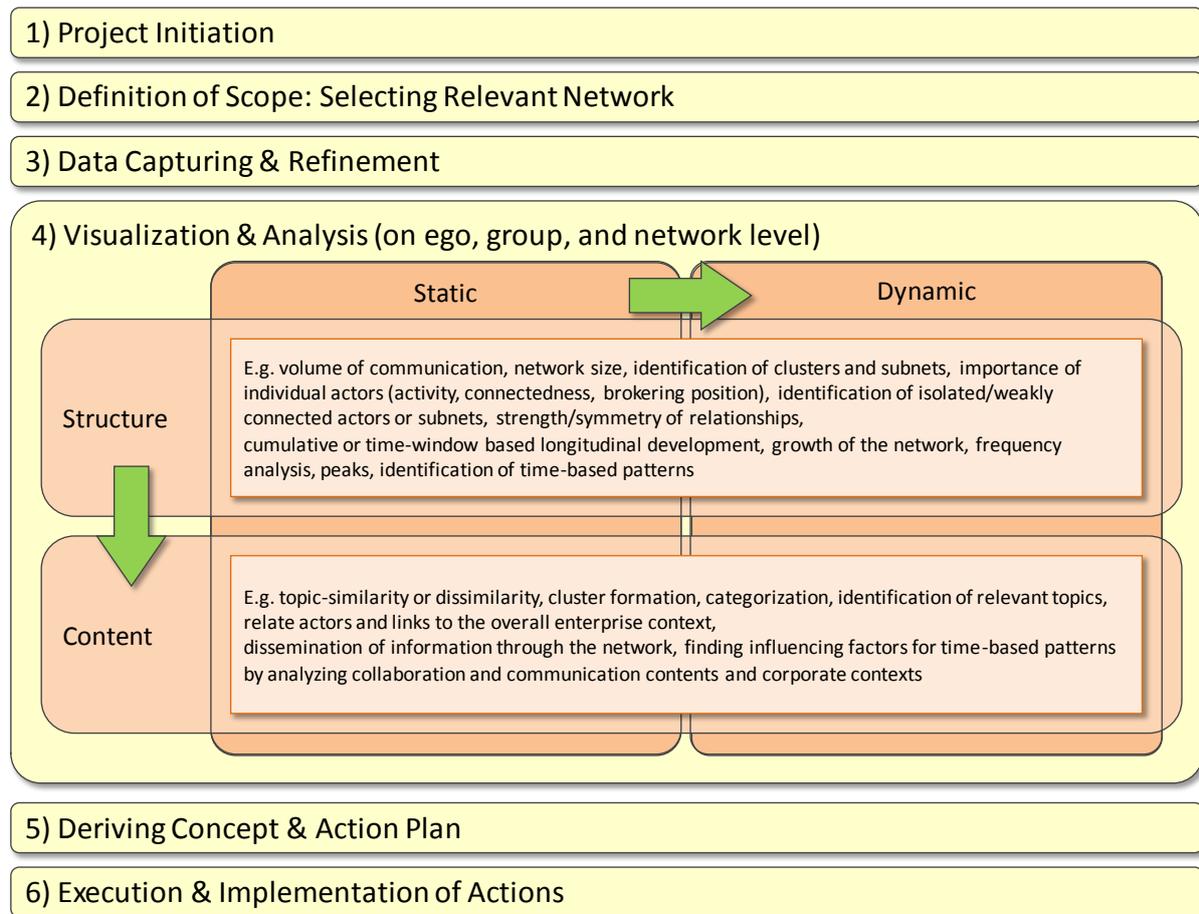


Figure 2-26: SNI Process. Extending conventional SNA. Based on: Trier and Bobrik (2007c)

In practice, there are two approaches how to conduct a IT-supported, process-oriented knowledge management project to analyze knowledge-intensive business processes: either as part of the general *process-oriented systems analysis* project or as independent *network-oriented systems analysis* project (see Trier and Bobrik 2007c: 390). If formal as well as informal structures of collaboration and communication are the main objects of investigation to e.g. analyze the influence of informal structures on business processes the SNI process will be a complementing part of the as-is analysis. If the complex informal structures of collaboration and communication are the main objects of investigation to identify and support determining factors and key players of information dissemination in the enterprise the SNI process can be regarded as an independent procedure. In this section, the six consecutive steps of the SNI process are explained.

2.2.3.1 Project Initiation

The project initiation phase usually includes the formation of the research team and the definition of the actual research problem or question. The purpose of the study and its objectives must be clear and all parties involved must consent to it. A precise definition of the goals is crucial for the application of the correct methodology and the validity of the results. Typical subjects of organizational network analyses are (see e.g. Cross et al. 2000; Cross et al. 2002; Krebs 2002; Anklam 2005a; Anklam and Wolfberg 2006; Laseter and Cross 2006; Johnson-Cramer et al. 2007):

- Organizational change during or following major restructuring, for instance through mergers or acquisitions.
- Knowledge management including the identification of knowledge and the improvement of necessary connections.
- Staffing of projects or task forces.
- Detection and measurement of communities of practice.
- Human resources strategy including leadership development, succession and retention planning.
- Assessing and improving current collaboration.

This first phase of the SNI process also involves the determination of supporting factors such as the allocation of time, space, authority and support for the research team as well as potential compensation (see Frank and Trier 2007: 192). In case of an analysis within an organization, this step also comprises negotiating agreements with the client and informing the necessary committees or the works council of the organization.

Furthermore, the research team must confirm that the management is committed to the study. This commitment is indispensable to receive the necessary authority and to make sure that it is understood that the results may require substantial changes in work practices, organizational structures and policies (Anklam 2005a).

As sensitive personal data is part of the analysis the data security and data privacy protections have to be carefully incorporated. Therefore, all ethical issues need to be addressed during this first phase. The respondents must be informed beforehand about how the study is conducted, what data is collected, and how the information is processed and published. Borgatti and Molina (2005) discuss some ethical problems of network studies in organizations:

- Lack of anonymity because all respondents must state their names. Anonymization through pseudonyms or id codes can be ineffective in small organizations.
- No true opt-out because employees are directed to participate and noncompliance can be detected by checking the appearance of each staff member in the results of the study.
- Opting-out is also prevented when other respondents mention an individual who decided not to participate.
- Fear of corrective actions through the management.

In sociological research – especially in a corporate context – it is necessary for the researcher to obtain permission from the respondents to collect, process, and publish their data. Borgatti and Molina (2005: 112-113) therefore suggest using a truly informed consent (TIC) form. It includes an account of what data is being presented to the management and how the results of the social network analysis will be used in the organization. The TIC form serves as a contract between the participants of the study, the researchers, and the management.

2.2.3.2 Definition of Scope: Selecting the Relevant Network

Similar to the definition of the scope and the elements of the relevant system in a process-oriented systems analysis, the SNI process starts with the definition of scope to identify the

relevant network (see Jansen 2003). There are two possible ways to select the relevant network: the *actor-oriented approach* or the *context-based approach*. The first approach starts with identifying the relevant set of actors. The relations between them will be determined due to e.g. structural or temporal criteria (e.g. the time frame of the study, the question of singular or repeated surveys), or the level of investigations. Furthermore, the level of detail (e.g. individual, group, or entire network) has to be considered. These criteria are related to the SNI dimensions which are explained in detail in section 2.2.2. For example, the focus of the analysis could be on certain project teams, or business units, or all key players in a certain knowledge domain, or even only on a few information leaders. The second approach starts with a thorough definition of the relevant context based on the information items available. Here, relevant topics will be considered first, corresponding information items will be identified and actors will be assigned to these items. In case of a very large population with indistinct boundaries researchers can also use sampling methods to define their set of relevant actors (see section 2.2.3.3.1).

In general, these decisions are usually not trivial and need careful consideration. They have to be made before the actual collection of the data with regard to the problem at hand and will strongly influence the process and the outcome of the entire analysis as well as the suitable means of data collection, refinement and analysis: Errors made during this phase are directly reflected in the network data and can lead to incorrect analyses and conclusions.

In an organizational context, the identification of the relevant actors is usually simple as they are often made up of members of one or several departments. However, when informal organizational networks are to be analyzed specifying the boundaries is much more difficult or even impossible. Often the researchers do not even know the relevant actors and their number beforehand. The network then evolves parallel to the collection of data. In either case, since circumstances may change and cannot be foreseen, researchers should be prepared to move the pre-defined boundaries if necessary. Researchers can generally adopt one of the following two strategies to boundary specification (see Knoke and Kuklinski 1983: 22): the *realist approach* or the *nominalist approach*. Using the first approach the boundaries of the network are consciously perceived and acknowledged by the actors themselves, for instance in case of a family, classroom or club. In contrast, the nominalist approach refers to the application of a conceptual framework that serves the researchers' theoretical concerns. In this case the researcher constructs the set of actors from units that bear the specific characteristics necessary for the study.

2.2.3.3 Capture and Refine Data

There are two possible methods for data collection (see Krallmann et al. 2007): the access of existing secondary data and the generation (or capturing) of novel data using primary data collection techniques. As human collaboration, communication and interaction are becoming more and more IT-supported there are a large number of electronically stored data available for analysis. Although they are a rich source of information they usually have to be complemented with background information and personal feedback using primary data collection techniques. Section 2.2.3.3.1 gives a brief introduction to data collection techniques in the context of this work. Trier and Müller (2004) propose a method for capturing knowledge-intensive business processes which involves both types of data collection (see

section 2.2.3.3.2). It emphasizes on a standardized, semi-structured interview to identify knowledge-intensive business tasks and relate topics, documents, and actors to them. Most often, data conversion and refinement have to be performed to enable an IT-supported analysis and to improve the quality of the data (see section 2.2.3.3.3). If the Commetrix software (see section 2.3.2) is used to support the analysis the characteristics of the SNI data model (see section 2.2.1) should also be taken into account during this step.

2.2.3.3.1 Data Collection for Social Network Analysis

For a simple graphical representation of the actors and relations of the network, questionnaires are generally sufficient. Cross et al. (2002: 27) state that the most pragmatic means to collect data in organizational settings are surveys. Further methods of primary data collection are interviews, workshops as a special form of group interview, or observations (see Krallmann et al. 2007).

Table 2-8: Survey to evaluate the particular needs of an organization. Source: Krebs (1996)

Assessed Dimensions	Sample Questions
Mission and vision	With whom do you discuss the company vision and business strategy? With whom do you discuss what is important and valued in the organization?
Work interactions	With whom do you work to get your job done [exchange information, documents and other resources]?
Grapevine	With whom do you discuss what is going on at work, and who is doing what in your organization?
Decision making	From whom do you seek inputs, suggestions and feedback before making a decision?
Innovation	With whom do you discuss ideas, innovations, and better ways of getting things done?
Expertise	To whom do you go for expert advice in doing your work?
Customer knowledge	With whom do you discuss customer needs and market demands?

The questions can cover any aspect from communication patterns to friendships and depend on the main research questions of the study. In business context the interviewee is usually asked which colleagues directly provide information or act as an information broker when solving a certain problem. Additionally, one can learn from the respondents who is well connected to whom and how they assess the competence of their colleagues in different knowledge domains. Section 2.2.3.3.2 presents the approach of Trier and Müller (2004) for a guided interview. Cross and Parker (2004) have compiled a list of archetypal questions in four categories to reveal collaboration, information-sharing, rigidity, or well-being and supportiveness in a network, and economical considerations of the whole study. Krebs (1996) provides sample questions related to seven dimensions that should be assessed when evaluating the business needs of an organization (see Table 2-8).

Given a set of actors the spectrum of possible strategies for capturing the relations among them ranges from full network methods to snowball sampling methods (Hanneman and Riddle 2005). In general, there is not a single method suitable for all research questions and problems.

To capture ego-centric networks *snowball sampling* is often applied (Hanneman and Riddle 2005): This procedure starts with interviewing a pre-selected set of actors (e.g. a list of domain experts, a certain department, a project group, or some key players already identified from a preceding analysis or by recommendation). They form the first-order zone and are asked to name people who they have ties with. Those who are not already included in the first zone constitute the second-order zone. Their members are then asked to specify further people to whom they are linked and so forth (Wasserman and Faust 1994: 34). The researchers must decide on a cut-off value, e.g. when the percentage of newly named people drops below a certain value or for reasons of time and resources.

The snowball method can be particularly helpful for locating and describing specialized and often numerically small sub-sets of people mixed with large numbers of others, e.g. business contact networks, community elites, or deviant sub-cultures. Those subgroups usually tend to be of small size due to the limitations on the numbers of strong relations that most actors have and the tendency for relations to be reciprocated (see e.g. section 2.1.3).

The quality of snowball approaches can be improved by a thorough selection of the initial nodes. In many studies, there may be a natural starting point. In studies of community power, for example, it is common to begin snowball searches with the chief executives of large economic, cultural, and political organizations. This approach is very likely to capture the elite network quite effectively but will miss the isolated members.

There are two major drawbacks of snowball methods: First, actors who are not connected are not located by this method but may be essential for some analytic purposes. Additionally, the snowball method tends to overstate the connectedness and solidarity among actors. Laumann et al. (1983) state that “it is scarcely informative to learn that a network constituted by a snowball sampling procedure is wellconnected”. Second, there is no guaranteed way of finding all of the connected individuals. Here, the set of actors to start with is essential as one may miss whole sub-sets of actors who are connected but not attached to the starting points. Further issues of sampling methods are discussed e.g. in Knoke and Kuklinski (1983: 26-30) and Scott (1991: 58-65). In general, snowball sampling is a trade-off between completeness and accuracy of the network data. The resulting entire network is constructed from the retrieved ego-centric networks.

In contrast to snowball methods, using *full network methods* the information about relations between all pairs of actors is collected (Hanneman and Riddle 2005). Many measures of the structural concepts of network analysis rely on full network data, e.g. betweenness centrality (see section 2.1.4). Unfortunately, full network data can also be very expensive and difficult to collect. Directly questioning every member of a population, and having every member rank or rate every other member can be a very challenging task in any but the smallest groups. However, most people, groups, and organizations tend to be able to maintain only a limited numbers of relations. At least the number of strong relations is usually restricted due limited resources, energy, time, and cognitive capacity. Furthermore, social structures can develop a considerable degree of order and solidarity with relatively few connections.

Although the snowball procedure or full network questionnaires were a common means for data collection in the beginning of SNA they are too laborious to be applied to large networks (see Wasserman and Faust 1994: 43). As collaboration, communication and interaction are

becoming more and more IT-supported there are a large number of electronically stored data available for analysis, e.g. work schedules, project reports, entries in document management systems, e-mail server log files, VoIP archives, instant messaging systems, or more recently blogs, newsgroups, and wikis. Using personal data as part of the analysis the relevant data security and data privacy protections have to be carefully incorporated. Compared to the results obtained from interviews and observations the effort of capturing and refining these data can be drastically reduced while simultaneously increasing the number of actors which are included in the analysis. Additionally, they tend to be less affected by personal estimations and sentiments. They also better allow capturing the overall picture as they also include isolated persons and loose contacts which might have been forgotten during the interview or cover topics that were not identified to be relevant for the analysis at first, or even reveal problems which have not yet become apparent.

Despite these benefits it is important to check for the consistency and the context of the data and if necessary to manually interpret and correct them. For example, managers often do not directly participate in communication and collaboration media but by their representatives, e.g. assistants or secretaries (see e.g. Trier and Bobrik 2008a: 330). Furthermore, networks derived from electronic data tend to quickly grow too large and to exceed a level that can still be processed and understood by human beings. Additionally, the data is very likely to contain interactions that are irrelevant for the research problem. Grippa et al. (2006) examined the validity of electronic data. They discovered that e-mail communication does not necessarily reflect the actual network structure. The authors were able to identify typical characteristics and biases, e.g.:

- Overestimation of communication between peers with technological skills and their central role within the network.
- Lacking ties between peripheral and co-located team members.
- Underestimation of individuals playing a gatekeeper role.
- The risk to interpret the communication pattern as dominated by a core group.

Thus, electronic data alone is seldom suitable to uncover why actors hold certain positions or exhibit particular patterns of relations. Only the context – organizational or social – can provide for a comprehensive picture. Therefore complementing background information and feedback from the people involved are beneficial. Capturing organizational units, process activities and work packages as well allows comparing the results from the network analysis with the formal business structures. Furthermore, if the SNI framework is applied to business investigations it will be necessary to add business context of non-archived interaction to the data. To complement relational data with attribute data, surveys should therefore include questions on demographics (e.g. gender or age), and in an organizational setting also on job function, tenure, hierarchy and location. The questionnaire could also include items on cultural values (Johnson-Cramer et al. 2007) or competencies (Anklam 2005b: 31). Thus, a thorough analysis will always involve both types of data collection: existing secondary data and the capturing of novel data. This data is essential in the interpretation of the findings during the next two stages.

2.2.3.3.2 Data Collection for Knowledge-intensive Business Processes

Trier and Müller (2004) propose a method for capturing knowledge-intensive business processes by means of evaluating available documents and personal interviews. It is designed to allow insights into questions like how to comprehensively capture advanced knowledge processes, how to store the collected information in a structured way, and how to design supporting instruments and materials in order to ensure a complete documentation of the existing processes and a structured discussion to generate a high quality data set, which subsequently is utilized to derive appropriate measures.

Based on the experience of practical project work with virtual communities of practice a generic knowledge management (KM) entity model has been developed by the authors (see Figure 2-27). It assembles all relevant objects together with their interrelationships on the most aggregated level to define an abstract meta-structure for the data.

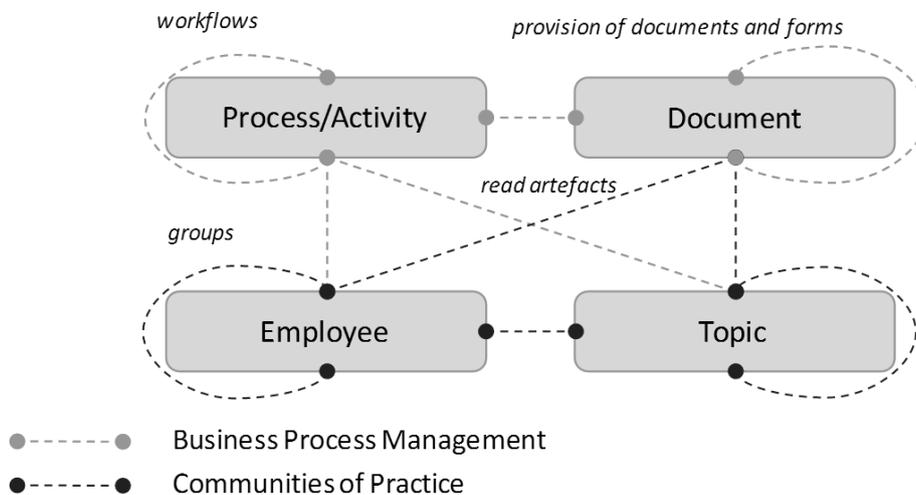


Figure 2-27: KM entity model. Source: Trier and Müller (2004: 243)

This approach helps to create transparency about the KM entities as it models the process (as a sequence of activities), the attached responsible person (employee), the documents necessary for the transactions, and the topics related to them. Insights that can be derived are e.g. which author is connected whom, who is responsible for a process, or what documents about which topics are needed to carry out an activity. Understanding the KM entities enables the systematical support of capturing knowledge-intensive processes.

First, a secondary data collection using the inventory method is conducted based on the collection and analysis of all relevant documents that exist in the company. The written information helps the project team to acquire a first understanding of the basic structures and processes of the enterprise. Documents such as organization charts, intranet pages about the different departments and their services, or existing job descriptions can give a first orientation for the capturing of knowledge objects.

Additionally, the non-standardized, semi-structured interview method is employed. The interviews with the respondents are supported by a guideline for the interviewing person. It allows detecting soft issues and adapting to the actual situation described by the respondent. Methodically quite similar to a workshop the interview is shifting towards a guided discussion. The authors suggest a necessary duration of about 60 to 90 minutes. The authors

recommend using the actual workplace as preferred location in order to create a better impression of the interviewee’s environment.

To guide the interviewers and to assure the completeness and comparability of each protocol, an interview guide should be created, released and consistently be used by the team. The following interview guide has been applied and tested in practice (see Figure 2-28). It includes five sections: The first and introductory section of the interview introduces the project and its objectives as well as the benefits for the employee. It can be useful to communicate several advantages of participating in the project, e.g. public recognition of the respondent’s fields of expertise, new knowledge about how problems can be solved, or opportunities to collaborate and exchange knowledge with people having a similar background.

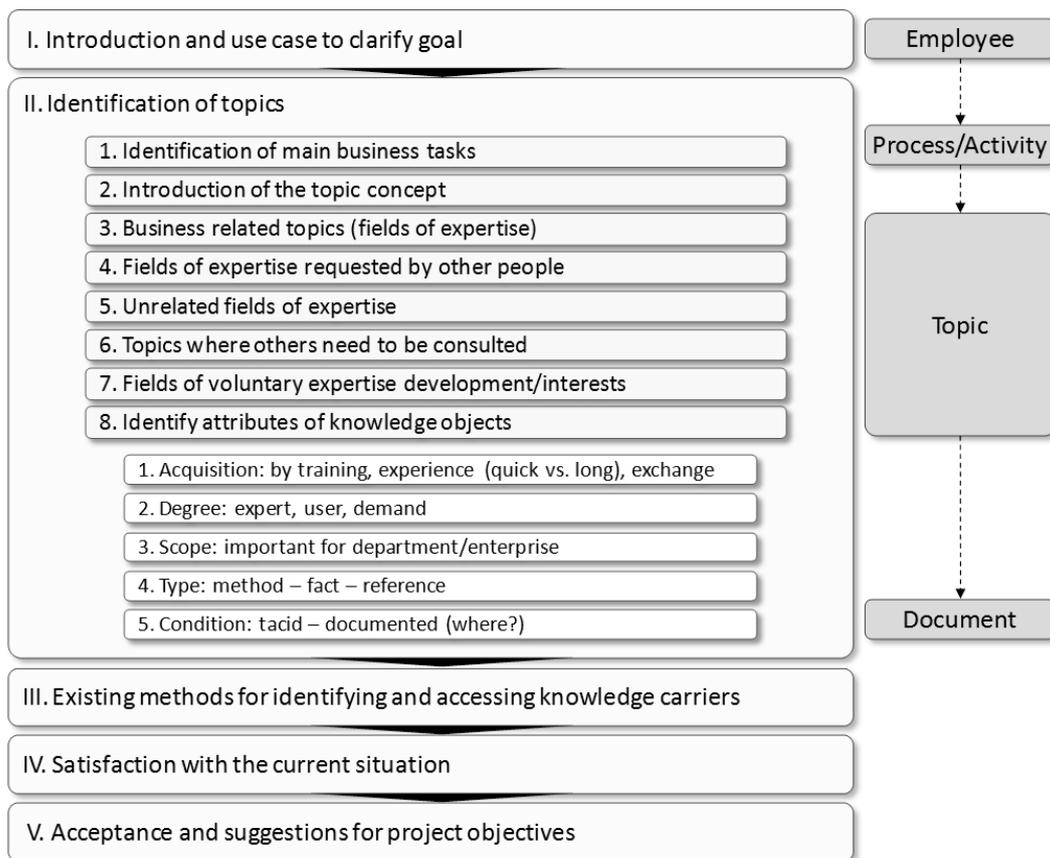


Figure 2-28: Procedural guide for capturing knowledge-intensive business processes and its relation to the KM entity model. Source: Trier and Müller (2004: 245)

The second section constitutes the main part and is broken down into eight subsections. Their main objective is to find out business tasks and relate topics to it. Next to the topics, the properties of the knowledge domains have to be documented in order to prepare for the derivation of management instruments in the last stage of the project. The three elements “process activities”, “related topics” and “topic attributes” should be captured in parallel in order to track the complete information about relations between the items. Often, the required topics cannot simply be asked and collected directly but have to be approached with a set of investigative questions. In parallel to eliciting all topics, the properties of these knowledge domains will be collected for every proposed item to improve the derivation of management interventions. The selected properties to be captured included five elements: the way of

acquisition, the degree of expertise, the scope of relevance, the type of knowledge, and the condition of the topic.

In the third section the respondent is interviewed about existing methods for identifying and accessing important knowledge carriers. This helps to understand the employment of different communication channels in the company. Additionally, it allows a first insight about probable media which can be used for the resulting solution.

In the fourth section the respondents evaluate the existing information structures and infrastructures. This is a very project specific question which requires preparatory supply of necessary documents from the preceding inventory method.

In the final section the employees should express their expectations for the project which can be used to identify new aspects, derive priorities, and receive feedback to ideas proposed by the interviewer in the introductory sections.

Trier and Müller (2004) suggest a structured analysis and interpretation of the data using tabular instrument to capture properties of topics and their connection to processes.

2.2.3.3.3 Data Conversion and Refinement

In order to employ an IT-supported analysis all collected data must be electronically captured, processed, and transferred to the network analysis software. If data is collected through interviews and questionnaires it can be stored in spreadsheets and imported into the network analysis software. Some SNA programs also offer a feature to directly capture the data. Instead of entering the data into the computer manually, computer-based surveys can be used, which save the responses in a database. For instance, Johnson-Cramer et al. (2007: 92) employ online surveys to collect the network data, which take about 15 minutes to fill in. Using secondary data derived from electronic archives the network analysis software may not natively support these data models. Thus, the information must likewise be processed and altered into a form which allows importing it into the software. An overview of suitable software can be found in Huisman and van Duijn (2005). It is presented in section 2.3. In the context of this work the software Commetrix⁶ using the SNI data model is recommended which has been developed by the IKM Research group (see section 2.3.2).

As already mentioned above it is important to check for the consistency and the context of the data and if necessary manually interpret and correct them. Suitable data refinement methods usually depend on different parameters, e.g. the data itself (see e.g. challenges of text characteristics, chapter 3.1.6), the scope and general purpose of the analysis as defined in section 2.2.3.1 and 2.2.3.2, data security and privacy restrictions, and the data format of the selected analysis software. Furthermore, structural or content-based filtering criteria can already be employed in this step. If the data has to be made anonymous the potentials of the analysis and the recommendations for action are limited. Nevertheless, even the general structure, the content and the evolution of the network can provide important insights.

⁶ <http://www.commetrix.com>

2.2.3.4 Visualization and Analysis

Analyzing a network is usually a process of several consecutive steps, beginning with information reduction by a primarily structural analysis; and ending with the examination of the content and the resources that are shared among the actors. The process and outcome of the analysis depends on the representation of the data as well as the availability of suitable metrics.

2.2.3.4.1 Data Representation for Information Visualization and Measurement

It is good practice to employ both graphical and matrix representations (see section 2.1.2). They can foster the understanding of the network in different stages of the analysis.

The visual representation of the data helps to gain a first impression of the network and to intuitively decide where to start first investigations. In general, IT-supported information visualization allows an interactive, visual representations of abstract data to amplify cognition in many respects (see Card et al. 1999): Search time can be reduced by visually grouping related data. Recognizing information is often easier than recalling information. Complex information can be simplified and organized by grouping certain information, abstracting the information in an insightful way and selectively omitting certain details. Visualization can help people to make a large amount of graphical inferences which else would be very hard to derive (perceptual inference). It can also enable humans to make specialized graphical computations, e.g. using diagrams to generate hypotheses about data. Finally, a configurable medium allows the user to explore the parameter space and the data. The user can change parameters and directly see the changes in the graphical output. In summary, information visualization serves two purposes: communication of ideas and creation and discovery of new ideas. Based on Jacques Bertin's semiology of graphics to encode information (Bertin 1967/1983), Trier (2008) and Trier and Bobrik (2008b) propose the SNI data model and the use of the *communigraph* as means of graphical representation (see section 2.2.1). The *communigraph* is basically a dynamic visual model which extends the static sociogram used in SNA (see section 2.1.2). It therefore incorporates Bertin's visual variables and the temporal variation as an additional variable on both nodes and links.

In addition to the graphical representation, a matrix notation facilitates a more detailed analysis of single actors or groups and allows calculating more complex metrics. Furthermore, to identify important actors and significant substructures. Meaningful patterns of collaboration ranking list can be established due to different criteria, e.g. number of events, number of direct contacts, relationship strength, or brokering activities (see section 2.1.4 for a detailed overview of SNA metrics which can be used as criteria).

2.2.3.4.2 Strategies for Analysis

Based on the SNI data model (see section 2.2.1) the SNI framework provides three SNI dimensions whose methods and metrics allow a thorough analysis of the network. They can be combined in any order and either directly applied to the data or used as filters to generate a subsample of the network. These three dimensions are described in more detail in section 2.2.2. The level of detail dimension distinguishes between the analysis being conducted on ego, group and network level. The temporal dimension allows investigating the data as a static

network or a dynamic animation of its evolution over time: The static analysis of the structural properties helps to reveal the position and embeddedness of an actor in the broader structural context of the network whereas the dynamic analysis describes the evolution of these structures from the beginning to its final manifestation. The dynamic perspective helps to identify and explain important changes in the behavior of the participants of the network. For instance, the researcher can compare project milestones, or seasonal as well as exceptional events with structural hot spots in the network evolution or the general information flow through the network. The level of investigation dimension distinguishes between the analysis of the network structure and its content: While the structural analysis can be characterized as descriptive explaining when and where significant changes e.g. due to group formation processes occur, the analysis of the content is of explorative nature investigating why these changes occur. Section 2.1.4 provides a detailed overview of SNA metrics that can be applied to all three SNI dimensions and help to guide and structure the process of analysis.

Anklam (2005c: 34) emphasizes that the goal of the initial analysis is to ask questions, not to find answers, and then to investigate further. She suggests examining different scenarios and working with “what-if” views: The researchers could, for instance, examine the changes in the network if certain actors or ties are removed. Trier and Bobrik (2007c: 391) recommend that the researcher first has to gain a thorough understanding of the overall structure of the network to allow a subsequent more detailed analysis. Therefore, the analysis should start with static analysis on different levels of detail, followed by a dynamic structural analysis comparing the results from the static analysis with the evolving network structure, and then incorporate the content of the data into the analysis as well. This approach is only meant as a basic guide of conducting this phase. It has to be customized and refined depending on what research questions need to be answered in a specific network study. Moreover, it often depends on the intuition and experience of the analysts whether plausible results can be extracted or not. As discussed in section 2.2.2 the three SNI dimensions level of detail, level of investigation and temporal dimension help to structure and guide the analysis. The following exemplary questions are related to these dimensions and help to foster the analysis (see Trier and Bobrik 2007c: 392).

Static structural analysis concentrates on question about structural position and impact like:

- How can the overall network structure be described due to number of nodes and links, density, diameter, number of isolated nodes, etc.?
- Who are the most prominent actors (“key players”) or groups of actors due to communicational activity, number of contacts, link strength, brokering activity, etc.?
- Whose activities have the highest structural impact?
- How far does the influence of an actor reach based on its network position?
- How are the actors integrated in the overall network structure?
- Can densely connected subgroups (“cluster”) be identified that are less connected with the rest of the network?

Dynamic structural analysis covers question about the evolution and the lifecycle of the network and its actors like:

- Who gets central?
- Has the actor a long steady or a short but quick growth in his degree?
- Who are the most important actors over time?
- Does the position and role of an actor change over time?
- Is the network structure evolving properly? Is it stable or eventually already decaying again?
- Whose activities have network impact (on the development of its structure)?
- How and where did the network change?
- How did the network structure react to a special situation or event? Is a cluster stable or just an additive artifact?
- Can recurring patterns on network level („network lifecycle“), group level (“cluster lifecycle”) or ego level (“actor lifecycle”) be identified?

Static and dynamic content-based analysis covers questions about the context of the network:

- Who are the core people in key topics?
- How have information and ideas diffused across the network via which core people?
- Which clusters form and decay around topics?
- How did the network react to a special situation or event (identified by the content)?
- Did the actor react to external events?
- How far does the influence of an actor reach based on its content (e.g. topic similarity to direct and indirect contacts)?

Initially, the entire network is analyzed by means of structural metrics and filters. Starting with a *static network analysis* all data from the sample period is aggregated into a single network. The analysis is led by questions concerning the overall structure of the network. One will apply structural and social-structural metrics which help to identify prominent actors or groups of actors based on social status and influence. Metrics involved here provide insight about the social status of an actor or a group of actors within the network. Filters based e.g. on link strength or number of events (activity) per actor can be employed to reduce the network.

Next, removing cutpoint nodes or bridging links would increase the number of unconnected components in the graph. In such a way important nodes and links can be identified that work as brokers and have a high impact on the overall network structure. Furthermore, it can be examined if structural holes exist which may negatively influence the distribution of information. The presence of structural holes can explain insufficient network structures that already exist or may evolve. This is an important indicator for necessary improvements that shall be conducted.

In summary, in context of network analysis within organizations the following network characteristics (and the related metrics) provide important information about its structure and important key players (see Cross et al. 2000: 39; Cross et al. 2002: 6):

- *Bottlenecks* which can disrupt the information flow and thus become the reason for slow responsiveness or inefficiency.

- *Central actors* and *brokers* who are often essential to the effectiveness of the network and who would leave a gap if they were lost.
- *Number of links* indicating e.g. insufficient or excessive ties among departments that have to coordinate their work.
- *Average distances* as a measure of how much steps it takes for a resource to reach the recipient in a network.
- *Peripherals* and *isolates* who may not only represent underutilized skills, but must also be regarded carefully in high turnover situations.
- Links across *fragmentation points* which refer to a difference in actor attributes such as skills, cultural values, function, hierarchy, tenure, or physical location.
- *Subgroups* that are coherent with the formal organizational structure, or informal cliques, which can both develop own cultures with negative effects to other subgroups or the whole organization.

In contrast to the static analysis which assumes that node appearances and links between nodes are persistent during the entire sample period, the *dynamic network analysis* incorporates the temporal dimension as an additional strategy for analysis (for more details see section 2.2.2.3 and section 2.2.2).

Therefore, analyzing the structural properties by means of static and dynamic measures and methods is the initial point of entry to gain an understanding of the data. Examining the content objects created, distributed and exchanged between the network participants provides a powerful instrument for a more detailed analysis. Here, employing text mining and cluster analysis offers the possibility to uncover hidden sources of information and knowledge domains and investigate on their relations to the actors of the network. The affiliation of single actors or groups of actors to knowledge domains identified by their communicational and collaborative content can reveal important insights about the complex informal structures that organically evolve alongside formal organizational hierarchies, workplace descriptions and narrowly defined business processes. However, most collaborative networks are too complex at first sight to allow starting with a *content analysis* on ego or group level. Nevertheless, only the analysis of the content of the network allows to assess the context of the network and to identify triggering external events.

It is very important not to lose sight of the (organizational) context during this phase. Often, contextual data on the background of each individual and the entire data, e.g. the enterprise or organization, can contribute to the analysis. For example, informal structures can be compared to formal organizational entities, or important events and milestones within the sample period can provide useful information to direct the analysis.

Additionally, one should keep in mind that each metric is only an indicator (Anklam 2002). They reveal patterns and places to ask further questions. For example, zero percent betweenness centrality does not mean “no information flows” but “no *frequent* information flows” and one should examine the reasons. In this way the analysis yields creative, insightful approaches to the challenges of organizations and people move into committed action very quickly.

Based on the insights obtained from the network visualization and analysis concepts and recommendations for action can be derived in the next step of the SNI process.

2.2.3.5 Deriving Concepts and Action Plan

Based on the previously defined goals of the analysis and its results organizational and technical improvements have to be planned and executed. However, before concepts of improvement and a suitable action plan can be derived the results of the (semi-)automated data analysis have to be validated. This includes manually cross checking the data and its context. Furthermore, another set of feedback sessions should be held with the participants of the study. The purpose of these meetings is to provide a person or small group with individual feedback on their network positions and how these positions affect the whole network. It helps to validate intuitive perceptions, i.e. what is implicitly already known. The feedback of the participants therefore allows to overcome existing inhibitions of problem awareness and to define solutions (Cross et al. 2000). Additionally, the participants should also be asked how they would respond to certain problems that have been identified. As a consequence of being involved in the process, people are more inclined to change their own behavior. For example, Cross (2004) suggests dividing a feedback session into two parts: first, one should start with a general overview of the analysis aimed at the whole group. It includes the most important results of the analysis. The second part is conducted in smaller groups, which should devise ways to adjust problematic conditions and network patterns. The subgroups will then present their ideas to the whole group and discuss which ideas should be carried out.

This approach has the advantage that it allows feedback and increases commitment of the participants. The participants are directly informed of the role they play in the network, either positive or negative. They are given the opportunity to implement changes of their own. The risks of making unfortunate or detrimental decisions can be minimized if they receive detailed feedback and advice by the researchers. Furthermore, this procedure ensures the participants' commitment in the execution of the planned actions as they were directly involved in their development. This participative approach thus results in higher identification with and internalization of the weaknesses in the network structure (for more details on participation see e.g. Krallmann et al. 2007).

However, not all situations are suited for this bottom-up approach, particularly not those studies with a high number of participants. In some cases, it may be preferable or necessary that only the management reviews the results of the network analysis and decides on intervention strategies. For example, not all findings may be appropriate to be disclosed openly when personal information have been collected.

After validating the results from the network the next step is the development of concepts and actions plans for improving e.g. the innovative capacity, the responsiveness of trouble shooting, and the productivity by eliminating knowledge barriers and deficits. This can be achieved by enabling efficient identification of information objects and knowledge workers.

Cross et al. (2000) indicate that four factors positively influence the effectiveness of network with respect to knowledge creation and use, and thus indirectly improve innovation and performance of an organization:

- (1) *Knowledge awareness*: Knowing what someone else knows.

- (2) *Access*: Gaining timely access to people and their knowledge.
- (3) *Engagement*: Active commitment of others in problem solving.
- (4) *Safety*: Relationships must be built on trust in order for information-sharing to be safe.

Anklam et al. (2005) conclude that important predictors of individual and organizational performance are increased awareness of each other's knowledge and the ability to access it. The selected intervention strategies therefore need to reflect these four aspects. Cross et al. (2006: 47-48) state that knowledge awareness can be achieved without substantial time or cost burden by two general interventions: by a skill-profiling system in order to make information on each actor available and through shifts in face-to-face and virtual meetings that break up existing groups and traditions and allow for more diversity in communication.

The right choice of actions depends on the organizational environment including geographical and cultural factors, the network structure and what goals should be accomplished by the change initiatives. According to Anklam (2002) actions tend to fall into three categories which allow to make "new paths for collaboration":

- *Organizational change*. An analysis may reveal a missing or a misplaced role. The modification of organizational structures includes establishing new contacts between existing subgroups as well as creating a new role, e.g. knowledge broker or information flow manager.
- *Knowledge management practices* to improve the network. Depending on the scope of the network the variety of KM practices ranges from inventing new communicational strategies to implementing expertise location systems, holding knowledge fairs or running seminars.
- *Individual and personal change*. Managers and individuals take notice of their place in a network, and often take private or public action. For example, after realizing their position in the network central people can reduce their information and work overload by delegating decision making on certain types of topics. Or people who are not connected to the network can be systematically brought into projects, gaining opportunities to give talks, etc.

Table 2-9 contains an overview of typical strategies and suggested actions for organizational network analysis. Prominent strategies include the implementation of yellow pages, fostering contact through face-to-face meetings and virtual forums as well as deliberately breaking off groups of peers and a joint staffing of projects (e.g. Laseter and Cross 2006). The interventions must be coordinated and complement each other. Furthermore, they must be suitable to achieve the objectives that have been determined after the review of the results of the network analysis. It is advisable to check the effectiveness and efficiency of the proposed modifications at a later date and therefore identify and define suitable KPIs based on SNA metrics during this conceptual step (see section 2.1.4).

The baseline of all these interventions is to create opportunity: the opportunity for people to understand and discuss the deficiencies in the current network structure, to change their own behavior, and to create the necessary kind of connections for a more beneficial network structure. The first aspect can be achieved by making the results of the SNA available to all people involved. But opportunity alone is not sufficient to achieve real change. It takes several well-targeted impulses. Similar to the validation process this can happen in form of

moderated meetings or concentrating on brokers in the network to promote the change process. In this context, brokers are prominent actors with the legitimacy and credibility to informally introduce new ideas (Cross et al. 2006: 43). They are also often not involved in the daily business of any distinct subgroup which lets them appreciate their different knowledge and values (Johnson-Cramer et al. 2007: 93).

Table 2-9: Strategies and recommendations for actions to improve organizational networks. Based on: Anklam (2006)

Strategies	Recommendations for actions
Create more connections	Make introductions through meetings and webinars; face-to-face events (like knowledge fairs); implement social software or social network referral software; social network stimulation.
Increase the flow of knowledge	Establish collaborative workspaces; install instant messaging systems; make existing knowledge bases more accessible and usable; enable expert search.
Discover connections	Increase visibility of knowledge domains; implement knowledge and expertise location and/or discovery systems, social software and social networking applications.
Decentralize	Social software, blogs, wikis; shift knowledge to the edge.
Fill in structural holes	Establish knowledge brokering roles; expand communication channels.
Strengthen weak ties	Assign people to work on projects together.
Judiciously balance the use of direct and indirect ties	Network goal setting; network analysis; establish roles and responsibilities.
Alter the behavior of individual actors	Create awareness of the impact of an individual's place in a network; educate employees on personal knowledge networking.
Increase diversity	Add nodes; connect and create networks; integrate isolated participants; encourage people to bring knowledge in from their networks in the world.
Support active discussion	Analyze contents and follow meaningful dialogues; identify hot topics; foster and maintain participation with valuable feedback.

In summary, the basic strategy is to bring people in contact with each other and increase their relationships with the other participants of the network. However, the more connected an actor is, the more time it takes to maintain those contacts. With only limited time and energy to spend this will inevitably result in neglecting the important relationships (Cross et al. 2002: 40). Thus, the management must ensure that the network members build the right kind of relationships to the right kind of people. The essential benefit of SNA in general and the SNI framework in particular is that it allows implementing of these targeted interventions and monitoring the development through regular follow-up analyses. If the network-oriented analysis is intended to complement a conventional systems analysis their strategies and action plans have to be coordinated.

2.2.3.6 Execute and Implement Actions

A network study ends after the identification of necessary changes and planning of concrete actions to implement them. However, a social network analysis is often only one step in a greater change process. A first study can provide baseline metrics against which further analyses can be measured. Thus, the SNI framework can be a diagnostic tool for the

management to stimulate and monitor change over time (Krebs 2002; Anklam and Wolfberg 2006). The final step of implementation includes organizational as well as technological activities. As they are usually not network-specific they are not covered in detail in this work (for more information see e.g. Krallmann et al. 2007). Again, if the network-oriented analysis is intended to complement a conventional systems analysis all activities have to be coordinated.

2.3 IT Support

Software programs play an important role in the analysis of networks. Not only do they construct the initial networks from the data but they also offer the possibility to change and customize them by removing and adding actors and links and simulating the possible behavior of the network to these changes. Additionally, the network can be compared to artificial networks of equal size with known statistical properties, e.g. scale-free networks or random networks, to be able to identify meaningful patterns. Color, size, shape, thickness, or position of nodes and links can be altered to emphasize certain characteristics of the structure and provide new insights. Furthermore, most SNA programs offer a mathematical and statistical analysis of the network. They calculate the values of whole-network metrics, measure certain aspects of individual nodes, identify nodes with specific roles, or identify cliques and clusters.

In the first part of this section a general overview of available SNA software based on is given. In the second part the Commetrix software developed by the IKM Research group is presented.

2.3.1 Overview of SNA Software

A very comprehensive list of programs can be found in Huisman and van Duijn (2010) and is available online⁷. Huisman and van Duijn (2005) provide a review of 23 programs for the analysis of social networks. Both commercial and freely available packages are considered. An overview of these software packages can be found in Table 2-10. All programs are examined according to the type of data they can process, their functionalities and availability of support. According to Huisman and van Duijn (2005) there are three groups of procedures SNA software does or does not possess: (1) data entry and data manipulation, (2) visualization techniques, (3) SNA routines. SNA routines can be further distinguished into descriptive methods to calculate (simple) network statistics (e.g. centrality or transitivity), procedure-based analysis based on more complex (iterative) algorithms (e.g. cluster analysis or eigendecompositions), and statistical modeling based on probability distributions (e.g. exponential random graph models or QAP correlation). The categorization of SNA routines that were inspected are based on Wasserman and Faust (1994):

- (1) Structure and location: centrality and cohesive subgroups (cliques).
- (2) Roles and positions: structural equivalence, blockmodeling, eigendecompositions.
- (3) Dyadic and triadic methods.
- (4) Statistical methods: exponential random graph models, QAP correlation, statistical analysis of network evolution.

⁷ <http://www.gmw.rug.nl/~huisman/sna/software.html>

Table 2-10: Overview of selected programs for social network analysis: objective, data format (type, input format, missing values), functionality (visualization techniques, analysis methods) and support (availability of the program, manual, online help). Source: Huisman and van Duijn (2005)

Program	Objective	Data Format			Functionality		Support		
		Type ¹	Input ²	Miss.	Visual.	Analyses ³	Avail. ⁴	Man.	Help
Agna 2.1.1	general	C	m	no	yes	d, sl, seq.	free	yes	yes
Blanche 4.6.5	network dynamics	C	m	no	yes	simulation	free	yes	yes
FATCAT 4.2 ⁵	contextual analysis	C	ln	yes	no	d, s	free ⁵	no	yes
GRADAP 2.0 ⁵	graph analysis	C	ln	yes	no	d, sl, dt	com ⁵	yes	no
Iknow	knowledge networks	E	n	-	yes	d, sl	free	yes	yes
InFlow 3.0	network mapping	c, e	ln	no	yes	d, sl, rp	com	yes	yes
KliqFinder 0.05	cohesive subgroups	C	m, ln	no	yes	sl, s	-	yes	no
MultiNet 4.38	contextual analysis	c, l	ln	yes	yes ⁸	d, rp, s	free	no ¹¹	yes
NEGOPY 4.30 ⁵	cohesive subgroups	C	ln	yes	yes	d, sl, rp	com ⁵	yes	yes
NetDraw 1.0	visualization	c, e, a	m, ln	yes	yes	d, sl	free	yes	no
NetMiner II 2.4.0	visual analysis	c, e, a	m, ln	no	yes	d, dl, rp, dt, s	com ⁹ 10	yes	yes
NetVis 2.0	visual exploration ⁶	c, e, a	m, ln	no	yes	d, sl	free ^{6,9}	no	yes
Pajek 1.0	large data visual.	c, a, l	m, ln	yes ⁷	yes	d, sl, rp, dt	free	no	no
PermNet 0.94	permutation tests	C	m	yes	no	dt, s	free	no	yes
PGRAPH 2.7	kinship networks	C	ln	-	no	d, rp	free	no ¹²	yes
ReferralWeb 2.0	referral chains	E	ln	-	yes	d	- ⁹	yes	yes
SM Link-Alyzer 2.1	hidden populations	E	ln	-	yes	d	com ¹⁰	yes	yes
SNAFU 2.0	general for MacOS ⁶	C	m, ln	no	yes	d, sl	free	no	no
Snowball ⁵	hidden populations	E	ln	-	no	s	free ⁵	yes	no
StOCNET 1.4	statistical analysis	C	m	yes	no	d, dt, s	free	yes	yes
STRUCTURE 4.2 ⁵	structural analysis	c, a	m	yes	no	sl, rp	free ⁵	yes	no
UCINET 6.55	comprehensive	c, e, a	m, ln	yes	yes ⁸	d, sl, rp, dt, s	com ¹⁰	yes	yes
visone 1.1	visual exploration	c, e	m, ln	no	yes	d, sl	free	no	no

¹ c=complete, e=ego-centered, a=alignment, l=large networks. ² m=matrix, ln=link/node, n=node.
³ d=descriptive, sl=structure and location, rp=roles and positions, dt=dyadic and triadic methods, s=statistical, seq. = sequential.
⁴ com=commercial product, free=freeware/shareware. ⁵ DOS-program which is no longer updated.
⁶ Open source software. ⁷ Only missing value codes for attributes.
⁸ No graph drawing routines. ⁹ Freely accessible on the internet (some with reduced functionality).
¹⁰ An evaluation/demonstration version is available. ¹¹ The manual of some modules is available.
¹² The manual is available after registration.

Huisman and van Duijn (2005) explore five programs in more detail: Multinet, NetMiner, Pajek, StOCNET, STRUCTURE, and UCINET. The example data used for comparison are Freeman's EIES network (Freeman and Freeman, 1979). The data comes from a computer conference among social network researchers and was collected as part of a study of the impact of the Electronic Information Exchange System (EIES). It consists of 32 actors. Two types of relations were recorded: the number of messages sent and acquaintanceship. The results from the comparison of the five programs can be found in Table 2-11.

Although UCINET gains good and excellent marks on all categories, the authors give no clear recommendation but suggest choosing the software which is most suitable for the problem at hand. For example, if one is looking for software with the primary aim to obtain many descriptive network measures, UCINET or NetMiner would be good candidates. On the other hand, if network visualization is an important objective, Pajek and NetMiner are competing

packages, where MultiNet and UCINET (with NetDraw) also give the opportunity for visual exploration.

Table 2-11: Scores for selected SNA software from Table 2-10. Source: Huisman and van Duijn (2005)

Program	Functionality					Support		User-friendliness
	Data	Visual.	Descr.	Proc.	Stat.	Manual	Help	
MultiNet	+-	+	+-	+	+-	+-	++	+
NetMiner	++	++	++	++	+-	+	+	++
Pajek	+	++	+	++	0	-	0	+-
StOCNET	+-	0	+-	0	++	+	+	+
STRUCTURE	-	0	+-	++	+	++	0	+-
UCINET	++	+ ¹	++	++	+-	+	+	+

¹UCINET does not contain graphical procedures to visualize networks, but it has a speedbutton to execute the program NetDraw

++ = very good, + = good, +- = undecided, - = has shortcomings, 0 = is lacking

Further reviews of SNA software can be found in e.g. Kirschner (2008), Loscalzo and Yu (2008), or Xu et al. (2010). A up-to-date overview of SNA tools for business analysis has been started by Roberto Dandi. It is now group curated using a wiki spreadsheet⁸.

2.3.2 Commetrix

Commetrix⁹ is a java-based tool for event-based dynamic network visualization and analysis. It was first introduced in Trier (2005a; 2005b) and since then further developed by the IKM Research group to meet the requirements of the SNI framework (see Figure 2-29). The ongoing development of this tool yields a comprehensive set of software-based methods for exploratory static and dynamic visualization with integrated analysis of social network measures. Thus, Commetrix provides easy exploratory yet comprehensive access to network data and allows for (Commetrix 2008):

- Extracting virtual communities in electronic communication networks.
- Analyzing dynamic network change, properties, lifecycles, and structures.
- Creating rich knowledge network maps or recommendation systems from communication logs or other network data sources (including surveys).
- Searching, filtering and navigating different types of social corpora.
- Understanding and utilizing social networks.
- Tracing dissemination of topics or properties through the network.

It is extendable to all sources of network data (e.g. collaborative work on electronic documents or contents, electronic project collaboration, VoIP telephony/contact centers, instant messaging, e-mail, discussion groups, etc.).

⁸ <https://sites.google.com/site/businesssna/>

⁹ <http://www.commetrix.de>

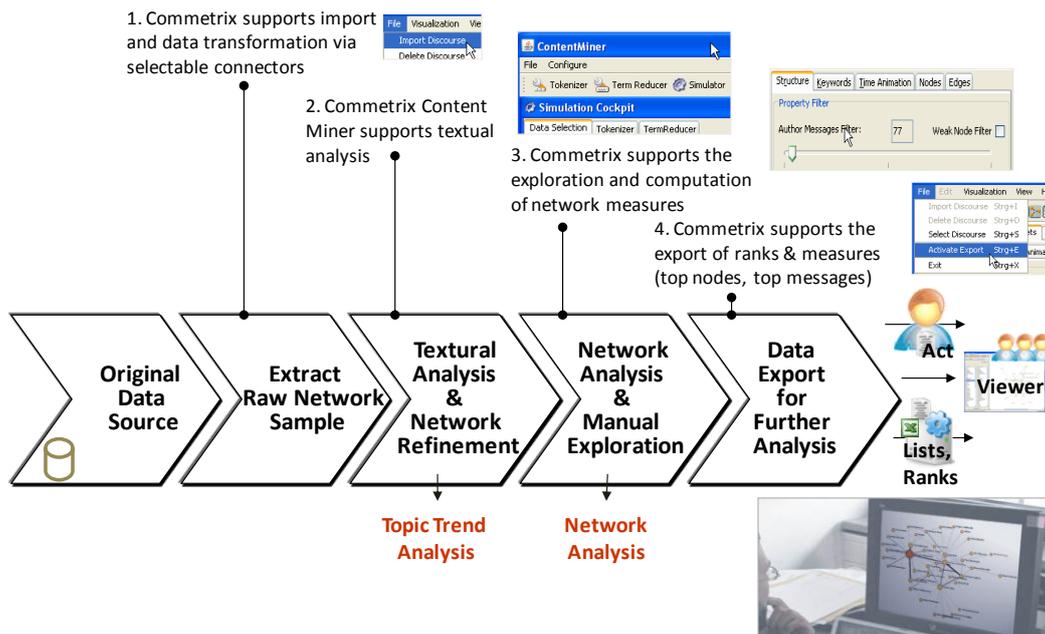


Figure 2-29: Commetrix: IT support from original data to network insights. Source: IKM Research group (2011)

The Commetrix software consists of two elements (see Trier et al. 2007). First, using the SNI data model (see section 2.2.1) together with the according mining algorithms it captures as much information from individual data as possible in a systematic and standardized way to prepare the data for subsequent analysis. Second, there are visual model specifications which utilize the underlying data to enable insights into the complex structures, activities, and contents of electronic collaboration via node size, node color, link length, link color, rings, orientation, etc. As in a sociogram (see section 2.1.2), the software visualizes actors as nodes in a graph. Links represent the relationships as flexible aggregations of events. The sociogram has been extended with additional means for information visualization and the capability to adapt to longitudinal network change yields a dynamic graph termed “communigraph”. Utilizing Bertin’s (1967/1983) concept of visual variables to encode information (see section 2.2.3.4.1), properties can be visualized by label, node size, node color (brightness, transparency), or a number of rings around each node. Figure 2-30 provides a screenshot of the CMXAnalyzer 2.0, the current Commetrix application for public use¹⁰, showing an exemplary network as communigraph with various visual variables like size and color.

To be able to implement the novel technique of event-driven dynamic network analysis a dynamic version of the spring embedder layout algorithm by Fruchterman and Reingold (1991) had to be developed (Trier 2008: 12). This algorithm can accommodate new nodes into an existing network layout. In contrast to the limited visualization of linear transitions between rendered frames that are provided by other SNA tools Commetrix creates smooth transitions between time frames. New event-based communication elements (nodes or links) are thrown into the network layout at the according time to let them find their natural place resulting into a very organic view on network evolution (see e.g. supplemental videos on www.commetrix.de/enron). A major reduction in unnecessary node movement has been

¹⁰ available at www.commetrix.de

achieved by relating the inactivity and structural stability of a node to its number of contacts. As a result, larger structures become more inert and less connected nodes quickly move towards them. This keeps established parts as stable as they should appear while drawing the user's full attention to moving areas where the actual change happens. Therefore, movements in the evolving graph directly represent structural changes. This allows visualizing the social network as a living system of interactive elements in a network relation. Analogously, links and nodes older than the observed time window can be dynamically taken out of the layout procedure. As a result, Commetrix provides visualizations that directly show the recent changes in the evolution of the network.

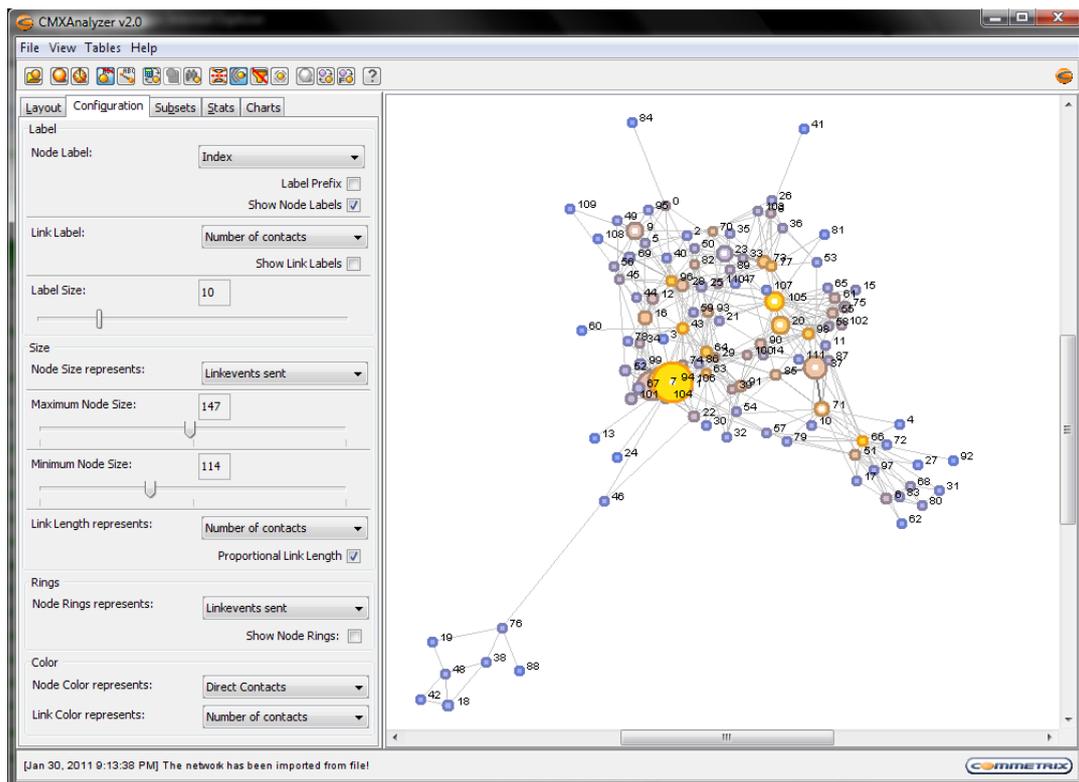


Figure 2-30: CMXAnalyzer 2.0. Screenshot. Node label: index; node size: linkevents sent; node color: direct contacts; link length: number of contacts; link color: number of contacts

To allow a flexible and detailed analysis Commetrix interprets the SNI data models and manages the massive set of computations for a large number of time windows via node-based and link-based filters. It provides very sophisticated functionality for animating the community formation as an evolving graph to visually inspect the actors' activities. This helps to actually represent and visually trace change in a network and adds additional insights to the quantitative results. The complex options for time, actor, link and linkevent filtering help to select relevant subsamples of the network. For any time period and for any selection, typical social network measures can be computed, analyzed and exported.

With regard to the social network analysis routines suggested by Huisman and van Duijn (2005) Commetrix is strong on descriptive methods but has some weaknesses in procedure-based analysis and statistical modeling. However, in addition to the Commetrix software the IKM Research group provides several additional software packages based on the SNI data model, e.g. text mining software or software for calculating the random graphs and triad census. Their results can easily be integrated in the analysis routines of Commetrix.

3 Text Mining

This chapter provides an overview of concepts and methods for automated mining of text samples in general and their application on social corpora as part of a SNA study in particular. First, section 3.1 gives a brief introduction to text mining covering its definition and related research areas as well as application areas and current challenges. Afterwards, sections 3.2 to 3.4 then concentrate on text preprocessing techniques to generate the knowledge base. Finally, section 3.5 gives an overview of text mining applications in social corpora.

3.1 Introduction to Text Mining

The way people distribute and gain information has drastically changed since the 1990s. Studies in that period of time showed that most people preferred getting information from other people by direct contacts instead of computer systems (Manning et al. 2008: xv). During the last two decades information systems and web search has become more and more important as the effectiveness of available algorithms has increased. Today, the amount of electronically stored documents and, more general, information items increases dramatically. The growth of the web can be seen as an expanding public digital library collection. Online digital information extends far beyond the web and its publicly available information. Huge amounts of information are private and are of interest to local communities, such as the records of customers of a business (Weiss et al. 2004). This information is mainly text. There are estimates that about 85% of business-relevant information originates in unstructured form (Grimes 2010). Although its primary purpose is record-keeping an automated analysis might be desirable to find patterns in the stored records. Therefore, methods for automated text processing can be employed to deal with the information overload of modern communication systems and access the hidden patterns, e.g. by information extraction and document summarization (Weiss et al. 2004). Analogous to data mining on structured data text mining also finds patterns and trends in information samples that are far less structured but have greater immediate utility for users.

3.1.1 Definition

Text mining is an interdisciplinary field that draws on information retrieval, data mining, machine learning, statistics, and computational linguistics. In order to define text mining one can refer to related concepts and research areas. Due to the different perspectives of this research area text mining has been termed text data mining (Hearst 1999; Merkl 2000), textual data mining (Losiewicz et al. 2000), text knowledge engineering (Hahn and Schnattinger 1998), knowledge discovery in texts (Kodratoff 1999) or knowledge discovery in textual databases (KDT, Feldman and Dagan 1995) with slightly different definitions of text mining.

In this work text mining is considered as *text data mining* focusing on algorithms and methods that extract useful patterns from texts in order to categorize or structure text collections or to extract useful information (Hearst 1999; Hotho et al. 2005: 23): The basic concept of data mining is to find valuable patterns in highly structured data (Weiss et al. 2004: 1). Thus, data mining deals with classification and prediction problems expecting either

ordered numerical or categorical data (see chapter 4.2.1). However, using document collections structured data is seldom available and techniques have to be involved to impose some structure on the data. Here, text mining is the essential means to preprocess the data and allow further analysis.

Current research in text mining covers problems of text representation, classification, clustering, information extraction, sentiment analysis, document summarization or the search for and modeling of hidden patterns (Hotho et al. 2005). Therefore, algorithms and procedures of related research areas have to be applied and will be presented in this chapter. In particular, modern text mining is deeply rooted in the area of information retrieval (IR) (Weiss et al. 2004) and sometimes no distinction is made between both research areas (Hotho et al. 2005: 23). The term *information retrieval* was introduced by Calvin Mooers (1950). Information retrieval is the finding of documents which contain answers to questions (i.e. rediscovering of existing knowledge), not the finding of answers itself (i.e. discovering of new knowledge) (Hearst 1999). More general, due to Salton and McGill (1984) IR is concerned with the retrieval, representation, storage, organization, and accessing of information items. Information retrieval techniques are often involved in information systems to support human activities or augment human knowledge. Therefore, information retrieval is linked to computer science as well as behavioral science (Salton and McGill 1984: xii). Today, the scale at which IR systems operate ranges from web pages over personal information systems, to enterprise, institutional and domain-specific information systems (Manning et al. 2008: 2).

Document classification in text mining is similar to document indexing in IR (see e.g. Luhn 1957; Maron and Kuhns 1960). The clustering of documents and measuring of document similarities which are often used in text mining were first introduced to IR in the 1970s (Jardine and van Rijsbergen 1971). In the same period of time, the representation of documents as a bag of words using the vector space model and term weighting approaches became popular in IR applications (Salton et al. 1975). Artificial intelligence methods of IR systems, which became popular in the 1980s, are also often found in text mining application, e.g. for text categorization (Hayes and Weinstein 1990).

3.1.2 Data Source

The variety of information collected in digital form in databases and in flat files ranges from simple numerical measurements and text documents to more complex information items such as spatial data, multimedia channels and hypertext documents. A non-exclusive list of possible data sources includes business transactions, scientific data, medical and personal data, surveillance videos and pictures, satellite sensing, games, digital media, CAD and software engineering, virtual worlds, text reports and memos as well as the internet. Text mining usually deals with some kind of narrative information, like letters, documents of all kinds, newspaper (see Salton and McGill 1984: 2), and nowadays also electronic communication and collaboration content like e-mail, wiki articles, or newsgroup postings. In practice, these information items are usually called *documents* (Salton and McGill 1984: 2). The group of documents is called (*document*) *collection* or *corpus* (Manning et al. 2008: 4). In this work documents are also more generally termed *content* or *content objects* which are grouped together in a *content collection*. This wording is more similar to Salton and McGill's information items and does not limit the view to persistently stored text corpora.

A collection of information items has to fulfill at least two criteria to be regarded as useful: currency and completeness (Salton and McGill 1984: 2). The criterion of *currency* demands that new items have to be constantly added to the data collection whereas the criterion of *completeness* demands that the collection contains all, or at least a larger proportion, of the items of interest. Currency and completeness affects the data acquisition, storage, and collection maintenance procedures. Limited resources and information overload require a thorough balance between these crucial requirements.

Content objects can range from completely unstructured data, semi-structured data to highly structured data (Manning et al. 2008: 178). *Unstructured data* refers to raw text without any markup, e.g. text documents without any meta information like author, date, etc. Given *semi-structured data* the raw text is enhanced with some markup. Relational databases are considered as *structured data* as they consist of sets of records that have values for predefined attributes. Application scenarios using structured data include digital libraries, patent databases, blogs, outputs from office suites that save documents as marked up text, and any kind of text enriched with *named entity tagging*, i.e. documents in which entities like persons and locations have been tagged. Most digital documents have additional machine-recognizable structure like *metadata* associated with each document (Manning et al. 2008: 102). Metadata are specific forms of data which provide additional information about the document or some part of the document. For example, the author, date of creation or format of the document are encoded as additional information together with the document, and some parts of the document, like title and abstract, are marked as well. According to Manning et al. (2008: 179) the Extensible Markup Language (XML) is the currently most widely used standard for encoding structured documents. The XML standard involves complex structures like nesting of attributes. However, as social software applications are build on different data models with varying degree of structuredness, all three types of data may be relevant in this study.

3.1.3 Data Collection

A text mining process will usually start with the collection of the data (Weiss et al. 2004: 15). In many text mining scenarios the relevant data is already given or it is part of the problem description itself. For example, a web page retrieval application for an intranet implicitly specifies the relevant documents to be web pages. In this case, the main issue of the data collection step is data cleansing to ensure that they are of high quality. Often, this process is only semi-automated and some manual human intervention is necessary. Here it is important to assure the integrity of the document collection process (Salton and McGill 1984: 390). Sometimes, the documents may be obtained from document warehouses or databases. In this case, data cleansing is usually already performed before deposit. In many other applications one may need a more or less detailed data collection process that not only involves data cleansing but also data retrieval techniques. For instance, for a web application comprising a number of autonomous web sites, a web crawler has to be employed that collects the documents (Manning et al. 2008: 405). Another example is an e-mail audit application that logs all incoming and outgoing messages on a mail server for a period of time. If the set of documents is extremely large data sampling techniques can be used to select a manageable subset of relevant documents, e.g. more recent documents according to a time stamp.

3.1.4 Knowledge Discovery Process for Text Data Mining

According to Hotho et al. (2005) data can be understood as a *quantity of facts*, which may be more or less structured data in a database as well as data in a simple text file with no structure available. *Knowledge discovery* or *knowledge discovery in databases* (KDD) aims at finding patterns and connections in these data (Klösgen and Zytkow 1996). Therefore, KDD is defined as the nontrivial extraction of implicit, previously unknown and potentially useful information from data (Fayyad et al. 1996). Similar, knowledge discovery from text (KDT) deals with the machine supported analysis of text and is not limited to structured data (Feldman and Dagan 1995). Thus, text mining can be related to the knowledge discovery process illustrated in Figure 3-1.

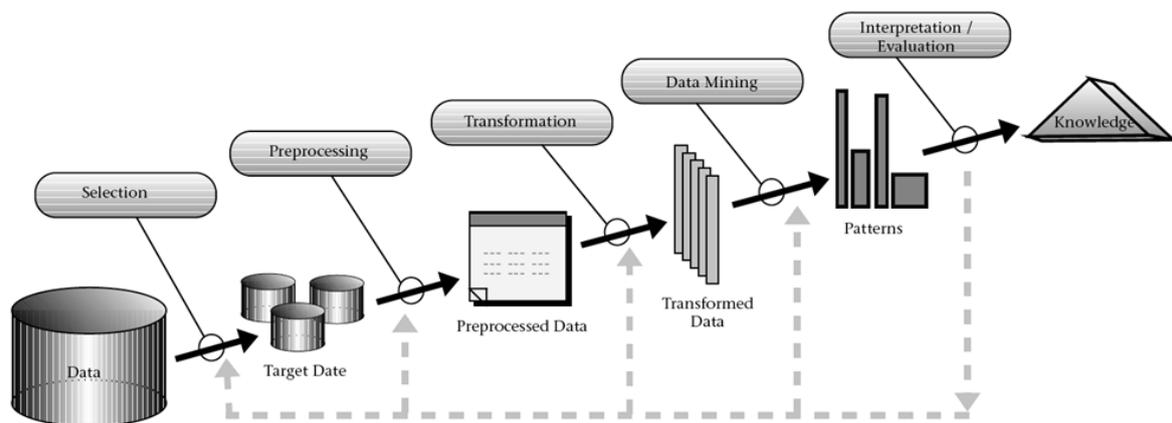


Figure 3-1: Knowledge discovery process. Source: Fayyad et al. (1996)

According to Fayyad et al. (1996) there are five steps of knowledge discovery which are passed through iteratively and require the supervision of the user:

- *Selection.* Significant data gets selected or created. This *target data* also includes metadata or data that represents background knowledge.
- *Preprocessing.* Important elements of the provided target data have to be detected and filtered out. The less noise contained in the *preprocessed data* the higher is the efficiency of data mining.
- *Transformation.* The preprocessed data also needs to be transferred into a data-mining-capable format. Additionally, the data is manually or automatically reduced via lossless aggregation or a lossy selection of only the most important elements. Being a representative selection the *transformed data* can be used to draw conclusions to the entire data.
- *Data Mining.* Data mining techniques are employed on the transformed data for knowledge discovery. The output of this step is the set of detected *patterns*.
- *Interpretation/Evaluation.* The interpretation of the detected patterns reveals whether or not the patterns are interesting, i.e. contain desired *knowledge*.

The detected knowledge out of the KDD process is usually used to support the decisions of the management. Therefore it flows into a Decision Support System (DSS) or into marketing automation for direct marketing purposes.

Similarly, the Cross Industry Standard Process of Data Mining (CRISP-DM) model involves the following steps which are iteratively performed and usually require interactive feedback from a user (CRISP-DM 1999): (1) business understanding, (2) data understanding, (3) data preparation, (4) modeling, (5) evaluation, (6) deployment. Text mining is especially involved in the data preparation step (3). Data mining algorithms will be employed in step (4). In contrast to the KDD process the CRISP-DM model explicitly involves the context of the data in the steps business understanding (1) and data understanding (2).

A survey of further knowledge discovery process models with detailed descriptions of the individual steps can be found in Kurgan and Musilek (2006).

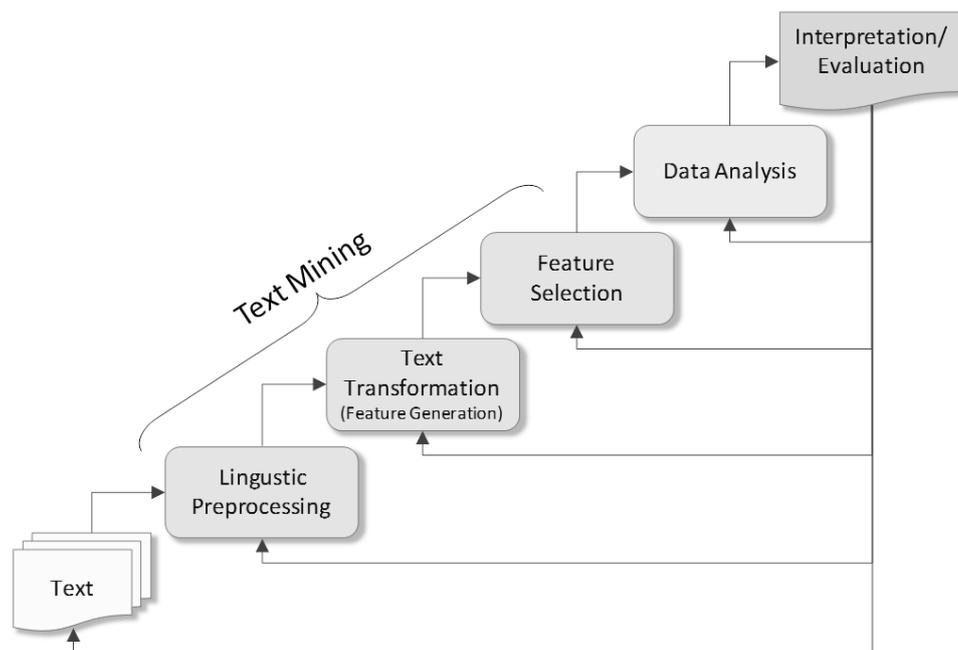


Figure 3-2: Text mining process

The KDD process by Fayyad et al. (1996) is designed for the analysis of data in general. To structure the text mining methods presented in this chapter the text mining process illustrated in Figure 3-2 is used which more precisely is designed for text corpora. The text mining process includes linguistic preprocessing (see section 3.2), text transformation for feature generation (see section 3.3) and feature selection (see section 3.4). As the text mining process is often performed before a data analysis it is also called *text preprocessing*. The subsequent steps are then data analysis including methods for data classification and cluster analysis and interpretation and evaluation of the final results. Again, this process is iteratively performed and only semi-automated as it usually requires interactive feedback from a user.

3.1.5 Application Areas

The application areas of text mining are numerous. Text mining algorithms for classification of text corpora to retrieve meaningful patterns or for trend detection to predict future behavior can be found in business, security, health, (web) news, or market research applications. They are also often used as basic methods to extract useful information for further analysis, e.g. cluster analysis (see chapter 4) or social network analysis (see chapter 1). In this section only a selection of these application scenarios can be presented.

Result snippets help to present a results list that will be informative to the user without forcing him to examine all the returned documents (Manning et al. 2008: 157). A result snippet is a short summary of the document, which is designed to meet the users' information need. Typically, the snippet consists of the document title and a short summary which is automatically extracted. *Text summarization* was initiated by Kupiec et al. (1995), and more recent work can be found in Barzilay and Elhadad (1997) as well as Jing (2000). A *dynamic text summary* consists of one or more extracts ("windows") from the document, aiming to present the pieces that have the most utility to the user by evaluating the document with respect to their information need. The advantages of dynamic text summarization are discussed in Tombros and Sanderson (1998). Turpin et al. (2007) provide some information of how to generate those snippets efficiently.

The field of *sentiment analysis* deals with the analysis of opinions found in documents. One of the basic tasks is sentiment classification. Here, units of text are sorted into classes which correspond to e.g. positivity or negativity of expressed opinion. For example, in Pang and Lee (2004) an empirical study can be found which employs dimensionality reduction techniques from text mining to solve the following two classification problems: (1) document polarity classification, which classifies documents representing complete reviews as positive or negative, and (2) sentence polarity classification, which deals with polarity classification of individual sentences. Such an analysis may require a labeled data set or labeling of the affectivity of words. Godbole et al. (2007) provide a resource of the affectivity of words for the open source thesaurus WordNet (Valitutti et al. 2004). They have developed a system for large scale sentiment analysis for news and blogs consisting of a sentiment identification phase, a sentiment aggregation and scoring phase and the evaluation of the significance of their scoring techniques.

Motivated by the Deutsche Presse-Agentur (dpa) and a group of leading German broadcasters Paaß and deVries (2005) evaluate the performance of seven commercial text classification systems on two test corpora of about half a million news stories to investigate the use of *automated story tagging* (e.g. categories, names, organizations, places) in real-world press archives. As a result, the Deutsche Presse-Agentur now is routinely using a text mining systems in its news production workflow.

Trend detection and *trend prediction* are popular applications of text mining methods, especially in *market research*. Here, one research field is analytical customer relationship management: Coussement and Van den Poel (2008) improve predictive analytics models for customer complaint management by automatic e-mail classification. Text mining also helps to automatically evaluate surveys of customer behavior (Henning 2010).

The foundation of modern capital market theory is the hypothesis of *efficient information* (Spiliopoulou et al. 2003): Stock prices immediately adapt to new information that are relevant for assessing the value. In contrast to analyzing well-structured data, ad hoc information is normally plain text without any structure. Knowledge discovery in textual databases and text mining allows automatically analyzing this type of economically important data and predict future trends.

Web 2.0 provides gathering places for internet users in e.g. blogs, forums, chat rooms and social networking platforms. These data include consumers' thoughts, beliefs, experiences

and even interactions. Feldman et al. (2010) present an approach to transform the Web 2.0 to a large field test for *monitoring market structures*. As a result, a better understanding of the market structure, the competitive landscape, and the features of the discussed product range can be gained. In order to transform the raw qualitative data from users' postings into meaningful knowledge an advanced text mining approach has been developed by the authors and combined with semantic network analysis tools.

Another common application area of text mining is *anti-spam filtering* of e-mails. Besides black-lists and hand-crafted rules machine learning techniques of text classification help to quickly adapt to new spam types. For example, Mozilla's e-mail client involves a naïve Bayes classifier in its spam filter. A comparison of different classifier methods on spam detection can be found in Michelakis et al. (2004).

Text mining is also often used in SNA, e.g. to extract named entities and relational data from text corpora to establish a network and its context. For more details see e.g. chapter 2.2.2.2.

3.1.6 Challenges

There are several text characteristics which make the retrieval of information from texts via text mining a complex and laborious task. Some of these characteristics are presented in this section.

User dependency for the data: Often, the text mining process is only semi-automated and some manual human intervention is necessary. Here it is important to assure the integrity of the document collection process (Salton and McGill 1984: 390).

Large data sets: Modern text mining is driven from the availability of cheap and fast computing as well as of enormous amounts of text in digital form, especially available through the internet (Weiss et al. 2004: 13). If the set of documents is extremely large data sampling techniques can be used to select a manageable subset of relevant documents.

High dimensionality: As each word or phrase is considered as a dimension in the feature space (or element in the dictionary) text mining has to cope with high dimensionality (see section 3.3.2).

Dependency: Relevant information is not only a single term but a complex conjunction of words or phrases. Many complex and technical concepts as well as organization and product names are multi-word compounds or phrases (Manning et al. 2008: 36).

Ambiguity: Ambiguity deals with semantic, syntactic and morphological ambiguity. To resolve these ambiguities knowledge of word structure (morphology), knowledge of phrase structure (syntax), and knowledge of meaning (semantics) is necessary (Bird et al. 2007: 30). *Synonymy* and *polysemy* are two classic problems of semantic ambiguity arising with natural language (Manning et al. 2008: 378). *Synonymy* refers to a case where two different words (e.g. "car" and "automobile") have the same meaning. *Polysemy* refers to the case where a term such as "charge" has multiple meanings. An overview of syntactic ambiguities can be found in section 3.2.3.3.

Abbreviations: Abbreviations substitute for fully expanded terms (e.g. World Wide Web) through the use of shortened term-forms (e.g. WWW). In general, abbreviations are much

more ambiguous than ordinary terms. Liu et al. (2002) report that 81.2% of abbreviations in Unified Medical Language System (UMLS) were ambiguous with an average of 16.6 senses.

Noisy data: Being mainly used for informal communication online chat, blogs, wikis and web pages are inherently noisy containing spelling errors, special characters, non-standard word forms, grammar mistakes etc. Non-intentional spelling errors as well as intentional abbreviations and distortions can be observed (Subramaniam et al. 2009). Choudhury et al. (2008) provide an overview of features of SMS noise that can be extended to other informal communication media:

- *Deletion of characters:* The commonly observed patterns include deletion of vowels (“msg” for “message”), deletion of repeated character (“tomorow” for “tomorrow”) and truncation (“tom” for “tomorrow”).
- *Phonetic substitution:* Words or letters are substituted. Examples of this type of noise are “2” for “to” or “too”, “lyk” for “like”, “rite” for “right”.
- *Abbreviation:* Some frequently used abbreviations are “tb” for “text back”, “lol” for “laughing out loud”.
- *Dialectal and informal usage:* Often multiple words are combined into a single token following certain dialectal conventions. For example, “gonna” is used for “going to”, “aint” is used for “are not”, etc.
- *Deletion of words:* Function words (e.g. articles) and pronouns are commonly deleted. “I am driving back home” for example may be typed as “driving home” or simply “drvng hm”.

As there is often no clear distinction between formal and informal communication, these types of spelling-errors can also be found in business context.

3.2 Linguistic Preprocessing

Often text mining methods may be applied without further preprocessing. However, additional linguistic preprocessing may be used to enhance the available information about terms (Manning and Schütze 1999). In contrast to later text mining steps, where only separate terms are available, this text preprocessing step uses full text representation (e.g. sentences) to analyze the semantic and syntactic impact of terms. These preprocessing algorithms include Natural Language Processing techniques like word sense disambiguation (see section 3.2.1), part-of-speech tagging (see section 3.2.2), parsing (see section 3.2.3) and chunking (see section 3.2.4).

For many text mining tasks advanced linguistic preprocessing is of limited value compared to the simple bag-of-words approach with basic preprocessing (see Hotho et al. 2005: 29). The reason is that the co-occurrence of terms in the vector representation itself serves as an automatic disambiguation, e.g. for classification (Leopold and Kindermann 2002). However, locating and ignoring information in sentences (e.g. keeping only nouns or verbs for further analysis) helps to identify some higher-level patterns (Bird et al. 2007: 115).

3.2.1 Word Sense Disambiguation

Word sense disambiguation (WSD), or simply *disambiguation*, deals with the semantic ambiguity of words, i.e. a word may have several meanings or senses (Manning and Schütze

1999: 229). Disambiguation aims at determining which of the senses of a particular word is invoked in a particular use.

Word sense disambiguation techniques can be distinguished as supervised disambiguation (section 3.2.1.1), dictionary-based disambiguation (section 3.2.1.2), and unsupervised disambiguation (section 3.2.1.3). Section 3.2.1.4 presents the one-sense-per-discourse hypothesis which provides powerful properties for WSD.

3.2.1.1 Supervised Disambiguation

Supervised disambiguation uses an already disambiguated corpus for training. The training set assigns the contextually appropriate sense as a semantic label to each occurrence of an ambiguous word. Therefore, supervised disambiguation is an instance of statistical classification. The basic idea is to build a classifier which correctly classifies new cases based on their context of use (Manning and Schütze 1999: 235). There are several classifiers available which employ very different sources of information. For example, the Bayesian classification approach treats the context of occurrence as a bag of words in the context window (see Gale et al. 1992) whereas an information-theoretic approach only looks at one informative feature in the context (out of a large number of candidates), which may be sensitive to text structure (see Brown et al. 1991).

3.2.1.2 Dictionary-based Disambiguation

Training data provide context-specific sense categorizations of the words in the data. Using a dictionary or thesaurus, only a general characterization of the senses is available (Manning and Schütze 1999: 241). There are several methods for *dictionary-based disambiguation*. For example, disambiguation can be directly based on the sense definitions in the dictionary (see Lesk 1986). Thesaurus-based disambiguation applies semantic categorization of words obtained from a thesaurus to the semantic categorization and disambiguation of contexts (see Yarowsky 1992). Other disambiguation algorithms make use of e.g. word correspondences in a bilingual dictionary (see Dagan and Itai 1994).

Using a thesaurus, each term can be automatically expanded with synonyms and related words. The following methods can be used for building a thesaurus (Manning et al. 2008: 174):

- *Controlled vocabulary*. A controlled vocabulary provides a canonical term for each concept. It is maintained by human editors. The use of a controlled vocabulary is quite common for well resourced domains, like biomedical research.
- *Manual thesaurus generation*. Human editors have built up sets of synonymous names for concepts without designating a canonical term.
- *Automatic thesaurus generation*. Word co-occurrence statistics over a collection of documents in a domain are used for automatic thesaurus generation (see 1993; 1998).

Traditionally, *Roget's Thesaurus* has been the best known manually generated English language thesaurus (Roget 1946). For example, Yarowsky (1992) uses this thesaurus for word sense disambiguation. In recent computational work, people almost always use the open

source thesaurus WordNet¹¹ (see Fellbaum 1998) which provides rich link structure. There is a high cost to manually producing a thesaurus and then updating it for scientific and terminological developments within a field. In general a domain-specific thesaurus is required: General thesauri and dictionaries give far too little coverage of the rich, domain-particular vocabularies of most scientific fields.

A thesaurus can be automatically generated by analyzing a collection of documents (Manning et al. 2008: 175). One approach is simply to exploit word co-occurrence (see e.g. Newman 2004a). Words co-occurring in a document or paragraph are likely to be in some sense similar or related in meaning. Thus, text statistics can be used to find the most similar words (see section 3.4.3). A more sophisticated approach is to use a shallow grammatical analysis of the text and to exploit grammatical relations or grammatical dependencies. Simply using word co-occurrence is more robust but using grammatical relations is more accurate.

Table 3-1: An example of an automatically generated thesaurus employing Latent Semantic Indexing. Based on: Manning et al. (2008: 176)

Word	Nearest Neighbors
Absolutely	absurd, whatsoever, totally, exactly, nothing
Bottomed	dip, copper, drops, topped, slide, trimmed
captivating	shimmer, stunningly, superbly, plucky, witty
Doghouse	dog, porch, crawling, beside, downstairs
Makeup	repellent, lotion, glossy, sunscreen, skin, gel
Mediating	reconciliation, negotiate, case, conciliation
Keeping	hoping, bring, wiping, could, some, would
lithographs	drawings, Picasso, Dali, sculptures, Gauguin
pathogens	toxins, bacteria, organisms, bacterial, parasite
Senses	grasp, psyche, truly, clumsy, naive, innate

An example of an automatically generated thesaurus is shown in Table 3-1. Latent Semantic Indexing (LSI) was employed for dimensionality reduction. The quality of the associations varies. Term ambiguity easily introduces irrelevant statistically correlated terms.

In Web 2.0 applications collaborative tagging systems like del.icio.us based on folksonomies contain nested groups of tags related to common topics which provide a rich data source for domain-specific thesaurus generation (Heymann and Garcia-Molina 2006; Damme et al. 2007). However, they particularly suffer from the problem of ambiguity when individuals use the same tag in different contexts.

3.2.1.3 Unsupervised disambiguation

Unsupervised disambiguation assumes that similar senses occur in similar contexts. Senses can then be induced from text by clustering word occurrences using some measure of similarity of context (see e.g. Brown et al. 1991; Schütze 1998). Then, new occurrences of the word can be classified into the closest induced cluster of senses. To compare its performance

¹¹ <http://wordnet.princeton.edu>

to other WSD approaches the induced senses have to be mapped to a known dictionary of word senses. Alternatively, if a mapping to a set of dictionary senses is not desired, cluster-based evaluations can be performed. Seo et al. (2004) describe a sense disambiguation method using the context words surrounding the target noun and its WordNet relatives, such as synonyms, hypernyms and hyponyms. The result of sense disambiguation is a relative that can substitute for the target noun in a context. The selection is made based on the co-occurrence frequencies between candidate relatives and each word in the context.

3.2.1.4 One Sense per Discourse, One Sense per Collocation

Disambiguation algorithms can be improved by exploiting the distributional properties of senses. Yarowski (1995) presents an unsupervised algorithm that can accurately disambiguate word senses in a large, completely untagged corpus, which avoids the need for costly hand-tagged training data by exploiting two properties of human language:

- (1) *One sense per collocation*: Nearby words provide strong and consistent clues to the sense of a target word, conditional on relative distance, order and syntactic relationship.
- (2) *One sense per discourse*: The sense of a target word is highly consistent within any given document.

The observation that words strongly tend to exhibit only one sense in a given discourse or document is due to Gale et al. (1992).

Table 3-2: Accuracy and applicability of the one-sense-per-discourse hypothesis. Source: Yarowsky (1995: 189)

Word	Senses	Accuracy	Applicability
Plant	living/factory	99.8%	72.8%
Tank	vehicle/contr	99.6%	50.5%
Poach	steal/boil	100.0%	44.4%
Palm	tree/hand	99.8%	38.5%
Axes	grid/tools	100.0%	35.5%
Sake	benefit/drink	100.0%	33.7%
Bass	fish/music	100.0%	58.8%
Space	volume/outer	99.2%	67.7%
Motion	legal/physical	99.9%	49.8%
Crane	bird/machine	100.0%	49.1%
Average		99.8%	50.1%

Yarowski (1995) provides a test for the one-sense-per-discourse hypothesis. It is performed on a set of 37,232 hand-tagged examples. For these words, Table 3-2 measures the claim's accuracy (when the word occurs more than once in a discourse, how often it takes on the majority sense for the discourse) and applicability (how often the word does occur more than once in a discourse).

This hypothesis is employed in an unsupervised disambiguation algorithm which does not need a labeled set of training examples. Its tested accuracy exceeds 96% (Yarowsky 1995: 188).

3.2.2 Part-of-Speech Tagging

Part-of-speech-tagging (POS) is the process of marking up the words in a text as corresponding to a particular part-of-speech, e.g. noun, verb, or adverb, based on both its definition as well as its context, i.e. the relationship with adjacent and related words in a phrase, sentence or paragraph. It is also called *grammatical tagging*, *word-category disambiguation*, or simply *tagging* (Manning and Schütze 1999: 139, 341). POS makes use of a tag set which contains the possible grammatical forms of a word in a sentence. It is a case of limited syntactic disambiguation, thus a complete understanding of the structure inherent in language is not necessary. An example of POS using the Brill tagger (see Brill 1995) is given in Table 3-3.

Table 3-3: Part-of-speech tagging. Example on Enron data set¹² using the Brill tagger

Original Text	Annotated Text
Stan and Danny, here is a recap of activities in Gas Logistics this last week including Gas Control activity.	Stan/NNP and/CC Danny/NNP here/RB is/VBZ a/DT recap/NN of/IN activities/NNS in/IN Gas/NNP Logistics/NNP this/DT last/JJ week/NN including/VBG Gas/NNP Control/NNP activity/NN

Historically, the Brown tag set used for tagging the American Brown corpus (Francis and Kučera 1982) and the CLAWS¹³ series of tag sets developed by the University of Lancaster called CLAWS1 through CLASW5 tag set (Garside et al. 1987; more recently described in Garside 1995) have been the most popular tag set. Today, the Penn Treebank tag set is the most widely used tag set. A general introduction to this tag set can be found in Marcus et al. (Marcus et al. 1993) whereas Santorini (Santorini 1990) provides a more detailed description. It is a simplified version of the Brown tag set. A comparison of the CLAWS5 (), the Brown and the Penn tag set can be found in Table A-1 and Table A-2 in the appendix A. Using fine grained classes like e.g. „plural proper noun (PPN)“ from the Penn Treebank tag set up to 96% per-tag accuracy can be obtained (Manning et al. 2008: 37).

Taggers are usually trained by machine learning methods with hand-tagged text. The accuracy depends on the amount of training data available, the tag set, the difference between training corpus and dictionary versus corpus of application, and the number of unknown words (Manning and Schütze 1999: 372). In general, rule-based and stochastic taggers can be distinguished. A *rule-based tagger* involves a tag set and a vocabulary containing possible word-tag-combinations (see Klein and Simmons 1963). A trivial approach would be that given a sentence each word is annotated with the tag out of all possible tags which most often appears in the vocabulary (see e.g. Charniak et al. 1993). Due to word ambiguities, this approach has an inaccuracy of about 33% (DeRose 1988). The empirical study of DeRose

¹² see chapter 5, section 5.3.1

¹³ CLAWS = Constituent-Likelihood Automatic Word-tagging System

(1988) on English language shows that only 11.5% of all words have more than one possible tag but these words make up to 40% of the text corpus. Therefore, rule based-tagging involves linguistic rules to decide how ambiguous words should be tagged. In a first step, all possible tags are applied to each word. In a second step, the ambiguous words with more than one possible tag are tagged due to a set of linguistic rules involving the knowledge about the unambiguous tags. For example, if the preceding word is an article and the present word might be a noun, it should be tagged as a noun. The definition of the tagging rules involves advanced knowledge and cannot be transferred to other languages (Manning and Schütze 1999: 371).

Stochastic taggers automatically generate the annotation rules based on the notion that some tag combinations are more likely than others. In the mid 1980s, hidden Markov models (HMMs) were first used to disambiguate parts of speech (see Garside et al. 1987; Marshall 1987). HMMs involve counting cases and then build a table of the probabilities of certain sequences. For example, once an article such as 'the' has appeared in the text, the next word might be a noun with 40% probability, an adjective with 40%, and a number with 20%. Thus, the tagging algorithm can decide which tag is more likely. Furthermore, this method is enhanced by knowing the possible tags of the following words. Together, the decision making process of the algorithm can be improved. Stochastic taggers that only use the preceding tag and the current tag are called *bigram taggers*. A *trigram tagger* takes more context information into account to predict the accurate tag on two preceding tags and the current tag (Church 1988).

Calculating the most probable tags is called decoding. Besides a simple brute force approach the more efficient Viterbi algorithm (see Forney 1973) can be employed to reduce the computational effort. Stochastic taggers suffer from the manual annotation of the text corpus for training. Brill (1995) suggests a *transformation based tagger* which only requires a vocabulary with the possible tags of each word.

Compared to parsing (see section 3.2.3) tagging is much easier to solve due to its limited effort. Additionally, the accuracy of the results is quite high (Manning and Schütze 1999: 342).

3.2.3 Parsing

Parsing, or *syntactic parsing* or *syntax analysis*, is the process of analyzing a sentence to determine its grammatical structure with respect to a given more or less formal grammar. The parser gets a sentence as input, identifies the words and assigns a grammatical structure (“*parse*”) to the sentence. As output of the parser, this structure is usually represented as a *parse tree* which shows the syntactic structure of the sentence. Each node represents a grammatical element (“*non-terminal*”) and each leaf represents a word (“*terminal*”) in the sentence.

3.2.3.1 Grammar

A *grammar* is a formal system that specifies which sequences of words are well-formed in the language (Salton and McGill 1984; Manning and Schütze 1999). It provides one or more phrase structures for well-formed sequences. Apart from their compactness, grammars usually

capture important structural and distributional properties of the language. They can be used to map between sequences of words and abstract representations of meaning.

Table 3-4: Example of syntactic categories. Source: Bird et al. (2007: 175)

Symbol	Meaning	Example
S	sentence	<i>the man walked</i>
NP	noun phrase	<i>a dog</i>
VP	verb phrase	<i>saw a park</i>
PP	prepositional phrase	<i>with a telescope</i>
...
Det	determiner	<i>the</i>
N	noun	<i>dog</i>
V	verb	<i>walked</i>
P	preposition	<i>in</i>

A *context-free phrase structure grammar*, or simple *context-free grammar* (CFG) is a collection of *productions* of the form $S \rightarrow NP \rightarrow VP$. This means, that a constituent S can consist of sub-constituents NP^{14} and VP^{15} . Similarly, the production $V \rightarrow \textit{help}$ means that the constituent V can consist of the string *help*. Based on its production rules, the CFG decomposes a sentence into nested and juxtaposed sentence portions. A phrase structure tree is termed well-formed relative to a grammar if each non-terminal node and its children correspond to a production in the grammar. A simple overview of syntactic categories is given in Table 3-1. Using this grammar to parse the sentence “The man hit the ball.” the sequences “the man” and “the ball” are identified as noun phrases and the sequence “hit the ball” as a verb phrase (see Figure 3-3).

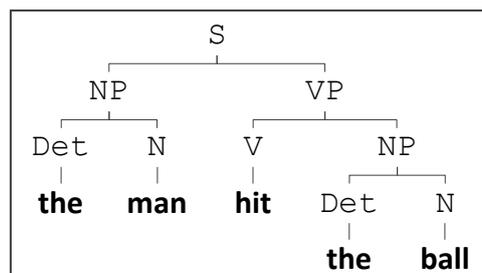


Figure 3-3: Example of Context-free Grammar. Source: Salton and McGill (1984: 90)

This simplified language analysis suffers from some severe disadvantages (Salton and McGill 1984: 91). First of all, if the structure of a sentence is not of the basic phrase structure type it cannot be analyzed by the phrase structure model. Second, a unique pattern is not obtainable for all sentences. In this case, multiple parse trees will be generated with no hint which one fits semantically best. Third and most important, the phrase structure model is not sufficiently

¹⁴ A *noun phrase* (NP) is a phrase whose head is a noun or a pronoun, optionally accompanied by a modifier set, e.g. determiners like articles or numerals, adjective, etc.

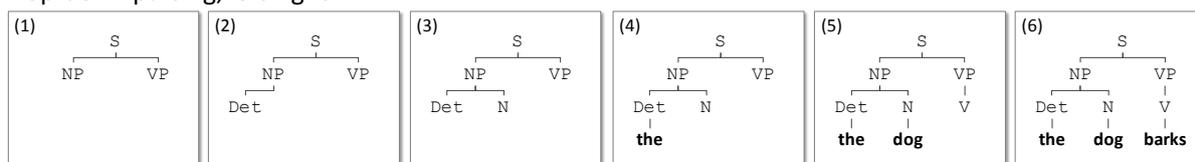
¹⁵ A *verb phrase* (VP) is a syntactic unit that corresponds to the predicate. In addition to the verb, this includes auxiliaries, objects, object complements, and other constituents apart from the subject.

rich to make it possible to recognize semantic relationships between sentence components that may not be reflected by some sort of physical juxtaposition of the components in the sentence. Thus, this may lead to an overspecification producing an underassignment of phrases for the respective documents. To overcome this last problem, *context-sensitive grammar* (CSG) was introduced by Noam Chomsky (1956). The left-hand sides and right-hand sides of any production rules may be surrounded by a context of terminal and non-terminal symbols which allow further specifications. However, Pullum and Gazdar (1982) have argued that despite a few non-context-free constructions in natural language the vast majority of forms in natural language are indeed context-free.

3.2.3.2 Types of Parsing

In general, the analysis of a sentence starts at the beginning and incrementally looks at each word from left to right (*unidirectional left-right-parsing*). This is the standard procedure for languages where sentences are read from left to right (Carstensen et al. 1986: 304). Another approach is the *head-corner parsing* (Bouma and van Nourd 1993) which starts with lexical head of the sentence (e.g. the finite verb) and proceeds bidirectional to the left (beginning of the sentences) and right (end of the sentence).

Top-down parsing, left-right



Bottom-up parsing, left-right

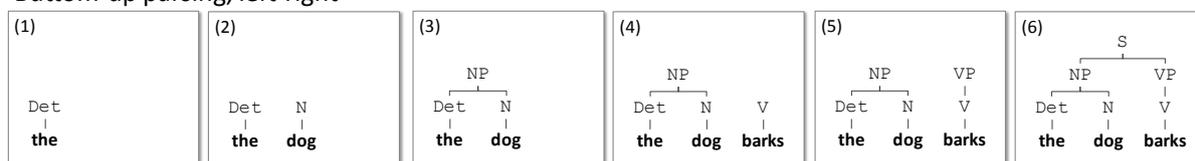


Figure 3-4: Top-down versus bottom-up parsing. Based on: Carstensen et al. (1986: 305-306)

The basic task of the parser is to determine if and how the input can be derived from the start symbol of the grammar as the root node of the parse tree. Basically, there are two strategies which are illustrated in Figure 3-4 (Carstensen et al. 1986: 305): *Top-down parsing*, e.g. *recursive descent parsing*, analyzes unknown data relationships by hypothesizing general parse tree structures and then considering whether the known fundamental structures are compatible with the hypothesis. Therefore, this approach starts with the root of the parse tree and tries to find the path to the leaves (words), i.e. it expands the structure. LL parsers which precede from left-to-right and construct a leftmost deviation of the sentence are examples of this approach. *Bottom-up parsing*, or *shift-reduce parsing*, analyzes unknown data relationships by trying to identify the most fundamental units first, and then to infer higher-order structures from them. Therefore, this approach starts with the leaves (words) of the parse tree and tries to build trees upward toward the start symbol, i.e. it reduces the structure. LR parsers which precede from left-to-right and construct a rightmost deviation of the sentence are examples of this approach. Both strategies can be combined, e.g. the *left-corner parsing strategy* (see Rosenkrantz and Lewis 1970; Demers 1977). All these strategies

involve search strategies, like breadth-first, depth-first or best-first search, which determines how the parse tree will be traversed.

3.2.3.3 Syntactic Ambiguities

Syntactic ambiguity is a severe problem for parsing (Carstensen et al. 1986: 307). In *syntactic ambiguity*, the same sequence of words is interpreted as having different syntactic structures (Allen 1962; Allen and Caldwell 1963). In contrast, in *semantic ambiguity*, the structure remains the same, but the individual words are interpreted differently (see WSD, section 3.2.1). There are three types of syntactic ambiguity which can cause parsing difficulty: *Attachment ambiguity* arises when a phrase can be attached to more than one node in the parsing tree (Manning and Schütze 1999: 278). Especially, *PP-attachment ambiguity* has received much attention in NLP research. It occurs when a sentence contains a prepositional phrase (PP), after a verb complemented by a noun, making it syntactically ambiguous to determine what the PP attaches to. For example, given the sentence “I saw the girl with the telescope”, the phrase “with the telescope” may belong to the verb “saw” or to the noun “girl”.

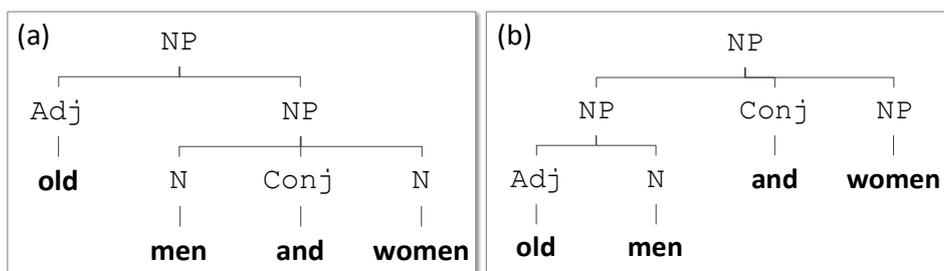


Figure 3-5: Example of syntactic ambiguity: coordination ambiguity. Source: Bird et al. (Bird et al. 2007: 165)

Coordination ambiguity is a very common form of syntactic ambiguity in English (Resnik 1999). Words and phrases of all types can be coordinated with the external modifier being a word or phrase of almost any type and appearing either before or after the coordination (Okumura and Muraki 1994). Therefore, in the phrase “old men and women” the word “old” may either be applied to both “men and women” or to just the “men” (see Figure 3-5). This problem has received little attention in the literature compared with other structural ambiguities such as PP-attachment (Chantree et al. 2006).

Noun phrases consist of a head noun that has more than one modifier. In this case the phrase needs bracketing to separate head and modifiers, either left bracketing or right bracketing. *Noun-phrase (NP) bracketing ambiguity* occurs when both types of bracketing are possible. NP bracketing is not identified by parsers trained on the Penn Treebank tag set (Vadas and Curran 2007).

Advanced parsing techniques involve *chart parsing* and *probabilistic parsing* which help to choose between several possible parses when syntactic ambiguity occurs. A *chart parser* is a parser for context-free grammars. It uses the dynamic programming approach and stores partial hypothesized results in a structure called a *chart* so that they can be re-used. Chart parsing can use a top-down as well as bottom-up parsing strategy. One of the most popular chart parser is the *Earley parser* which uses a top-down strategy (see Earley 1970). Modern chart parsers are often based on the work of Martin Kay (1973; 1980). Another chart parsing

algorithm is the Cocke-Younger-Kasami (CYK) algorithm (Kasami 1965; Younger 1967; Cocke and Schwartz 1970).

Probabilistic parsing greatly increases the ability to develop accurate, robust, broad coverage parsers (Charniak 1997). It converts parsing into a classification task using statistical and machine learning methods. *Probabilistic context-free Grammar* (PCFG) is also known as stochastic grammar and dates back to Booth (1969). Applications of probabilistic parsing can be found in Collins (1997), Ratnaparkhi (1997) and Charniak (2000). PCFGs are essentially the same as ordinary CFGs except that each rule has associated with it a probability. These probabilities are used to calculate the probability of a given derivation to choose between competing derivations.

3.2.4 Chunking

Chunking is an efficient and robust method for identifying short phrases in text. These chunks are groups of contiguous, non-overlapping spans of related tokens in the input text (Abney 1991). They usually consist of a head word (e.g. a noun) and the adjacent modifiers and function words (e.g. adjectives and determiners). The basic idea of chunking is to locate and ignore information by recognizing higher level units of structure to compress the description of a sentence, e.g. keeping only chunks containing nouns (Manning and Schütze 1999: 407).

Chunkers often operate on tagged texts, e.g. from part-of-speech tagging (see section 3.2.2), and use the tags to make chunking decisions. The following sentence shows a Wall Street Journal text with noun phrase chunks marked using brackets (see Bird et al. 2007: 115):

```
[ The/DT market/NN ] for/IN [ system-management/NN software/NN ]
for/IN [ Digital/NNP ] [ 's/POS hardware/NN ] is/VBZ fragmented/JJ
enough/RB that/IN [ a/DT giant/NN ] such/JJ as/IN [ Computer/NNP
Associates/NNPS ] should/MD do/VB well/RB there/RB ./.
```

Chunking is related to parsing (see section 3.2.3) as it can be used to build hierarchical structure over text. However, there are several important differences. First, chunking is not exhaustive and typically omits items in the surface string. Second, parsing constructs deeply nested structures of unlimited depth, chunking creates structures of fixed depth, typically depth 2. These chunks often correspond to the lowest level of grouping identified in the full parse tree. The relationship between chunking and parsing is illustrated in Figure 3-6.

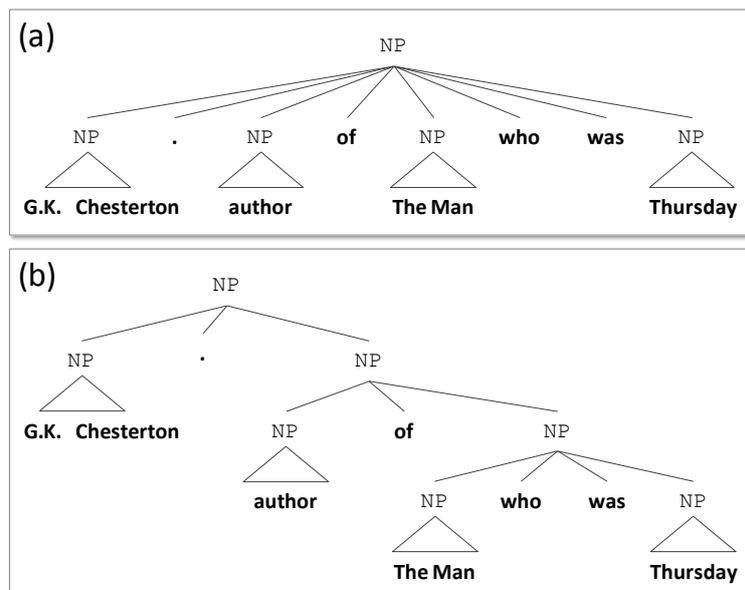


Figure 3-6: Chunking versus parsing. (a) tree representation of chunking; (b) tree representation of parsing. Source: Bird et al. (2007: 117)

Compared to parsing, chunking is more robust and efficient (Bird et al. 2007: 117): Parsing uses recursive phrase structure grammars and arbitrary-depth trees. Parsing has problems with robustness, given the difficulty in getting broad coverage and in resolving ambiguity. Regarding efficiency, the time taken to parse a sentence grows with the cube of the length of the sentence, while the time taken to chunk a sentence only grows linearly. As an intermediate between tagging and parsing, chunk structures can be represented using either tags or trees.

3.3 Text Transformation (Feature Generation)

To detect meaningful patterns in a document collection each document is typically represented by a *bag of words*, i.e. the words it contains and their occurrences. In order to transform the documents a *tokenization process* is employed which parses each document, splits sentences and returns a collection of *tokens*, or terms. Statistics like term frequency, document frequency, term proximity, or document length can be used to add a numerical dimension to unstructured text to determine the relevance of a term or a document (see section 3.4.3). The set of terms that can be extracted from a document collection is also called the *dictionary*, or *term vocabulary*. The dictionary contains the tokens extracted from the entire document collection. Thus, a zero term frequency indicates that the token does not appear in the current document.

Figure 3-7 gives an example of the bag-of-words representation of a document after the tokenization process. The tokenization process in this example also involves linguistic preprocessing (e.g. part-of-speech tagging) as well as feature selection techniques (e.g. stemming) to extract only singular nouns (see sections 3.2 and 3.4).

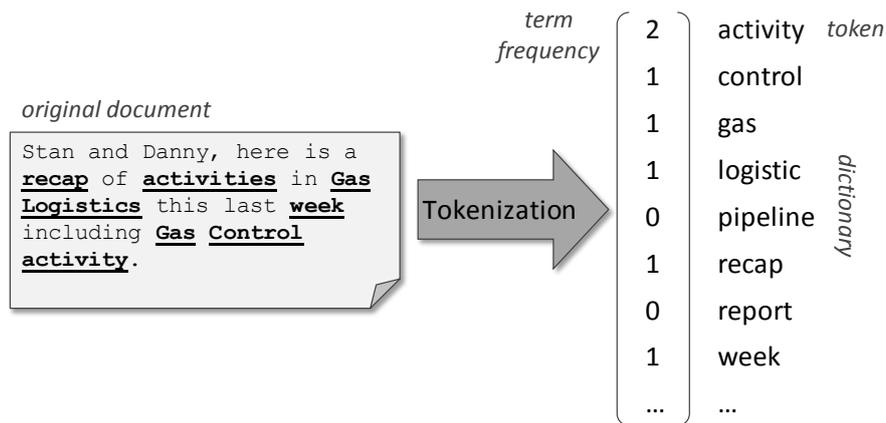


Figure 3-7: Example of bag-of-words representation after tokenization

This section deals with the general tokenization process and common text transformation techniques to generate such a bag of words.

3.3.1 Tokenization

The process of assigning documents to terms is called *indexing* (Manning et al. 2008: 3). *Terms* are indexed units, which can be words but also abstract information elements. They are stored in a *dictionary* of terms (or *vocabulary*, or *lexicon*). An *index*, also called *inverted index* (or *inverted file*), always maps back from terms to the part of a document where they occur. A term vector can be retrieved storing only the occurrence, or *postings*, of a term in a document. The process of creating an inverted index consists of the following four steps (Manning et al. 2008: 18):

- (1) Collect the documents to be indexed.
- (2) Tokenize the text, turning each document into a list of tokens.
- (3) Do linguistic preprocessing, producing a list of formalized tokens which are the indexing terms.
- (4) Index the document that each term occurs in order to create an inverted index consisting of a dictionary and postings.

Step (1) includes the choice of the *document unit*. For example, the document unit of a file storage system can be each folder including several files or each file separately (Manning et al. 2008: 21). Searching on an e-mail database, the e-mails and their attached files can be regarded as separate document units. A single document like a book can be split into chapters. Therefore, the indexing process starts with the issue of *indexing granularity* and must be able to handle the need to simultaneously index documents at multiple levels of granularity.

Step (2) and (3) of the indexing process result in splitting up the text into tokens which determine the term vocabulary of the dictionary (Manning et al. 2008: 21). The text is split up at certain characters like punctuation or whitespaces. This process is called *tokenization*. A *token* is an instance of a sequence of characters that are grouped together as a semantic unit. Depending on the tokenization techniques applied a token can consist of more than one word. Although terms and tokens can be the same, terms are usually derived from *tokens* by various normalization processes.

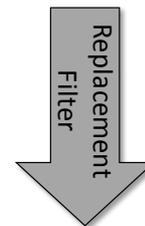
3.3.1.1 Replacement Filter

Especially with online media like e-mails as data source the text corpus may suffer from artifacts that do not belong to the content at all, e.g. xml markups for text formatting and metadata as well as hyper links. Here, a *replacement filter* can be employed to identify these elements using predefined *search patterns*. The identified patterns are then replaced by the *replacement pattern*. To optimize the search patterns and reduce their number regular expressions should be employed, e.g. “*” for any number of letters.

original e-mail

```
----- Forwarded by Shawn Davis/Hou-ComOps/EnergyTrading/PEC
on 07/27/200103:22 PM ----- Remind me To: Shawn
Davis/Hou-ComOps/EnergyTrading/PEC@PEC 07/23/2001 cc: 02:12 PM
Subject: 2001 NNG Winter Ops MeetingI am pl=anning to attend
the NNG Winter Ops Mee=ting the second week ofSeptember as=20we
discussed. However, my NNG rep Robert Benningfield==20informsme
that he and o=ther reps can=01,t be allowed to attend. Th=is
was==20thesame case as last year.
```

Rule no.	Search pattern	Replacement pattern
1	\-{4,}Forwarded by(.*)\-{4,}(.*) (PM AM)	<Whitespace>
2	To:(.*)Subject:	<Whitespace>
3	(=20)	<Whitespace>
4	(= 01,)	<EmptyString>
5	(\p{Blank} \p{Punct})+	<Whitespace>



resulting e-mail

```
Remind me 2001 NNG Winter Ops MeetingI am planning to attend
the NNG Winter Ops Meeting the second week ofSeptember as we
discussed However my NNG rep Robert Benningfield informsme that
he and other reps cant be allowed to attend This was thesame case
as last year
```

Figure 3-8: Replacement filter. Example on Enron e-mail corpus¹⁶

The resulting text in Figure 3-8 shows that with noisy data (see section 3.1.6) some successive analysis like word splitting may also be employed. Although some rules can be defined for each type of data, e.g. e-mail, web page etc., the accuracy and efficiency of this step can be improved by an initial inspection of the data to modify the replacement filter with appropriate search patterns. As the replacement filter makes use of the full text representation of the data it has to be performed before the tokenization process.

3.3.1.2 Where to Split the Text?

The major problem with tokenization is where to split the text (Manning et al. 2008: 22). One example is the use of the apostrophe for possession and contraction. Splitting the contraction

¹⁶ See chapter 5, section 5.3.1

“aren’t” can lead to four different tokenization strategies: “aren’t”, “arent”, “are” and “n’t”, “aren” and “t”.

Particular domains have unusual specific tokens like abbreviations, and especially computer technology introduced new types of semantic character sequences that should not be split up, like e-mail addresses, e-mail dates, web URLs, numeric IP addresses, package tracking number and many more (Manning et al. 2008: 23). These items have a clear semantic meaning. They should be identified and indexed separately. *Hyphenation* is used for various purposes ranging from splitting up vowels in words (“co-education”) to joining nouns as names (“Hewlett-Packard”) to a copyediting device to show word grouping (“the hold-him-back-and-drag-him-away maneuver”). The complex problem of handling hyphens can be a classification problem or solved by some heuristic rules (Manning et al. 2008: 23). For example, short hyphenated prefixes on words can indicate that the two words should actually be kept together in one token. Splitting on whitespaces also includes the problem of splitting semantic compounds with *internal spaces*, like names (“Los Angeles”), dates (“March 11, 1983”) or phone numbers. Additionally, some compounds have several different types of spelling, like “whitespace” and “white space”. A good tokenizer should be able to treat them as the same single token. Hyphenation and internal spaces can also occur in several combinations. Depending on the type of language additional techniques can be applied, like *compound splitter* to identify and split up compound nouns without any spacing often used in e.g. German, or *word segmentation* when no spaces are used at all like in East Asian Languages (Manning et al. 2008: 24). Word segmentation techniques vary from having a large vocabulary to using machine learning methods like hidden Markov models.

As the problem of tokenization is highly language specific and most languages have distinctive signature patterns, *language identification techniques* based on classifiers that use short subsequences of characters as features can be involved to improve the results. Language identification was first explored in cryptography. A n -gram algorithm can be found in Konheim (1981) and most researchers regard character n -gram techniques as highly successful means of language identification (Cavnar and Trenkle 1994; Dunning 1994; Beesley 1998). Written language identification is regarded as much easier than spoken language identification (Hughes et al. 2006).

3.3.1.3 Size of the Dictionary and Index Compression

There are two statistics for estimating the size of the dictionary: Heaps’ law and Zipf’s law. *Heaps’ law* helps to estimate the size of the dictionary M as a function of the entire collection size, i.e. number of tokens T (Manning et al. 2008): $M = kT^b$ with $30 \leq k \leq 100$ and $b \approx 0.5$. With the Reuters-RCV1 (see section 3.4.5) the fit between collection and vocabulary size is excellent for $T > 10^5$ with parameters $b = 0.49$ and $k = 44$. *Zipf’s law* can be used for modeling the distribution of terms across documents (Heaps 1978). It states, if t_1 is the most common term in the collection, t_2 is the next most common, and so on, then the collection frequency cf_i of the i th most common term is proportional to $1/i$: $cf_i \propto \frac{1}{i}$. This can be written as $cf_i = ci^k$, with $k = -1$ and c as a constant and is therefore a power law.

Most often it will be necessary to reduce the size of the dictionary. In general, this is called *index compression*. Index compression techniques can be lossless, if all information is

preserved, or lossy, if some information is discarded (Manning et al. 2008: 80). Case folding, stemming, stop word removal or latent semantic indexing as well as the vector space representation itself are forms of lossy compression. Lossy compression is also known as *preprocessing* or *pruning*. All of these techniques discard some information that cannot fully be restored from the compressed data representation. However, there are several studies concluding that lossy compression can achieve good compression with no or no significant decrease in retrieval effectiveness (see Büttcher and Clarke 2006; Blanco and Barreiro 2007; Ntoulas and Cho 2007).

3.3.1.4 Linguistic Preprocessing versus Tokenization

Linguistic preprocessing implicitly involves some tokenization as it has to identify the words (= tokens) in the sentences. However, the syntactic and semantic relationships are still available. Thus, this chapter deals with the tokenization process which results in a bag of words model.

3.3.2 Vector Space Model

Given the results from the tokenization process, the representation of a set of documents as vectors in a common vector space is known as the *vector space model*. Its first use was in the SMART information retrieval system as explained in section 3.4.3.2 (see Salton 1971a; van Rijsbergen 1986). Other common representations are the probabilistic model (Robertson 1977) and the logical model (van Rijsbergen 1986). A *document vector* is denoted as $\vec{V}(d)$. The values of each document vector are *term weights*, e.g. number of occurrences in the document. The position refers to a specific term in the dictionary. In this context, terms are also called *features*. This representation loses the relative ordering of the terms in each document, thus syntactic analysis has to be performed before (Manning et al. 2008: 111).

The similarity between two documents d_1 and d_2 can be calculated by the *cosine similarity* between their vector representations $\vec{V}(d_1)$ and $\vec{V}(d_2)$ (Manning et al. 2008: 111):

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| |\vec{V}(d_2)|} = \vec{v}(d_1) \cdot \vec{v}(d_2) \quad \text{with} \quad \vec{v}(d_i) = \frac{\vec{V}(d_i)}{|\vec{V}(d_i)|}$$

The numerator represents the *dot product* (also known as the *inner product*) of the two vectors, whereas the denominator is the product of their Euclidean lengths (see section 4.3.3). The effect of the denominator is to length-normalize the vectors to unit vectors. This is necessary to compensate the effect of document length as the relative distribution of terms can be identical in the two documents but the absolute term frequencies can be quite different.

Figure 3-9 represents the two documents of Figure 3-10 as vectors in the three-dimensional vector space. As illustrated in Figure 3-9 the similarity between two vectors can be expressed by the cosine of the angle θ between the two vectors (Manning et al. 2008: 112).

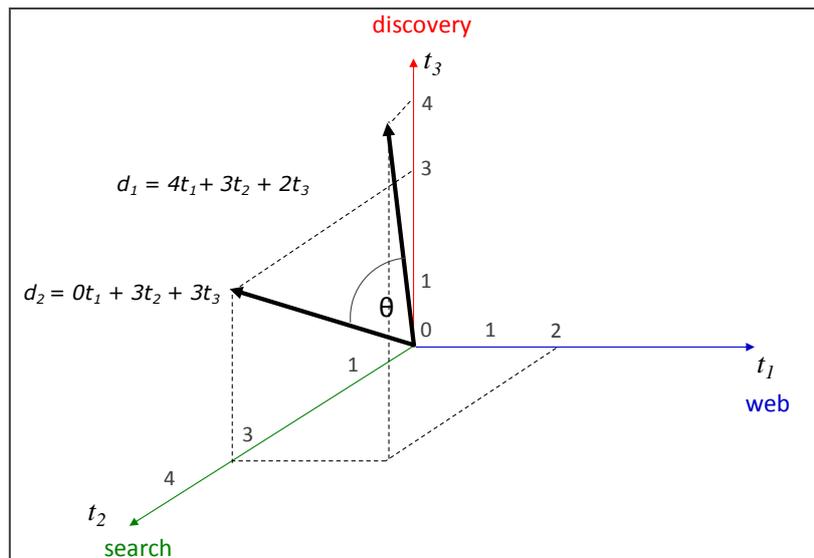


Figure 3-9: Vector space representation and illustration of the cosine similarity $\text{sim}(d_1, d_2) = \cos\theta$

A collection of N documents represented by vectors of size M where each value corresponds to a term in the dictionary can also be written as an $M \times N$ *term-document matrix* which is a sparse matrix whose rows represent terms (= dimensions) and whose columns represent documents (Manning et al. 2008: 113). The number of terms in the dictionary determines the dimension of the vector space. Text mining operates on high dimensional vector spaces but many missing values (“*sparseness*”) makes it effective and efficient (Weiss et al. 2004: 6). In contrast to cluster analysis (see chapter 4, section 4.2.3.1) missing values in text are no issue, as words are either there or not.

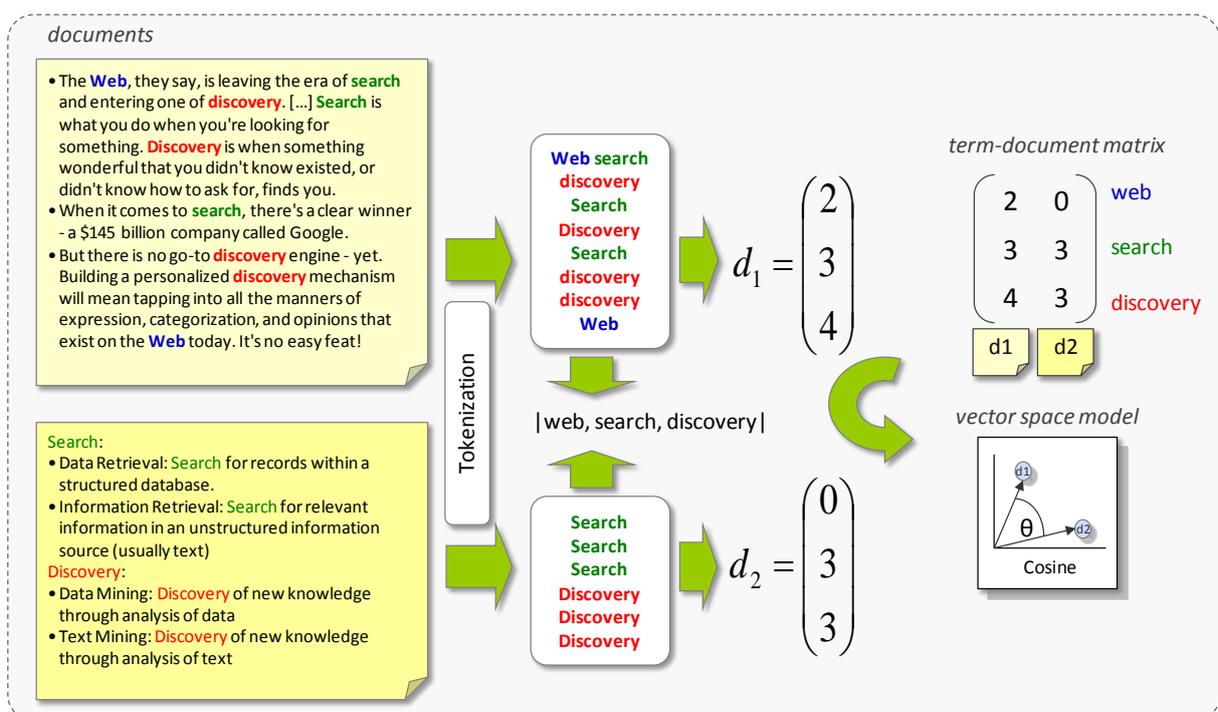


Figure 3-10: Text transformation for feature generation. Overview

Most often, the term weights are calculated using the *tf.idf* weighting scheme but other variants are also possible (see section 3.4.3). The weighting scheme determines the influence and significance of a term. Thus it is part of the feature selection process.

Figure 3-10 illustrates the text transformation process including tokenization, term-document-matrix representation as well as vector space representation and similarity calculation.

3.4 Feature Selection

The number of terms in the dictionary determines the dimension of the vector space (Weiss et al. 2004: 6). As depicted in Figure 3-11 feature selection aims at reducing the dimensionality of the feature space and improving the resolving power of the words. According to Luhn (1958) only medium-frequency terms provide a good resolving power for a document in the document collection comparing it with other candidate documents.

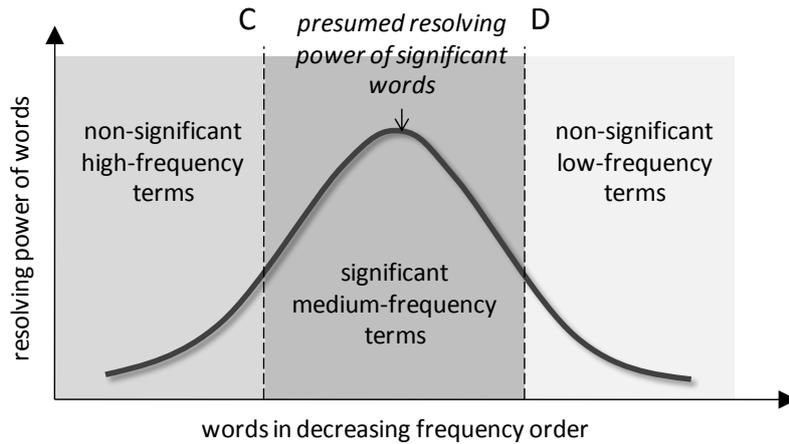


Figure 3-11: Resolving power of significant medium-frequency words. Based on: Salton and McGill (1984: 62) and Luhn (1958)

This section starts with the motivation of the feature selection process on performance measures (section 3.4.1) and presents several feature selection techniques for eliminating features (section 3.4.2), weighting features (section 3.4.3) and normalizing features (section 3.4.4) to improve the feature space. An overview of all feature selection methods described in this section is given in Figure 3-12.

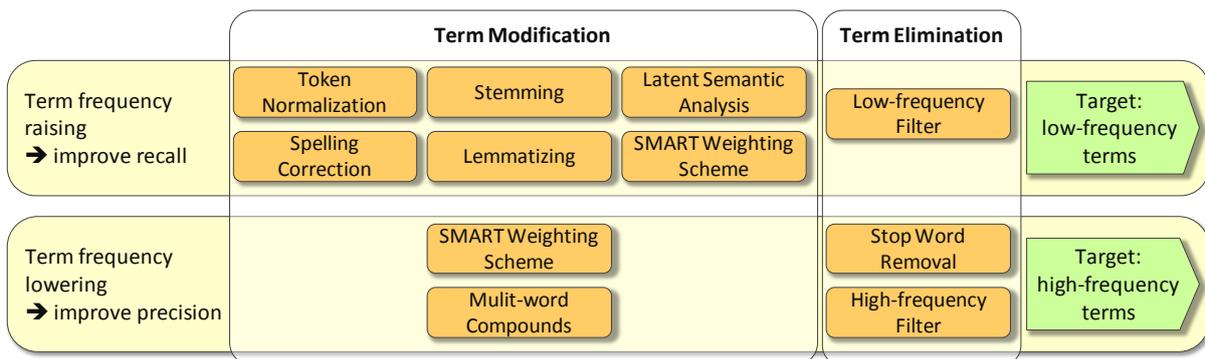


Figure 3-12: Feature selection methods. Overview

The particular strategy how to combine the different feature selection techniques depends on the data set and the intention of the analysis. However, usually one will decide for a weighting scheme first, and then perform some stop word removal and tag selection, e.g. keeping only nouns if part-of-speech tagging was performed in the linguistic preprocessing step, normalize features, e.g. using stemming, and then remove low and high frequency terms due to certain threshold.

3.4.1 Motivation: Performance Measures

There are a wide number of evaluation criteria to measure the performance of an information system. Among those concerning the users' needs six of them can be identified as critical (Cleverdon et al. 1966; Lancaster 1979):

- (1) *Recall*. The ability of the system to present all relevant items.
- (2) *Precision*. The ability of the system to present only the relevant items.
- (3) *User effort*. Intellectual or physical effort that is required from the users in formulating the queries, conducting the search and screening the output.
- (4) *Response time*. The time interval which elapses between receipt of a user query and the presentation of the systems response.
- (5) *Form of presentation*. The presentation of the search output which influences the user's ability to utilize the retrieved materials.
- (6) *Collection coverage*. The extent to which all relevant items are included in the system.

All six measures are originally designed for measuring information retrieval effectiveness where a (usually short) query vector is compared with the document collection to retrieve matching documents. Nevertheless, they can be related to the text mining process when documents are compared with each other in the succeeding data analysis. Criterion (6) is related to currency and completeness (see section 3.1.2). According to Salton and McGill (1984: 163) these criteria are easy to measure except for precision and recall. Precision and recall are the most frequent and basic measures for information retrieval effectiveness and are based on the work of Kent et al. (1955).

Precision is the percentage of relevant documents that have been retrieved:

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved})$$

Recall is the percentage of retrieved items that are in fact relevant for the user:

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant})$$

Figure 3-13 provides an example of the different combinations of high and low precision and recall values in a result set.

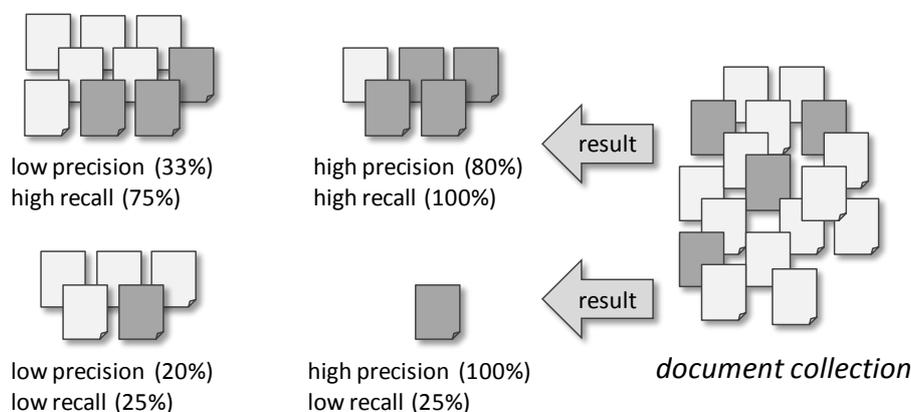


Figure 3-13: Precision and recall. Example

The difficulty of calculating these two measures is based on the problem how relevance can be defined and interpreted. An objective view is to define the relevance of an item to a query

as the degree to which the item deals with the subject of the user's information need (Cooper 1971; Saracevic 1975). This notion of relevance of a result is also called topic relatedness (Salton and McGill 1984: 163). A subjective view on the problem includes also the state of knowledge of the user at the time of search and the items that the user already knows about. This notion of relevance depends on the utility of the item to the user. Thus, using this subjective view an information system should retrieve those pertinent sets of items that are appropriate to the user's information need at the time of retrieval (Goffman 1964; Goffman and Newill 1964).

Unfortunately, precision and recall trade off against one another (Manning et al. 2008: 144). Usually, an increasing number of relevant documents retrieved (high recall) is biased by an increasing number of non-relevant documents retrieved as well (low precision). A measure which trades off precision and recall is the *F measure* which is formalized as the weighted harmonic mean of both measures:

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad \text{with } \beta^2 = \frac{1-\alpha}{\alpha}, \alpha \in [0,1], \beta \in [0, \infty]$$

The default balanced F measure with $\alpha = 0.5$ and $\beta = 1$ equally weights precision and recall:

$$F_{\beta=1} = F_1 = \frac{2PR}{P+R}$$

Values of $\beta < 1$ emphasize precision, whereas values of $\beta > 1$ emphasize recall.

The F measure, or rather its complement $E = 1 - F$, was first introduced by van Rijsbergen (1979) providing an extensive theoretical discussion. He describes the principle of decreasing marginal relevance where the user will be unwilling to exchange a unit of precision for an added unit of recall which leads to the harmonic mean being the appropriate method for combining both precision and recall.

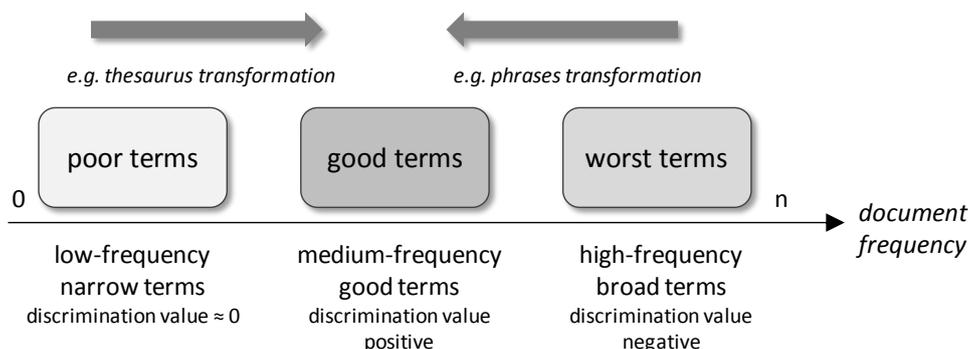


Figure 3-14: Term characterization in frequency spectrum. Based on: Salton and McGill (1984: 87)

Precision and recall are directly related to the feature selection process. Low-frequency terms tend to improve precision but reduce recall, whereas high-frequency terms improve recall but reduce precision. As illustrated in Figure 3-14 narrow low-frequency terms have a discrimination power which tends to zero and thus being regarded as poor terms. Broad high-frequency terms have even a negative discrimination power and thus are regarded as worst terms (Salton and McGill 1984: 86). Therefore, it is desirable to use feature selection methods to improve the feature space by reducing the number of features (= dimensionality). Salton and McGill (1984) suggest thesaurus transformation to reduce low-frequency terms to a common synonym and phrase transformation to generate multi-words to reduce the number of

high-frequency terms. In general, there are two options which involve eliminating, normalizing, weighting or transforming features:

- (1) Normalize, weight or transform low and high frequency terms to medium frequency terms.
- (2) Eliminate x percent of terms with the lowest and highest frequency.

The methods may depend on the problem type. For classification and filtering problems information from example documents is often used to guide the selection. The second option includes the definition of an appropriate threshold x which may be difficult to obtain (Salton and McGill 1984: 61).

3.4.2 Eliminating Features

Eliminating features usually deals with high-frequency and low-frequency terms and involves the global criterion of eliminating certain, predefined words (section 3.4.2.1) or the local criterion of eliminating those words with highest and lowest frequency in the document collection (section 3.4.2.2).

3.4.2.1 Stop Word Filter

Extremely common words with little semantic impact, called *stop words*, will be of little value for text mining and thus can be eliminated from the vocabulary. Besides ranking all terms by their total number of appearance in the collection (collection frequency) and omitting the most frequent terms some IR systems also use a fixed *stop word list*. An example from Reuters-RCV1 (see section 3.4.5) is shown in Figure 3-15.

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	The
to	was	were	will	with					

Figure 3-15: A stop word list of 25 semantically nonselective words that are common in Reuters-RCV1. Source: Manning et al. (2008: 25)

Figure 3-16 illustrates the removal of predefined stop words on the Enron data set. Obviously, mostly high-frequency terms are reduced but also some terms with medium or low frequency.

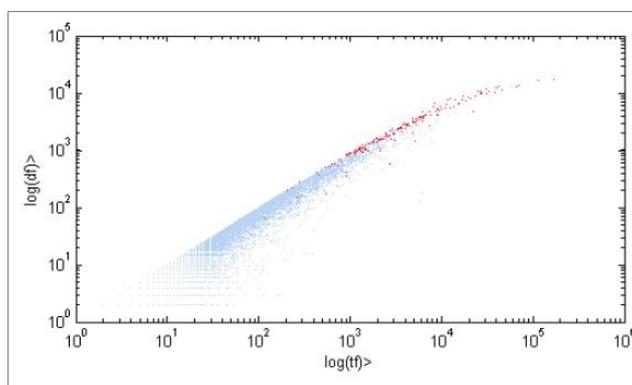


Figure 3-16: Stop word removal on Enron data set. Eliminated stop words are indicated as red dots

Excluding stop words from the vocabulary reduces the memory space of the storage of the posting lists drastically. However, especially Boolean retrieval is negatively affected by

omitting stop words as some meaning of the query may be lost and some phrases solely consists of stop words (“to be or not to be”).

According to Manning et al. (2008: 26) the general trend has been from large stop lists (200-300 terms) to very small or no stop lists at all. Web search engines in general do not use stop lists.

Some words can be considered stop words in a specific context; e.g. “AI” might be a stop word in a set of papers published in the proceedings of a conference about Artificial Intelligence. Including this term in the dictionary will not provide new insights as the scope of the conference is already known.

3.4.2.2 High-Frequency and Low-Frequency Filter

Instead of using a fixed list of stop words, stop word elimination can also be performed by eliminating a fixed number or percentage of the most common words (Manning et al. 2008). The *rule of 30* states that the 30 most common words account for 30% of the tokens in written text. This type of stop word filter can be called high-frequency filter (see Figure 3-17 (a)). Similarly, a low-frequency filter will remove the fixed number or percentage of the words with lowest frequency (see Figure 3-17 (b)).

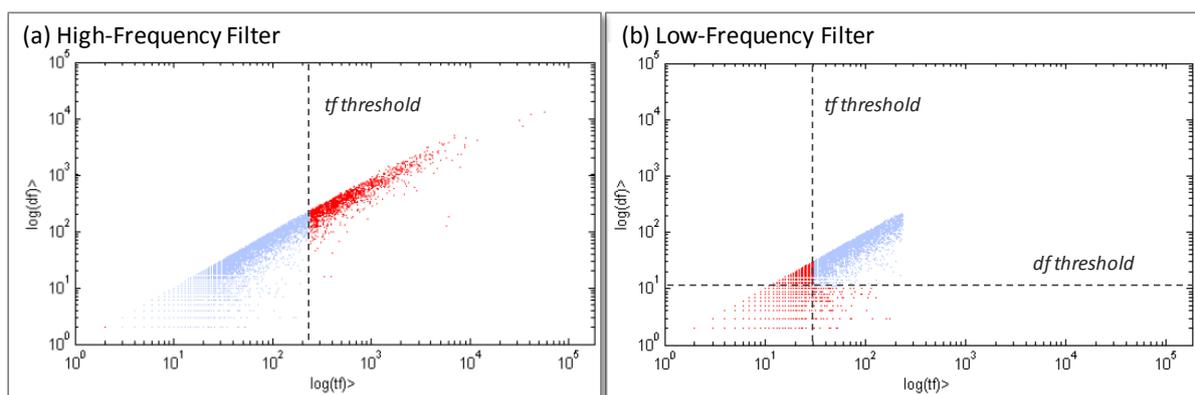


Figure 3-17: High-frequency and low-frequency filter on Enron data set. Eliminated stop words are indicated as red dots. Term frequency (tf) and document frequency (df) thresholds are indicated as dashed lines.

3.4.3 Weighting Features

Some of the earliest applications of term weighting are reported by Luhn (1957; 1958) with special focus on the importance of medium-frequency terms (see section 3.4). These terms are neither too commonplace nor too rare. Therefore, Luhn can be regarded as anticipating the *tf.idf* and related weighting schemes. Given a representation of the terms in a vector space (see section 3.3.2) the weighting schemes are also called vector space scoring.

3.4.3.1 Term Frequency and Weighting

For each term in a document a weight can be assigned that depends on its number of occurrences. The *term frequency* (tf) weighting scheme is the simplest approach of using term occurrence for document ranking. It is denoted as $tf_{t,d}$, with t denoting the term and d denoting the document. For each document the set of weights determined by any weighting function that maps the number of occurrences of each term in the document to a positive real

value is a quantitative digest of this document. Thus tf weighting makes the more frequent words in the document more important, i.e. more indicative of the topic. This view of a document is known as bag of words model (see section 3.3). However, this approach has one major disadvantage (Manning et al. 2008: 108): All terms are considered equally important regardless of their discriminating power in determining the relevance of a document. Terms that occur too often in the document collection are less indicative of overall topics (see section 3.4.1). Consequently, the term weights with high *collection frequency* (cf) should be scaled down. However, a high collection frequency can occur if a term occurs extremely often in a few documents but is not equally distributed over the entire collection of size N . Therefore, instead of directly using the collection-wide statistic of collection frequency it is common to use the document-level statistic *document frequency* (df) for this purpose. The document frequency is the number of documents in the entire collection containing the term t , denoted as df_t . The *inverse document frequency* (idf) if a term can be denoted as:

$$idf_t = \log \frac{N}{df_t}$$

The idf values are high for rare terms and low for frequent terms. It is an indicator of the discrimination power of a term. The idf weighting makes rare words across documents more important.

The use of inverse document frequency in term weighting was evaluated by Spärck Jones' work based on Luhn (see Spärck Jones 1972) and other researchers (Spärck Jones 1972; Robertson and Spärck Jones 1976; Croft and Harper 1979; Salton and Buckley 1987; Papineni 2001).

Both term frequency and inverse document frequency together are used as a measure of term weights (Manning et al. 2008: 109). The tf.idf weighting scheme assigns to term t a weight in document d given by:

$$tf.idf_{t,d} = tf_{t,d} \cdot idf_{t,d}$$

Thus, a term gets higher weights if it occurs many times in only a few documents, and lower weights if it occurs in many documents but only a few times in each document. Now, each document can be represented as a vector with one component corresponding to each term in the dictionary, together with a weight for each component calculated. Terms that do not occur in the document gain zero weights. This notion captures the relative importance of the terms in a document.

3.4.3.2 SMART Weighting Scheme

There are several variations from vector space scoring methods which apply different weights to term occurrences (Manning et al. 2008: 116). For example, as it seems unlikely that the significance of a term increases proportionally with the number of occurrences in a document a common modification is to use the logarithm of the term frequency allowing *sublinear tf scaling*. Longer documents contain higher term frequencies because longer documents tend to repeat the same words. Replacing the tf values by the *maximum tf normalization* eliminates this anomaly. To smooth the distribution of term frequency on document-level the tf weights of all terms occurring in the document can be normalized by dividing them by the tf weight of the most frequent term in each document. However, this approach suffers from some major

disadvantages: First of all, a change in the stop word list can dramatically alter term weightings, leading to unstable rankings. Furthermore, outlier terms with unusually large number of occurrences which are not representatives of the content will disturb the ranking (see chapter 4, section 4.2.3.3). Finally, documents with almost equal occurrences of all terms should be treated differently from those with more skewed distribution of term frequency.

An overview of the principle weighting schemes in use for a document vector is presented in Table 3-5 together with a mnemonic for representing specific combinations of weights. This system of mnemonics is called SMART notation. It is presented in Salton and Buckley (1988) and Singhal et al. (1996) with some difference in notation. Early experiments by Gerard Salton and colleagues on the SMART system can be found in Salton (1971b; 1991). The system presented in Table 3-5 is based on Singhal et al. (1996). A larger palette of schemes for term and document frequency weighting can be found in Moffat and Zobel (1998).

Table 3-5: SMART notation for tf.idf variant. Source. Manning et al. (2008: 118)

Term Frequency		Document Frequency		Normalization	
n (natural)	$tf_{t,d}$	n (none)	1	n (none)	1
l (logarithm)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \cdot tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max(0, \log \frac{N - df_t}{df_t})$	u (pivoted unique)	$\frac{1}{u}$
b (Boolean)	$\begin{cases} 1, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$			b (byte size)	$\frac{1}{CharLength^a}, a < 1$
L (log ave)	$\frac{1 + \log(tf_{t,d})}{1 + \log(ave_{t \in d}(tf_{t,d}))}$				

CharLength := the number of characters in the document

The mnemonic takes the form of *ddd.qqq*: the first triplet gives the weighting in the document vector and the second triplet the weighting in the query vector. The first letter in each triplet refers to the term frequency component of the weighting, the second letter to the document frequency component and the third letter to the type of normalization. It is not uncommon to use different normalization functions for query and document vectors. A standard weighting scheme is *lnc.ltc*: the document vector uses log-weighted term frequency, no document weighting (for both effectiveness and efficiency reasons), and cosine normalization, whereas the query vector uses log-weighted term frequency, idf weighting as document frequency and cosine normalization. Comparing documents with each other one should use the same weighting schemes for all documents.

3.4.3.3 Learning Weights

In general, learning weights cover approaches of scoring and ranking that are known as machine-learned relevance and use a set of training examples. The weights are then learned from these examples approximating the relevance judgments in the training examples. Especially if the collection changes frequently (e.g. the internet) the user-generated relevance judgments of the training set are the expensive component of this method. Learning of ranking functions is subject to IR research since the late 1980 and pioneering work can be

found in Fuhr (1989), Fuhr and Pfeifer (1994), Cooper et al. (1994), Bartell (1994), Bartell et al. (1998), and Cohen et al. (1998).

3.4.4 Normalizing Features

In contrast to weighting features where the number of terms is preserved and eliminating features where the feature space is reduced and some information is lost, normalizing features aims at reducing the feature space while keeping all information by combining different words into a single representation. This includes morphological analysis like stemming and lemmatizing, spelling correction or multi-word generation. Advanced linguistic methods like word sense disambiguation (see section 3.2.1) can also be used for feature normalization.

3.4.4.1 Token Normalization

Token normalization, or *term normalization*, is the process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens (Manning et al. 2008: 26). In general, there is no clear guidance to what amount of normalization is sufficient and appropriate. However, too much normalization can be harmful for the quality of the resulting set of tokens as wrong equivalence classes can be established and semantic meaning can get lost.

The most standard way is to implicitly create *equivalence classes*, which are normally named after one member of the set. Equivalent tokens are then represented by their equivalence class. They can be tokens with different spelling of the same word (“colour” and “color”) or grammatical variations (“window” and “windows”). Establishing term relationships between unnormalized tokens can help to construct lists of synonyms such as “car” and “automobile”. This can be done by indexing unnormalized tokens and maintain a query expansion list of multiple vocabulary entries which has to be considered for a certain query term. A query term is then a disjunction of several posting lists. The alternative is to perform the expansion during index construction. As there are more postings to be stored equivalence classes are much more effective than either of these two methods. The first method requires more processing at query time, the second method more space for storing postings.

Diacritic normalization and case-folding are commonly employed in IR systems. *Diacritics* can be used to spell the same word in different ways, such as “cliché” and “cliche”. This is often the case with English texts. However, in other languages like Spanish, *accents* on characters distinguish completely different words, like “pena” (English: sorrow) and “peña” (English: a cliff). Today, users enter queries often without diacritics for reasons of speed, limited software and problems to use non-ASCII text in many computer systems (Manning et al. 2008: 28). Most tokenizers will perform *case-folding* by reducing all letters to lower cases (Manning et al. 2008: 28). Although most often it makes no differences case-folding can equate words that should be regarded as separate tokens. For example, many proper nouns like company names, governmental organizations and person names are derived from very common nouns and can only be distinguished by case (e.g. “General Motors”, “the FED”, “Bush”). In English most ordinary words are written with lower case one solution is to lowercase only some tokens, like those at the beginning of a sentence. Midsentence capitalization usually indicates proper nouns and therefore stays unaffected. *Truecasing* is the use of machine learning sequence models for case-folding (Lita et al. 2003).

Most normalization techniques depend as highly on the language as on the spelling habits of the users which can be quite different to traditional usage on computer systems and web search (Manning et al. 2008: 28). About 60% of all web pages are in English (Gerrand 2007) but only 10% of the world's population primarily speak English and only about one third of blog posts are in English (Sifry 2007). Therefore, document indexing has to be able to deal with several languages. Document collections can include documents from many different languages and one document, like a scientific article or an e-mail, can contain quotes in one or more other languages than the rest of the text (Manning et al. 2008: 30). Although the language can be detected and language-particular tokenization and normalization rules can be applied to different parts of the document using different vocabularies for indexing brief quotations are still hard to automatically identify and handle. However, most often the same word will not appear in different languages with different meanings. More problems arise with the different spellings of foreign names due to variant transliteration standards (e.g. "Chebyshev" and "Tschebycheff"). One can use spelling correction heuristics like the Soundex algorithm (see section 3.4.4.2) or expand terms with phonetic equivalents.

3.4.4.2 Spelling Correction

There are two basic principles how to implement spelling corrections (Manning et al. 2008: 52):

- (1) Given a misspelled term and several alternative correct spellings choose the nearest alternative.
- (2) Given a misspelled term and several tied or nearly tied alternative correct spellings choose the alternative that is more common.

The first method demands some notion of nearness or proximity. The second method demands some statistic of term occurrence in documents or term usage by users. To apply spelling correction to each term will increase the calculation effort dramatically. Thus, some methods take only a limited number of candidate terms into account.

Besides these basic principles one can distinguish two specific forms of spelling corrections (Manning et al. 2008: 53): *isolated-term* correction and *context-sensitive* correction. Isolated-term correction corrects a single term at a time, even if it is in fact a multi-term. Therefore, isolated-term correction can fail as it misses the semantic, or context, of the term. *Edit distance*, also known as *Levenshtein distance*, and *k-gram overlap* are isolated-term correction techniques. Given two character strings, the edit distance between them is the minimum number of *edit operations* required to transform the first string into the other string (Gusfield 1997). This can be done by inserting a character, deleting a character, and replacing a character by another character. Spelling correction using edit distance will compute the edit distance between a misspelled term and the terms in the dictionary and use the term for correction that has a minimum edit distance (Manning et al. 2008: 55). Different weights can be applied to different kinds of edit operations, e.g. depending on the likelihood of letters substituting for each other. However, this approach is extremely expensive. Therefore, a number of heuristics can be employed. Most simply, one can restrict the comparison to those terms beginning with the same letter supposing spelling errors will not appear at the beginning of the term. A *k-gram index* is applied to retrieve only those vocabulary terms that

naturally have a low edit distance to the term under consideration and compute the minimum edit distance only on this limited set of terms (Manning et al. 2008: 55). The idea is to scan the postings for matching k -grams with the given term and retrieving those that have many k -grams in common, e.g. for a string of length four and a bigram index this could be two out of three possible matches. However, a major disadvantage of this approach is that implausible corrections of the query term containing the same k -grams will be enumerated as well. This problem can be overcome by the overlap between the two sets of k -grams of the given term and each vocabulary term. A common measure of overlap is the *Jaccard coefficient* (see chapter 4, Section 4.3.4). If the coefficient exceeds a predefined threshold the vocabulary term is considered as spelling correction candidate. One of the first implementations of spelling correction techniques can be found in Damerau (1964). Zobel and Dart (1995) provide a survey of several methods concluding that k -gram indexing is very effective for finding candidate mismatches but should be combined with additional techniques like edit distance to determine the most likely misspelling.

In contrast of isolated-term correction, *context-sensitive spelling correction* deals with the context (i.e. surrounding terms) of a term generating a multi-term (Manning et al. 2008: 58). A simple but expensive approach is to enumerate possible corrections for each term and to successively substitute the original terms determining the number of matching results. This approach can be improved employing several heuristics, like k -grams or the logs of queries issued by user of an IR system, to reduce the number of combinations of alternatives.

Phonetic spelling correction techniques deal with misspellings that occur because the user types a term that sounds like the target term (Manning et al. 2008: 58). Such algorithms are especially applicable to proper nouns, like the name of a company or a person. In general, phonetic correction algorithms generate a *phonetic hash* for each term so that similar-sounding terms hash to the same value. Phonetic hashing algorithms are commonly known as *Soundex algorithms*. The original Soundex algorithm is attributed to Margeret K. Odell and Robert C. Russell and described in Bourne and Ford (1961). An evaluation of phonetic matching algorithms can be found in Zobel and Dart (1996).

Probabilistic models for spelling correction, also known as noisy channel models, were first introduced by Kernighan et al. (1990) and further developed by Brill and Moore (2000) and Toutanova and Moore (2002). In these models, the misspelled phrase is viewed as a probabilistic corruption of a correct phrase. Cucerzan and Brill (2004) show how this work can be extended to learning spelling correction models based on query reformulations in search engine logs.

3.4.4.3 Multi-Word Compounds

Many complex technical concepts as well as organization and product names are multi-word compounds or phrases (Manning et al. 2008: 36). For example, search engines nowadays support *phrase queries* by double quote syntax (e.g., “Stanford University”). About 3% (Kammenhuber et al. 2006) to 11.7% (Johnson et al. 2006) of web queries are phrase queries and even more are implicit phrase queries such as person names, entered without double quotes (Silverstein et al. 1999). Therefore, simple one-word dictionaries containing single terms should be replaced by more elaborate approaches. The idea of k -gram indices can not

only used for spelling correction applied to the letters in a word but also for collocation analysis of words in a sentence to generate multi-words. *Biword* indices consider every pair of consecutive terms in a document as a phrase (Manning et al. 2008: 37). In general, nouns and noun phrases have a special status in describing the information need of a user. However, related nouns can be divided by several function words. Generating multi-words is a feature normalization technique which helps to improve the significance of high-frequency terms. However, as it will need information about the structure of the sentences and the results from a preceding linguistic analysis (see section 3.2) it has to be performed as part of the tokenization process (see section 3.3.1). For example, a part-of-speech tagger can be employed to assign the lexicographical unit to each term depending on its position in the sentence (see section 3.2.2). Now, apart from the nouns all other terms can be omitted and biwords are only established on consecutive nouns which might be separated by functional words in the original text. More extended part-of-speech patterns can help to further improve the generation of biwords. The concept of biword indices can be extended to longer sequences of words. If the index includes word sequences of variable length it is generally referred to as a *phrase index*. Storing large phrase indices, the size of the dictionary will drastically expand (see section 3.3.1.3).

3.4.4.4 Stemming and Lemmatization

Automated stemming and lemmatization are two methods of computational morphological analysis using Natural Language Processing (NLP) techniques (see e.g., Sproat 1992; Beesley and Karttunen 2003). Different inflectional related forms of a word can be found in the document collection, such as “organize”, “organizes”, and “organizing”. Additionally, there are families of derivationally related words with similar meanings, such as “democracy”, “democratic” and “democratization”. Stemming and lemmatization are two methods to reduce inflectional and sometimes derivationally related forms of a word to a common form. *Stemming* refers to a heuristic process of removing derivational affixes. *Lemmatization* usually involves the use of a vocabulary and morphological analysis of words. It removes inflectional endings and returns the base or dictionary form of a word, the *lemma*. For example, a stemmer might reduce the token “saw” just to “s”, whereas a lemmatizer would return either “see” or “saw” depending on whether the use of the token was either a verb or a noun. Most commonly, lemmatizing collapses only the different inflectional forms of a lemma, whereas stemming also collapses derivationally related words (Manning et al. 2008: 31). Stemmers are language-specific but require less knowledge than a lemmatizer (Manning et al. 2008: 32). A popular and empirically proven effective stemmer for English texts is Porter’s algorithm consisting of five phases of word reduction (Porter 1980). In Table 3-6, the results from the Porter stemmer on a sample text are compared with those obtained from the older, one-pass Lovins stemmer (Lovins 1968), and the newer Paice/Husk stemmer (Paice 1990).

Table 3-6. A comparison of the stemming algorithms by Lovins, Porter and Paice on a sample text.
 Source: Manning et al. (2008: 32)

Stemmer	Text	Source
Sample text	Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation	-
Lovins stemmer	such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpre	Lovins (1968) ¹⁷
Porter stemmer	such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret	Porter (1980) ¹⁸
Paice/Husk stemmer	such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret	Paice (1990) ¹⁹

According to Manning et al. (2008: 32) doing full morphological analysis provides only modest benefits for information retrieval and text mining, especially for languages with few morphology like English. In contrast, results for the European CLEF evaluations have repeatedly shown large improvements from the use of stemmers and compound splitters for German documents. Although some term comparisons can be improved by lemmatization, it equally hurts the performance of others. In general, stemming increases recall but reduces precision, especially if distinctive tokens are merged with those that are more commonly used.

3.4.4.5 Latent Semantic Analysis

Although the vector space representation has a number of advantages including the induced document comparison based on cosine similarity it suffers from its inability to cope with two classic problems arising in natural languages (Manning et al. 2008: 378): synonymy and polysemy. *Synonymy* refers to the case where two different words have the same meaning, e.g. “car” and “automobile”. The vector space representation fails to capture the relationship between such synonymous terms and assigns each to a separate dimension in the vector space. Consequently, the computed similarity between documents containing synonyms underestimates the true similarity that a user would perceive. *Polysemy* on the other hand refers to the case where a term such as “charge” has multiple meanings so that the computed similarity overestimates the similarity that a user would perceive.

One possibility to solve these problems is to employ the co-occurrences of terms to capture the latent semantic associations between them (Manning et al. 2008: 279). *Latent Semantic Analysis* (LSA) is a NLP technique which makes use of this approach. It analyzes the relationships between a set of documents and the terms they contain by producing a set of *concepts* related to the documents and terms. LSA is based on the term-document matrix which describes the occurrences of terms in documents (see section 3.3.2). However, even for

¹⁷ <http://www.cs.waikato.ac.nz/~eibe7stemmers/>

¹⁸ <http://www.tartarus.org/~martin/PorterStemmer/>

¹⁹ <http://www.comp.lancs.ac.uk/computing/research/stemming/>

a collection of modest size, the term-document matrix is likely to have several tens of thousands of rows and columns. Therefore, LSA finds a low-rank approximation to the term-document matrix using the singular value decomposition (SVD) of the matrix. As a result, some dimensions are combined and depend on more than one term. The rank lowering is expected to merge the dimensions associated with terms that have similar meanings. Thus, synonyms can be identified.

There are some disadvantages of LSA. For example, the resulting dimensions might be difficult to interpret. Additionally, LSA may fail to capture polysemy. Each occurrence of a polysemous word is treated as having the same meaning joining the word and its contexts to a single concept which is represented as a single point in space. This results in the vector representation being an average of all the different meanings of the word in the corpus, which can make it difficult for comparison. However, often polysemous words have a predominant sense throughout a corpus which also dominates in the related concept.

Wei and Croft (2006) present the first large scale evaluation of LSA. In IR, this approach is also called *Latent Semantic Indexing* (LSI). The connection between IR and the low-rank approximations of the term-document matrix employed by LSA was introduced in Deerwester et al. (1990) with a subsequent survey of results in Berry et al. (1995). For IR applications, Dumais (1993; 1995) describes experiments on TREC benchmarks using the commonly used Lanczos algorithm to compute the SVD which give evidence that at least on some benchmarks, LSI can produce better precision and recall than standard vector-space retrieval. This suggests that, for a suitable value of k , LSI addresses some of the challenges of synonymy. However, the computational cost of the SVD is significant. Manning et al. (2008: 382) mention about one million documents as the upper limit. This has been the biggest obstacle to the widespread adoption to LSI. One approach to this problem is to build the LSI representation on a randomly sampled subset of the documents in the collection. Schütze and Silverstein (1997) evaluate LSI and truncated representations of centroids for efficient K -means clustering. Applications of LSI to cross-language data sets are developed in Berry and Young (1995) and Littman et al. (1998). Hofmann (1999) provides an initial probabilistic extension of the basic LSI technique based on a multinomial model. A more satisfactory formal basis for a probabilistic latent variable model for dimensionality reduction is the Latent Dirichlet Allocation (LDA) model (Blei et al. 2003), which is generative and assigns probabilities to documents outside of the training set. This model is extended to a hierarchical clustering by Rosen-Zvi et al. (2004).

3.4.5 Evaluation of Feature Selection Methods

The effectiveness of feature selection methods depends on the size and type of data as well as the language (Manning et al. 2008: 80). In this section two studies are presented which give insights in the effectiveness of commonly used feature selection methods.

Hollink et al. (2004) evaluated the effectiveness of advanced feature selection methods of eight European languages in terms of change in mean average precision over a baseline system. As presented in Table 3-7 diacritic removal was especially helpful with Finnish, French, and Swedish. Among the linguistically motivated techniques, morphological normalization and compound splitting was evaluated. Stemming was remarkably helpful with

Finnish and Spanish but for most other language only small improvements could be achieved. Results from lemmatization were even poorer still. To increase comparability across languages, the authors of the study applied a single family of lemmatizers to as many of the eight languages as possible. The lemmatizing that they finally used was the part of tree-tagger (see Schmid 1994), a probabilistic part-of-speech tagger based on decision trees available for English, French, German, and Italian. Compound splitting was only performed on compound-rich languages. Swedish and German could benefit from compound splitting.

Table 3-7: Comparison of the effectiveness of advanced feature selection methods applied to European languages measured in change in mean average precision. Source: Hollink et al. (2004)

Language	Diacritic Removal	Morphological Normalization		Compound Splitting		k-Grams		
		Stemming	Lemmatization	Compound Splitting	Compound Splitting +Stemming	4-Gram	4-Grams +Stem	4-Gram +Lemma
Dutch	+9.6%	+1.2%	-	+4.0%	+3.6%	+0.3%	-3.6%	+3.3%
English	+2.1%	+4.0%	-10.2%	-	-	+7.3%	-12.2%	-
Finnish	+23.4%	+30.0%	-	+18.7%	+9.8%	+37.4%	+10.2%	-
French	+18.5%	+1.2%	-4.2%	-	-	+6.7%	-6.4%	+8.2%
German	+1.9%	+7.3%	+6.0%	+12.2%	+15.5%	+20.3%	+8.8%	+18.2%
Italian	+7.6%	+4.9%	+2.4%	-	-	+7.6%	-8.6%	-1.9%
Spanish	+15.0%	+10.5%	-	-	-	+1.5%	-10.9%	-
Swedish	+19.3%	+1.7%	-	+6.0%	+25.3%	+27.4%	+23.2%	-

The knowledge-free approach of indexing character k -grams could provide often as good or better results as language-particular methods. Depending on the language some further improvements could be made by combining k -grams with morphological normalization techniques.

Similar results were obtained by Tomlinson (2003). Detailed experiments and discussion on the impact of stemming in English can be found in several other studies (see Salton 1989; Harman 1991; Krovetz 1995; Hull 1996).

Table 3-8 presents an overview of the effects of feature selection on the number of terms, non-positional postings and tokens for Reuter-RCV1 by Manning et al. (2008: 80). Reuter-RCV1 is a collection of 806,791 documents and 391,523 distinct terms that were sent over the Reuters newswire during a 1-year period between August 20, 1996, and August 19, 1997 (see Lewis et al. 2004). Stemming was performed with the Porter stemmer (see Porter 1980). Stop word elimination was not performed with a fixed list of stop words but by eliminating the 30, or 150, most common words. The *rule of 30* states that the 30 most common words account for 30% of the tokens in written text (31% in Table 3-8). The combination of all steps helps to reduce the number of terms, non-positional postings and tokens by 33%, 42%, and 52% respectively.

Table 3-8: The effect of feature selection on the number of terms, non-positional postings and tokens for Reuter-RCV1. Source: Manning et al. (2008: 80)

	Distinct Terms			Non-positional Postings			Tokens ¹⁾		
	Number	$\Delta\%$	T%	Number	$\Delta\%$	T%	Number	$\Delta\%$	T%
unfiltered	484,494			109,971,179					
no numbers	473,723	-2	-2	100,680,242	-8	-8	179,158,204	-9	-9
case folding	391,523	-17	-19	96,969,056	-3	-12	179,158,204	-0	-9
30 stop words	391,493	-0	-19	83,390,443	-14	-24	121,857,825	-31	-38
150 stop words ²⁾	391,373	-0	-19	67,001,847	-30	-39	94,516,599	-47	-52
stemming	322,383	-17	-33	63,812,300	-4	-42	94,516,599	-0	-52

¹⁾ number of position entries in postings

$\Delta\%$:= reduction in size between successive filters

²⁾ uses case-folding as reference line for $\Delta\%$

T%:= cumulative reduction from unfiltered data

Although the effects of text mining techniques will differ depending on the size, language and document type the deltas in Table 3-8 are in the range typical of large collections (Manning et al. 2008: 80).

3.5 Applications of Text Mining in Social Corpora

The numerous application scenarios of text mining on social corpora like author-centric documents, e-mails, postings in user forums, wikis or collaborative tagging systems cover such different problems like labeling, categorization and classification, ontology extraction, knowledge identification or trend detection. To highlight the potentials of text mining methods applied to social corpora in general a non-exclusive selection of recent approaches is presented in this section. There is a number of publications concerning searching and identification of important knowledge workers in social corpora, some of them only considering the content objects, some of them additionally exploiting network structures to enhance the analysis. Understanding an enterprise's workforce and skill-set can be seen as the key to understand an organization's capabilities. In today's large organizations it has become increasingly difficult to find people that have specific skills or knowledge or to explore and understand the overall picture of an organization's portfolio of topic expertise (Brunnert et al. 2007).

Patterns of Topic Usage and Lifecycles: Co-authorship networks provide rich data of collaborative writing. Molka-Danielsen et al. (2007) provide a historical reflection and analysis of the social network of researchers associated with the IRIS (Information Systems Research in Scandinavia) Conference in the period from 1978 to 2006. Using text mining techniques and subsequent data analysis two remarkable temporal effects can be found: First, keywords being popular in earlier years show decreasing popularity over time. This indicates increased pluralism and heterogeneity in terms of keywords usage. Over time the main topics differentiate to a larger set of different terms. The second observation is that a general lifecycle of the whole discipline from general issues to more diverse and application oriented issues can be observed.

shown in Figure 3-18 the methods exploits two feature of the graph: the terms related to the main topics of the document (e.g. “Apple Inc.” and “Blindness”) tend to bunch up into densely interconnected subgraphs or communities while non-important (e.g. “Time”, “Month”, “Massachusetts” or “Consumer”) or even disambiguation mistakes (e.g. “Home Office”, “Free agent”, “Grocery store”) tend to fall into peripheral or weakly connected communities or even become isolated vertices in a semantic graph. Using the graph-based GN algorithm (see chapter 4.7.2) for community detection the graph is partitioned into thematically cohesive groups of terms. To weight terms and determine their semantic relatedness the authors exploit information extracted from Wikipedia. Involving the Wikipedia knowledge base, this approach does not require any training. The results are evaluated using human judgments showing that the produced key terms have higher precision and recall than existing methods.

Exploiting Wikipedia-based Semantic Relatedness: Similar to the work of Gabrilovich and Markovitch (2006; 2007) and Grineva et al. (2009), there are several other publications which deal with the semantic relatedness in Wikipedia data (see e.g. Strube and Ponzetto 2006; Milne and Witten 2008; Turdakov and Velikhov 2008). A comparison of Wikipedia-based semantic relatedness measures can be found in Milne (2007).

Adding Semantics to E-mail Clustering: Li et al. (2006) cluster e-mails according to their contents and the sentence styles of their subject lines. NLP and frequent itemset mining techniques are utilized to automatically generate meaningful generalized sentence patterns (GSPs) from subjects of e-mails. The GSPs are used as pseudo class labels. According to the authors the algorithm outperforms the popular K -means clustering algorithm and improves cluster naming readability.

Using Text Mining to Analyze User Forums: According to Feldman et al. (2008) product discussion boards are a rich source of information about consumer sentiments about products which is being increasingly exploited. Most sentiment analyses have looked at single products in isolation but users often compare different products, stating which they like better and why. Feldman et al. (2008) present a set of techniques for analyzing how consumers view product markets. Specifically, they perform relative sentiment analysis and extract comparisons between products to understand what attributes users compare products on and which products they prefer on each dimension. These methods are illustrated in an extended case study analyzing the sedan car markets.

Ontology Extraction by Collaborative Tagging with Social Networking: Hamasaki et al. (2008) propose the integration of social networks with collaborative tagging for ontology extraction. They extend the common tripartite models of emergent ontologies based on three dimensions (i.e. users, tags and instances) by a new dimension, the user-user relations. Examples are the “friend” relation in social networking services and the “knows” relation in Friend-of-a-Friend (FOAF) documents. The communicational relationships within a community are an important source of information that can be used to improve the emergent ontology. The authors evaluate the performance of their algorithm using extracted ontologies for information recommendation in two case studies.

Understanding of Organizational Phenomena: Gloor and Zhao (2006) present a novel software system (“iQuest”) which permits to gain new insights into organizational behavior. It

allows addressing issues such as tracking information while respecting privacy, comparing different interaction channels, network membership as well as correlating organizational performance and creativity. It extends automatic visualization of social networks by mining communication archives such as e-mail and blogs and including the analysis of the contents of those archives.

Structuring Cross-Organizational Knowledge Sharing: According to White and Lutters (2007) ontology development is fundamental to most knowledge management efforts. When approached in a formal knowledge engineering manner the resulting ontology usually becomes insufficient even if only a modest number of groups within a single organization is involved. It breaks entirely when scaled up to multiple, heterogeneous organizations. A promising alternative is the bottom-up approach such as can be found in social tagging systems (e.g. del.icio.us). The authors extend their field work with IT helpdesk staff to examine the drivers for natural ontology development. They examine the user-centered design criteria for both mid-level ontology development and related expert profile management. As one of their findings a balance between some degree of external order is required while at the same time maintaining local flexibility.

Matching Human Actors based on their Texts: Bringing together human actors with similar interests, skills or knowledge is a major challenge in community-based knowledge management. Similar to the motivation of this thesis, Reichling et al. (2005) use writing or reading textual documents as an indicator for a human actor's interests, skills or knowledge. Using text mining methods like language recognition, stop word removal and stemming they extract and match user profiles based on a large collection of documents. The ExpertFinder Framework measures the similarity of these profiles by means of LSI. The quality of the algorithmic approach is evaluated by comparing its results with judgments of different human actors. Later, Reichling and Wulf (2009) evaluate the performance of an expert recommender systems in practice. Unlike other approaches, the system involves users in selecting textual documents for semi-automatic profile generation.

Searching for Experts in the Enterprise: Employees depend on other people in the enterprise to gain rapid access to important information. Therefore, Ehrlich et al. (2007) combine text and social network analysis. They infer content networks and dynamic social networks from e-mail and chat logs. The paper provides a user study of SmallBlue, a social-context-aware knowledge search system that can be used to identify experts, see dynamic profile information and get information about the social distance to the expert, before deciding whether and how to initiate contact. A similar approach can be found in Brunnert et al. (2007). Their approach is based on techniques like k -grams, clustering, and visualization for improving the user search experience for people and skills. However, they do not explicitly involve a network perspective.

Social Search - Searching and Exploring Social Corpora: Current developments of internet usage provide mutual advice, collaboratively filter important information and create virtual networks of trust. Understanding this environment requires additional knowledge about the patterns and processes of group interaction. Trier and Bobrik (2008b) address this objective by introducing a visualization-based procedure for searching and exploring social corpora. It involves text analysis as well as dynamic network visualization. Promising results have been

derived on two samples of an electronic discussion and a corporate e-mail network to demonstrate search and retrieval in networks. This approach provides novel insights about the actual dynamics of content dissemination and network evolution in social corpora. An application of the social search approach to customer relationship management (CRM) can be found in Trier et al. (2009). In this example, CRM incorporates knowledge into the direct customer contact. There are rich networks of electronic interaction available. To be able to meet complex customer requests the authors suggest combining knowledge management with social network analysis. Based on the SNI approach (see chapter 2.2) and its application to social search, the authors present the IT-supported Social Network Intelligence-based Routing (SNIBR) approach which involves text analysis as well as SNA techniques to identify the best matching contact person and the best routing paths through the network. They provide a case study which helps to improve a customer interaction center of a company in the telecommunication market.

4 Cluster Analysis

Cluster analysis provides methods and techniques to identify group memberships in a data set. This chapter is organized as follows: First, a general introduction to cluster analysis is given (section 4.1) and the consecutive steps of a clustering methodology are explained (section 4.1.3). The following sections discuss the methods and techniques that are involved in the clustering methodology steps including the initial data collection, the clustering process itself as well as some final tests for cluster tendency and validity (section 4.2 to 4.6). Finally, this chapter presents applications of cluster analysis in social corpora with special focus on the popular graph-based edge-betweenness clustering algorithm (section 4.7).

4.1 Introduction to Cluster Analysis

This section provides a definition of the terms clusters analysis and cluster (section 4.1.1). Afterwards, a categorization of clustering methods (section 4.1.2) and a general methodology how to perform a cluster analysis (section 4.1.3) are discussed. Finally, a brief overview of common application areas (section 4.1.4) and important challenges (section 4.1.5) in cluster analysis are given.

4.1.1 Definition

Cluster analysis, or *clustering*, is the method of classifying objects into meaningful sets (Aldenderfer and Blashfield 1984: 5). In contrast to discriminant analyses like pattern recognition and decision analysis which assign predefined labels to the data, establishing rules and separate future data into these categories cluster analysis is about discovering groups inherent to the data (Jain and Dubes 1988; Everitt et al. 2001: 6). The development of cluster analysis is interdisciplinary and includes researchers like taxonomists, psychologists, social scientist and engineers. It therefore subsumes numerical methods of classification from different sciences, e.g. numerical taxonomy (biology), Q analysis (psychology), unsupervised pattern recognition (artificial intelligence), or segmentation (market research) (Everitt et al. 2001: 4).

The basic elements of cluster analysis are the data consisting of observations, objects or patterns p described by a d -dimensional set of variables, or features. Using a specific clustering method these objects are assigned to groups, classes or *clusters* according to the similarity of their feature values. An overview of different definitions of a cluster can be found in Everitt (1974): A general approach defines a cluster by a set of entities which are *alike*, in contrast to entities from different cluster which are not alike. A second definition defines alikeness as the *distance* between two points in the test space. A cluster is then an aggregation of points where the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it. The idea of distance and proximity can also be found in a third definition: Clusters are connected regions of a multi-dimensional space containing a relatively *high density* of points, separated from other such regions by a region containing relatively low density of points. The last definition of clusters does not depend on any type of data analysis (Aldenderfer and Blashfield 1984: 35).

Cormack (1971) and Gordon (1980) define a cluster by its internal cohesion and external isolation. Figure 4-1 shows clusters of different size and shape to illustrate that a formal definition of these two properties has led to the development of numerous criteria to identify and validate the quality of a cluster (see chapter 4.6.5).

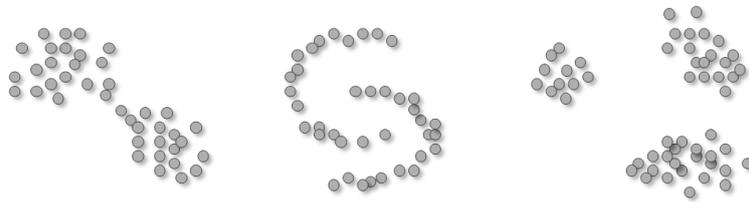


Figure 4-1: Clusters with internal cohesion and/or external isolation. Source: Gordon (1980)

Using a suitable visual representation clusters can easily be recognized by the human mind and one can give a functional definition of a cluster (Jain and Dubes 1988: 2). However, it is difficult to give an operational definition because data objects can be grouped into different clusters with different purposes in mind. Thus, clusters can vary in shape (clumsy or linear clusters), number and size (many small or few large clusters). Additionally, cluster membership can change over time and it is not an easy task to track them. Furthermore, the number of clusters often depends of the aggregation level of the data, e.g. fine or coarse, local or global level.

4.1.2 Categorization of Clustering Methods

According to Jain and Dubes (1988: 56) clustering is a special kind of classification. The relationship between classification in general and clustering has been discussed by Kendall (1988). Lance and Williams (1967) provide a tree of classification types where each leaf of the tree defines a different type of classification problem (see Figure 4-2).

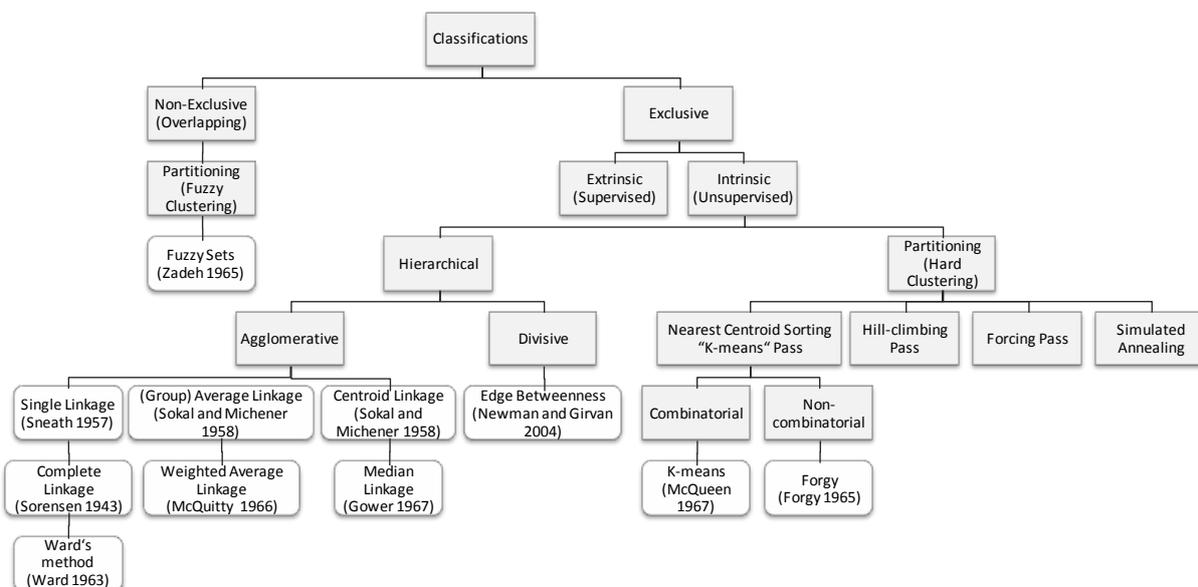


Figure 4-2: Tree of classification types. Based on: Lance and Williams (1967) and Jain and Dubes (1988: 56)

Classification problems can be either exclusive or non-exclusive. Exclusive problems can be extrinsic or intrinsic. Clustering algorithms belong to intrinsic classification problems and can be either hierarchical or partitioning:

- *Exclusive versus non-exclusive.* An exclusive classification is a non-overlapping partition of the set of objects where each object belongs to exactly one subset. Non-exclusive classification generates overlapping partitions where an object can be assigned to several classes, e.g. fuzzy clustering algorithms (e.g. Zadeh 1965).
- *Intrinsic versus extrinsic.* An intrinsic classification uses only the proximity matrix to perform the classification and it is known as “unsupervised learning” in pattern recognition. In contrast, extrinsic classification uses category labels as well as the proximity matrix. To assign the data objects to the categories, a “teacher” is used and the classifier must learn the proper categorization from a training data set. This approach is also called “supervised learning”.
- *Hierarchical versus partitioning.* Hierarchical and partitioning approaches reflect the type of structure imposed on the data. A hierarchical classification produces a nested series of partitions (see chapter 4.4), whereas a partitioning classification produces a single partition (see chapter 4.5). Hard partitioning clustering methods can be further characterized by the type of pass employed. For more details see chapter 4.5.2.4.

The clustering algorithms categorized as exclusive, intrinsic classification problems can be further distinguished due to their primary algorithmic (Jain and Dubes 1988: 57):

- *Agglomerative versus divisive.* Agglomerative hierarchical clustering initially assigns each object to a cluster of its own and gradually merges these clusters into growing clusters until all objects belong to the same cluster (see section 4.4.1). In contrast, divisive hierarchical clustering starts with all objects belonging to one large cluster and then gradually splitting this cluster into smaller pieces (see section 4.4.2). This option is rather a choice of procedure than a different kind of classification.
- *Serial versus simultaneous.* As outlined by Clifford and Stephenson (1975) serial clustering procedures handle the patterns one by one, whereas simultaneous procedures are performed with the entire set of patterns at the same time.
- *Monothetic versus polythetic.* If the data is represented as points in a space a monothetic clustering algorithm uses the features one by one whereas a polythetic clustering algorithm uses all features at once.
- *Graph theory versus matrix theory.* The distinction between graph theory and matrix algebra covers the question which mathematical formalism is appropriate for expressing a clustering algorithm. If classifications are defined by properties such as connectedness and completeness graph theory is applied, otherwise matrix algebra is used to express an algorithm in terms of algebraic constructs, e.g. mean square error. Some algorithms have convenient expressions under both options.

Aldenderfer and Blashfield (1984: 35) distinguish and discuss seven families of clustering methods more or less detailed: (1) hierarchical agglomerative, (2) hierarchical divisive, (3) partitioning, (4) density search, (5) factor analysis, (6) clumping and (7) graph theoretic. Type (1) is often used in biological sciences, type (5) often in psychology. The most popular methods used in social science are (1), (3) and (5). In general, the choice of the clustering method depends on the classification problem itself, the application area as well as the data and the similarity measure used. Similarly, Everitt (2001: 141) give a brief overview with the

following non-exclusive categorization in addition to the popular hierarchical and partitioning clustering methods including:

- *Density search clustering*. Clusters are concentrated in relatively dense patches in a metric space, e.g. mode analysis (Wishart 1969), nearest-neighbor clustering (Wong 1982; Wong and Lane 1983).
- Methods which allow *overlapping clusters*, e.g. clumping using cohesion coefficients (Needham 1967; Spärck Jones and Jackson 1967; Parker-Rhodes and Jackson 1969), graph-based methods like the B_k technique (Jardine and Sibson 1968; Rohlf 1975), additive clustering (Shepard and Arabie 1979), pyramids (Diday 1986).
- *Direct clustering* of variables and objects simultaneously by using the data matrices instead of the proximity matrices, e.g. hierarchical classes (De Boeck and Rosenberg 1998), error variance techniques (Eckes and Orlik 1993).
- *Constrained clustering*. The cluster memberships are determined partly by external information, e.g. spatial contiguity (Gordon 1999).
- *Fuzzy methods*. Objects have a membership function due to the strength of membership to each cluster, e.g. fuzzy set theory (Zadeh 1965), grade-of-memberships analysis (Woodbury and Manton 1982), or fuzzy K -means (Bezdek 1974), measures assessing fuzziness by Rousseeuw (1987) (silhouette plot, see section 4.6.4.3) and Dunn (1974) (Dunn's partition coefficient, see section 4.5.3).
- *Neural networks*. Pattern recognition algorithms to imitate the computational capabilities of large, highly connected networks, e.g. Kohonen self-organizing map (Kohonen 1982).

4.1.3 Clustering Methodology

The methodology of cluster analysis presented in this section is based on Jain and Dubes (1988). It is more a proposal than an accepted standard. Case studies that follow this methodology are difficult to find because each application has its special needs. Nevertheless, the method by Jain and Dubes (1988) has shown to be most suitable in the course of this work.

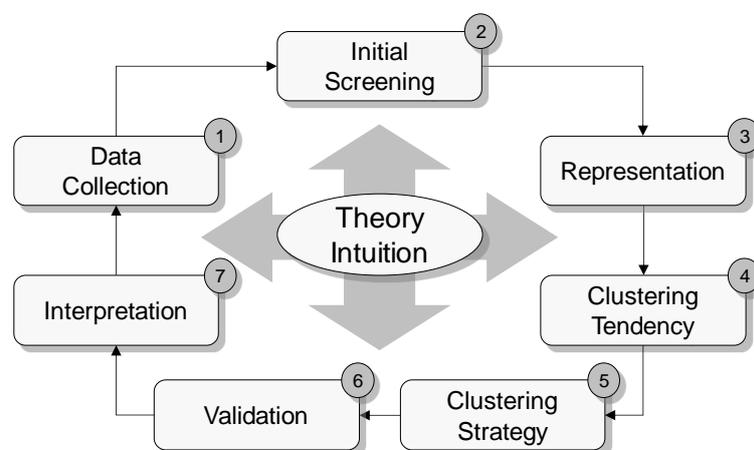


Figure 4-3: Clustering methodology. Source: Jain and Dubes (1988: 135)

Figure 4-3 contains the major steps to be considered when undertaking an exploratory data analysis whose central component is a cluster analysis (Jain and Dubes 1988: 135): (1) data

collection, (2) initial screening, (3) representation, (4) clustering tendency, (5) cluster strategy, (6) validation and (7) interpretation. The process is an endless loop in which new insights are obtained and new ideas are generated at each iteration of the loop. However, the needs and special circumstances of an individual problem may make parts of the methodology unnecessary or even impossible.

The first step in cluster analysis is the careful recording of data. In *data collection* it is essential to follow the standards and recommended methods in the area of application. The amount and type of data will strongly influence the strategies available for analyzing the data. In the second step, *initial screening*, the raw data has to be adapted for formal analysis. Besides some normalization the data should be reviewed in a rough manner. Outliers and features with no contribution to the analysis can be removed (see section 4.2.3). Next, the data has to be transformed into a suitable *representation* for further analysis (see sections 4.2.1 and 4.3). This includes e.g. choosing a proximity index, projecting the data to a suitable feature space, examining the intrinsic dimensionality and performing multidimensional scaling. The suitability of any of these procedures depends on the problem at hand. The result should either be a pattern matrix or a proximity matrix. An important question is whether the data are actually random or whether there is a structure naturally included in the data. Therefore, the step of *clustering tendency* is crucial for the quality of the results of the analysis (see section 4.6). Nevertheless, it is often ignored or carelessly treated. The step of *cluster strategy* deals with the major question of the appropriate clustering procedure including the choice between hierarchical and partitioning procedures (see sections 4.4 and 4.5). Each type of clustering allows choosing between different algorithms, output representations or parameters and has its own requirements and restrictions, like the amount of data, as measured by the number of patterns and the number of features. Another strategic decision is whether to cluster only the patterns or the features or both. Although *cluster validation* can become even more time and resource consuming than collecting the data and the clustering itself, it is essential for generating reliable results and should never be neglected. The validation can involve studying the stability of the analysis by imposing small changes in the data and then repeating the analysis. The data can be disturbed by adding random noise or by removing some patterns or features (Strauss 1973; Gnanadesikan et al. 1977; van Ness 1983). The process of cluster analysis proposed by Jain and Dubes (1988) concludes with the *interpretation* of the results. Here, one will compare the results with those of previous studies in related as well as complementary application scenarios and draw conclusions about the data. The interpretation of exploratory analysis is mainly based on comparison, experience, the data and the purpose of the analysis. However, the fact that cluster analysis is exploratory by nature does not mean that only ad hoc procedures can be adopted.

Another framework for cluster analysis can be found in Milligan (1996). It is based on the findings of several studies about cluster analysis and consists of the following seven steps: (1) objects to cluster, (2) variables to be used, (3) variable standardization (optional), (4) proximity measure, (5) clustering method, (6) number of cluster and (7) replication, testing and interpretation. Step (7) includes tests for the clustering tendency of the data. Although each cluster analysis should start with a test for the absence of structure in the data, they hardly provided in any practical applications (Everitt et al. 2001: 180). One reason is simply

the lack of appropriate tests. Compared to Milligan's framework the cluster methodology suggested by Jain and Dubes (1988) is much more faceted.

4.1.4 Application Areas

Cluster analysis is a subject of multiple disciplines like computer sciences, biology, sociology, linguistics and physics. In general, clustering algorithms can be employed to classify or categorize similar objects of any kind.

Cluster analysis is widely used in *strategic management* and *marketing research*. Working with multivariate data from surveys and test panels market researchers use cluster analysis to partition the general population of consumers into market segments and to better understand the relationships between different groups of consumers or potential customers. This approach belongs to the area of data mining. Application scenarios are market segmentation and determining target markets, product positioning, new product development or selecting test markets (see e.g. Green et al. 1967; Punj and Stewart 1983; Arabie and Hubert 1994; Dolcinar 2003; Chye et al. 2004).

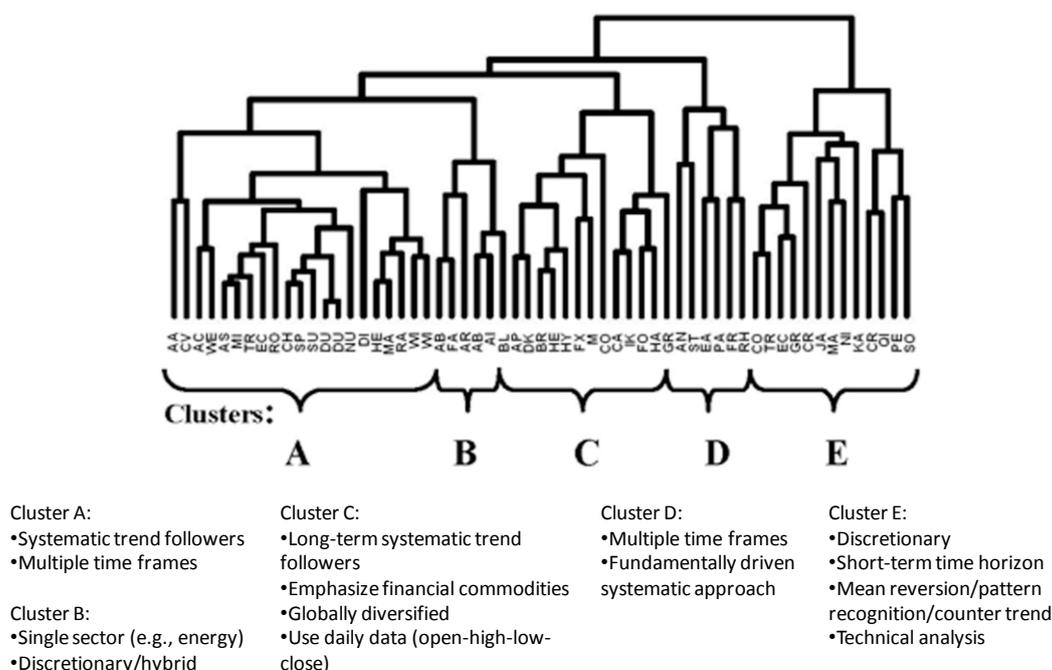


Figure 4-4: Complete linkage agglomerative cluster analysis on investment strategies of 59 Barclay Managed Futures (CTA) fund managers based on their investment strategies January 2004 - December 2007. Cluster labels assigned after clustering. Source: Noma and Shtrapeina (2010)

Cluster analysis is also applied to *credit scoring* and *account management* (e.g. Edelman 1992; Hsieh 2005). To improve credit scoring, cluster analysis treats each borrower as an individual case and attempts to find cluster of borrowers that are of financial benefit. Miceli and Susinno (2003; 2004) classify various hedge funds based on their fund returns. Here, clustering can be used to monitor potential style drifts, conduct peer group analysis and identify a proper benchmark for groups of funds. Applied to investigations on portfolio management behavior, Gibson and Gyger (2007) conclude that cluster analysis shows that certain managers do not follow their investment style consistently over time. A clustering-based analysis of the effects of changing market conditions during the financial crisis of 2008–2009 on 59 Barclay Managed Futures (CTA) fund managers and their responses can be

found in Noma and Shtrapeina (2010) comparing the clustering results of their investment strategies 2004-2007 (see Figure 4-4) versus 2008-2009.

Another common application area of cluster analysis is *information retrieval*. Here, the objects are documents and the features are topics. Some preliminary text mining procedures have to be employed to retrieve meaningful topics from each document (see chapter 3). For example, cluster analysis can then be used for *document clustering* to divide or classify a set of documents, such as newspaper articles or web pages, into genres (see e.g. Schütze and Silverstein 1997; Hatzivassiloglou et al. 2000; El Sayed et al. 2008). Document clustering is also employed for *search result grouping*: In the process of intelligent grouping of the search results, clustering may be used to create a more relevant result. Clustering can be used to group all the shopping items available on the web into a set of unique products. Other application areas of cluster analysis in information retrieval belong to research on computational linguistics, e.g. *word sense discrimination* (see e.g. Schütze 1998; Savova et al. 2006) or the generation of *taxonomies* using hierarchical clustering algorithms (see e.g. Sokal and Sneath 1963).

Cluster algorithms like partitioning K -means clustering can be used for *image segmentation* based e.g. on areas of similar color (see e.g. Ittner et al. 1995; Turi and Ray 1998; Ray and Turi 1999; Shi and Malik 2000).

In social network analysis clustering may be used to recognize communities within large groups of people. The use of cluster analysis in SNA and on social corpora is explained in section 4.7

4.1.5 Challenges

Using clustering methods one should be aware of some problems of cluster analysis in general (1984: 14). First, most clustering methods are regarded as heuristics which means that they are not supported by an extensive body of statistical reasoning like, for example, the factor analysis. Nevertheless, they can have some important mathematical properties (Jardine and Sibson 1971). Second, cluster analysis methods have evolved from different disciplines. Each clustering method is influenced by the biases and preferences of the discipline it was originally developed for. Third, different clustering solutions will be obtained from different clustering methods. For this reason, it is important to carefully choose the method and thoroughly validate the solution. Finally, clustering algorithms are structure-imposing: They will always find some clustering structure in the data even if they are totally random. This is important to notice as it contradicts the strategy of cluster analysis which is structure-seeking (Aldenderfer and Blashfield 1984: 16). Therefore, if a method is said to find spherical clusters, like the complete or average linkage method, it rather imposes spherical clusters on the data regardless of the natural structure (Everitt et al. 2001: 64). This is important to be kept in mind when performing cluster analysis and interpreting the results.

4.2 Preparation of the Data Set and Initial Screening

During the process of data collection various error conditions, like outliers or noise features, can be imposed on the data. Thus, most often an initial screening of the raw data will be necessary to prepare it for a formal analysis. In this section the preparation of the data set and

its initial screening is explained in some detail. First, different techniques how to select and represent the data (see sections 4.2.1 and 4.2.2) are presented including the choice of data types and scales, existing approaches of graphical data representation and cluster visualization as well as standardization and weighting of variables. Afterwards, the important issue of how to detect and cope with missing data, uninformative data, or outliers is discussed (see section 4.2.3).

4.2.1 Representation

Usually, one will decide early in the clustering process not only on which data to collect but also on how to represent the results. The appropriate representation of the data as well as the final clustering results include decisions about the data type and scale (data representation, see section 4.2.1.1), data and cluster visualization (see section 4.2.1.2) as well as the choice of cluster representative (see section 4.2.1.3).

4.2.1.1 Data Representation: Data Types and Scales

Jain and Dubes (1988: 12) use the categorization of data types and scales by Anderberg (1973) which fits the needs of cluster analysis. Figure 4-5 illustrates a taxonomy of data representation types.

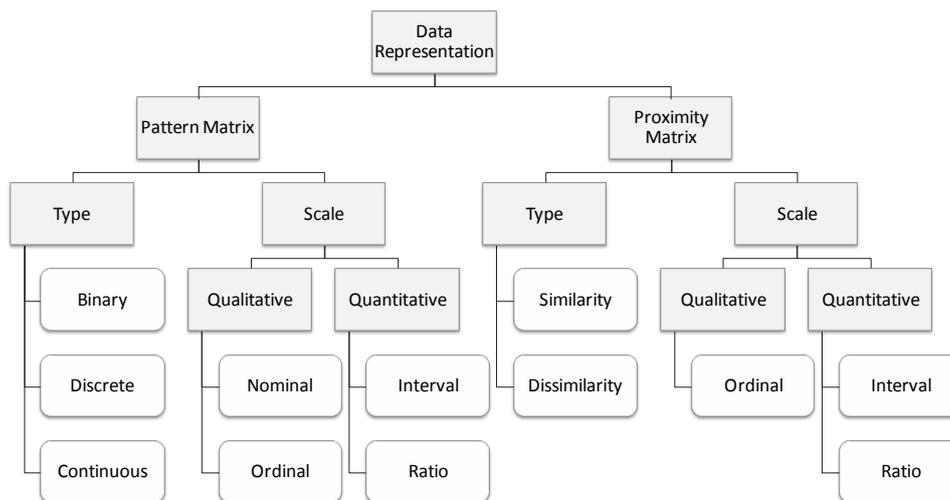


Figure 4-5: Taxonomy of data representation: formats, types, and scales of data. Source: Jain and Dubes (1988: 13)

The *format* of the data refers to the representation as pattern matrix or proximity matrix. The *pattern matrix* is a $n \times m$ -matrix which contains the raw data as n objects, or patterns, as rows and the m recorded features as columns. The *proximity matrix* is a $n \times n$ -matrix which contains the relationships between each pair of the n objects expressed by the proximity index calculated on the raw data or the pattern matrix.

Features as well as proximity indices can be distinguished due to their *type* of data: *binary*, *discrete*, or *continuous*. The data contained in a proximity matrix can be typed either a *similarity* or *dissimilarity*. In cluster analysis similarity measures as well as dissimilarity measures, often called distance measures, are used to describe the relationship between two data objects. In this work they are uniformly referred to as *proximity measures* if it is unnecessary to make a distinction between similarity and dissimilarity.

Besides the concrete type of the data, features and proximity indices can be characterized by their data *scale*. The data scale indicates the relative significance of numbers and can be dichotomized into *qualitative scales* and *quantitative scales*. Stevens (1946) introduces the “theory of scales of measurements” and he proposes four levels of measurement, which today are uniformly accepted in mathematics and statistics to classify measurements. Following his terminology scales can be nominal, ordinal, interval or ratio (Stevens 1946): Nominal and ordinal measurements belong to the category of qualitative scales, interval and ratio measurements to the category of quantitative scales (Jain and Dubes 1988: 12). The conversion of one scale to another is explained by Anderberg (1973). This might be necessary as clustering methods use quantitative measures of distance to assign cluster with labels, so a nominal scale can be generated from a quantitative scale. Multidimensional scaling changes ordinal scale into ratio scales (see section 4.2.1.2).

4.2.1.2 Graphical Data Representation and Cluster Visualization

The graphical representations of the raw data and the clustering results typical for cluster analysis are of minor interest in the course of this work as the results are projected onto the graphical representation of the network as a graph. Therefore, only a brief overview is given in this section.

The graphical representation of the original data can provide insights into the structure of the data. The human visual system is able to detect patterns which can help to decide whether the data may be clustered or not (Everitt et al. 2001: 11). The complex recognitions process involves the assessment of relative distances between points. An account of how humans visually detect clusters can be found in Feldman (1995). There are a number of methods to visualize group memberships in univariate or bivariate data. Using suitable projection methods they can also be applied to multivariate data. Popular methods are histograms and scatterplots (see Figure 4-6). Non-parametric density estimators can be employed to discover clustering structures in the data visualization (Silverman 1986; Scott 1992; Wand and Jones 1995; Simonoff 1996).

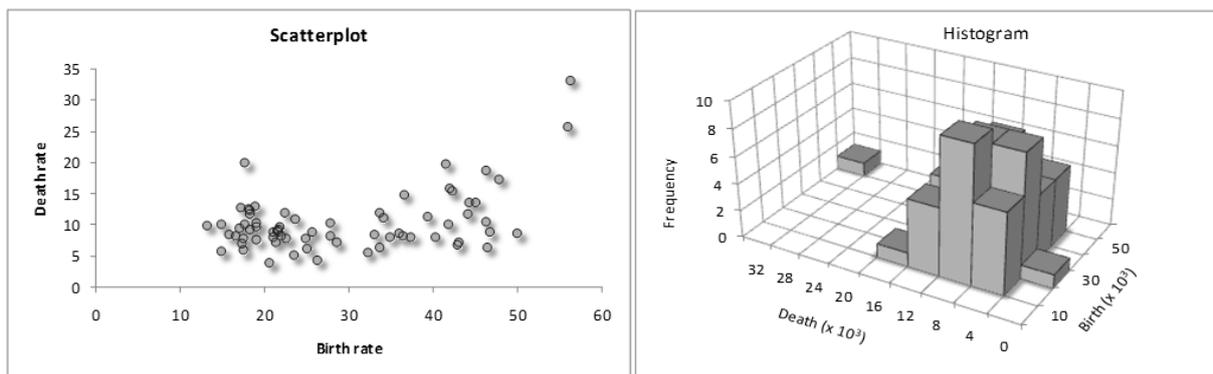


Figure 4-6: Graphical representation of bivariate data as scatterplot or histogram: Birth and death rates for 69 countries. Based on: Everitt et al. (2001: 14)

To visualize multivariate data as histograms or scatterplots projection methods like principle component analysis and multidimensional scaling have to be applied (see e.g. Jackson 1991; Everitt and Dunn 2001). These methods transform the dimensions to one or two new dimensions while keeping the clustering structure inherent in the original data.

Hierarchical clustering techniques organize data into a nested sequence of clusters (Jain and Dubes 1988: 90). The results can be visualized as dendrograms showing how the objects are being merged into clusters or split at successive levels of proximity. After calculating the entire hierarchy it has to be decided which level of proximity represents best the true structure of the data. However, dendrograms get impractical with more than a few hundred patterns (Jain and Dubes 1988: 90). Clustering visualizations using dendrograms can be found in sections 4.4.1.6 and 4.6.4.3.

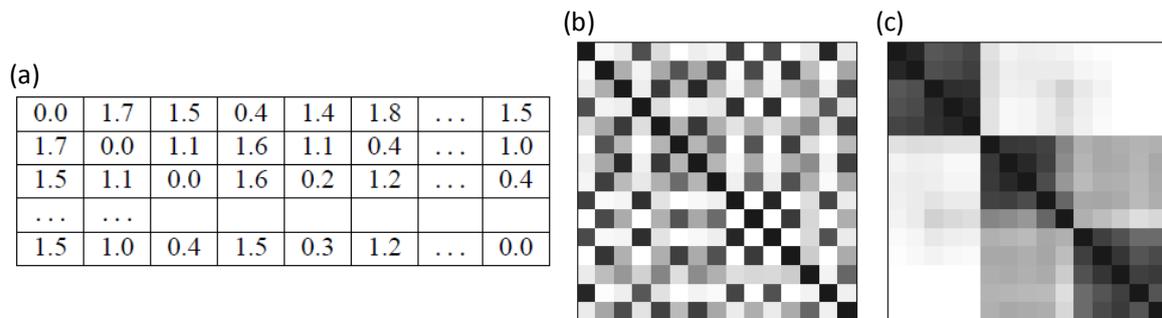


Figure 4-7: Illustration of matrix shading: (a) proximity matrix based on Iris data set by Merz et al. (1997) using Euclidean distance; (b) randomly ordered shaded proximity matrix; (c) reordered shaded distance matrix using a seriation algorithm. Source: Wang et al. (2002)

With partitioning clustering, especially when overlapping group memberships are allowed, there is no such visual representation like the dendrogram. Among the existing methods to represent partitioning clustering results are the graphical representation of the proximity matrix based on matrix shading (Sokal 1966) or unclassed choropleth mapping (Gale et al. 1984), representing objects as icons with size and shape being related to their feature values (Wainer 1983), taxometric maps (Carmichael and Sneath 1969), silhouette plots (Rousseeuw 1987), clustergrams (Schonlau 2002), distance graphs between cluster centroids (Chen et al. 1974; Gnanadesikan et al. 1977; Chambers and Kleiner 1982; Gnanadesikan et al. 1982), or cluster validity profiles (Bailey and Dubes 1982). An example of matrix shading is given in Figure 4-7. Rousseeuw et al. (1989) suggest computing principal components of cluster membership coefficients for fuzzy clusters. Here, the cluster memberships are used instead of the original measurements. Methods that visualize clusters as single aggregated objects or only by their cluster representatives do not allow insights into the individual objects (Kaufman and Rousseeuw 1990: 121).

4.2.1.3 Cluster Representatives

For each cluster a *representative* object, also called *centrotype* or *prototype*, can be calculated which summarizes various aspects of the structure of the data (Jain and Dubes 1988: 90; Kaufman and Rousseeuw 1990: 68). Using a partitioning algorithm representatives are recalculated at each iteration and then the data objects are assigned to the cluster with the most similar representative. Thus, the choice of the representative has some influence on the results of this clustering procedure. Additionally, cluster representatives can be used for data reduction or characterization of the cluster solution.

One way to choose a cluster representative is to select one object from each cluster so that it is centrally located in the cluster it defines. A *medoid* is the object in a cluster with minimal average distance to all other objects in the same cluster (Kaufman and Rousseeuw 1990: 40).

Compared to other representatives like centroids medoids are more robust against outliers and can deal with all kinds of proximity indices (Kaufman and Rousseeuw 1990: 41).

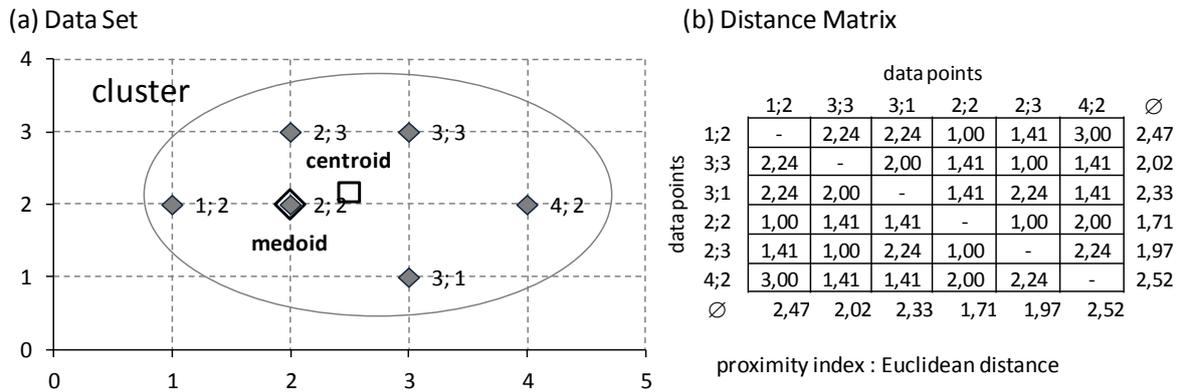


Figure 4-8: Cluster representatives: medoid versus centroid

If the objects can be described by data points in a d -dimensional metric space the representative can be calculated by averaging the measurement values along each dimension. In this case, the representative is called *centroid* and it is not necessarily one of the objects in the data. It can only be calculated on interval-scaled data (Kaufman and Rousseeuw 1990: 112). The popular partitioning K -means type algorithms FORGY by Forgy (1965) and the K -means method by McQueen (1967) are variance minimization techniques using the square-error criterion with centroids (Kaufman and Rousseeuw 1990: 112). Figure 4-8 illustrates the difference between medoid and centroid as cluster representative. Choosing the representative among the data points leads towards using point (2;2) as medoid. Centroid (2.5;2.17) has no representation as data object.

4.2.2 Data Measurement

The choice of the variables and measurement units is one of the most critical but also most troublesome and least understood steps in the research process (Aldenderfer and Blashfield 1984: 19; Gnanadesikan et al. 1995: 114). Depending on the quality of the selection process it can notably facilitate cluster recovery as well as prevent the detection of even obvious clustering structures. According to Gnanadesikan et al. (1995: 114) there are two different approaches in the choice of variables for cluster analysis: variable selection and variable importance. *Variable selection* refers to the selection of the appropriate initial variables out of the set of all possible variables. *Variable importance* consists of assigning weights to the variables according to their relative importance derived from statistics computed on the data. Often this includes a preliminary standardization of the variables. Therefore, this section is organized as follows: First, an introduction to the variable selection approach is given (section 4.2.2.1) and then standardization and weighting techniques are explained which refer to the variable importance approach (sections 4.2.2.2 and 4.2.2.3).

4.2.2.1 Variable Selection

Reducing the number of variables early in the clustering process is economic for future measurement. A significant advantage of variable selection compared to variable importance is that there are no difficulties of understanding or interpreting the meaningfulness of the derived weights (Gnanadesikan et al. 1995: 114). Ideally, when performing cluster analysis an

explicitly stated theory to support the classification should be established which will also lead the rational choice of the variables to be used in the study (Aldenderfer and Blashfield 1984: 19). In practice, a theoretical foundation of the classification problem is often implicit, which makes it difficult to assess the relevance of the variables to the problem. Furthermore, cluster analysis is often performed by collecting and analyzing as many variables as possible in the hope that there will emerge a structure if only enough data is available. Aldenderfer and Blashfield (1984: 20) use the term “naïve empiricism” to distinguish this erratic approach from valuable empirical studies. Due to Everitt (1979) the heuristic nature of cluster analysis and its many unsolved problem add to the danger adherent to thoughtless use of variables in cluster analysis. Variable selection can also be regarded as a special case of variable weighting where variables with weights 0 being excluded from further analysis (Gnanadesikan et al. 1995: 114).

4.2.2.2 Standardization of Variables

Sometimes the purpose and the requirements of the analysis and the type of the data requires a *standardization*, also called *normalization*, of the data. Standardized or normalized data are unitless and thus independent from the choice of measurement units (Kaufman and Rousseeuw 1990: 6). The j th feature for the i th pattern in the $n \times d$ pattern matrix is denoted x_{ij}^* , the standardized value is denoted as x_{ij} . To standardize these data the *mean value* and the *standard variance* of each feature j is used (Jain and Dubes 1988: 24; Kaufman and Rousseeuw 1990: 8):

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_{ij}^* \text{ (mean value),} \quad \sigma_j = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_{ij}^* - \mu_j)^2} \text{ (standard variance)}$$

The standard deviation is a measure of spread, or dispersion, of the data (Kaufman and Rousseeuw 1990: 8).

However, wrongly recorded large values will have strong influence on the standard variance. Furthermore, using this measure will smooth the effect of outliers in such ways that they will become less obvious. Therefore, Kaufman and Rousseeuw (1990: 8) and Hartigan (1975: 299) suggest a dispersion measure that is not too sensitive to outliers. According to Hampel et al. (1986) the *mean absolute deviation* is more robust against outlying observations:

$$\sigma_j = \frac{1}{n} \sum_{i=1}^n |x_{ij}^* - \mu_j| \text{ (mean absolute deviation)}$$

Using this measure outliers can still be recognized in the standardized data (Kaufman and Rousseeuw 1990: 9). However, Milligan and Cooper (1988) recommend the use of a non-robust deviation measure for exactly the opposite reason.

There are two ways of normalizing the data, regardless of which standardization measure is used: (1) Normalizing the data with only the mean value makes the feature values invariant to rigid displacement of the coordinated, (2) normalizing the data with the mean value and the variation measure translates and scales the axes so that all features have zero mean and unit variance (Jain and Dubes 1988: 24):

$$x_{ij} = x_{ij}^* - \mu_j \text{ (1),} \quad x_{ij} = \frac{x_{ij}^* - \mu_j}{\sigma_j} \text{ (2)}$$

The second formula is sometimes called Z-score (Kaufman and Rousseeuw 1990: 9). Other types of normalization include scaling by range (Carmichael et al. 1968) and heterogeneity measure (Hall 1969).

There is some controversy about the benefits and effect of standardization on the results of cluster analysis (Aldenderfer and Blashfield 1984: 20; Kaufman and Rousseeuw 1990: 10). Especially when the variables have an absolute meaning standardization should not be used. Additionally, standardization will reduce the differences between objects and thus dampen the clustering structure inherent in the data (Everitt 1980; Jain and Dubes 1988: 24; Kaufman and Rousseeuw 1990: 10). Edelbrock (1979) and Matthews (1979) state that standardization has only little effect on the clustering structure, whereas others like Milligan (1979) could show a negative effect of standardization compared to a known optimal classification. The decision whether to use standardized data or not also depends on the proximity measure used (Jain and Dubes 1988: 24). For example, using the Euclidean distance features with large range get more weighting than those with small range. Sometimes this effect can be desirable, expressing a natural weighting on different features. If not, normalizing these features, or rescaling them to the same measurement unit, will be a good solution.

Sometimes the variables used are highly correlated. According to Aldenderfer and Blashfield (1984: 21) the “uncritical use of highly correlated variables [...] is essentially an implicit weighting of these variables”. Here, factor analysis or principal component analysis would be appropriate to reduce the dimensionality of the data and obtain uncorrelated variables. However, both approaches will blur the relationships between clusters.

Depending on the choice of measurement unit different weights are assigned to the variables. Variables with small units can have values of a relatively large range which will have a large effect on the resulting structure and vice versa (Kaufman and Rousseeuw 1990: 10). Standardization will remove the effects of measurement units from the data and, consequently, will remove at least part of the clustering structure as well. To decide whether the impact of different measurement units on the structure of the data is desirable or not is up to the researcher and strongly depends on experience and the particular application (Kaufman and Rousseeuw 1990: 10). Everitt et al. (2001: 102) mention the circular problem of standardization where one needs to know the group memberships to standardize variables and at the same time one needs to know the standardized variables to estimate group memberships. This problem can be overcome with employing an adaptive clustering algorithm (see for example Diday and Govaert 1977; Lefkovitch 1980; De Sarbo et al. 1984; Lefkovitch 1987; Fowlkes et al. 1988; Gnanadesikan et al. 1995).

4.2.2.3 Weighting of Variables

Sometimes proximity measures are calculated from weighted variables (see also chapter 3, section 3.4.3). *Weighting* means to change the impact of a variable on the cluster analysis from what has been recorded by the values of this variable to what the researcher believes to be more appropriate due to some a priori knowledge (Williams 1971; Aldenderfer and Blashfield 1984: 21). Applying weights to the raw data according to their perceived importance has the same effect as changing the measurement units and rescaling the data (Kaufman and Rousseeuw 1990: 13). The concept of weighting variables is related to the concept of standardization and is equally controversially discussed. Although there may be

situations where weighting the variables after recording will be useful and necessary it is no easy task to perform and few guidelines exist (Aldenderfer and Blashfield 1984: 23).

In general, there are two possible reasons for weighting variables (Kaufman and Rousseeuw 1990: 14). Either it turns out after recording the data that a wrong measurement unit has been used or some variables are regarded as more important than other variables according to some background knowledge. In the first case it is preferable to standardize these variables instead of using weighted proximity measures. In the second case, different weights can be imposed on the data or the measurement unit can be changed.

Gnanadesikan et al. (1995) evaluate nine different weighting techniques which can be described in terms of a weighting matrix that is part of the squared distance function. The weighting matrix was calculated by using (1) equal weight scaling using Euclidean distance, (2) standardization of variables dividing them by their own standard deviation (autoscaling), (3) standardization of variables with sample range (range scaling), (4) variance decomposition based on between-cluster pairs, (5) variance decomposition with farthest neighbors (weighting based on estimates of within-cluster variability), (6) estimation of between to within sums-of-squares ratios, (7) weighted Euclidean distances that are optimally suited for representation by an ultrametric tree or dendrogram, (8) combining the weighting and clustering process using a weighted version of K -means clustering and (9) separation of clusters by analysis-of-variance statistic in reference to the estimated expected value in the absence of clustering (forward variable selection). Method (6) involves a priori knowledge about the structure and methods (8) and (9) involve cluster analysis. The results from the experiments can be summarized as follows (Gnanadesikan et al. 1995: 134):

- Equal weight scaling (1), autoscaling (2) and range scaling (3) of the data are generally ineffective but range scaling is preferable to autoscaling.
- Weighting based on estimates of within-cluster variability (4, 5), chosen with some care, work well overall and dominate (1), (2) and (3) in all test scenarios.
- Weighting aimed at pulling apart clusters by factoring in estimated between-cluster variability (6) can enhance performance when some variables have strong cluster structure.
- Weighting to optimize the fitting of an ultrametric tree (7) is often even less effective than (1) and (2).
- Weighting that is intertwined with K -means cluster optimization (8) is more effective overall than (7) but never a top performer.
- Forward variable selection (9) is often, but not always, among the better performers.

The study concludes that weighting schemes based upon carefully chosen estimates of within-cluster and between-cluster variability, method (6), are generally more effective and do not require any knowledge of the cluster structure. However, there is still no overall satisfying procedure (Gnanadesikan et al. 1995: 135). In practice, interpreting functions derived from the data and assessing the relative influences of initial variables on them (e.g., discriminant functions) is highly subjective.

4.2.3 Dealing with Noise

Besides the choice of the initial variables and their representation the process of data preparation and initial screening also deals with noise in the data captured. Noise refers to missing values (see section 4.2.3.1) and uninformative variables (see section 4.2.3.2) as well as to objects that are too far removed from the others measured by some proximity index (see section 4.2.3.3).

4.2.3.1 Missing Data

If not all measurements are available in the recorded data set the data matrix contains *missing values* (Kaufman and Rousseeuw 1990: 14). This can have several reasons (Jain and Dubes 1988: 19; Kaufman and Rousseeuw 1990: 14):

- (1) *Inapplicable question*: There is no value which could be measured (e.g. hair color of a bald person).
- (2) *Unavailability*. There is a value but the information is not available (e.g. patient cannot remember the answer).
- (3) *Unwillingness*. There is a value and the information is available but the questioned subject does not give any answer.
- (4) *Equipment failure & recording error*. There is a value but the information cannot be recorded (e.g. malfunction of measurement instruments).
- (5) *Carelessness*. There is a value and the information is available but has been lost or has not been recorded at all (due to oversight or lack of time).
- (6) *Ambiguity*. There are several possible values.

The first case will often occur in clustering keyword sets derived from text or documents. Additionally, values can be missing because they are already wrongly recorded in the original data, like for example misspelled keywords. Sometimes it is not easy to decide whether a value is missing for some reason or there is simply no value of a feature for some of the objects. This decision can have a strong influence on the choice of the proximity measure, especially if matching coefficients are used (see Aldenderfer and Blashfield 1984: 29).

Missing values can be encoded in the data matrix by some indicator, e.g. an unusually large or a negative value in a data set with only positive values. Thus, a clustering algorithm can be designed which recognizes the missing values (Kaufman and Rousseeuw 1990: 14). How to deal with missing data has been discussed by Sneath and Sokal (1973), Dixon (1979), Kittler (1978), and Zahoruiko and Yolkina (1982). Possible solutions strongly depend on the size of the data set and the type of analysis (Jain and Dubes 1988: 19). Dixon (1979) discusses several simple and computationally inexpensive techniques for handling missing data. In general, there exist three approaches: eliminate part of the data, estimate the missing values, or compute an estimated distance between to data objects with missing values. The simplest but not most efficient solution is to delete the patterns or features that contain missing data (elimination). This method is only recommended when many or all values of a pattern or feature are missing (Kaufman and Rousseeuw 1990: 14). If a value of the data object x_i is missing, another method is to find the K nearest neighbors of x_i and replace the missing value with the average of the corresponding values of the neighbors (estimation). The value of K should be a function of the size of the pattern matrix. A third method is to compute the

proximity between two data objects with missing value by computing the proximity only from those features with values present for both objects (Kaufman and Rousseeuw 1990: 15). Similarly, the standardization of a feature with some values missing should only take present values into account. If there is a pair of objects with no common measured variables either object could be removed or some average proximity of the rest of the data could be used as a proximity index. Alternatively, the missing values of a feature could be replaced by the mean value of this feature (see for a detailed algorithm Jain and Dubes 1988: 98; Kaufman and Rousseeuw 1990: 15). Based on the results from his experiments Dixon (1979) recommends the third method as the best overall.

4.2.3.2 Uninformative Variables

Uninformative variables are recorded in the data set but contain no relevant information for the task performed from the cluster analysis. According to Kaufman and Rousseeuw (1990: 14) these variables are “worse than useless” as they flatten the effect of clustering tendencies in the data. Thus, these “trash variables” add some randomness to the proximities between objects and the clustering obtained from these proximities becomes less apparent. In the worst case no meaningful clustering can be obtained. While preparing the data set for the cluster analysis one should carefully overlook the data for uninformative variables and delete them from the feature vector. To decide whether a variable belongs to this group of noise is often a nontrivial task. It depends on the specific purpose of the analysis and the application area and the decision involves trial and error as well as prior knowledge and common sense (Kaufman and Rousseeuw 1990: 14). How to identify and deal with uninformative data is also an important task in information retrieval and text mining, especially during feature generation and feature selection, e.g. removing stop words (see chapter 3).

4.2.3.3 Outlier Detection

Sometimes a pattern is far removed from the rest of the data so it might have been included by error, such as a mistake in data entry due to noise in the measurement process or error in data coding. These patterns are regarded as *outliers* (Jain and Dubes 1988: 98). Forcing an outlier to belong to a cluster seriously disturbs the shape of that clusters and will have a strong influence on the whole clustering structure. For example, an outlier can force a clustering algorithm to put two compact and well-separated groups into the same cluster. Often outliers can be found in large data sets (Kaufman and Rousseeuw 1990: 153) as the possibility of error and noise in the data increases with the size of the objects and features recorded. Obviously, partitioning clustering algorithms are much more affected by outliers than hierarchical clustering algorithms (Al Hasan et al. 2009: 2): Partitioning clustering assign all objects to a predefined number of clusters at the same time. Hierarchical algorithms produce a nested series of partitions by merging (agglomerative), or splitting (divisive), only two clusters at each level. Thus, objects far removed from all other objects will either be merged on one of the last levels, or be split on one of the first levels.

The existence of outliers can be due to aberrant patterns as well as representatives of poorly sampled group of data (Aldenderfer and Blashfield 1984: 61). In the first case it is best to identify an outlier and remove it from further considerations (Kruskal 1977). In the second case the data generation method itself should be revised. Therefore, outliers should always be

treated carefully in a separate or preliminary analysis (Jain and Dubes 1988: 98). Some clustering algorithms also treat small clusters as outliers. Outliers can be identified by using a threshold. If the similarities of an object to all other objects fall below this threshold the object can be regarded as an outlier and therefore removed from the data set. Singleton clusters and too small clusters can also be identified by a threshold of the minimum number of objects in a cluster. This threshold is used by the partitioning square-error minimization algorithm (Forgy 1965; Jain and Dubes 1988: 101). The threshold which divides anomalous and non-anomalous data numerically is crucial for the clustering results. Methods for univariate outlier detection are often based on estimation of location and scatter (e.g. mean ± 2 ·scatter) or on quantiles of the data. However, they are independent from the sample size and no distinction is made between outliers and extremes of a distribution. The basis for multivariate outlier detection is the Mahalanobis distance. The standard method for multivariate outlier detection is robust estimation of the parameters in the Mahalanobis distance (see section 4.3.3) and the comparison with a critical value of the χ^2 distribution (Rousseeuw and Van Zomeren 1990). Again, values larger than this threshold are not necessarily outliers but can still belong to the data distribution. In order to distinguish between extremes of a distribution and outliers, Garrett (1989) introduced the χ^2 plot, which draws the empirical distribution function of the robust Mahalanobis distances against the χ^2 distribution. A break in the tail of the distributions is an indication for outliers, and values beyond this break are iteratively deleted.

4.3 Proximity Measures

Besides the more popular proximity measures, like the Euclidean or Manhattan distance, the Jacquard coefficient or Salton's cosine similarity, a large number of distance measures have been designed for different clustering problems and application areas. A detailed overview can be found in Anderberg (1973). The choice of the best suited proximity measure depends on the data, the application area and the actual purpose of the analysis. It is an important part of the representation of the data and the cluster strategy and will strongly influence the results of the cluster analysis (Jain and Dubes 1988: 3). There are few guidelines how to choose the appropriate proximity measures. Knowledge of the context and the type of data will lead the decision (Everitt et al. 2001: 179).

Sneath and Sokal (1973) propose a classification of proximity measures which subdivides them into four groups: correlation coefficients, distance measures, association coefficients, and probabilistic similarity measures. Probabilistic measures can only be used with binary data. They are more popular in numerical taxonomies and ecology than in social sciences and therefore not further discussed in this chapter. Another distinction is the type of data the measure can be used with. Additionally, there is a distinction between proximity measures whether they can be interpreted as metrics. This chapter is organized into an introduction of the mathematical properties and benefits of metrics as proximity measures (see section 4.3.1) followed by an overview of commonly used correlation coefficients (see section 4.3.2), distance measures (see section 4.3.3) and association coefficients (see section 4.3.4). Additionally, the proximity matrix calculated before and updated during the clustering procedure may contain ties, which are proximities between pairs of objects having the same values. Possible solutions to this problem will be discussed at the end of this chapter (see section 4.3.5).

4.3.1 Proximity Measures as Distance Metrics

The four standard criteria to decide whether a proximity measure is a true metric are symmetry, triangular inequality, distinguishability of non-identicals and indistinguishability of identicals. Let i , j and k be three entities. Then, the four criteria can be formalized as follows:

- | | |
|---|---|
| (1) <i>Symmetry</i> | $d(i, j) = d(j, i) \geq 0$ |
| (2) <i>Triangle inequality</i> | $d(i, j) \geq d(i, k) + d(k, j)$ |
| (3) <i>Distinguishability of non-identicals</i> | <i>if $d(i, j) \neq 0$, then $i \neq j$</i> |
| (4) <i>Indistinguishability of identicals</i> | $d(i, i) = 0$ |

Criterion (2) is also called *metric inequality*. Only proximity measures that fulfill this criterion can be regarded as distance measures with a geometric interpretation, otherwise they are termed similarity or dissimilarity measures. Gower and Legendre (1986) argue that only property (3) and (4) are required for a metric; other properties can be derived from these two (Jain and Dubes 1988: 15). The criteria are important mathematical properties. Many researchers argue against the use of proximity indices that are not metrics (e.g. Jardine and Sibson 1971; Clifford and Stephenson 1975). A number of correlation measures will not fulfill all criteria. For example, the popular Pearson product-moment correlation coefficient fails to meet the third criterion and in some applications the second as well. However, it was argued by Tversky (1977) that a geometric interpretation of similarity, such as a metric distance, bears little relation to perceptual similarity. Human perception frequently violates the metric properties of symmetry and triangular inequality.

4.3.2 Correlation Coefficients

Some researchers, like Strauss et al. (1973) or Cliff et al. (1995), suggest the use of the correlation between the d -dimensional observations, or *profiles*, of two objects i and j as a measure of similarity. According to Cronbach (1953) the similarity between profiles can be characterized by its *shape* (pattern of highs and lows across all variable values), its *scatter* (dispersion of the scores around their average) and its *elevation* or *size* (mean score of the case over all variables). Unfortunately, correlation coefficients are only sensitive to the shape of the profiles compared (Aldenderfer and Blashfield 1984: 23). Thus, two profiles are regarded as identical that can in fact differ by scatter and elevation as only part of the available information is used. Beside this, there are other reasons why there is some dispute about the use of correlation coefficients as proximity measures (e.g. Jardine and Sibson 1971). Often correlation coefficients fail to satisfy the triangular inequality (Aldenderfer and Blashfield 1984: 23). Additionally, it is claimed that it makes no statistical sense to calculate the correlation of objects instead of variables for which correlation coefficients originally have been designed. The meaning of the mean value across the variables of two objects is far from clear. However, correlation coefficients are among the most frequently used proximity measures in the social sciences (Aldenderfer and Blashfield 1984: 22) and they outperform other measures when only shape effects are relevant for the classification of objects (see e.g. Hamer and Cunningham 1981). Correlation coefficients belong to the category of proximity measures for continuous data. An overview of the most commonly used correlation coefficients is given below in Table 4-1:

Table 4-1: Correlation coefficients. Overview

Measure	Formula
Pearson correlation coefficient (product-moment correlation coefficient)	$r(i, j) = \frac{\sum_{f=1}^d (x_{if} - m_i)(x_{jf} - m_j)}{\sqrt{\sum_{f=1}^d (x_{if} - m_i)^2} \sqrt{\sum_{f=1}^d (x_{jf} - m_j)^2}}$
Angular separation (Cosine similarity/Salton's cosine)	$r(i, j) = \frac{\sum_{f=1}^d x_{if} x_{jf}}{\sqrt{\sum_{f=1}^d x_{if}^2} \sqrt{\sum_{f=1}^d x_{jf}^2}}$
Spearman rank correlation coefficient	$r(i, j) = 1 - \frac{6 \sum_{f=1}^d (\text{rank}(x_{if}) * \text{rank}(x_{jf}))^2}{n(n^2 - 1)}$
Kendall's τ rank correlation coefficient	$r(i, j) = \tau = \frac{S_+ - S_-}{n(n - 1)}$
Goodman-Kruskall γ rank correlation coefficient	$r(i, j) = \gamma = \frac{S_+ - S_-}{S_+ + S_-}$

 S_+ : number of concordant states S_- : number of discordant states

In statistics the *Pearson product-moment correlation coefficient* named after Karl Pearson is a common measure of the correlation between two variables in a sample (see Pearson 1901). It reflects the strength and direction of a linear relationship between them and therefore it is also known as *linear correlation coefficient* (Jain and Dubes 1988: 16). The Pearson correlation normalizes the values of the vectors to their arithmetic mean (Egghe and Leydesdorff 2009). As it is calculated from standardized values it allows comparison of observations from different normal distributions. It ranges from +1 (perfect positive linear relationship) to -1 (perfect negative linear relationship). A correlation of 0 means there is no linear relationship between the two variables. Applied to cluster analysis it groups together objects with similar behavior recorded as variables. The Pearson correlation coefficient belongs to the parametric correlation coefficients assuming normal distributions (Kaufman and Rousseeuw 1990: 17). For non-linear relationships it may not adequately describe similarity. Another drawback is that it is sensitive to noise. The *angular separation* or *Salton's cosine similarity* is a correlation coefficient related to the Pearson correlation coefficient but measures the angular separation of two normalized data vectors measured from zero (Salton and McGill 1984). The cosine similarity can be seen as a method of normalizing document length during comparison. This similarity ranges from -1 meaning exactly opposite, to +1 meaning exactly the same, with 0 usually indicating independence.

Salton (1975) suggests both correlation measures for comparing document similarities in the vector space. For text matching, the data vectors are usually the term frequency vectors of the documents (Egghe and Leydesdorff 2009). As term frequencies cannot be negative the correlation of two documents will range from 0 to 1. Salton's cosine and Pearson correlation have been explained in geometrical terms by Jones and Furnas (1987) and compared to other proximity measures like Jaccard and Dice (see section 4.3.4). In geometrical terms, using the Pearson correlation means that the origin of the vector space is located in the middle of the set, while the cosine constructs the vector space from an origin where all vectors have a value of zero. Leydesdorff (2005) discuss the benefits of cosine versus Pearson correlation and Jaccard index for SNA on co-citation networks and text mining. As a result the differences are

marginal in practice because the Pearson correlation measure can also be considered as a cosine between normalized vectors (see also Jones and Furnas 1987)

Pearson correlation and cosine similarity require ratio-scaled data. If the data is at least on ordinal scale rank correlation coefficients can be applied (Jain and Dubes 1988: 16). Each variable is ranked separately by putting the values of the variable in order and numbering them with increasing ranks. The calculation is then carried out on the ranks of the data. There are a number of rank correlation coefficients that are calculated on the matches (concordance), or mismatch (discordance), of the ranks of two data sequences (Jain and Dubes 1988: 153). Ties can exist in one or both of the sequences.

The *Spearman rank correlation coefficient* named after Charles Spearman measures the correlation between two sequences of values. It is used as a measure of monotone relationship between two sets of ranked data. That is, it measures how tightly the ranked data clusters around a straight line (Kaufman and Rousseeuw 1990: 18). Applied to cluster analysis it groups together objects whose expression profiles (variables) have similar shapes or show similar general trends (e.g. increasing expression with time) but whose expression levels may be very different. It ranges from -1 (perfect negative relationship) to $+1$ (perfect positive relationship). The Spearman correlation coefficient belongs to the non-parametric correlation coefficients (Kaufman and Rousseeuw 1990: 18).

Kendall tau rank correlation coefficient by Maurice George Kendall, also called *Kendall's τ* , is another non-parametric measure of the correspondence of two objects in their ranking (Kendall 1938). It measures the degree of association between the two sequences normalized by the total number of pairs. If there is a large number of ties the total number of pairs should be adjusted accordingly. Kendall's τ ranges from $+1$ (the two rankings are the same) to -1 (one ranking is the reverse of the other). Values near 1 suggest that one sequence increases when the other one does. Values near 0 indicate no relation between the patterns of increase and decrease in the sequence. In contrast to Spearman correlation this coefficient does not assume that subsequent ranks indicate equi-distant positions on the variable measured.

Goodman-Kruskal gamma rank correlation coefficient by Leo A. Goodman and William Henry Kruskal, also called *Goodman-Kruskal γ* , measures the degree of association between the two sequences in terms of the number of concordant and discordant pairs (Goodman and Kruskal 1954). It varies from Kendall's τ only by the denominator. The γ statistic is invariant of monotone transformations of either sequence (Jain and Dubes 1988: 154). The actual distribution of γ depends on the null hypothesis, the sample size and the number of ties.

Hubert (1974) suggests using γ to measure correspondence between rank order proximity matrix and the cophenetic matrix. The idea behind this statistic is attributed to Mantel (1967). The basic idea behind *Hubert's Γ* in this context is to compare two $n \times n$ proximity matrices on the same n objects which have no in-built or implied relationships (Hubert 1974; Jain and Dubes 1988). In normalized form, Γ is the product-moment correlation coefficient between entries of the two matrices ranging from -1 to $+1$. The Γ statistic measures the degree of linear correspondence between the entries in both matrices. Unusually large values suggest that the two matrices agree with each other.

The values of a correlation coefficient range from -1 (strong negative relationship) to $+1$ (strong positive relationship). There are three possibilities to convert a correlation coefficients $r(f, g)$ into a dissimilarities $d(f, g)$ ranging from 0 to 1 (Kaufman and Rousseeuw 1990: 19). The first way is setting

$$d(f, g) = (1 - r(f, g))/2.$$

This formula assigns dissimilarities close to zero to variables with high positive correlation and high dissimilarities to variables with a strongly negative correlation. Another formula is

$$d(f, g) = 1 - |r(f, g)|$$

which uses only the absolute correlation values and thus assigns small dissimilarities to strongly positive as well as strongly negative correlations. A similar effect can be obtained by a third formula:

$$d(f, g) = 1 - r(f, g)^2.$$

Here, strong correlations are getting more weight than weak correlations. With this formula objects tend to be dissimilar rather than similar. Comparing these three possibilities on real data Lance and Williams (1979) concluded that the first formula outperformed the second, although the latter still did relatively well. The third possibilities could not provide satisfactory results.

4.3.3 Distance Measures

The second category of proximity measures by Sneath and Sokal (1973) involves similarity measures that satisfy the criterion of triangular inequality and are thus termed distance measures or distance metrics (see section 4.3.1). An overview of the most commonly used distances measures is given in Table 4-2. They all are applied to continuous data.

The *Euclidean Distance* is by far the most popular distance measure, especially in engineering work, and thus also termed *standard distance* (Jain and Dubes 1988: 15). It calculates the shortest distance between the variables of objects which is called the Pythagorean distance (Gower 1971). It can be interpreted as physical distances between two data points in a d -dimensional space (Everitt et al. 2001: 42).

The *city block distance* is also known as *Manhattan distance* (Larson and Sardiq 1983), *taxicab distance* (Krause 1975) or *rectilinear distance* (Brandeau and Chiu 1988) because it measures distances as travelling in a street configuration. When all features are binary, this metric is also known as *Hamming distance* (Jain and Dubes 1988: 15).

The *Chebyshev distance*, or *maximum value distance*, is named after Pafnuty Chebyshev and calculates the distance between two objects represented as data points by the maximum distance between the points in any single dimension (Jain and Dubes 1988: 15). It is also known as *chessboard distance* since in the game of chess the minimum number of moves a king needs to go from any square on a chessboard to some other square is proportional to the Chebyshev distance between those squares in two dimensions. This distance may be appropriate if the difference between attributes is reflected more by differences in individual variables rather than all variables considered together. It is very sensitive to outliers.

Minkowski distance of order r or L_r norm is the generalized form of the distance measures based on the difference of the corresponding attributes (Jain and Dubes 1988: 14). Both the city block distance with $r = 1$ and the Euclidean distance with $r = 2$ are special cases of Minkowski distance and thus formally termed L_2 and L_1 norm respectively. As r increases, the metric tends towards the Chebyshev distance with $r \rightarrow \infty$. Therefore, by increasing r a numerical value is placed on the largest distance between variables. A disadvantage of the Minkowski method is that if one element in the vectors has a wider range than the other elements this range may dilute the distances of the small-range elements (Aldenderfer and Blashfield 1984: 26).

Table 4-2: Distance measures. Overview

Measure	Formula
Euclidean distance (standard distance)	$d(i, j) = \sqrt{\sum_{f=1}^d (x_{if} - x_{jf})^2}$
Manhattan distance (city block/taxicab/rectilinear distance)	$d(i, j) = \sum_{f=1}^d x_{if} - x_{jf} $
Chebyshev distance (maximum value/chessboard distance/ sub distance)	$d(i, j) = \max_f x_{if} - x_{jf} $
Minkowski distance	$d(i, j) = \sqrt[r]{\sum_{f=1}^d (x_{if} - x_{jf})^r} \quad (r \geq 1)$
Canberra distance	$d_f(i, j) = \begin{cases} 0, & x_{if} = x_{jf} = 0 \\ \frac{ x_{if} - x_{jf} }{ x_{if} + x_{jf} }, & x_{if} \neq 0 \text{ or } x_{jf} \neq 0 \end{cases}$
Mahalanobis distance	$d(i, j) = (X_i - X_j)' \Sigma^{-1} (X_i - X_j)$

The *Canberra distance* sums up a series of ratios between usually non-negative variables of a pair of objects (Lance and Williams 1966). It accounts for the distance between two points as well as their relation to the origin. Each term of fraction difference ranges between 0 and 1. If one coordinate is zero the term becomes unity thus the distance will not be affected. This distance is very sensitive to small changes when both coordinate are near to zero (Everitt et al. 2001: 41).

Another distance measure used in cluster analysis is the *Mahalanobis distance* which has been introduced by Prasanta Chandra Mahalanobis (see Mahalanobis 1936). It is a dissimilarity measure between two objects where Σ is the pooled within-groups variance-covariance matrix and X_i and X_j are feature vectors for objects i and j . Unlike the Minkowski distances this measure incorporates the correlation between features and standardizes each feature to zero mean and unit variance. If the correlation between variables is zero the Mahalanobis distance is equivalent to squared Euclidean distance (Aldenderfer and Blashfield 1984: 24).

4.3.4 Association Coefficients

Proximity measures for objects measured in binary or nominal variables are called *association coefficients* (Aldenderfer and Blashfield 1984: 28). The outcomes of comparing the binary values for two individuals i and j can be represented in an 2×2 association table (see Table 4-3).

Table 4-3: Association table for comparison of binary data

		Individual i		
		Outcome	1	0
Individual j	1	a	b	a+b
	0	c	d	c+d
	Total	a+c	b+d	a+b+c+d

One major distinction between association coefficient can be made regarding how negative matches are treated (Aldenderfer and Blashfield 1984: 29; Kaufman and Rousseeuw 1990: 27). Invariant associated association coefficients work on symmetric binary data where all types of outcome carry equal information. Asymmetric binary variables do not treat all outcomes as equally important. The most important outcome is then coded by 1 and the less important outcome by 0. The agreement of two 1s (positive match) is considered more significant than the agreement of two 0s (negative match). Therefore, association coefficients on asymmetric binary data do not take negative matches into account. They are also called non-invariant measures. An overview of invariant as well as non-invariant association coefficients is given in Table 4-4.

Table 4-4: Association coefficients. Overview

Measure	Formula	Type
Simple matching coefficient	$s(i, j) = \frac{a + d}{a + b + c + d}$	Invariant
Jaccard coefficient Jaccard (1908)	$s(i, j) = \frac{a}{a + b + c}$	non-invariant
Rogers and Tanimoto (1960)	$s(i, j) = \frac{a + d}{a + 2(b + c) + d}$	Invariant
Sokal and Sneath (1963)	$s(i, j) = \frac{a}{a + 2(b + c)}$	non-invariant
Gower and Legendre (1986)	$s(i, j) = \frac{2(a + d)}{2(a + d) + (b + c)}$	Invariant
Dice coefficient Dice (1945); Sorensen (1948)	$s(i, j) = \frac{2a}{2a + b + c}$	non-invariant

Most of the numerous association coefficients were first invented in biological systematics (Aldenderfer and Blashfield 1984: 28). Among these only a few have been thoroughly tested and not rejected because of questionable features (see Sneath and Sokal 1973; Clifford and Stephenson 1975; Everitt 1980). However, according to Aldenderfer and Blashfield (1984: 29) two association coefficients are extensively used: the *simple matching coefficient* and its non-variant version, the *Jaccard coefficient* (Jaccard 1901). The simple matching coefficient measures the percentage of matches between two objects. It is sometimes called *M-coefficient*

or *affinity index*. The *Dice coefficient* is used by Smadja (1993) for the extraction of collocations from text corpora. Dias et al. (1999) introduce an extension of the Dice coefficient to larger n -grams under the name *mutual expectation*. Both Jaccard coefficient as well as Dice coefficient are frequently used in information retrieval and text mining.

Binary data can also be treated as interval-scaled such that distance metrics like the Euclidean distance can be applied (Kaufman and Rousseeuw 1990: 22). Using nominal variables there are more than two states the variable can adopt (Kaufman and Rousseeuw 1990: 28). M different outcomes are then coded as states $1, 2, \dots, M$. A simple matching coefficient on nominal data is given by

$$s(i, j) = \frac{u}{p}$$

where u denotes the number of matches and p the total number of variables. Corresponding to the formula in Table 4-4 more weight can be assigned to matches than to no-matches (Rogers and Tanimoto 1960; Sokal and Sneath 1963). Additionally, matches in variables with a large number of states can get more weights (Hyvärinen 1962) as well as matches corresponding to rare states (Lingoes 1967). Collapsing the states until only two of them remain converts nominally scaled variables into binary variables (Kaufman and Rousseeuw 1990: 28).

Association coefficients can be converted into dissimilarities by setting (Kaufman and Rousseeuw 1990: 25):

$$d(i, j) = 1 - s(i, j)$$

One major drawback of association coefficients is that they produce only a few distinct values and thus a proximity matrix based on one of these measures will contain a significant number of ties. Some clustering procedures perform poorly with many ties present in the proximity matrix (Aldenderfer and Blashfield 1984: 31).

4.3.5 Ties in the data

Tied proximities are subject to hierarchical as well as partitioning clustering procedures. Using a hierarchical clustering algorithm two new clusters are never formed at the same level and the algorithms produce unique dendrograms if there are no ties in the proximity matrix. A tie implies that two or more edges are added to the proximity graph at once and that the minimum and maximum functions required in matrix updating are not unique (Jain and Dubes 1988: 76). Besides choosing randomly between two candidate clusters a simple solution is to join the pair of objects or clusters which are entered first in the proximity matrix (Kaufman and Rousseeuw 1990: 109). A slightly more elaborate solution with a similar effect is to (randomly) add and subtract small amounts from the tied proximities in such a way that the ranks between the untied proximities stay unchanged (Jain and Dubes 1988: 76). In general, hierarchical structures can change dramatically with small changes in the rank orders of the proximities (Jain and Dubes 1988: 78). Nevertheless clustering procedures implemented in software packages do not typically check for ties and generate only one clustering, even though a number of clusterings might be equally justifiable. Jardine and Sibson (1971) show that the single-linkage method does not suffer from ambiguities due to ties because it has a continuity property. As long as the ranked order of the proximities are not changed by the added amounts this procedure of treating ties in proximities applies to all single linkage

algorithms. By contrast, several complete linkage dendrograms can be obtained by breaking ties in this way. However, according to Jain and Dubes (1988: 78) comparative studies suggest that the complete linkage method outperforms the single linkage method in generating useful hierarchies, even though proximity ties make it ambiguous.

4.4 Cluster Strategy: Hierarchical Clustering

All *hierarchical clustering algorithms* produce a nested sequence of clusters which can be pictured on a dendrogram. Partitions obtained by this type of cluster algorithm cannot be resolved in a later partition and clusters cannot overlap (Jain and Dubes 1988: 63). According to the taxonomy classification types proposed in chapter 4.1.2 hierarchical methods can be further classified due to the direction of the hierarchy. While agglomerative clustering methods start on the bottom of the hierarchy with each object assigned to a cluster of its own and then merge these singleton clusters (see section 4.4.1), divisive clustering methods start from the opposite direction with all objects in one cluster and then split this cluster (see section 4.4.2). This chapter discusses both types of hierarchical clustering but the major focus is on agglomerative methods. It concludes with an overview of desirable mathematical properties of hierarchical clustering methods (see section 4.4.3). A fundamental problem with agglomerative as well as divisive hierarchical clustering procedures is to decide which partition is best. This question is explored in detail in chapter 4.6 together with other cluster validation techniques.

4.4.1 Agglomerative Hierarchical Clustering

According to Everitt (2001: 56) agglomerative clustering procedures are the most widely used hierarchical clustering methods. Starting with all objects as singleton clusters they produce a nested series of clusterings merging the two most similar clusters at each level. At the last level all objects are grouped together in a single cluster. The wide range of agglomerative hierarchical clustering methods can be distinguished by their linkage rule which determines how the proximity between objects and clusters is calculated. Figure 4-9 illustrates the calculation of the proximity between two clusters using single, complete, average and complete linkage.

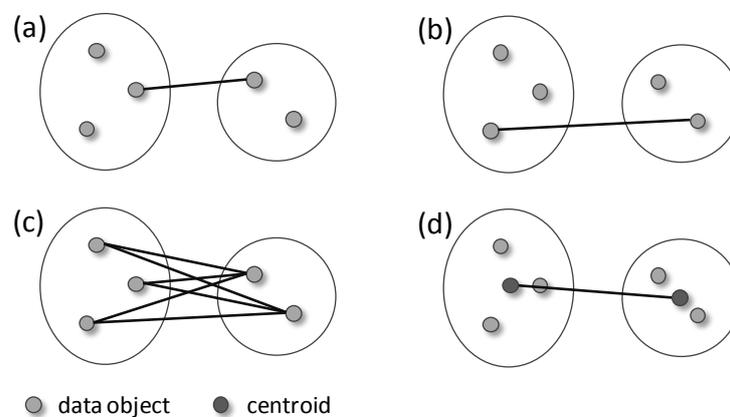


Figure 4-9: Illustration of hierarchical agglomerative clustering methods: (a) single linkage; (b) complete linkage; (c) average linkage; (d) centroid linkage

In this section the most commonly used linkage methods are presented (see sections 4.4.1.1 to 4.4.1.7): single linkage, complete linkage, weighted and unweighted group average linkage,

centroid and median linkage as well as Ward's method. An overview of these methods is given in Table 4-5 in section 4.4.1.6. Afterwards, the general algorithm for the updating procedure at each level is explained and applied to all seven linkage methods 4.4.1.7.

4.4.1.1 Single Linkage

The *single linkage* method by Sneath and Sokal (1973) calculates the similarity between two clusters C_r and C_s , with $r \neq s$, by the similarity of the two objects i and j – each from every cluster – which are most similar. At each level, two clusters will join if the shortest distance between such pair of objects is minimal for all clusters, i.e. $i \in C_p$ and $j \in C_q$ (see Figure 4-9 a)). If the proximities are dissimilarities $d(i, j)$ the linkage rule can be formalized as

$$\min_{i \in C_p, j \in C_q} \{d(i, j)\} = \min_{r \neq s} \{ \min_{i \in C_r, j \in C_s} \{d(i, j)\} \} \quad (\text{Jain and Dubes 1988: 65})$$

This method is also called *nearest neighbor method* (Everitt et al. 2001: 62) or *minimum method* (Johnson 1967) but this terminology can be confusing if similarities are used as proximities (Jain and Dubes 1988: 65).

Single linkage clustering has some desirable mathematical properties (Jardine and Sibson 1971; Aldenderfer and Blashfield 1984: 38; Jain and Dubes 1988: 65): invariance under monotonic transformation and unaffectedness of ties in the data. The first property means that the relative ordering of the proximity matrix is unaffected by data transformation. Almost all other hierarchical clustering algorithms miss this property. The major drawback of this method is the shape of clusters obtained. Single linkage tends to find long, unbalanced and straggly clusters (Aldenderfer and Blashfield 1984: 39; Everitt et al. 2001: 61). This typical effect of single linkage clustering is called “chaining” (Jain and Dubes 1988). Besides the shape of the clusters there are some more objections against this method (e.g. Wishart 1969; Hubert 1974; Jain and Dubes 1988: 65). According to Hansen and DeLattre (1978) single linkage clusters will not only chain but also have little homogeneity. In graph theory, single linkage clusters is defined in terms of maximally connected subgraphs and are characterized by minimum path length among all pairs of objects in the cluster (Jain and Dubes 1988: 62).

4.4.1.2 Complete Linkage

The *complete linkage* method by Sorensen (1948) calculates the similarity between two clusters C_r and C_s , with $r \neq s$, by the similarity of the two objects i and j – each from every cluster – which are least similar. At each level, two clusters will join if the largest distance between such pair of objects is minimal for all clusters, i.e. $i \in C_p$ and $j \in C_q$ (see Figure 4-9 a)). If the proximities are dissimilarities $d(i, j)$ the linkage rule can be formalized as

$$\max_{i \in C_p, j \in C_q} \{d(i, j)\} = \min_{r \neq s} \{ \max_{i \in C_r, j \in C_s} \{d(i, j)\} \}. \quad (\text{Jain and Dubes 1988: 65})$$

The complete linkage method is also called *furthest neighbor method* (Everitt et al. 2001: 60) or *maximum method* (Johnson 1967) but this terminology can be confusing if similarities are used as proximities (Jain and Dubes 1988: 65).

This method is the logical opposite of single linkage clustering (Aldenderfer and Blashfield 1984: 40) which becomes obvious in many contrasting properties of the two methods. Complete linkage clusters are regarded as *conservative* as all pairs of objects have to be related before a cluster can be formed (Jain and Dubes 1988: 65). Clusters obtained by

complete linkage clustering are relatively compact, of spherical shape, and contain highly similar objects with equal distances (Aldenderfer and Blashfield 1984: 40; Everitt et al. 2001: 61). But according to Hansen and DeLattre (1978) complete linkage clusters may not be well separated. Additionally no account of the cluster structure is taken (Everitt et al. 2001: 62). In graph theory, complete linkage clustering is defined in terms of maximally complete subgraphs, or cliques (Jain and Dubes 1988: 62). As the diameter of a complete subgraph is the largest proximity among all pairs of objects in the subgraph, the complete linkage method is also known as the *diameter method* (Jain and Dubes 1988: 65). The level of clustering at which a cluster is formed corresponds with the diameter of the cluster. However, this method does not generate clusters with minimal diameter.

4.4.1.3 Group Average Linkage

The (*group*) *average linkage* clustering method was invented by Sokal and Michener (1958) as an intermediate between the extremes of single and complete linkage clustering methods (Aldenderfer and Blashfield 1984: 40; Everitt et al. 2001: 62). Using the terminology by Rohlf (1970) and Sneath and Sokal (1973) this method is also called the *unweighted pair-group method using the average approach (UPGMA)*. This method calculates the average of the similarities between an object under consideration and all other objects in a cluster (see Figure 4-9 c)). Thus, the entire cluster structure is taken into account and not only the minimum and maximum distance. The object is joined to the cluster with maximum average similarity or minimum average dissimilarities respectively. There are several variants of this method. For example, the arithmetic average of proximities between objects as well as objects and cluster representatives can be calculated (Aldenderfer and Blashfield 1984: 41). According to Everitt et al. (1980: 62) this method is relatively robust against outliers and noise in the data. Hierarchical clustering methods using (*group*) average linkage tend to find roughly ball-shaped clusters (Kaufman and Rousseeuw 1990: 47) and were first extensively used in biological science and later became also popular in social sciences (Aldenderfer and Blashfield 1984: 41).

There is a variant of this method called *weighted average linkage* or *weighted pair-group method using the average approach (WPGMA)* by McQuitty (1966) that weights inter-cluster distances according to the inverse of the number of objects in each cluster. This approach tries to avoid small clusters getting dominated by more numerous clusters after being merged.

4.4.1.4 Centroid Linkage & Median Linkage

The *centroid linkage* method was first invented by Sokal and Michener (1958). As illustrated in Figure 4-9 d) this method uses a data matrix rather than a proximity matrix and merges clusters with the most similar mean vectors (centroids). Merging two clusters the more numerous of the two will dominate the merged cluster (Everitt et al. 2001: 60). The commonly used proximity measure for centroid linkage methods is the Euclidean distance. Other proximity measures are possible as well but they would lack an interpretation in terms of the raw data (Anderberg 1973).

There is a variant of this method called *median linkage* by Gower (1967) that tries to avoid small clusters getting dominated by more numerous clusters after being merged. It therefore weights inter-cluster distances according to the inverse of the number of objects in each

cluster. Using this method new group intermediates are positioned between merged groups (Everitt et al. 2001: 62). Like centroid linkage methods, median linkage methods should only be used when the objects are represented as objects in a multidimensional metric space and the Euclidean distance is used as distance measure (Jain and Dubes 1988: 80; Everitt et al. 2001: 62).

The centroid method is also known as the *unweighted pair-group method using the centroid approach (UPGMC)*, while the median method also called *unweighted pair-group method using the centroid approach (WPGMC)* (Lance and Williams 1967).

4.4.1.5 Ward's method

Ward's method is designed for optimization of the minimum variance within clusters (Ward 1963). Therefore, this method is also called the minimum variance method or minimum sum of squares method as it is based on the square-error criterion often used in statistical procedures, especially for analysis of variance (Jain and Dubes 1988: 82). Ward's method is design for data objects that appear as patterns in a pattern space and thus has a geometric interpretation (Everitt et al. 2001: 62). Suppose the n objects are patterns in a d -dimensional metric space. The patterns have been partitioned into K non-overlapping clusters $\{C_1, C_2, \dots, C_K\}$ where cluster C_k has n_k objects. Let $x_{ij}^{(k)}$ be the value of or variable, j of or object i in cluster k . The center of a cluster, $m^{(k)} = (m_1^{(k)}, m_2^{(k)}, \dots, m_d^{(k)})$, is defined as the centroid. The f th coordinate of the centroid can then be calculated by

$$m_f^{(k)} = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{if}^{(k)}.$$

The square-error of this cluster is the sum of the squared Euclidean distances between each object in the cluster and the centroid of the cluster:

$$e_k^2 = \sum_{i=1}^{n_k} \sum_{j=1}^d (x_{ij}^{(k)} - m_j^{(k)})^2$$

The square-error for the entire clustering is the sum of the within-cluster variations:

$$E_K^2 = \sum_{k=1}^K e_k^2$$

At each clustering level the algorithm merges the two clusters that minimize the error sum of squares of the entire data set for a fixed number of clusters. The resulting partition is also called *minimum variance partition*. The error represents the deviations of the cluster members from their centroids (Jain and Dubes 1988: 93). Jain and Dubes (1988: 93) describe the partition obtained as a collection of K "spherically shaped swarms". The square-error criterion function tries to make these swarms as compact and separated as possible. Minimizing square-error, or within-cluster variation, can be shown to be equivalent to maximizing the between-cluster variation (Dubes and Jain 1980: 92). Using the square-error criterion Ward's methods tends to find same size, spherical clusters. Unfortunately it is quite sensitive to outliers (Everitt et al. 2001 62). Although the square-error used in Ward's method is a familiar criterion in engineering it might not be appropriate to impose such a priori criterion as the objective of cluster analysis is to investigate the structure of data. Additionally, the limitation to data objects as patterns can hinder its usage. Nevertheless there are several comparative

studies that conclude that Ward's clustering method outperforms other hierarchical clustering methods (Jain and Dubes 1988: 82).

4.4.1.6 Comparison of Agglomerative Hierarchical Clustering Methods

Applying hierarchical clustering methods to the same data one will get very different hierarchies and often there is no single best clustering method. One way of comparing agglomerative hierarchical clustering methods is to examine in what way these methods transform the relationships between the points in the multivariate space (Aldenderfer and Blashfield 1984: 45). Here, space-contracting, space-dilating, and space-conserving methods are distinguished. With *space-contracting* methods like single linkage, new data points are assigned to existing clusters rather than forming new clusters and thus reducing the space between the clusters. In contrast, *space-dilating* methods like complete linkage and Ward's method tend to create new clusters of hyperspherical form and roughly the same size. Thus, more distinct clusters are formed that appear to recede on formation. *Space-conserving* methods like average linkage preserve the original space. Some authors argue against space-contracting methods (e.g. Williams et al. 1971) whereas others emphasize their desirable mathematic properties (e.g. Jardine and Sibson 1968). A review of agglomerative hierarchical clustering methods can be found by Milligan (1981). Several empirical studies recommend Ward's method for clusters with equal numbers of objects; otherwise group average and complete clustering (Cunningham and Ogilvie 1972; Blashfield 1976) or centroid clustering (Hands and Everitt 1987) will be suitable. Hubert (1974) and Hartigan (1975) focus on the effect of outliers and could show that single linkage is much more affected by the presence of outliers than complete linkage. Hubert (1974) recommends complete linkage as superior over single linkage in practical situations with ordinal proximities.

Duflou and Maenhout (1990) reject centroid and median linkage because of reversals when studying the chemical concentrations in brains. They prefer Ward's method and complete linkage. In archaeology empirical studies generally favor Ward's method and average linkage, but Ward's method will impose spherical structures where none exists (Everitt et al. 2001: 67). The popularity and success of a clustering method depends on the application area, the problem and the data as well as the a priori expected type of clusters, but also on the availability of software and the research tradition (Aldenderfer and Blashfield 1984: 45; Everitt et al. 2001: 67).

Table 4-5 gives a brief overview of the seven agglomerative hierarchical clustering methods presented in this chapter. The acronym "PGM" refers to the "pair group method"; the prefixes "U" and "W" refer to unweighted and weighted, respectively (Jain and Dubes 1988: 79). Generally an "unweighted" method treats each object in a cluster equally. In contrast, a "weighted" method weights all clusters the same. Thus, objects in small cluster are weighted more heavily than objects in large clusters. The suffices "A" and "C" refer to which type of cluster representatives is used: either "arithmetic averages" or "centroids". "UPGMA", for example, stands for "unweighted pair group method using arithmetic averages" (Jain and Dubes 1988: 80). This terminology has been introduced by Rohlf (1970) and Sneath and Sokal (1973).

Table 4-5: Standard agglomerative hierarchical clustering methods. Overview. Following: Everitt (2001: 62)

Clustering Method	Alternative names*	Usually used with	Distance between clusters defined as	Remarks	Source
Single Linkage	Nearest neighbor/ minimum method	Similarity/Distance	Minimum distance between pair of objects belonging to different clusters	Tends to produce unbalanced and straggly clusters ("chaining"), especially in large data sets; takes no account of cluster structure. Requires proximity matrix without ties.	Sneath (1957)
Complete Linkage	Furthest neighbor/ maximum/diameter method	Similarity/Distance	Maximum distance between pair of objects belonging to different clusters	Tends to find compact clusters with equal distances (maximum distance between objects); takes no account of cluster structure. Requires proximity matrix without ties.	Sorensen (1948)
(Group) Average Linkage	UPGMA	Similarity/Distance	Average distance between pair of objects belonging to different clusters	Tends to join clusters with small variances; tends to find roughly ball-shaped clusters; intermediate between single and complete linkage; takes account of cluster structure; relatively robust.	Sokal and Michener (1958)
Weighted Average Linkage	WPGMA	Similarity/Distance	Weighted average distance between pair of objects belonging to different clusters	Similar to (group) average linkage but weights inter-cluster distances according to the inverse of the number of objects in each cluster.	McQuitty (1966)
Centroid Linkage	UPGMC	Distance (requires raw data)	Squared Euclidean distance between mean vectors (centroids)	Assumes points can be represented in Euclidean space for geometrical interpretation; the more numerous of two groups clustered dominates the merged cluster; subject to reversals.	Sokal and Michener (1958)
Median Linkage	WPGMC	Distance (requires raw data)	Squared Euclidean distance between weighted centroids	Assumes points can be represented in Euclidean space for geometrical interpretation; new group intermediate in position between merged groups; subject to reversals.	Gower (1967)
Ward's method	Minimum sum of squares/minimum variance method	Distance (requires raw data)	Increase in sum of squares within clusters, after fusion, summed over all variables	Assumes points can be represented in Euclidean space for geometrical interpretation; tends to find same size, spherical clusters; sensitive to outliers	Ward (1963)

*U: unweighted; W: weighted; PG: pair group; A: arithmetic average; C: centroid

Single linkage and complete linkage clustering can be regarded as the two extremes of the spectrum of agglomerative hierarchical clustering methods (Jain and Dubes 1988: 80). Arithmetic averaging (UPGMA and WPGMA) tries to avoid the extremes of the single linkage and complete linkage methods in measuring the proximities between an existing cluster and a prospective cluster. Single linkage and complete linkage clustering assume an ordinal proximity matrix with no ties (Jain and Dubes 1988: 61). Single linkage generally suffers from the chaining effect which joins clusters with noise objects between them. However, this effect can also be used to detect outliers as it leaves them as singletons if they are sufficiently far away (Gnanadesikan et al. 1977). As already mentioned above complete as well as average linkage clustering imposes spherical clusters on the data, regardless of the natural structure.

In contrast to the arithmetic averaging methods the centroid methods have direct geometric interpretations when the objects are represented as patterns in a multidimensional space (Jain and Dubes 1988: 80). The centroid methods assess the dissimilarity between two clusters as the distance between centroids. The UPMGC method measures distance in terms of the centroid computed from all patterns in each clusters, whereas the WPGMC method computes the centroids from the centroids of the two clusters that merge to form a new cluster. In contrast to the other methods presented in Table 4-7 the centroid methods do not fulfill the criterion of monotonicity (Jain and Dubes 1988: 80). That means, later clusters can merge on a smaller level, or dissimilarity, than earlier clusters.

Figure 4-10 illustrates the results obtained by the algorithms in Table 4-7 on the same data set.

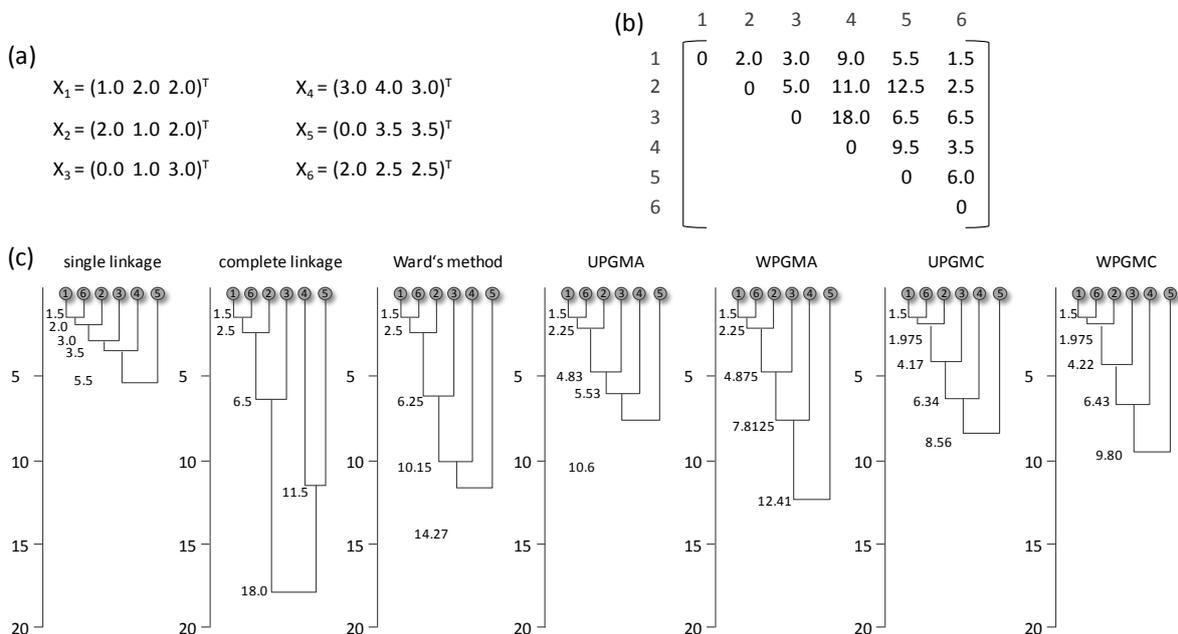


Figure 4-10: Comparison of different agglomerative clustering methods. (a) six three-dimensional pattern vectors; (b) proximity matrix using squared Euclidean distance; (c) dendrograms. Source: Jain and Dubes (1988: 82)

All dendrograms agree that (x_1, x_2, x_6) is a strong cluster. The results from the average and centroid clustering methods as well as from Ward's method suggest that pattern x_4 should be regarded as an outlier because it joins the cluster last and late with a significant gap between its own clustering level and the clustering level before. Using the single linkage method

pattern x_5 seems to be an outlier. Quantitative measures of the strength and quality of the clusters and partitions can help to decide which cutting level is the best and which cluster is indeed a good cluster (see section 4.6).

4.4.1.7 General Matrix Updating Algorithm

Comparing different agglomerative clustering methods there can be huge differences in the results obtained (see Figure 4-10 in section 4.4.1.6). However, the updating procedure is always the same and the differences in the calculation of the proximities between existing and prospective clusters can be generalized in one recurrence formula invented by Lance and Williams (1967) with concrete parameter values referring to the specific method (see Table 4-6 step 4)). An approach to matrix updating was suggested by King (1967) and formalized by Johnson (1967) for single linkage and complete linkage clustering. This algorithm uses the formula by Lance and Williams (1967) and can be generalized for all SAHN (sequential, agglomerative, hierarchical, non-overlapping) clustering methods (Jain and Dubes 1988: 79). The version of the algorithm presented in Table 4-6 requires a proximity matrix containing no ties (Jain and Dubes 1988: 72).

Table 4-6: General matrix updating algorithm. Based on: Johnson (1967) and Lance and Williams (1967)

General Matrix Updating Algorithm	
Prerequisites	Let n be the number of objects. Let $d(i, j)$ be the proximity between object i and j . Let $D = [d(i, j)]$ be a $n \times n$ proximity matrix calculated from all pairs of objects. Let $m = 0, 1, \dots, (n - 1)$ be the sequence number of a partition. Let $L(k)$ be the level of the k th partition. Let (m) denote a cluster with sequence number m . Let $d[(r), (s)]$ the proximity between clusters (r) and (s) .
Step 1	Begin with the disjoint partition having level $L(0) = 0$ and sequence number $m = 0$.
Step 2	Find the least dissimilar pair of clusters in the current partition according to $d[(r), (s)] = \min\{D\}$ (or find the most similar pair of clusters in the current partition according to $d[(r), (s)] = \max\{D\}$ if a similarity measure is used as proximity index.)
Step 3	Set $m \leftarrow m + 1$. Merge clusters (r) and (s) into a single cluster to form the next partition m . Set the level of this partition to $L(m) = d[(r), (s)]$
Step 4	Update the proximity matrix, $D = [d(i, j)]$, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r, s) and old cluster (k) is defined as follows. $d[(k), (r, s)] = \alpha_r d[(k), (r)] + \alpha_s d[(k), (s)] + \beta d[(r), (s)] + \gamma d[(k), (r)] - d[(k), (s)] $
Step 5	If all objects are in one cluster, stop. Else, go to step 2.

Step 4 of that algorithm specifies how the proximity matrix is to be updated by defining the formula for the proximity between a newly formed cluster, (r, s) and an existing cluster (k) with n_k objects. The recurrence formula in step 4 was first proposed by Lance and Williams (1967). It helps to define a flexible clustering scheme with various characteristics that (Everitt et al. 2001: 61). Table 4-7 shows the parameter values for the seven clustering algorithms discussed in the previous sections.

Table 4-7: Lance-Williams parameters for SAHN Matrix Updating Algorithms. Source: Jain and Dubes (1988: 80).

Clustering Method	α_r	α_s	β	γ
Single Linkage	1/2	1/2	0	-1/2
Complete Linkage	1/2	1/2	0	1/2
Group Average Linkage (UPGMA)	$\frac{n_r}{n_r + n_s}$	$\frac{n_s}{n_r + n_s}$	0	0
Weighted Average Linkage (WPGMA)	1/2	1/2	0	0
Centroid Linkage (UPGMC)	$\frac{n_r}{n_r + n_s}$	$\frac{n_s}{n_r + n_s}$	$\frac{-n_r n_s}{(n_r + n_s)^2}$	0
Median Linkage (WPGMC)	1/2	1/2	-1/4	0
Ward's method	$\frac{n_r + n_k}{n_r + n_s + n_k}$	$\frac{n_s + n_k}{n_r + n_s + n_k}$	$\frac{-n_k}{n_r + n_s + n_k}$	0

U: unweighted; W: weighted; PG: pair group; A: arithmetic average; C: centroid
 n_r : number of objects in cluster r n_s : number of objects in cluster s

4.4.2 Divisive Hierarchical Clustering

Divisive hierarchical clustering methods are the logical opposite to agglomerative hierarchical clustering methods (Aldenderfer and Blashfield 1984: 50). They start with all objects grouped together in one cluster and then successively split clusters that are least similar. Again, a hierarchy of nested partitions is obtained. An agglomerative clustering algorithm starts with a pairwise comparison of all n objects. This will lead towards $n(n-1)/2$ combinations. This number grows quadratically with n . In contrast, a divisive clustering algorithm will need to compute $2^{n-1} - 1$ different partitions of two non-empty subsets to select the first division into two clusters. This number grows exponentially. Thus, even for medium-sized data sets a complete enumeration is computationally prohibitive (Kaufman and Rousseeuw 1990: 254). Although divisive clustering methods are less commonly used than agglomerative methods they have the advantage that they start from a more natural point of view as most users will be interested in the main structure of the data, e.g. dividing the data in only a few clusters (Kaufman and Rousseeuw 1990; Everitt et al. 2001: 67).

A popular divisive clustering algorithm in SNA is the edge betweenness clustering method by Girvan and Newman (2002; 2004). It is based on the representation of the data as a graph. In contrast to the hierarchical as well as partitioning clustering method presented in this chapter it assigns objects, represented as nodes in the graph, to non-overlapping clusters with respect to their relationship among each other, represented as links between the nodes, instead of taking the objects attribute into account. The algorithm is explained in the last section of this chapter (see section 4.7.2) and the content-based clustering method proposed in this work will be compared to it.

While agglomerative hierarchical as well as partitioning clustering methods commonly use a polythetic clustering strategy, there are some divisive clustering strategies that are monothetic (see section 4.1.2). A monothetic clustering algorithm uses the recorded features one by one, whereas a polythetic clustering algorithm uses all features at once. Monothetic approaches usually work with binary data (Aldenderfer and Blashfield 1984: 50). In monothetic divisive clustering methods an optimization criterion is used to decide which cluster should be split.

The criterion reflects either cluster homogeneity or association with other variables. Most often the χ^2 -square statistic or an information statistic are used as divisive criteria for monothetic clustering procedures (Clifford and Stephenson 1975; Everitt 1980). An example for an information statistics on cluster homogeneity is the information content criterion by Lance and Williams (1968) which splits the cluster into subgroups with preferably identical attributes (Everitt et al. 2001: 67).

Monothetic clustering methods have some advantages in the classification of new members and the tolerance to missing values. Missing values are replaced by the value of the non-missing with the largest absolute association (Everitt et al. 2001: 68). Typical application areas of this monothetic clustering approach are medical and archaeological studies and especially for association analysis in ecology. In social sciences its use has been somewhat restricted to archaeological applications (Aldenderfer and Blashfield 1984: 50).

Using the association of a variable with all other attributes is also known as *association analysis* (Williams and Lambert 1959). Association analyses use χ^2 -criterion functions (Kaufman and Rousseeuw 1990: 304). The split at each clustering level is made according to the presence or absence of the attribute which is maximal associated with the others.

The polythetic divisive clustering procedure by MacNaughton-Smith et al. (1964) tries to avoid part of the computational effort by not considering all possible splits. The algorithm starts with all objects in one cluster and then selects the object which is least similar to all other objects as the seed for the “splinter” group. Objects are assigned to the splinter group if they are closer to the splinter group than to the original group. At the succeeding levels the cluster with the two least similar objects is split by the same rule.

4.4.3 Mathematical Properties of Hierarchical Clustering Methods

The mathematical property of *ultrametricity* was first introduced by Hartigan (1967), Jardine et al. (1967) and Johnson (1967). In cluster analysis, the *ultrametric property* is related to the ability of represent the hierarchy by a dendrogram. It is defined as

$$d(i, j) \leq \max\{d(i, k); d(j, k)\} \text{ for all objects } i, j, k.$$

That means, for any three objects the two largest distances are equal (Everitt et al. 2001: 74). Although this property may not hold for the elements of the proximity matrix it holds for the levels at which two objects are joined in the same cluster in many hierarchical clustering procedures. If this property is failed, *inversions* or *reversals* can occur: later clusters can merge on a smaller level than earlier clusters. Inversions can help to identify clustering levels with no clear structure (Gower 1990). Centroid linkage and median linkage both can produce reversals (Everitt et al. 2001: 74). A proximity matrix where all entries fulfill the ultrametric inequality is hard to find in practice as the matrix would need to contain large numbers of ties (Jain and Dubes 1988: 165).

The tendency of *space distortion* is another feature of clustering procedures related (Everitt et al. 2001: 74). There are three types of space distortion: space contraction, space dilation and space conserving. Here, *space contraction* refers to the effect of merging dissimilar clusters and thus moving the merged clusters closer to each other, like the chaining effect with single linkage clustering. *Space dilation* is the opposite type of space distortion and can be found in

complete linkage clusterings. It refers to the moving the merged clusters further from each other. The third type of distortion is *space conserving* which occurs with group average linkage clustering procedures. These methods average the distances to the clusters merged.

Fisher and van Ness (1971) have introduced a number of admissibility properties, which describe desirable properties of clustering methods and thus can help, to choose the appropriate hierarchical clustering method:

- (1) *Clump admissibility*. There exists a clustering such that all within-cluster distances are smaller than all between cluster distances.
- (2) *Convex admissibility*. If the objects can be represented in Euclidean space the convex hulls of partitions never intersect.
- (3) *Point proportional admissibility*. Replication of points does not alter the boundaries of partitions.
- (4) *Monotone admissibility*. Monotonic transformation of the elements of the proximity matrix does not alter the clustering.

The first property was originally termed (*k-group*) *well-structured admissibility*. It is related by Mirkin (1996) to the concepts of space conversion and ultrametricity and the recurrence parameters of Lance and Williams in Table 4-7 and termed clump admissibility. Although a desirable feature in cluster analysis, standard procedures like single and complete linkage do not fulfill the second property (Everitt et al. 2001: 76). Point proportionality is important with samples containing replicated observations, like in systems for automatic monitoring of keywords contained in web pages (see e.g. Kirriemuir and Willett 1995). Allowing the weighting of objects will reflect replication in the data. The monotone property should be fulfilled by the clustering procedure chosen when only rank-order information is reliable, like subjective ratings in market research. The four admissibility properties are given in Table 4-8 for the seven agglomerative hierarchical clustering methods discussed in chapter 4.4.1.

Table 4-8: Admissibility parameters for agglomerative hierarchical clustering methods. Source: Everitt (2001: 63).

Clustering Method	<i>U</i>	<i>C</i>	<i>P</i>	<i>M</i>
Single Linkage	No	No	Yes	Yes
Complete Linkage	No	No	Yes	Yes
Group Average Linkage	No	No	No	No
Weighted Average Linkage	No	No	Yes	No
Centroid Linkage	Yes	No	No	No
Median Linkage	Yes	No	Yes	No
Ward's method	No	Yes	No	No

U: ultrametric; C: convex; P: point proportional; M: monotone

4.5 Cluster Strategy: Partitioning Clustering

Partitioning clustering techniques, also known as *optimization clustering techniques*, try to recover natural groups present in the data by generating a single partition of a fixed number of clusters. They are used frequently in engineering applications when single partitions are important. These methods are well suited for the efficient representation and compression of

large data sets (Jain and Dubes 1988: 91). According to the taxonomy classification types proposed in chapter 4.1.2 partitioning clustering algorithms can either assign each object to a single non-overlapping cluster, or assign each object to several overlapping clusters according to some membership value. The first type of partitioning clustering procedures is also called *hard clustering*, whereas the second type is also known as *fuzzy clustering* (Kaufman and Rousseeuw 1990: 164). Discussing both types of partitioning clustering algorithms this chapter is organized as follows: first, the basic idea partitioning clustering procedures is presented together with its major advantages and disadvantages (see section 4.5.1). Afterwards, a general algorithm for non-overlapping partitioning clustering is introduced and its main parameters are discussed (see section 4.5.2). In the last section, the main objectives of fuzzy clustering algorithms are explained (see section 4.5.3).

4.5.1 General Objectives of Partitioning Clustering Procedures

To allow the optimizing of the clustering results overlapping, or non-exclusive, as well as non-overlapping, or exclusive, partitioning clustering algorithms are iterative procedures. In contrast to hierarchical clustering procedures partitioning clustering algorithms do not produce a nested series of partitions ranging from 1 to n clusters but iteratively optimize an initial partition of a fixed number of K clusters according to some statistical criterion (Aldenderfer and Blashfield 1984: 46). As a result, a data set containing n data objects is organized into an optimal partition assigning each object to one of K clusters with the objects in one cluster being more similar to each other than to any other object in a different cluster. If fuzzy clustering is used the objects have different membership values in different clusters and an objective function is optimized to gain an optimal distribution of cluster memberships (Zadeh 1965; Jain and Dubes 1988: 131). To decide whether an optimal partition has been found a clustering criterion must be adopted and optimized. Such criteria can be classified as global and local. A *global* criterion represents each cluster by a representative, like the centroid of the cluster, and assigns the objects to the cluster with the most similar representative (Kaufman and Rousseeuw 1990). A *local* criterion forms clusters by utilizing local structure in the data, e.g. by identifying high-density regions in the metric space or by assigning a pattern and its k nearest neighbors to the same cluster. Most often, the statistical criterion will be to reduce the square-error of the cluster assignments (Jain and Dubes 1988: 96). Generally, the square-error tends to decrease as the number of clusters increases and can be minimized only for a fixed number of clusters. Similarly to the wide range of hierarchical clustering procedures and proximity measures, different algorithms of the general method for iterative partitioning clustering, especially the use of different statistical criteria, will lead to different partitions when applied to the same data (Aldenderfer and Blashfield 1984: 47; Jain and Dubes 1988: 96).

In theory, the global optimum for a given number of clusters can be achieved by calculating all possible partitions and evaluating them with the clustering criterion. However, in practice this solution raises some difficulties. The choice of the appropriate criterion for the problem at hand is part of the clustering strategy. It should be suitable to recover the natural structure of the data as well as the researcher's notion of a "cluster" and therefore depends highly on the application area and the purpose of the analysis. Additionally, it should be simple to compute but at the same time complex enough to reflect various data structures. Moreover, even for a

data set of moderate size the possible number of partitions is extremely large. The number of different partitions of n objects into K clusters is given by

$$N(n, K) = \frac{1}{K!} \sum_{m=1}^K (-1)^{K-m} \binom{K}{m} m^n \text{ (Liu 1968).}$$

For example, assigning 15 objects to three clusters would mean to examine 217,945,728,000 unique partitions (Aldenderfer and Blashfield 1984: 46). Thus, evaluating a criterion for all possible partitions is an extremely time consuming task depending on the size of the data set. Therefore, heuristics are used to sample a small subset of all possible partitions hoping to find or at least approximate the optimal solution.

Apart from this limitation, iterative partitioning clustering procedures have some advantages (Aldenderfer and Blashfield 1984: 46). Unlike hierarchical clustering procedures they do not need to pre-compute and store proximity matrix but work directly upon the raw data. Thus, iterative methods can be applied to distinctly larger data sets than hierarchical methods. Additionally, they can compensate poor initial partitions at least to some extent as they make more than one pass through the data, which is one of the major drawbacks of hierarchical agglomerative algorithms. The result obtained from partitioning algorithms is one single, optimal partition instead of a hierarchy of partitions. Most iterative methods can only produce non-overlapping clusters. However, some algorithms like the fuzzy K -means algorithm by Dunn (1974) and Bezdek (1974) also generate overlapping clusters.

4.5.2 Hard Clustering

As there can only be one assignment of each object to a cluster non-overlapping partitioning clustering is also called *hard clustering* to distinguish it from overlapping partitioning, or fuzzy, clustering (Kaufman and Rousseeuw 1990: 164). According to Aldenderfer and Blashfield (1984: 46) non-overlapping iterative partitioning clustering methods can be characterized by (1) the choice of the initial partition, (2) the type of pass, and (3) the statistical criterion. The term pass, or cycle, is used for the iteration through the data.

Table 4-9: General algorithm for iterative partitioning clustering by Anderberg. Source: Jain and Dubes (1988: 96)

General algorithm for iterative partitioning clustering by Anderberg (1973)	
Prerequisites	Let K be the number of clusters.
Step 1	Select an initial partition with K clusters. Repeat steps 2 through 5 until the cluster memberships stabilize.
Step 2	Generate a new partition by assigning each pattern to its closest cluster center.
Step 3	Compute new cluster centers as the centroids of the clusters.
Step 4	Repeat steps 2 and 3 until an optimum value of the criterion function is found.
Step 5	Adjust the number of clusters by merging and splitting existing clusters or by removing small clusters or outlier clusters.

Table 4-9 provides a general algorithm for iterative partitioning clustering. This algorithm has been suggested by Anderberg (1973) who, as well as Dubes and Jain (1980), or Aldenderfer and Blashfield (1984: 47), has also discussed its parameters and options: the choice of the initial partition and the proximity measure, the updating procedure depending on the statistical criterion, the cluster representatives and the type of pass, the number of clusters, the

convergence criterion, and the computational complexity. These options are discussed in the following sections.

4.5.2.1 Initial Partition

The *initial partition* can be obtained either by selection of an appropriate starting partition or by definition of seed points (Aldenderfer and Blashfield 1984: 47; Jain and Dubes 1988: 97). In a starting partition all objects are already assigned to one of K clusters. The partition can either be randomly chosen or be pre-computed by some algorithm, like for example a partition obtained by cutting a hierarchical clustering tree. This starting partition is then refined and optimized by the partitioning clustering algorithm. The other option is to choose K objects as seed points and then compute a first pass through the data to assign the other objects to the closest seed point measured by some proximity index (Aldenderfer and Blashfield 1984: 47). Although seed points can be randomly chosen it is advisable to use K well-separated objects: One way is to take the centroid of the whole data and then selecting successive seed points which are at least a certain distance away from those already chosen (Jain and Dubes 1988: 97). Applications of the different methods can be found in McQueen (1967), Beale (1969b; 1969a), Thorndike (1953), McRae (1971), Friedman and Rubin (1967) and Blashfield (1976).

The choice of the initial partition is crucial for the clustering result (Friedman and Rubin 1967; Blashfield 1976). Different initial partitions can lead to different final clusterings as the algorithms can converge to local minima, especially if the clusters are not well-separated (Jain and Dubes 1988: 97). Running the partitioning algorithm with several different initial partitions can help to validate a final partition as the global optimum and detect the appropriate parameters of the algorithm for the actual data and the purpose of the analysis. A study about the connection of local and global optima can be found in Hartigan (1975). According to Marriott (1982) a slow convergence with many changes in group memberships indicates that a wrong number of clusters has been chosen as there is no clear evidence of clustering. Some algorithms, like the K -means type algorithms, are very sensitive to the choice of the initial partition. Several Monte Carlo studies have been performed on the influence of the seed parameters on the final solution. Blashfield and Aldenderfer (1978) and Milligan (1980) demonstrate that the major reason for obtaining a suboptimal solution is a poor starting partition. According to Blashfield and Aldenderfer (1978) a rational selection of the initial partition did have no significant influence on the result. However, Milligan (1980) use a K -means pass with a starting partition obtained from an average linkage clustering on a data set with known structure. Here, the results outperformed other partitioning and hierarchical clustering methods. Other studies suggest that an optimal solutions can be obtained regardless of the initial partition when the data is well-structured (Bayne et al. 1980; Everitt 1980). This again indicates that the performance of a cluster analysis strongly depends on the quality of the data and a preliminary evaluation of the clustering tendency (see section 4.6.6) can indicate if the data is suitable for cluster analysis or other types of analysis have to be considered. Additionally, it is advisable to use different initial partitions and compare the results (Everitt et al. 2001: 100).

4.5.2.2 Statistical Criterion

The next step after the selection of the initial partition is the updating of this partition according to some *statistical criterion* (Aldenderfer and Blashfield 1984: 46). Partitions are updated by reassigning patterns to clusters in an attempt to either explicitly or implicitly minimize a criterion function like the square-error criterion. In general, employing a different type of pass with the same criterion will lead to different final solutions (Aldenderfer and Blashfield 1984: 47).

The most common clustering criteria explicitly used on continuous data make use of the decomposition of the $p \times p$ covariance or dispersion matrix T of the variables (Everitt et al. 2001: 92). This dispersion matrix can be partitioned into the within-group dispersion matrix W the between-group dispersion matrix B with $T = W + B$. The minimization of $\text{trace}(W)$ ²¹ is then the minimization of the within-group sum-of-squares over all variables which is the same as minimizing the squared Euclidean distance between individuals and their cluster mean and can also be expressed as the minimization of the lack of compactness with $h_1(m)$ for $r = 2$ in Table 4-13, section 4.6.5.2 (Everitt et al. 2001: 93). Minimizing $\text{trace}(W)$ is equivalent to maximizing $\text{trace}(B)$. Ward (1963) invented this criterion to optimize the hierarchical building process. It is explicitly suggested by Singleton and Kautz (1965) and implicitly used by the clustering methods by Forgy (1965), Edwards and Cavalli-Sforza (1965), Jancey (1966) 1966 and McQueen (1967) as well as Ball and Hall (1967). Unfortunately, this criterion is scale-dependent, thus different solutions will be derived from the raw and standardized data (Everitt et al. 2001: 94). Additionally, it will identify spherical clusters even if the data itself would indicate clusters of a different shape.

The minimization of $\det(W)$, the determinant of the within-group of squares, as a clustering criterion is based on Friedman and Rubin (1967) who suggest to maximize $\det(T)/\det(W)$ as a clustering criterion to overcome some disadvantages of minimizing $\text{trace}(W)$ as it is scale-independent and does not impose only clusters with spherical shape. The basic idea is of this criterion is derived from multivariate analysis of variance where large values of this ration indicate that group mean vectors differ (see Krzanowski 1988). Since T remains the same for all partitions, only $\det(W)$ has to be considered as a clustering criterion. Detailed studies can be found in Marriott (1971; 1982). Friedman and Rubin (1967) also proposed the maximization of $\text{trace}(BW^{-1})$ as a clustering criterion which again is scale-independent but minimizing $\det(W)$ is more popular. Unfortunately, both criteria find groups of roughly the same size. The $\det(W)$ criterion allows different shapes in different clustering solutions but assumes that in one solution the clusters have the same shape. To overcome the similar shape problem of the $\det(W)$ criterion several other criteria have been suggested based on W (see for example Scott and Symons 1971; Maronna and Jacovkis 1974; Symons 1981).

Most of the clustering criteria based on T , B and W are heuristics for continuous data but some of them can be shown to be equivalent to more formal statistical criteria. Non-continuous data can be clustered using the dissimilarity matrix with a clustering criterion on

²¹ In linear algebra, the trace of an $n \times n$ square matrix A is defined to be the sum of the elements on the main diagonal.

proximities instead of raw data. Alternatively, the proximity matrix can be transformed into a Euclidean distance matrix and the clustering based on the representation of the objects in Euclidean space (Everitt et al. 2001: 99). Criteria on binary and ordinal data can be found in Gower (1974) and Späth (1985).

Besides the decomposition of the dispersion matrix T some indices for measuring the internal cluster quality can also be used for cluster optimization (see Table 4-13, chapter 4.6.5.2). Having chosen an index and its suitable aggregation over groups, e.g. the sum, the maximum or the minimum, it has to be minimized (lack of homogeneity/compactness) or maximized (separation/isolation). Additionally, both types of indices can be combined.

The most commonly used statistical criterion is the square-error criterion (Dubes and Jain 1980: 92). The final solution is a partition which minimizes the square-error of the entire data set for a fixed number of clusters. The definition of the square error criterion has already been explained in chapter 4.4.1.5. Several very popular criteria are versions of square-error (Dubes and Jain 1980: 90). Another global criterion finds a partition by fitting a mixture density model to the patterns. All these criterion functions find clusters which are globular or hyperellipsoidal (Jain and Dubes 1988: 95). There are also a number of partitioning clustering methods that are based on local criteria of density or mode estimation, graph connectivity and near-neighbor relationships (Jain and Dubes 1988: 92).

In contrast, clustering algorithms using a hill-climbing pass K -means procedures do not explicitly optimize a statistical criterion function but compare the proximities of objects and cluster centers of consecutive iterations. Nevertheless, they attempt to minimize the variance within each cluster using implicitly criterion function (Aldenderfer and Blashfield 1984: 47). Kaufman and Rousseeuw (1990) suggest the *K-medoid model* with their partitioning clustering algorithm PAM. This model uses medoid as cluster representatives and minimizes the sum of dissimilarities between the objects and the representative object of a cluster. Another approach minimizes the largest distance between any object to the representative object of its cluster. This is called the *K-center model* (Hakimi 1965). If a *covering model* is applied the objects are assigned to a cluster if they are within a given distance of its representative object. The objective is to minimize the number of clusters to achieve this aim (Kaufman and Rousseeuw 1990: 111).

Although a large number of criterion functions can be found in the literature of cluster analysis and its areas of science only a small number of criteria are truly independent and often the same criterion appears in different publications (Jain and Dubes 1988: 91). Some authors have evaluated similarities between different criteria function, e.g. Shaffer et al. (1979), or Urquhart (1982). To evaluate clustering criteria to find the best criterion for obtaining a partition a precise and workable definition of “cluster” has to exist (Jain and Dubes 1988: 91). However, clusters can vary in shapes and size in a multidimensional pattern space which makes a clear definition impossible. Each clustering criterion imposes a certain structure on the data and if the data happen to conform to the requirements of a particular criterion the true clusters are recovered.

4.5.2.3 Proximity Measure

The wide range of *proximity measures* presented in section 4.3 can be applied to hierarchical as well as partitioning clustering procedures. Although the choice of the appropriate proximity measure depends on the type of data, the application area and the problem, the Euclidean metric is the most common metric to measure the distance between an object and a cluster center. The Mahalanobis distance is also often used. However, it has the disadvantage that it is computationally expensive (Jain and Dubes 1988: 93). According to Diday and Govaert (1977) this measure can be used to modify K -means clustering to become an adaptive procedure for standardization (see chapter 4.2.2.2). If text documents have to be clustered the cosine similarity is commonly used (see section 4.3.2 and chapter 3, section 3.3.2).

Most of the partitioning clustering techniques assume continuous-valued feature vectors (Jain and Dubes 1988: 92). Thus the data objects can be viewed as patterns in a metric space. If the features are on a nominal or ordinal scale, Euclidean distances and cluster center are not very meaningful. With those scales hierarchical clustering methods are normally applied. Wong and Wang (1979) suggest a clustering method for discrete-valued data which is similar to the mode estimation procedure for continuous data. The main difference between this approach and the mode estimation procedure is that it approximates the high-order discrete probability distribution by a second-order product and uses Hamming distance between patterns (Jain and Dubes 1988: 92).

4.5.2.4 Type of Pass

Besides the *statistical criterion* function the updating procedure is characterized by its *type of pass*. The term pass or cycle refers to the process of examining the cluster label of every pattern once. The two basic types of a pass are the K -means pass and the hill-climbing pass (Aldenderfer and Blashfield 1984: 47). A third type of a pass is the forcing pass (Dubes and Jain 1980: 97). The basic idea behind employing a pass is this: The complete enumeration of every possible partition has been and is still not possible for larger data sets. Some criteria do require to enumerating all partitions to find the optimum (Gordon 1999). Optimization techniques are available which reduce the necessary number of enumerations, e.g. *dynamic programming* (Jensen 1969) or *branch and bound algorithms* (Koontz et al. 1975; Diehr 1985). This problem of time consuming and space expensive enumeration of all partitions can also be overcome by rearranging existing partitions in a pass through the data and keeping the new one only if it improves the criterion function.

The K -means pass is also called the *nearest centroid sorting pass* or the *reassigning pass*. Basically, the objects are reassigned to the cluster with the most similar (“nearest”) centroid. It is the most widely used type of pass (Everitt et al. 2001: 102). One of the first implementations can be found in Ball and Hall (1967) implicitly minimizing $trace(W)$ (see chapter 4.5.2.2). K -means passes can be combinatorial or non-combinatorial, exclusive or inclusive (Aldenderfer and Blashfield 1984: 47): A *combinatorial pass* recalculates the centroid of a cluster after each change in its membership. In contrast, a *non-combinatorial pass* recalculates the centroids only after a complete pass through the data. Exclusive methods compute the centroid of the parent cluster without the object under consideration, whereas inclusive methods include the object in the calculation. Additionally, the order in which single

objects are considered for relocation can be random or systematic (Everitt et al. 2001: 100). The K -means method by McQueen (1967) is one of the most popular non-hierarchical clustering methods (Kaufman and Rousseeuw 1990: 112). It employs a combinatorial K -means pass whereas the algorithm FORGY by Forgy (1965) is non-combinatorial. Both methods are variance minimization techniques using the square-error criterion with centroids as cluster representatives (Kaufman and Rousseeuw 1990: 112). FORGY starts with a randomly selected initial partition and then it calculates the centroid for each cluster. At each iteration a non-combinatorial pass through the data is calculated until no changes in the cluster assignments have occurred. The algorithm can also directly start with a set of pre-defined seed points as centroids. Using centroids with a non-combinatorial K -means pass, like the simple and straightforward FORGY algorithm by Forgy (1965) the final partition sometimes contains less than K clusters (Kaufman and Rousseeuw 1990: 113). This problem will not occur with medoids as representatives, or a combinatorial pass. Although very similar to Forgy's method the K -means method by McQueen (1967) uses a combinatorial K -means pass calculating both the changes in the old and the new cluster after each change in the cluster membership (Kaufman and Rousseeuw 1990: 113). In contrast to Forgy's method, the number of cluster cannot change as a combinatorial pass is used. K -means type algorithms are very sensitive to the choice of the initial partition (Aldenderfer and Blashfield 1984: 48). Especially the commonly used option of a randomly selected initial partition can hinder to gain the global optimum. Figure 4-11 illustrates the K -means clustering algorithm.

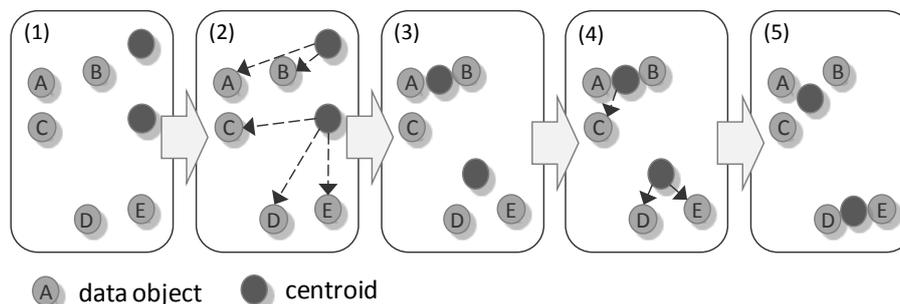


Figure 4-11: Illustration of K-means clustering algorithm: 1) initial randomized centroids and some data objects; 2) objects are associated with the nearest centroid; 3) centroids are moved to the center of their respective cluster; 4) step 2 & 3 are repeated until a suitable level of convergence has been reached; 5) final result

In contrast to K -means passes who use the proximity between object and cluster, a *hill-climbing pass* assigns an object to a new cluster if the change of the cluster membership will optimize the value of the statistical criterion function (Aldenderfer and Blashfield 1984: 47). Friedman and Rubin (1967) have defined a hill-climbing pass and a forcing pass in their clustering algorithm using the square-error criterion with centroids as cluster representatives (Jain and Dubes 1988: 97; Kaufman and Rousseeuw 1990: 114). Wishart (1987) use a hill-climbing pass where atypical objects can be allocated to an outlier group which is disregarded when the cluster criterion is evaluated.

A *forcing pass* perturbs the partition to avoid getting trapped at a local minimum (Dubes and Jain 1980: 98). The criterion function is recalculated after each test, the best partition found is retained and the forcing pass is repeated for the next cluster. These passes are applied repeatedly until convergence is obtained. *Simulated annealing algorithms* also try to avoid the problem of becoming trapped in an inferior local optimum. Instead of relocating an object

which reduces the quality of a partition a small probability is assigned (Klein and Dubes 1989; Selim and Asultan 1991; Sun et al. 1994). However, this type of algorithm suffers from the disadvantage that the cluster result is strongly affected by internal parameters (Everitt et al. 2001: 102).

4.5.2.5 Number of Clusters

Only a few partitioning methods provide a procedure to determine the *optimal number of clusters*. Most of them depend on a good estimation of the user (Kaufman and Rousseeuw 1990: 114). The partitioning clustering algorithms ISODATA by Ball and Hall (1964) can create new clusters or merge existing clusters if certain conditions are met. The capability of adjusting the number of clusters allows the algorithm to recover from poor initial partitions and lets it select the most suitable number of clusters (Jain and Dubes 1988: 98). After a first run where the number of clusters K is specified by the user, ISODATA removes outliers and too small clusters from the data set. Now, the algorithm either splits one cluster if the current number of clusters is more than $K/2$ or merges two clusters if the current number of clusters is $2K$. If neither condition is met the algorithm alternates between splitting and merging. Clusters are split if their average sum of dissimilarities is over the splitting threshold. Two clusters are merged if their cluster centers are sufficiently close according to the merging threshold. The thresholds for splitting and merging clusters are specified by the user. The algorithm is performed with the new number of clusters which replaces K unless a user defined maximum number of runs is reached or a clustering is obtained twice. According to Anderberg (1973) this method requires some a priori knowledge about the data and affords periodic human intervention.

Relative indices for measuring the validity of partitioning structures can be used to determine the number of clusters after several runs with different numbers of K . A detailed overview is given in section 4.6.5.3.

4.5.2.6 Convergence Criterion

An important part of any iterative partitioning clustering algorithm is the *convergence criterion*, which specifies under which condition the algorithm stops (Jain and Dubes 1988: 99). Partitioning algorithms naturally terminate when the criterion function cannot be improved. This means, the cluster labels for all patterns do not change anymore between two successive iterations. However, to prevent endless oscillations a maximum number of iterations should be specified. In practice, algorithms using the K -means pass converge rapidly. Selim and Ismail (1984) provide a study on the convergence of the K -means algorithm, whereas Pollard (1981) discusses the conditions for a reliable convergence of the K -means algorithm as the number of patterns increases.

4.5.2.7 Computational Complexity

The *computational complexity* of the general updating algorithm to generate one partition with a given set of parameters is $\mathcal{O}(ndKT)$, where n is the number of patterns, d the number of features, K the number of clusters desired and T the number of iterations (Jain and Dubes 1988: 100). The value of T depends on the initial centroids, the distribution of patterns and the

size of the data set (Jain and Dubes 1988: 101). In practice, a maximal number of iterations will be specified. Using parallel processing can reduce the computation time.

Computational complexity also refers to the problem of finding the global optimum. As already mentioned it would be too computationally expensive to calculate all possible solutions for a given number of clusters. Thus, some heuristic is used to preselect a small subset of “reasonable” partitions that has a good chance of containing the optimal partition (Jain and Dubes 1988: 91). The updating procedure of the iterative partitioning algorithm starts with an initial partition and then moves objects from one cluster to another cluster until the criterion function converges and no improvement can be achieved. Only a limited number of partitions are calculated and compared on the basis of the initial partition.

Using subsamples from the whole range of possible partitions leads to a major drawback of all iterative partitioning clustering procedures. There is always some possibility that the global optimum was not included in the sample and only the algorithm converges to a local optimum (Aldenderfer and Blashfield 1984: 48). Although even a thorough sampling procedure will not avoid this problem an appropriate validation procedure can help to determine whether the solution is globally optimal (see section 4.6).

4.5.3 Fuzzy Clustering

Using a non-overlapping clustering procedure each pattern belongs to exactly one cluster. Objects in one cluster are supposed to be more similar to each other than to patterns in other clusters. This method is appropriate if the clusters are compact and well separated and there is no ambiguity in assigning the objects (Jain and Dubes 1988: 130). An example of two well-separated non-overlapping clusters is provided in Figure 4-12 a). But sometimes the boundaries of clusters are not sharp. In this case the assignment of an object to only one cluster may be difficult and somehow inaccurate. This type of clusters are said to have “fuzzy” boundaries (Jain and Dubes 1988: 131). An example of two overlapping clusters is provided in Figure 4-12 b).

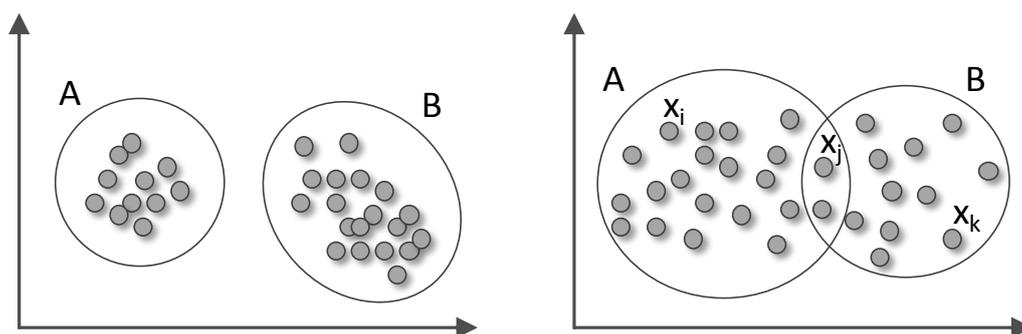


Figure 4-12: Examples of cluster structures: (a) well-separated clusters; (b) overlapping clusters. Source/Following: Jain and Dubes (1988: 130)

For this reason, Zadeh (1965) has developed the *fuzzy set theory*. It permits an object to belong to a cluster with a certain grade of membership. The degree of an object belonging to a cluster is expressed by a *membership coefficient* ranging from 0 (no membership) to 1 (full membership) (Jain and Dubes 1988: 131; Kaufman and Rousseeuw 1990: 164). For ordinary clusters, called “crisp” clusters, the membership is either 0 or 1. With fuzzy clusters, objects can belong to more than one cluster and their membership coefficients sum up to 1. The larger

the membership value in one cluster the more confidence exists the objects belongs to the cluster. However, membership grades are subjective in nature and are based on definitions rather than measurements (Zadeh 1984). Early works about fuzzy clustering algorithms can be found in Bellman et al. (1966), Ruspini (1969), Gitman and Levine (1970), Bezdek (1974), and Dunn (1974). Kim et al. (2004) propose a cluster validity index to determine the optimal partition and optimal number of clusters for fuzzy clustering. The proposed validity index is based on the inter-cluster overlap and the distance between fuzzy clusters (separation measure, see section 4.6.5.2).

A fuzzy partitioning clustering algorithm can be characterized by its membership function and its objective function. The performance of a fuzzy clustering algorithm depends critically on the definition of the *membership function* (Jain and Dubes 1988: 132). The construction of the membership proposed by Backer (1978) is based on similarity decomposition. According to Backer (1978) the *similarity* or *affinity function*, can be based on the distance concept, the neighborhood concept, or the probabilistic concept. Depending on the data this approach measures the relationship between a pattern and a cluster as a whole or between a pattern and one or more representatives of that cluster. Backer and Jain (1981) define an affinity function where the similarity is calculated as the Euclidean distance between the object and the centroid of the cluster.

Fuzzy partitioning clustering algorithms generate partitions that minimize induced fuzziness measured by a *criterion function* following the same steps as square-error clustering algorithms. The criterion function is also called *objective function* (Kaufman and Rousseeuw 1990). Criterion functions can be found in e.g. FANNY clustering by Kaufman and Rousseeuw (1990), fuzzy K -means clustering by Dunn (1974) and Bezdek (1974) and MND2 clustering by Roubens (1978). The fuzzy K -means clustering gains some of its popularity from the fact that it is a generalization of the classical K -means approach by McQueen (1967). Although all fuzzy clustering algorithms are partitioning procedures in nature a few generate hierarchies.

The output of a fuzzy algorithm not only includes a partition but also additional information in the form of membership values which have to be interpreted by the data analyst. One way of interpretation is to measure the *amount of fuzziness* of the clustering: The membership coefficients for all n objects and all K clusters can be stored in a membership matrix U . The membership coefficients of all objects in a fuzzy clustering can range from complete fuzziness, where the membership is $\frac{1}{K}$ for all memberships of an object, to an entirely hard clustering, where the membership for one cluster is 1 and 0 for all other clusters (Kaufman and Rousseeuw 1990: 171). To measure the amount of fuzziness the normalized Dunn's partition coefficient $F'_K(U)$ can be applied to the membership matrix (Dunn 1976). It ranges from 0 to 1. However, reducing the membership matrix U to only one value will not express the most important features of a fuzzy clustering (Kaufman and Rousseeuw 1990: 192). Sometimes, it can be interesting to compare the results from fuzzy clustering and hard clustering on the same data set. The closest hard clustering corresponding to a fuzzy one can be obtained by setting its membership to 1 for the cluster with the largest membership value and all other membership values of this object to zero (Kaufman and Rousseeuw 1990: 188). Then, a second function of the fuzziness of U is calculated which represents the average

squared error of a fuzzy clustering with respect to the closest hard clustering. If $D_k(U) = 0$ the fuzzy clustering is actually a hard clustering, if $D_k(U) = 1 - \frac{1}{K}$ it is completely fuzzy (Kaufman and Rousseeuw 1990: 192). The normalized version $D'_k(U)$ ranges from 0 to 1. These two indices can be used to represent a fuzzy clustering as a point in the (F'_K, D'_K) system (Trauwaert 1988).

According to Jain and Dubes (1988: 131) fuzzy clustering can be regarded as more appropriate than ordinary clustering for capturing human concepts such as “small”, “big”, “high” and “low”. Compared to hard clustering methods they reveal more detailed information about the structure of the data (Kaufman and Rousseeuw 1990: 165). Fuzzy sets are especially useful in applications involving imprecise and incomplete information, commonsense reasoning, and complex concepts (Zadeh 1984). However, it is not always clear that it offers any advantages over the classical and better understood clustering methods (Jain and Dubes 1988: 131). Fuzzy clustering methods are usually quite complicated and require a considerable amount of computation time which rapidly increases with the number of objects and clusters (Kaufman and Rousseeuw 1990: 165).

4.6 Clustering Tendency and Cluster Validation

In this chapter formal statistics as well as heuristics are summarized under the term *cluster validation index*. Section 4.6.1 provides a categorization to cluster validation indices whereas section 4.6.2 gives a brief introduction to cluster validation with hypothesis testing. Afterwards, a general overview is given about the different methods for validating hierarchies, partitions and clusters (see sections 4.6.3 to 4.6.5). The method of content-based clustering proposed in chapter 5 involves a precalculation of the initial partition by a suitable hierarchical clustering method which is then refined by a partitioning clustering procedure. Therefore, relative criteria for validating partitional structures (see chapter 4.6.4.3) are of major interest and therefore explained in more detail. Clustering algorithms have the major disadvantage that they will create clusters whether the data are naturally clustered or purely random. Therefore, a preliminary assessment of *clustering tendency* as discussed in section 4.6.6 should be an important part of clustering methodology to prevent the inappropriate application of clustering algorithm to data in which the clusters can only be artifacts of the clustering algorithms (Jain and Dubes 1988: 201).

4.6.1 Introduction to Cluster Validation

According to Jain and Dubes (1988: 160) an index of cluster validity is a measure of the adequacy of a structure obtained by cluster analysis. The adequacy of a clustering structure shows in how far the structure obtained reflects the intrinsic character, or actual structure, of the data. Jain and Dubes (1988: 160) mention three types of structures: hierarchies, partitions and clusters. For each type of structure the validity can be expressed by three types of criteria (Jain and Dubes 1988: 161):

- (1) *External criteria* measure the performance of a clustering algorithm by matching a clustering structure to a priori information. For example, an external criterion measures the degree of correspondence between cluster numbers, obtained from a clustering algorithm, and category labels, assigned a priori.

- (2) *Internal criteria* measure the fit between the structure and the data, using only the data itself. For example, an internal criterion measures the degree to which a partition, obtained from a clustering algorithm, is justified by the given proximity matrix.
- (3) *Relative criteria* decide which of two structures is better in some sense, such as being more stable or more appropriate for the data. For example, a relative criterion would compare the structure obtained by different clustering algorithms to decide which fits the data best.

Jain and Dubes (1988) use the term *criterion* to express the strategy by which a clustering structure is to be validated whereas an *index* is a statistic in terms of which validity is to be tested.

One special goal of cluster validation is to determine the right number of clusters in a data set (Dubes 1987; Jain and Dubes 1988). With partitioning clustering this question as to be answered before the actual clustering procedure as the number of clusters is one input parameter. With hierarchical clustering one has to answer the question after the clustering procedure to choose level of the clustering hierarchy which fits the structure of the data best (Everitt et al. 2001: 76). According to Aldenderfer and Blashfield (1984: 53) there are two major problems with determining the number of clusters. The first problem is the lack of a suitable null hypothesis and the complex nature of multivariate sampling distributions. The main difficulty in creating a suitable null hypothesis can be assigned to the lack of a proper definition of the structure and content of a cluster. As the concept of “no structure” in the data is not clear it is difficult to design suitable tests to decide if and what structure in the data is present or not. As a second problem real-world data samples can be composed of a mixture of different multivariate sampling distributions of unknown structure (Aldenderfer and Blashfield 1984: 54). Since there is no reliable theory to unravel these mixtures, formal tests of clustering tendency are not likely to be developed. The various application areas of cluster analysis are affected by this problem to different extents. As some fields like biological science mainly focus on examine the hierarchical trees for the general pattern of relationship between entities they are less affected by the problem of determine the right number of clusters. In contrast, social science depends heavily on a proper solution to this problem and thus two basic approaches have evolved there: heuristic procedures and formal statistic tests. Using heuristics, one has to be aware of them being experience-based techniques and only poorly understood yet, so the results have to be treated with caution (Aldenderfer and Blashfield 1984: 58). In addition, the procedure of determining the number of clusters should be used together with an appropriate validation index or be part of the cluster validation step. Otherwise the partition obtained with the chosen number of clusters might not be valid as measured by other criteria.

4.6.2 Cluster Validation with Hypothesis Testing

Tests of hypothesis use indices, or statistics, to determine if a recovered structure is appropriate for the data. In the case of external and internal criteria one will test whether the value of the index is either unusually large or unusually small. This requires that a baseline population is established. The general approach to cluster validity involves several steps (Jain and Dubes 1988: 147):

- (1) Define a null hypothesis expressing the idea of no structure which is appropriate for the data type.
- (2) Select a statistic, or index, sensitive to the presence of structure in the data and establish the distribution of the statistic under the null hypothesis.
- (3) Find a threshold that defines how large is “large” for the statistic selected.
- (4) The threshold establishes a formal test of hypothesis.
- (5) The power of the test evaluates the ability of the statistic to recognize the presence of a structure specified in an alternative hypothesis.

Defining a validation index (step 2) is much easier than defining a proper threshold which indicates an unusually large or small index (step 3) (Jain and Dubes 1988: 144).

Most commonly, the null hypothesis or randomness hypothesis is used (Aldenderfer and Blashfield 1984: 58). Under this hypothesis the data forms a random sample from a multivariate normal distribution. The term randomness refers to testing for “no structure” present in the data. The choice of the null hypothesis depends on the type of the data and the aspect of the data being tested (Jain and Dubes 1988: 144). However, these hypotheses are regarded as extremely limited in scope (Aldenderfer and Blashfield 1984: 54).

4.6.3 Validity of Hierarchies

The question considered in validating hierarchical structures is how well this structure, imposed by a hierarchical clustering method, fits the pattern matrix. Therefore, external, internal and relative validation indices can be employed (see sections 4.6.3.1 to 4.6.3.3).

4.6.3.1 External Indices

The problem of dealing with external indices is to determine if a hierarchy computed for a given data set fits an expected hierarchy, or a priori structure, which is assigned without regard to the measurement. For example, clustering numbers obtained by a clustering algorithm can be compared to category labels, assigned independent of the clustering. The null distributions for these statistics depend on a number of factors, especially the type of reference population, the number of objects, and the type of hierarchical clustering used. The major problem is that an expected hierarchy is not often available (Jain and Dubes 1988: 165).

4.6.3.2 Internal Indices

The question considered when using internal indices is whether a clustering hierarchy fits the data from which it was derived unusually well. This is important to be able to decide if one should be confident in the results of a hierarchical clustering or not (Jain and Dubes 1988: 165). The index used for answering these questions depends on the data scale. The *cophenetic correlation coefficient* (CPCC) has been proposed for quantitative data, and measures of rank correlation (see 4.3.2) have been used with qualitative data. The CPCC is the product-moment correlation (see chapter 4.3.2) between the entries of the proximity matrix and the cophenetic matrix²² which have to be on ratio or interval scales. The value of CPCC ranges from -1 (no

²² The *cophenetic proximity* between two objects is the level in the dendrogram for a particular clustering method at which the two objects are first placed in the same cluster. The *cophenetic matrix* is then a symmetric matrix containing the fusion levels of each pair of objects. The cophenetic matrix fulfills the property of ultrametricity.

match between both matrices) and +1 (perfect match). The CPCC has been widely used, especially in numerical taxonomy (Jain and Dubes 1988: 167). The major difficulty with CPCC in validating hierarchies is that its distribution depends on so many factors that one is forced into a Monte Carlo analysis to establish a baseline distribution for CPCC for each particular problem. Rohlf and Fisher (1968) provide a study of CPCC by Monte Carlo analysis when patterns are randomly chosen from uniform and Gaussian distributions and clustered by the UPGMA method. The average CPCC tends to decrease with the number of patterns and to be independent of the number of features. However, the CPCC values are more sensitive to the choice of the proximity measure than to the underlying distribution of the patterns. According to Rohlf (1970) the UPGMA method seems to produce consistently high values. He concludes that even a CPCC near 0.9 would not guarantee that the hierarchy fits the data. A rank correlation can measure the match between proximity and cophenetic matrix when the proximities are on an ordinal scale. Hubert (1974) proposed the Goodman-Kruskal γ statistic for this purpose and Cunningham and Ogilvie (1972) suggest Kendall's τ for comparing rank matrices (see chapter 4.3.2).

4.6.3.3 Relative Indices

In addition to the problem of validating a specific hierarchical clustering one is often faced with the problem of deciding which of two hierarchical clustering solutions fits the given data better. The indices are the same as those discussed for in section 4.6.3.2 applied in a different context. For example, Baker (1974) provides one of the first comparative studies on ordinal data for hierarchy validation. He starts with three “basal” dendrograms with different properties (chaining, binary, and arbitrary; see Figure 4-13) for $n = 16$ objects and then perturbs the ranks to include noise in the data.

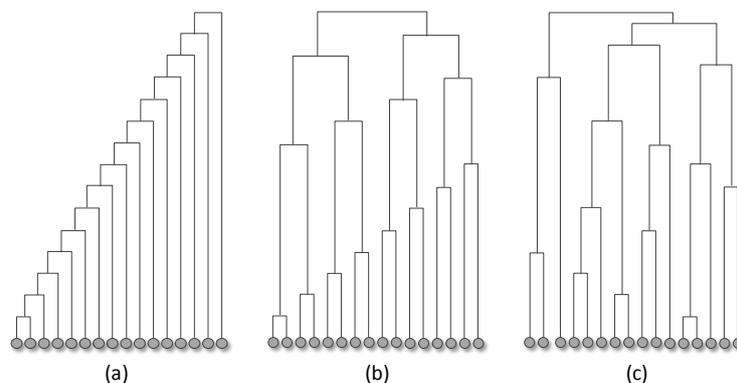


Figure 4-13: Different types of dendrograms. (a) chaining; (b) binary; (c) arbitrary. Source: Jain and Dubes (1988: 171)

The resulting proximity matrices are clustered by single and complete linkage methods and a Goodman-Kruskal γ is computed between each pair of proximity and corresponding cophenetic matrices. The mean γ for the complete linkage method is better than that for the single linkage method in all but one situation. The standard deviations of γ do not exhibit a consistent pattern. The fact that the complete linkage method provides a larger mean γ even under chaining is counterintuitive (Jain and Dubes 1988: 170). One would expect both methods performing equally well on an arbitrary dendrogram but the single linkage method to be sensitive to the chained dendrogram and the complete linkage method to be sensitive to the binary dendrogram. Hubert (1974) extends Baker's study and provides more evidence for the

superiority of the complete linkage over the single linkage method in practical situations with ordinal proximities.

4.6.4 Validity of Partitional Structures

In general, the partitional adequacy problem includes solutions to the question of the right number of clusters, the best cut of a clustering hierarchy and the comparison of partitions obtained by an iterative clustering procedure. It is therefore the most commonly encountered of all the validation problems (Jain and Dubes 1988). The estimation of the “true” number of clusters is still the fundamental problem of cluster validity (Duda and Hart 1973; Everitt 1979; Everitt et al. 2001). Due to Jain and Dubes (1988: 174) this question can be asked in at least three ways: An external index is appropriate for answering the question if the data contains the number of clusters one would expect when an a priori partition is available (see section 4.6.4.1). An internal index can help to decide whether a computed external index is unusually small (or large) so that the clustering is valid (see section 4.6.4.2). The two major difficulties here are the definition of a baseline distribution and the dependence of indices on problem parameters. Relative indices can be used to decide which one of two clusterings is better (see section 4.6.4.3).

4.6.4.1 External Indices

An external index of partitional adequacy assesses the degree to which two partitions agree. One partition U comes from a clustering solution. The second partition V is assigned a priori, independent of the data and the first partition, as from category labels. Hubert and Arabie (1985) define several indices for comparing two partitions. External indices can be expressed by indicators similar to association coefficients (see chapter 4.3.4). The *indicator functions* I_U and I_V indicate whether two objects are in the same cluster of partition U and V respectively. Table 4-10 contains the contingency table for these two indicator functions.

Table 4-10: Contingency table for indicator functions. Source: Jain and Dubes (1988: 173)

		I_V		
		1	0	Total
I_U	1	a	b	$m_1 = a + b$
	0	c	d	$M - m_1 = c + d$
	Total	$m_2 = a + c$	$M - m_2 = b + d$	$M = a + b + c + d$ $= n(n - 1)/2$

Table 4-11 lists some common external indices for comparing two partitions. Large values of these statistics imply close agreement between the two partitions (Jain and Dubes 1988: 174). The maximum value may not be achievable when the two partitions have different numbers of clusters. The Γ statistic is a special case of the Hubert Γ statistic (see section 4.3.2). The key step in applying these external indices of partitional adequacy is the formation of a baseline distribution. A clustering can then be termed “valid” if it has an unusually high value, as measured with respect to the baseline distribution. Hubert and Arabie (1985) suggest a baseline distribution in which the row and column sums are fixed but the partitions are selected at random.

Table 4-11: External indices of partitional adequacy. Based on: Jain and Dubes (1988: 174)

Name	Formula	Range
Rand (1971)	$\frac{a+d}{\binom{n}{2}} = 1 + \frac{Z - \frac{1}{2}(\sum_{i=1}^n n_i^2 + \sum_{j=1}^n n_j^2)}{\binom{n}{2}}$	[0, 1]
Jaccard	$\frac{a}{a+b+c} = \frac{Z-n}{\sum_{i=1}^n n_i^2 + \sum_{j=1}^n n_j^2 - Z - u}$	[0, 1]
Fowlkes and Mallows (1983)	$\frac{a}{\sqrt{m_1 m_2}} = \frac{\frac{1}{2}(z-n)}{\left[\sum_{i=1}^n \binom{n_i}{2} \sum_{j=1}^n \binom{n_j}{2}\right]^{1/2}}$	[0, 1]
Γ statistic	$\frac{(Ma - m_1 m_2)}{\sqrt{m_1 m_2 (M - m_1)(M - m_2)}}$	[-1, 1]

The external indices defined here can also be used to judge the relative merits of two partitions in recovering structure from the data and to test whether the data confirm a prior hypothesis, independent of any clustering (see e.g. Hubert and Subkoviak 1979).

4.6.4.2 Internal Indices

Internal indices of partitional adequacy measure the fit between the partition imposed by a clustering algorithm and the data themselves (Jain and Dubes 1988: 174). Several indices have been discussed and evaluated by Dubes and Jain (1979), Milligan (1981), and Milligan and Cooper (1985). Internal indices are strongly influenced by problem parameters, such as the number of patterns, features, and clusters, and the spread of the data (Jain and Dubes 1988: 178). Square-error, for example, naturally decreases as the number of clusters increases and increases with increasing number of objects and variables. Based on several experiments Jain and Dubes (1988) show that the effects of dimensionality and sample size are severe, even for idealized data that are well matched to the clustering algorithm. Although comparing an internal index to a baseline distribution is not advisable when validating a partition, this procedure is appropriate when trying to determine if a given set of data is better described with one or two clusters. This can be useful in deciding whether or not to split a cluster.

4.6.4.3 Relative Indices

Validating the adequacy of a partition with a relative index will help to answer the questions which of a set of partitions best matches the data given (Jain and Dubes 1988: 183). All of the internal indices could serve as relative indices. The sequence of clusterings can be obtained from repeated applications of a non-hierarchical clustering algorithm, but are usually obtained from cutting a dendrogram at successive levels.

A procedure for choosing the best level for cutting a dendrogram is called a *stopping rule*. Besides computing a statistic based on a baseline distribution one can also approximate this distribution and use a heuristic. Heuristic procedures are by far the most commonly used methods. In general, they look for some unusual aspect of a sequence of partitions, such as a maximum, minimum, or “significant knee”. They range from simple visual inspection of the dendrograms or two partitions to more formal but still heuristic methods involving some formalized criterion (Aldenderfer and Blashfield 1984: 54). There exist a number of methods

which are specific for hierarchical clustering procedures but also some that are independent from the type of clustering method (see Table 4-12).

According to Gordon (1999) stopping rules can be categorized into global and local rules. *Global rules* evaluate the index over the entire data set (e.g. within-cluster versus between-cluster similarities) and optimize it as a function of the number of clusters. *Local rules* are only based on pairwise cluster comparison, e.g. within-cluster similarities. In general, global rules are not defined for $K = 1$. Thus, they do not offer any evidence if the data should be clustered at all. Local rules need a threshold value or a significance level which depends on the specific data and in most cases has to be empirically determined. The indices presented in Table 4-12 are no discussed in some detail.

Table 4-12: Relative indices of partitional adequacy

Name	General Principle	Clustering Type	Source
Davies-Bouldin index	global rule	partitioning and hierarchical clustering	Davies and Bouldin (1979)
Modified Hubert's Γ	local rule	partitioning and hierarchical clustering	Theodoridis and Koutroubas (1999)
Upper Tail Rule	local rule	hierarchical clustering	Mojena (1977)
Moving Average Approach	local rule	hierarchical clustering	Mojena (1977)
Silhouette Coefficient	global rule	partitioning and hierarchical clustering	Kaufmann and Rousseeuw (1990: 87)
Calinski and Harabasz Stopping Rule	global rule	partitioning and hierarchical clustering	Calinski and Harabasz (1974)
Hartigan Stopping Rule	local rule	hierarchical clustering	Hartigan (1975)
Index by Duda and Hart	global rule	partitioning and hierarchical clustering	Duda and Hart(1973)
Index by Beale	global rule	partitioning and hierarchical clustering	Beale (1969b)
Gap statistic	local rule	partitioning and hierarchical clustering	Tibshirani et al. (2001)

The *Davies-Bouldin index* (Davies and Bouldin 1979) is plotted against a number of clusters and clustering is stopped when the index is minimized. The index is independent of the number of clusters and the clustering method. Given a partition of n objects into K clusters, one first defines the *within-to-between cluster spread* for all pairs of clusters (j, k) as

$$R_{j,k} = \frac{e_j + e_k}{m_{j,k}}$$

where e_j is the average error for the j th cluster and $m_{j,k}$ is the Euclidean distance between the centers of the j th and k th cluster. Based on the within-to-between cluster spread the index for the k th cluster is

$$R_k = \max_{j \neq k} \{R_{j,k}\}$$

and the *Davies-Bouldin index* for the K -cluster clustering is

$$DB(K) = \frac{1}{K} \sum_{k=1}^K R_k \text{ for } K > 1$$

The smaller $DB(K)$ the better the clustering. A value of zero will be obtained for the trivial clustering that places each object in an individual cluster. The index is not defined for $K = 1$ when all objects are in the same cluster. Plotting $DB(K)$ against K for successive values of K the partition is chosen that minimizes the index. Together with other validity measures Ingaramo et al. (2008) evaluate the Davies-Bouldin index by comparing their correlation with the external F -measure on three different short text corpora and testing their relative hardness. In general, the Davies-Bouldin index did not perform well in all data sets but had good results on relative hardness.

The *MH statistic* (modified Hubert's Γ , see chapter 4.3.2) is defined as the point serial correlation between two partitions (Jain and Dubes 1988: 186). Here the Euclidean distance between the centers of clusters to which the objects belong is used as a distance measure between two objects. The idea is that cluster centers estimate the "true" positions of the clusters in the pattern space and that deviations from the centers are due to errors and distortions. The *MH statistic* can be defined as follows (Theodoridis and Koutroubas 1999):

$$\Gamma = \frac{1}{m} \sum_{i=1}^{n-1} \sum_{j=i+1}^n p(i, j) \cdot q(i, j)$$

where n is the number of objects in a data set, m defined as $m = n(n - 1)/2$, $p(i, j)$ is the value from the proximity matrix for elements i and j and $q(i, j)$ is the value from an $n \times n$ matrix containing the distances between the corresponding cluster representatives. This index decreases monotonically as K decrease, so one must seek a significant knee in a plot of $MH(K)$ against K . A value of $MH(K) = 1$ refers to the trivial clustering in which all objects are in an individual cluster and is not defined for the clustering in which all objects are in the same cluster. A preliminary Monte Carlo study suggests that *MH* is a more reliable way to identify the true number of clusters than *DB*. However, *MH* requires some judgment as to the definition of a significant knee, whereas *DB* requires only that a minimum be identified.

One way of detecting the right partition in a hierarchical tree is to examine the value of the fusion coefficients for a significant "jump" (Aldenderfer and Blashfield 1984: 57; Everitt et al. 2001: 76). A jump implies that two relatively dissimilar clusters have been merged. Therefore, the level before this "bad" merge indicates at the right number of clusters. However, if many small jumps can be found, this test offers no way to select the "correct" one (Aldenderfer and Blashfield 1984: 57). In Mojena (1977) and Mojena and Wishart (1980) this approach has been extended to a heuristic procedure with a formula defining a sufficient jump. *Stopping rule #1*, also known as *upper trail rule*, by Mojena (1977) is based on the relative size of the different fusion levels. It states that an optimal partition of a hierarchical clustering solution must satisfy the inequality

$$\alpha_{j+1} > \bar{\alpha} + ks_{\alpha}$$

where α_{j+1} is the value of the fusion coefficient at level $j + 1$ of the clustering process. The terms $\bar{\alpha}$ and s_{α} are the mean and unbiased standard deviation of the fusion coefficient values of the j previous fusion levels, and k is a constant which is termed standard deviate. Mojena (1977) suggest values of k ranging from 2.75 to 3.50, whereas Milligan and Cooper (1985) suggest 1.25. Alternatively a t -distribution can be used (Everitt et al. 2001: 77). If no partition

fulfills this inequality then the data has only one cluster, or the optimal partition is at stage j where stage $j + 1$ yields the maximum standard deviate, or another heuristic has to be employed (Mojena 1977: 360). For the second case, the standard deviate k can be calculated for each level as

$$k_j = \frac{\alpha_{j+1} - \bar{\alpha}}{s_\alpha} \text{ (Aldenderfer and Blashfield 1984: 58)}$$

A visual approach then is to plot k_j against j and identify breaks in this plot (Everitt et al. 2001: 77). It is also possible to test the statistical significance of the results with the t-statistic with $n - 2$ degrees of freedom, where n is the number of fusion coefficients, by multiplying the square root of $n - 1$ and the value of the standard deviate k (Wishart 1982; Aldenderfer and Blashfield 1984: 58; Everitt et al. 2001: 77).

A variant of this methods proposed by Mojena (1977) is based on the *moving average approach* where the optimal partition corresponding to the first stage j , in a partial cluster sequence from $j = r$ to $j = n - 2$ clusters which satisfy

$$\alpha_{j+1} > \bar{\alpha} + L_j + b_j + ks_j$$

In this formula $\bar{\alpha}$ and s_j are the mean and standard deviation of the fusion values based on the previous t -values. L_j (trend lag) and b_j (moving last-squares slope of the fusion levels) are corrections to the mean of the upward trend in fusion levels. This method has the advantage that the fusion level under consideration is not part of the sample statistic (Wishart 1982). The value of r has to be chosen by the the investigator. With both stopping rules by Mojena one will choose the smallest value with the lowest number of clusters where the rule is satisfied (Everitt et al. 2001: 77).

Based on Rousseeuw (1987), Kaufmann and Rousseeuw (1990: 87) suggest the *silhouette coefficient* as a measure of the optimal number of clusters (compare different partitions of one hierarchical clustering algorithm) as well as the amount of clustering structure that has been discovered by the clustering algorithm (compare partitions obtained by different algorithms). The clustering coefficient is constructed as follows. First, $a(i)$ is defined as the average within-cluster dissimilarity of object i assigned to cluster A and $b(i)$ as the minimum dissimilarity to any other cluster. The cluster B which minimizes $b(i)$ is also called the *neighbor* of object i . This is the second-best choice of all clusters for object i . The quantity $s(i)$ of object i can be calculated as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ with } -1 \leq s(i) \leq 1$$

There are three extreme situations for values of $s(i)$: When $s(i) \rightarrow 1$ then $a(i) \ll \min(b(i))$ and object i can be regarded as well-classified. When $s(i) \approx 0$ there is no clear preference for any cluster A or B . When $s(i) \rightarrow -1$ object i is more similar to the objects in cluster B than in cluster A and it can be regarded as misclassified. The *silhouette* of a cluster is a plot of the $s(i)$ values of all of its members, ranked in decreasing order (Kaufman and Rousseeuw 1990: 86). A wide silhouette with uniformly large $s(i)$ values indicates a well-classified cluster. The *average silhouette width* can be calculated for a single cluster averaging all $s(i)$ values of the cluster or for the entire data set. The latter value, $\bar{s}(k)$, can be used as a measure of the quality

of a partition. A *silhouette coefficient* can then be calculated for different numbers of clusters k which can be used to select the optimal number value for k :

$$SC = \max_k \bar{s}(k), \text{ with } 2 \leq k \leq n - 1$$

Calinski and Harabasz (1974) propose a stopping rule for hierarchical clustering to find the best cut by measuring the similarity between internal and external distances. Here, n is the number of objects to be clustered and k the clustering step. With a hierarchical clustering algorithm, the number of the clustering step is also the number of clusters in this step as well as the number of the cluster created at this step. For each step two measures are calculated: $BGSS(k)$ is the sum of the dispersions between the k cluster centroids and the general centroid, called between-cluster or *between-group sum of squares*. The second measure, $WGSS(k)$, is the within-cluster or *within-group sum of squares*. It is calculated from the dispersion of the cluster members weighed by the number of cluster members. The dispersion is measured by the sum of squared distances between each member and the cluster centroid. $VRC(k)$ is then the *variance ratio criterion* of the external to internal distances for a cluster k derived from the between-group and the within-group sum of squares:

$$VRC(k) = \frac{\frac{BGSS(k)}{k-1}}{\frac{WGSS(k)}{n-k}}$$

The best cut can be found if $VRC(k)$ is maximized. The distance used is the Euclidean distance. $VRC(k)$ is analogous to the F -statistic in univariate analysis (Calinski and Harabasz 1974: 10). This method can also be employed to compare the results from a partitioning clustering algorithms for different values of k to choose the right number of clusters. $BGSS(k)$ and $WGSS(k)$ are the same as $trace(B)$ and $trace(W)$ in section 4.5.2.2.

The stopping rule by Hartigan (1975) is similar to the index by Calinski and Harabasz (1974) but it only employs $WGSS(k)$, the total sum of squared distances of cluster members from their cluster centroid in all k clusters. It also compares successive levels of a clustering hierarchy. At each level, it compares the current cluster k with the cluster from the previous step $k + 1$:

$$H(k) = \left(\frac{WGSS(k)}{WGSS(k+1)} - 1 \right) (n - k - 1)$$

Hartigan (1975: 91) suggests that as a rule of thumb $H(k) > 10$ indicates that the number of clusters should be increased from k to $k + 1$. Thus, the optimal solution is the smallest $k + 1$ such that $H(k) > 10$.

Instead of directly trying to find a knee point in the graph of the criterion function, the *gap statistic* by Tibshirani et al. (2001) creates a sample of reference data that represents the observed data as if it has no meaningful clusters in it and is simply made up of noise. The criterion function of the reference data is then compared to that of the observed data in order to identify the value of k in the observed data that is least like noise. This value therefore represents the best clustering of the data. Let

$$D_r = \sum_{i,j \in C_r} d(i,j)$$

be the sum of the pairwise within-cluster distances, and set

$$W_r = \sum_{r=1}^k \frac{1}{2n_r} D_r.$$

If the squared Euclidean distance measure is used then W_r is the pooled within-cluster sum of squares around cluster means. The gap statistic is then computed as

$$\text{Gap}_k(k) = E_n^*\{\log(W_k)\} - \log(W_k),$$

where E_n^* denotes expectation under a sample of size n from the reference distribution. The estimate of the optimal number of clusters is the value k where $\log(W_k)$ falls farthest below the reference curve. An example of the gap statistic is provided in Figure 4-14.

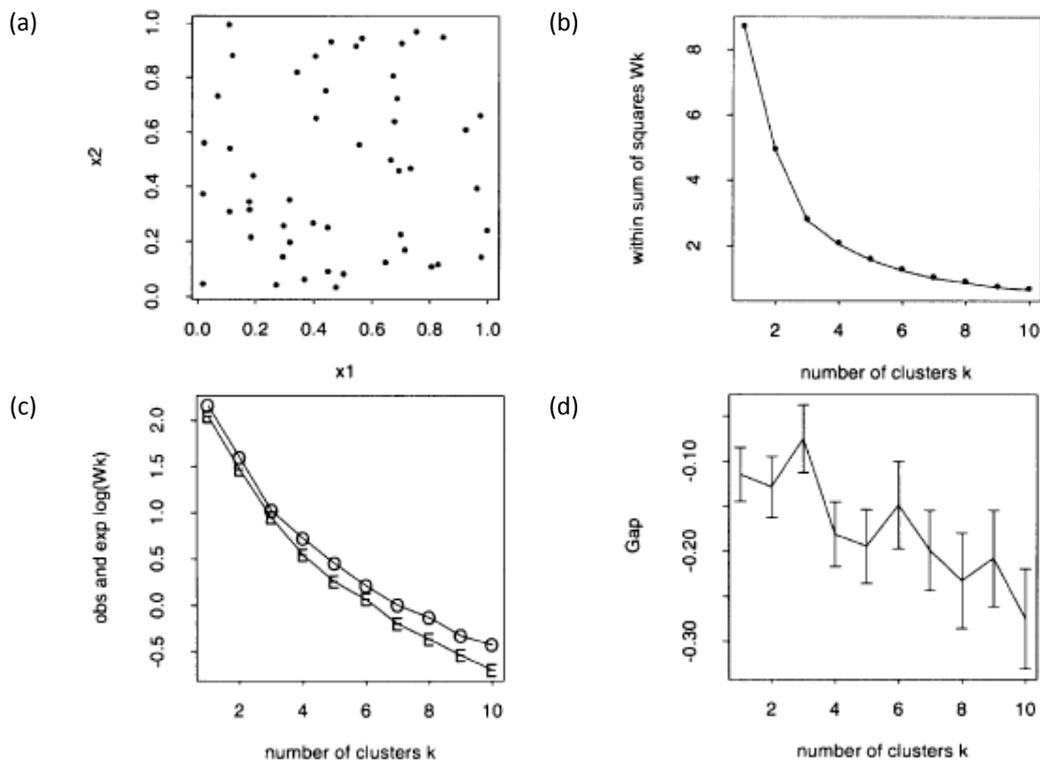


Figure 4-14: Illustration of the gap statistic: (a) original data set; (b) within sum of squares function W_k ; (c) observed and expected $\log(W_k)$; (d) gap curve. Source: Tibshirani et al. (2001: 416)

Pedersen and Kulkarni (2006) propose a variant of the gap statistic which allows to use any clustering criterion function to compare observed and reference data.

A number of formal approaches are evaluated by Milligan and Cooper (1985) and later Gordon (1998) and more recently e.g. Savova et al. (2006) on word sense discrimination, El Sayed et al. (2008) on document clustering and Vendramin et al. (2009; 2010) in general. Comparing 30 decision rules for hierarchical clusterings Milligan and Cooper (1985) identify five best-performing methods and recommend the global methods by Duda and Hart (1973) and Beale (1969b) as well as the stopping rule by Calinski and Harabasz (1974) for hierarchical clusterings. The Davies-Bouldin index and the MH index are among the top ten. Savova et al. (2006) compare the criterion functions by Hartigan (1975), Calinski and Harabasz (1974) and an adaption of the gap statistic in the context of word sense discrimination with respect to accuracy, quality (measured by the F -score) and predicted average number of sense. In this study, the stopping rule by Calinski and Harabasz (1974) performs best. Vendramin et al. (2009; 2010) suggest indices based on Hubert's Γ (see section 4.3.2). As the determination of the right number of clusters is an important problem when

designing an automatic clustering procedure several relative indices have been discussed in this section. Altogether no clear consensus but rather contrarily advises can be found in the literature which method performs best. According to Baxter (1994) informal and subjective criteria based on the investigators' experience will remain the most common approaches.

4.6.5 Validity of Internal Clusters

The two main properties of a cluster are compactness and isolation (Jain and Dubes 1988: 188). *Compactness* measures the internal cohesion among the objects in the cluster whereas *isolation* measures separation between the cluster and other patterns. A valid cluster is unusually compact and unusually isolated. Here again, one has to define a reference population and baseline distribution in terms of which "unusual" has meaning. All indices of internal cluster validity presented in the following sections 4.6.5.1 to 4.6.5.3 measure compactness and isolation.

4.6.5.1 External Indices

An external index validates an a priori cluster by contrasting the isolation and compactness of the cluster to the compactness and isolation of a randomly chosen cluster (Jain and Dubes 1988: 192). This is appropriate when the cluster labels are assigned without regard to the proximity matrix. If the data are on an ordinal scale the problem of validating clusters can be solved from probability theory (see Bailey and Dubes 1982). The reference population is implied by the random graph hypothesis which states that all ordinal proximity matrices are equally likely. Selecting a proximity matrix under the random graph hypothesis is equivalent to randomly placing links between distinct nodes on an n -node graph where all unfilled positions are equally likely at each step. The order in which the links are inserted establishes the ordinal proximities: A (n, N) random threshold graph is a graph formed by inserting N links at random into an n -node graph where each node represents one object. The *compactness* of a cluster at a specific proximity level N is the number of within-cluster links (inner links) in this graph. Similarly, the *isolation* of this cluster at the same proximity level is the number of links connecting the cluster with the rest of the graph. A perfect cluster of size k would be maximally isolated with no connecting links and maximally compact with $k(k - 1)/2$ inner links. A valid cluster should have an unusually large amount of inner edges and an unusually small amount of linking edges. Establishing an external index, the *isolation index* $I(A)$ of a cluster A is the probability that the observed number of links connecting objects in an a priori cluster to the rest of the objects is less or equal to a random variable denoting the number of connecting links in a (n, N) random threshold graph. The *compactness index* $C(A)$ of a cluster A is the probability that the observed number of internal edges between objects in an a priori cluster is more than or equals a random variable denoting the number of internal links in a (n, N) random threshold graph. Both indices should be small for a valid cluster at the particular proximity level. A *cluster profile* plots the compactness and isolation index of a cluster against the number of links in the threshold graph (see Figure 4-15).

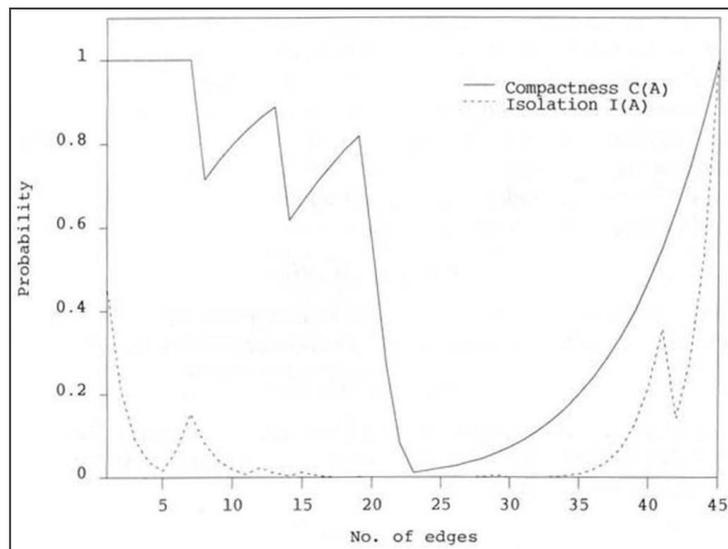


Figure 4-15: Cluster profiles for a priori cluster. Source: Jain and Dubes (1988: 193)

In the example given in Figure 4-15 it appears that the cluster is more isolated than compact. The evidence for compactness is weak, so one would not call cluster A a valid cluster.

4.6.5.2 Internal Indices

As almost any cluster identified by a clustering algorithm is unusually isolated and compact when compared to a randomly chosen cluster the validation scheme of an external index is not suitable for internal indices (Jain and Dubes 1988: 191). To avoid this problem there are two possibilities to define a reference population and a baseline distribution that provide fair tests of cluster validity with internal indices: the CM-reachable method and the best-case method. The CM-reachable method uses Monte Carlo analysis and the best-case method employs bounds on probabilities. The *CM-reachable method* limits the population of clusters to those clusters that can be “reached”, or achieved, by a particular clustering method when applied to proximity matrixes created under the random graph hypothesis. A sampling distribution can then be worked out over the limited population by Monte Carlo means, taking into account the number of objects, the level at which the cluster is formed, the size of the cluster and the clustering method being employed. Baker and Hubert (1976) develop such a test of cluster validity for complete linkage clustering.

The *best-case method* is independent of the clustering method and does not use a laborious Monte Carlo analysis (Jain and Dubes 1988: 193). The idea is to establish conservative bounds on the probabilities required in computing compactness and isolation indices. Given a (n, N) random threshold graph (see section 4.6.5.1) all best k -node subsets with minimum isolation index are assigned equal probability mass and all other k -node subsets are assigned probability mass zero. Similarly, a population of k -node subsets having the best, or largest, compactness index can be obtained as well. A measure of isolation for cluster A is then the probability that it is at least as isolated as the most isolated cluster in a random graph. Similarly, the compactness of cluster A is the probability of obtaining a cluster in a random graph that is more isolated than A . Exact expressions for these populations are not available but bounds can be used to define indices of cluster validity for k -node clusters. Thus, the *best-case isolation index* and the *best-case compactness index* for the k -node cluster with an

isolation value (number of connecting links) and a compactness value (number of internal links) are both functions of the proximity level N . Two relative indices of compactness and isolation indicate whether a cluster is compact among all clusters as isolated as the given cluster and whether a cluster is unusually isolated among all clusters as compact as the given cluster. A *probability profile* plots the four indices as functions of the proximity level N . If the best-case compactness and best-case isolation are both below a suitable level for an appropriate span of ranks the cluster under consideration is called valid. Since bounds are used large indices do not necessarily mean that a cluster is not valid. Judging a cluster valid from a probability profile does not always imply that all objects in the cluster have the same degree of isolation from other nodes in the graph or the same degree of compactness to other nodes in the cluster. For example, merging two very compact clusters might lead to an unusually compact cluster because of the individual clusters, not because of any homogeneity among all nodes in the cluster (Bailey and Dubes 1982).

A “good” cluster is both compact and isolated which means that it forms early in the dendrogram and is not absorbed into another cluster until late in the dendrogram. Ling (1972; 1973) has proposed an index of cluster validity for qualitative data as the probability that the lifetime of a cluster exceeds the observed lifetime under the random graph hypothesis. The observed lifetime of a cluster is defined as the difference between the rank at which the cluster is absorbed into another cluster and the rank at which the cluster is first defined. Valid clusters should “live” longer than randomly chosen clusters. The *Ling index* for a cluster containing a fixed number of objects is then defined to be the probability that a lifetime of a randomly selected cluster the observed lifetime of this cluster.

Table 4-13: Internal indices for internal cluster validation. Based on: Everitt et al. (2001: 91)

Measure		Index ($r \in \{1, 2\}$)
Lack of homogeneity	$h_1(m) = \sum_{l=1}^{n_m} \sum_{v=1, v \neq l}^{n_m} (\delta_{ml,mv})^r$	Sum of all (squared) dissimilarities between two objects from cluster m
Lack of homogeneity	$h_2(m) = \max_{\substack{l=1, \dots, n_m \\ v=1, \dots, n_k \\ v \neq l}} [(\delta_{ml,mv})^r]$	Maximum of all (squared) dissimilarities between two objects from cluster m
Lack of homogeneity	$h_3(m) = \min_{v=1, \dots, n_m} \left[\sum_{l=1}^{n_m} (\delta_{ml,mv}) \right]$	Minimum of the sum of the (squared) dissimilarities between all objects in cluster m and a single cluster member
Separation	$i_1(m) = \sum_{l=1}^{n_m} \sum_{k \neq m} \sum_{v=1}^{n_k} (\delta_{ml,kv})^r$	Sum of all (squared) dissimilarities between an object from cluster m and an object outside this cluster
Separation	$i_2(m) = \min_{\substack{l=1, \dots, n_m \\ k \neq m \\ v=1, \dots, n_k}} [(\delta_{ml,kv})^r]$	Minimum of the sum of all (squared) dissimilarities between two objects from cluster m

m := number of cluster under consideration
 n_m := number of objects in cluster m
 Δ := dissimilarity matrix

$\delta_{ql,kv}$:= elements in the dissimilarity matrix
 l := object in cluster m
 v := object in cluster k

Everitt et al. (2001: 91) suggest five heuristics as internal indices of the lack of homogeneity (lack of compactness) or separation (isolation) for internal cluster validation derived from the proximity matrix (see Table 4-13). With $r = 1$ and metric dissimilarities $h_2(m)$ is the diameter of the cluster and $h_3(m)$ is known as the *star index* as connecting all objects to the

reference objects will form a star-shaped graph with the reference object in the center for the smallest sum of dissimilarities (see Table 4-13). This reference object is than the medoid of the cluster (see section 4.2.1.3). Other measures can be found in Rubin (1967), Belbin (1987) and Hansen and Jaumard (1997). The notion of separation is used by Kim et al. (2004) to propose a new cluster validity index to determine the optimal partition and optimal number of clusters for fuzzy clustering (see section 4.5.3).

Having chosen an index to measure the validity of a cluster, criteria for comparing clusterings (see section 4.6.4.2) can also be defined by its suitable aggregation over groups, e.g. the sum, the maximum or the minimum, which then has to be minimized (lack of homogeneity/compactness) or maximized (separation/isolation). Both types of indices can also be combined.

4.6.5.3 Relative Indices

The problem of comparing two clusters has not received much attention yet (Jain and Dubes 1988: 200). Comparing the probability profiles of the two clusters is one approach. The data can be perturbed to test the stability of the cluster. The probability profiles before and after the perturbation can be compared to assess their stability.

4.6.6 Clustering Tendency

The term *clustering tendency* refers to the problem of deciding whether the data exhibit a predisposition to cluster into natural groups without identifying the groups themselves (Jain and Dubes 1988: 201). Clustering algorithms have the major disadvantage that they will create clusters whether the data are naturally clustered or purely random. This makes clusters validation very difficult. Therefore, a preliminary assessment of clustering tendency should be an important part of clustering methodology to prevent the inappropriate application of cluster analysis to data in which the clusters can only be artifacts of the clustering algorithms.

According to Jain and Dubes (1988: 201) testing for complete spatial randomness (see Diggle 1983), which is the same as a statistical test for the random position hypothesis (see section 4.6.1), can be used for testing for clustering tendency when data is characterized as samples of spatial point processes²³ restricted by a sampling window where the data come from (Ripley 1981; Diggle 1983). Most applications can be found with data in two dimensions (see e.g Hopkins 1954; Hines and Hines 1979; Ripley 1981; Osmer 1982). Some of these tests can be extended to the d -dimensional data. Unfortunately, this approach requires knowledge of the shape and size of the sampling window (Jain and Dubes 1988: 203): Depending on the sampling window chosen the data can appear as random or non-random which makes it a crucial part of the analysis.

Testing for clustering tendency will involve an internal criterion and no a priori information is used. The type of randomness and the clustering tendency test is determined by the type of data. Tests for spatial randomness can be used for data which can be represented as patterns in the d -dimensional feature space using a random position hypothesis. Such a test examines the

²³ A spatial point process is an arrangement of patterns, or points, scattered about a Euclidean space according to some probability model (Jain and Dubes 1988: 203).

spatial arrangement of the patterns and cluster analysis should only be applied to data that show the tendency to aggregate instead of being randomly arranged or regularly spaced (Panayirci and Dubes 1983; Smith and Jain 1984; Dubes and Zeng 1987). There are six categories of tests for randomness based on the type of information used: scan statistics, quadrat analysis, second moment estimates, interpoint distances, nearest-neighbor distances and graph structure (see Jain and Dubes 1988: 211). *Scan tests* count the number of patterns in the most populous subregion of the sampling window. An unusually large count accounts for the presence of some clustering structure (Naus 1966; Conover et al. 1979; Naus 1982). *Quadrat analysis* partitions a rectangular sampling window into rectangles of equal size (Greig-Smith 1964; Pielou 1969; Mead 1974; Rogers 1974). The number of points falling in each quadrat is count which will follow a Poisson distribution under randomness. Using *second moment structure* the test estimates the second moment of the spatial point process from the given data. The estimated function is compared to the theoretical function of a Poisson process to test the random position hypothesis (Liebertau 1977; Ripley 1977; Diggle 1983; Jain and Dubes 1988: 212). Structural relationships between patterns are reflected by *interpoint distances*. The observed distribution of interpoint distances can be compared to the theoretical distribution under random hypothesis to test for clustering tendency (Hammersley 1950; Alagar 1976; Cross 1980). Additionally, only the small interpoint distances can be observed. Here, many small distances will account for a clustering tendency (“clustered process”) whereas only a few distances indicate a regular process and a random process will fall between these extremes (Ripley and Silverman 1978; Silverman and Brown 1978). Studies about how to define the threshold for smallness can be found in Strauss (1975), Kelly and Ripley (1976) and Saunders and Funk (1977). Minimum spanning tree, Delauney tessellation and relative neighborhood graph depend on the global *graph structure*. The test statistic is based in the distribution of link length in the graph (Friedman and Rafsky 1979; Hoffman and Jain 1983; Smith and Jain 1984). The *nearest-neighbor distance* methods like pattern-to-pattern nearest-neighbor distances or sparse sampling measure the distance from each object to the closest object instead of mapping out the exact locations of all objects on a coordinate systems (Cross 1980; Panayirci and Dubes 1983).

Besides methods for data available as patterns in a d -dimensional space randomness can also be defined for data represented by an ordinal proximity matrix. (Jain and Dubes 1988: 221). Here, the random graph hypothesis serves as null hypothesis. Graph properties can then lead to tests for randomness. Fillenbaum and Rapoport (1971) and Rapoport and Fillenbaum (1972) propose three statistics based on graph properties. With clustered data, graphs will become connected only after all links between objects in the same cluster have been inserted (connectivity). The second idea is to compare the observed number of node degrees or cycles with their expected number under the null hypothesis. A component in a graph can be interpreted as a maximum single linkage cluster. The observed number of components can then be compared with the expected number to test for randomness.

4.7 Applications of Cluster Analysis in Social Corpora

Within the field of SNA community detection is an important task. The divisive hierarchical clustering algorithm by Girvan and Newman (2002) for graph-based community detecting has become a popular means to conduct such analysis. This section motivates community

detecting in social networks (see section 4.7.1), illustrates the clustering algorithm by Girvan and Newman (see section 4.7.2) and presents several studies employing this algorithm (see section 4.7.3).

4.7.1 Motivation: Detecting Community Structures

According to Newman (2006) many systems of scientific interest can be represented as networks, or graphs: sets of nodes connected by links. Examples include the worldwide web, metabolic networks, food webs, neural networks, communication and distribution networks as well as economic and social networks. This work focuses on social networks representing real-world systems. Social network graphs exhibit specific patterns that can be used to characterize them and accelerate algorithms. For example, social networks often have clustering tendency and community structure. Furthermore, the degree distribution usually follows a power law (Newman 2003b). Therefore, much research has been done on the graph-based view on social networks using graph theory (Scott 1991; Wasserman and Faust 1994).

To understand the structure of the social network within an organization, one is interested in determining how individuals interact and form groups that, in turn, interact with each other giving rise to higher order groups, i.e. groups of groups (Guimerá et al. 2003: 2). As community structures influence the way information is shared and the way actors behave (see Burt 1992; Burt 2001; Burt 2004) methods of community detection are an important means to understand the global structure of a network and the distribution of actors and activities (Scott 1991). Thus, within the field of SNA detecting and characterizing such community structures has become an important task (see e.g. Barabási and Albert 1999; Albert and Barabási 2002; Flake et al. 2002; Girvan and Newman 2002; Dorogovtsev and Mendes 2003; Newman 2003b; Newman and Girvan 2004; Danon et al. 2005; Newman 2006).

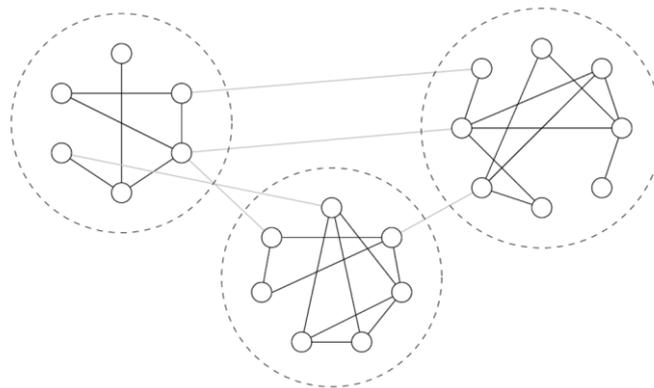


Figure 4-16: A schematic representation of a network with community structure. Communities denoted by dashed circles. Source: Newman and Girvan (2004d)

According to Scott (1991) there are three graph patterns that correspond to cohesive subgroups of actors and play an important role in community detection: *components* (isolated connected subgraphs), *cliques* (complete subgraphs) and *cycles* (paths returning to their point of departure). Based on these initial concepts alternative definitions are for example n-clique, n-clan and k-plex. In general, community structures can be defined as densely connected groups of nodes with only sparser connections between groups (Newman 2006: 8577). This notion of community structures is illustrated in Figure 4-16. In this case there are three

communities, denoted by the dashed circles, which have dense internal links but between which there are only a lower density of external link.

Methods for discovering groups in networks can be roughly divided into two lines of research: *Graph partitioning* has been pursued particularly in computer science (see e.g. Elsner 1997; Fjällström 1998). *Block modeling*, *hierarchical clustering* or *community structure detection* has been pursued by sociologists and more recently physicists and applied mathematicians with applications especially to social and biological networks (see e.g. White et al. 1976; Wasserman and Faust 1994; Newman and Girvan 2004). In contrast to graph partitioning methods where the number and size of the groups are often defined by the experimenter, community structure methods try to find the natural number and size of the groups inherent in the network. Here, even the possible outcome that no good division of the network exists is considered to be of interest when determining the topology of the network (Newman 2006: 8578). Within the second category, graph-theoretic methods make use of the representation of the network as a graph. Many of them use the notion of *edge separators*. There are various methods how an edge or link separator can be defined, e.g. conductance (Sinclair and Jerrum 1989; Kannan et al. 2004), edge betweenness (Girvan and Newman 2002) and modularity (Newman and Girvan 2004), or relative neighborhoods (Flake et al. 2000; Flake et al. 2002; Ino et al. 2005). Using *conductance*, a cut is regarded as dense if the number of links crossing the cut is high relative to the number of links within two sides of the cut. Similar measures are *edge expansion* and *sparsest cut* (Leighton and Rao 1999) as well as *normalized cut* (Shi and Malik 2000). *Modularity* is the number of inter-cluster links compared to the expected value in a random graph with the same degree distribution (see also 4.7.2). *Relative neighborhood* is the ratio of intra-cluster links versus inter-cluster links for each node. The strength of a cluster is then the minimum ratio over all nodes in the cluster. A survey of different graph-based clustering methods can be found in Schaeffer (2007). A comparison of the main principles as well as accuracy and running time of different algorithms for community detection can be found in Newman (2004c) and Danon et al. (2005).

4.7.2 Graph-based Community Detection by Girvan and Newman

Girvan and Newman (2002) propose a hierarchical divisive clustering algorithm which takes advantage of the structural relationship between nodes in a network graph (GN algorithm). The main issue of this algorithm is to find community structures in the network data. Thus, the algorithm divides the nodes of the network into groups within which the connections are dense but between which they are sparse. This approach corresponds to minimizing the within-cluster variance while maximizing the between-cluster variation. It has been extended to weighted links by Newman (2004a). The GN algorithm is widely used in cluster analysis applied to social corpora and according to the number of current publications can be regarded as the most popular clustering algorithm in graph-based social network analysis. For example, Falkowski et al. (2006; 2006) use this clustering approach to track the evolution of communities over time. Other application scenarios have been networks of e-mail messages, social networks of animals, collaborations of jazz musicians, metabolic networks and gene networks (see Newman 2004d). Some of these applications are presented in section 4.7.3.

Table 4-14: Divisive clustering algorithm for finding community structures (GN algorithm). Source: Newman and Girvan (2004: 4)

Divisive Clustering Algorithm for Finding Community Structures	
Step 1	Calculate betweenness scores for all edges in the network.
Step 2	Find the edge with the highest score and remove it from the network.
Step 3	Recalculate betweenness scores for all remaining edges.
Step 4	Repeat from step 2.

The algorithm uses the network structure to produce a nested series of partitions. The data objects are represented as a set of n nodes in a graph with m edges²⁴ established between them due to some kind of exchange or collocation. The general algorithm is given in Table 4-14. The algorithm is usually performed employing the edge betweenness or shortest-path betweenness. Additionally, Newman and Girvan (2004) discuss several metrics to calculate betweenness scores between pairs of nodes based on random walk and resistor networks theory. The edge betweenness is based on the concept of centrality by Freeman (1977) as presented in chapter 1, section 2.1.4.2. Calculating the shortest paths between all pairs of nodes weights each edge with the number of paths that go through it (Newman and Girvan 2004: 5). As outlined in Table 4-15 at first the edge weights have to be calculated.

Table 4-15: Edge weighting algorithm. Source: Newman and Girvan (2004: 5)

Edge Weighting Algorithm	
Step 1	The initial node s is given distance $d_s = 0$ and a weight $w_s = 1$.
Step 2	Every node i adjacent to s is given distance $d_i = d_s + 1 = 1$ and weight $w_i = w_s = 1$.
Step 3	For each node j adjacent to one of those nodes i one of three things are done:
Step 3 (a)	If j has not yet been assigned a distance, it is assigned distance $d_j = d_i + 1$ and weight $w_j = w_s$.
Step 3 (b)	If j has already been assigned a distance and $d_j = d_i + 1$, then the node's weight is increased by w_i , that is $w_j \leftarrow w_j + w_i$.
Step 3 (c)	If j has already been assigned a distance and $d_j < d_i + 1$, do nothing.
Step 4	Repeat from step 3 until no nodes remain that have assigned distances but whose neighbors do not have assigned distances.

Afterwards, the edge betweenness scores can be computed by the algorithm given in Table 4-16. This algorithm has to be repeated for all n source nodes s . Summing up the resulting scores on the edges gives total betweenness for all edges in time $O(mn)$. Repeating this calculation for each edge removed from the network the complete community structure algorithm based on the edge betweenness measure operates in worst-case time $O(m^2n)$. However, one has only to recalculate the betweenness scores of the edges that are in the same subgroup like the edge removed (Newman and Girvan 2004: 6). Meunier and Paugam-Moisy (2006) provide an extension of this algorithm for directed graphs applied to cluster detection in neural networks.

²⁴ Adopting the terminology of the authors the term *edge* is used in this section. In the remaining parts of this work the term *link* is preferred.

Table 4-16. Edge weighting algorithm. Source: Newman and Girvan (2004: 5)

Edge Weighting Algorithm	
Step 1	Find every “leaf” node t , i.e., a node such that no paths from s to other nodes go through t .
Step 2	For each node i neighboring t assign a score to the edge from t to i of w_i/w_t .
Step 3	Starting with the edges that are farthest from the source node s work up towards s . To the edge from node i to node j , with j being farther from s than i , assign a score that is 1 plus the sum of the scores on the neighboring edges immediately below it (i.e., those with which it shares a common node), all multiplied by w_i/w_j .
Step 4	Repeat from step 3 until node s is reached

Figure 4-17 illustrates the entire splitting process (Guimerá et al. 2003): Using a binary tree (see Figure 4-17 b)) communities are depicted as bifurcations (white nodes) and individuals as leaves (black nodes). Each branch in the binary tree corresponds to a community in the original network (see Figure 4-17 a)) and central nodes in a community, such as E , appear as the tips of the branches. At the beginning of the process, the network is a single entity, represented by bifurcation node 1 in the tree. After the removal of the edge, the network is split into two subnetworks, 2 and 3, containing the individual nodes A to D and E to I . Since the two offspring networks have no further internal community structure all the internal edges have the same betweenness. In this case, when iterating the edge removal procedure nodes will be separated randomly one by one by the GN algorithm, in such a way that each community will appear as a branch in the binary tree. As more central nodes will be separated late, this particular characteristic of the GN algorithm can be used with managerial purposes to detect those persons that act like hubs in the organization (Guimerá et al. 2003: 2).

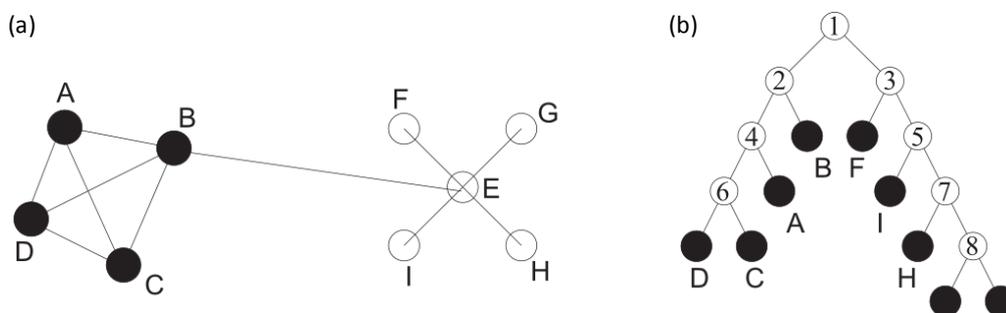


Figure 4-17: Community identification according to the GN algorithm: (a) a simple network with two communities; (b) binary tree generated by the GN algorithm. Source: Guimerá et al. (2003)

To identify the appropriate number of clusters a measure of the quality of a particular division, called *modularity*, is calculated at each level. The measure is based on a measure of assortative mixing by Newman (2003a). Suppose the network consists of n nodes and has been divided into k communities, or clusters. A symmetric $k \times k$ matrix C can be established where each element e_{ij} is the fraction of all edges in the network that link nodes in community i to nodes in community j . The fraction of edges from all communities connecting to community i can be written as $a_i = \sum_{j=1}^n e_{ij}$. The trace of this matrix $trace(C) = \sum_{i=1}^n e_{ii}$ gives the fraction of within-community edges. If the edges between nodes are not influenced by any community structure one would get $e_{ij} = a_i a_j$. Thus, the “non-randomness” of the community structure can be measured as

$$Q = \sum_{i=1}^n (e_{ii} - a_i^2) = Tre - \|e^2\| \quad (\text{Newman and Girvan 2004: 8})$$

Where $\|x\|$ denotes the sum of elements of matrix x . According to Newman and Girvan (2004: 8) “this quantity measures the fraction of the edges in the network that connect vertices of the same type (i.e., within-community edges) minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices”. With $Q = 0$ the number of within-community edges would be equal to a random distribution. Values near $Q = 1$ indicate strong community structures. In practice, the value of Q ranges from 0.3 to 0.7 for comparatively well structured data. The modularity measure always takes all edges into account even if they have been removed from the network during the clustering procedure (Newman and Girvan 2004: 8). Newman (2006) proposes an improved computer algorithm for maximizing the modularity value based on the eigenvalues and eigenvectors of the modularity matrix.

4.7.3 Applications

The applications of the GN algorithm (and its variants²⁵) are numerous. In this section, a selection of these applications is presented to highlight the use and the potentials of the algorithm in real-world social networks and enterprise 2.0 applications:

Ranking and recommendation: Park and Newman (2005) employ the GN algorithm to create a network-based ranking system for US college football.

Patterns of collaboration: Girvan and Newman (2002: 7825) and later Newman (2004b; 2004e) identify key players in (scientific) co-authorship networks.

Patterns of purchase: Applied to a large network of co-purchasing data from the online retailer Amazon.com the GN algorithm discovers communities that correspond to specific topics or genres of books or music (Clauset et al. 2004). The clustering results indicate that the co-purchasing tendencies of Amazon customers are strongly correlated with subject matter.

Patterns of human interaction: In order to investigate the formation and evolution of social networks by self-organized human interaction Guimerá et al. (2003) apply the GN algorithm to a social network built from e-mail communications within the University Rovira i Virgili (URV) in Tarragona, Spain, containing 1,133 users including faculty, researchers, technicians, managers, administrators and graduate students. In contrast to the skewed power-law degree distributions of scale-free networks (Barabási and Bonabeau 2003) that has been demonstrated in e.g. an Instant Messaging Network by Ebel et al. (2002), the cumulative degree of the e-mail network is exponential. However, the truncation of the scale-free behavior in real networks has also been found by e.g. Newman et al. (2002): In real networks, limitations and costs in establishing and maintaining active social acquaintances communications exist. The branching behavior of the e-mail network is compared with a randomly generated network with the same exponential degree distribution (see Figure 4-18).

²⁵ The GN algorithm has high computational costs. There are several variants which try to overcome this problem (see e.g. Wu and Huberman 2003; Newman 2004d; Radicchi et al. 2004) In this work all algorithm related to the original edge betweenness clustering algorithm by Girvan and Newman are subsumed under the term “GN algorithm”.

The absence of branches in Figure 4-18 b) represents the lack of community structure (Guimerá et al. 2003: 2)

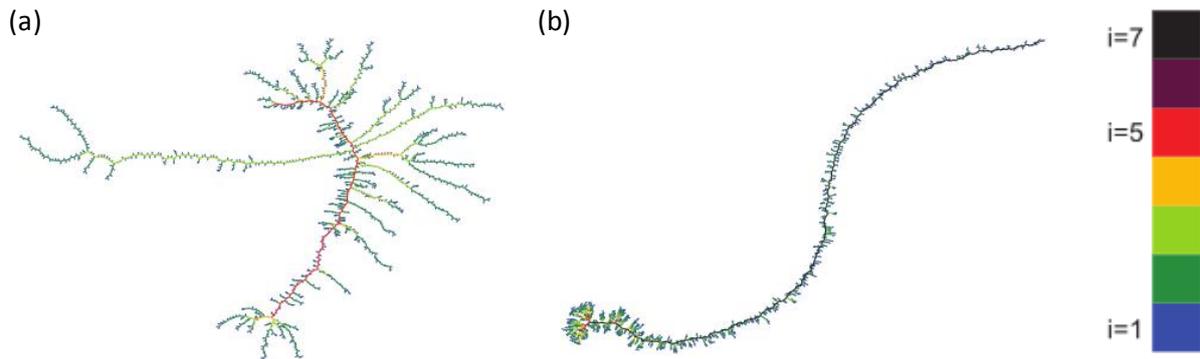


Figure 4-18: Comparison of community structures identified by applying the GN algorithm to the URV e-mail network represented as a binary tree: (a) complex: URV e-mail network, clustering coefficient $c=0.254$ and average shortest path length $d=3.606$; (b) trivial: randomly generated network, $c=0.028$ and average shortest path $d=3.317$. Binary trees without leaf nodes. Coloring due to Horton-Strahler index²⁶. Guimerá et al. (2003: 2)

Word clustering: More general, the GN algorithm can be used for any type of *word clustering* to enhance *automatic thesaurus construction*, *text classification*, and *word sense disambiguation* (see e.g. 2004a).

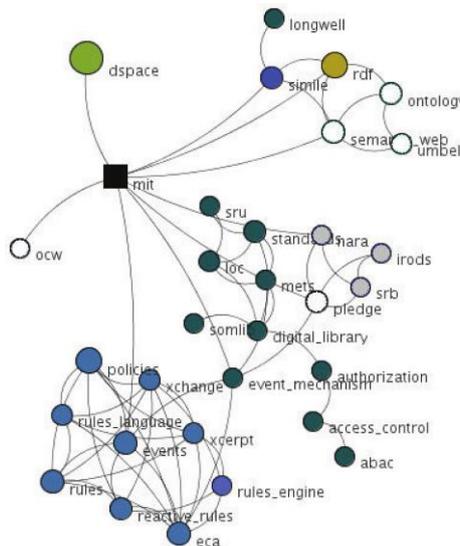


Figure 4-19: Contextualizing Tags in Collaborative Tagging System: Subgraph of internal bookmarking service, Labbies, used by a group of researchers at HPLabs (136 users, 95,155 bookmarks, 8,012 tags, 61,453 co-occurrences); root="mit"; cluster distance <2; minimum tag use >2; NCO²⁷>0.0. Source: Simpson (2008: 223)

Contextualizing tags in collaborative tagging system: Tagging systems like del.icio.us based on folksonomies contain nested groups of tags related to common topics (Heymann and Garcia-Molina 2006; Damme et al. 2007). They suffer from the problem of ambiguity when individuals use the same tag in different contexts. Simpson (2008) and similarly Yeung et al.

²⁶ The Horton-Strahler index was originally introduced for the study of river networks by Horton (1945), and later refined by Strahler (1952). The Leaves of the tree have index $i = 1$, joining branches with the same index gain $i = i + 1$, with different indices the maximum of both branches.

²⁷ Normalized co-occurrence (NCO) of two tags using the Jaccard index.

(2009) recommend the use of the GN algorithm to cluster tags to reveal the social context of a tag automatically without consulting any external resources (see Figure 4-19). Besides the visual inspection, Guimerá et al. (2003) propose the standard deviation of the bifurcation ratios as a quantitative measure of topological self-similarity. This measure tends to 0 when topologically self-similarity holds.

Modeling Enterprise Architecture: In order to reconcile changing business process requirements and information systems, Aier and Schoenherr (2007) propose a method to identify service domain clusters in complex scenarios. Business processes, information systems and information system interfaces are taken into account. They evaluate their method with a case study in a globally operating company.

Automated counter-terrorism SNA: Weinstein et al. (2009) describe an approach of automated modeling, detection and tracking of dynamically changing terrorist groups and their intents based on multimedia data. Their test corpus consists of 60 articles on a recent terrorist event, totaling about 200,000 words. In order to retrieve a social network from these source documents they employ NLP software to identify named entities and possible events and relations within as well as between document co-reference resolution and a final check for spelling and naming conventions is used to identify real-world entities as nodes.

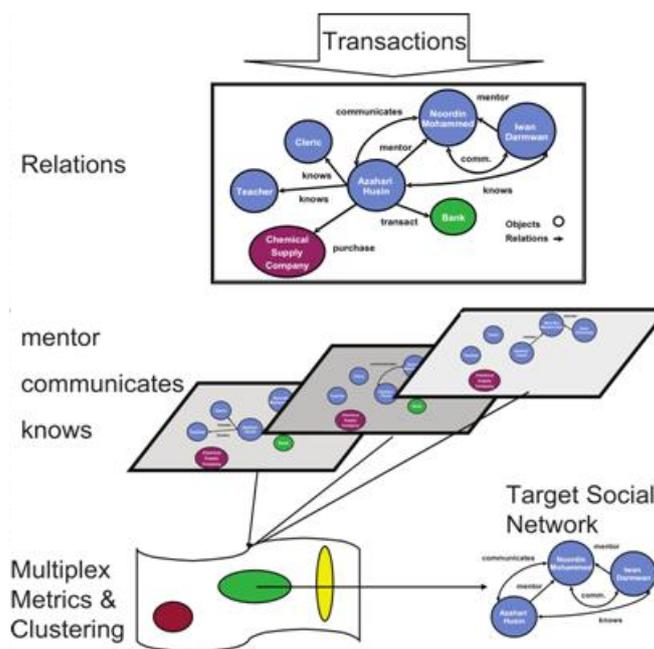


Figure 4-20: Multiplex social network graph generation. Link types are defined by the social network transaction types. Weights are provided by the content extraction algorithm as a confidence measure of the observation. For each link type a subgraph can be retrieved and analyzed. Source: Weinstein et al. (2009)

Finally, link analysis based on relations, events and sentence co-occurrence is performed to retrieve different types of links. The resulting network is used for a multiplex social network graph generation and subgraphs are extracted from the whole data with respect to the type of relation of interest (see Figure 4-20). The GN algorithm then identifies cohesive subgroups.

To evaluate the performance of their method, Weinstein et al. (2009) use a simulated clutter network with an embedded terrorist cell such that the terrorist actors communicate with the clutter actors but no new edges between the terrorist actors are introduced. Knowing the terror

cell actors in advance, after the clustering the community with the highest number of terrorists is selected and the number of terrorists (positive detection) and clutter actors (false alarm) are count so that precision and recall measures can be calculated. For smaller graphs the community detection performs quite well with high precision scores. However, as the graph gets larger, precision scores begin to drop dramatically. Recall remains about the same regardless of graph size. This indicates a fixed percentage of missed terrorist actor nodes.

Collusion in social moderation systems: Lou et al. (2009) propose a collusion-resistant automation scheme for social moderation systems. This kind of mechanism automatically summarizes reports from user moderators and bans misbehaving users or removes inappropriate content. However, some user moderators may collude and dishonestly claim that a user misbehaves (“*bad mouthing*”) to mislead the automatic summarization in order to obtain shared benefits. The scheme detects whether an accusation from a user moderator is fair or malicious based on the structure of mutual accusations of all users in the system. Based on simulation data the authors show that collusion attacks are likely to succeed if an intuitive count-based automation scheme is used. Compared with this simple scheme, the proposed scheme, which is based on the community structure of the user accusation graph detected by the GN algorithm, achieves a decent performance in most scenarios.

Monitoring community dynamics: In Spiliopoulou et al. (2006) clusters are identified in different time periods. A cluster transition at a given time point is a change experienced by a cluster that has been discovered at an earlier time point. This can be an internal transition (change in size, compactness and location of cluster) as well as an external transition (change of between-cluster links). Transitions may lead to changing cluster memberships of the nodes. A cluster survives from one time period to the next time period if the overlap of the cluster members is beyond a certain threshold. The lifetime of cluster is then the number of adjacent time periods a cluster survives. Transition triggers are organizational change, knowledge domain changes, community leadership changes, and external influences and information overload (Falkowski and Bartelheimer 2008).

Based on this work, Falkowski et al. (2006) propose a method called DENGGRAPH applying the GN algorithm to study active participation and community evolution over time (see also (see also Falkowski et al. 2007; Falkowski and Spiliopoulou 2007)). Here, a cluster can survive even if their instances (appearances) are more than one time period apart. The DENGGRAPH method involves two clustering steps (Falkowski et al. 2006): First, the weighted graph of interactions retrieved from the data set is partitioned into time periods and the GN algorithm is applied to each period to identify cohesive subgroups. Subgroup are then tracked over time by measuring the structural equivalence measured by stability, density and cohesion, Euclidean distance, correlation coefficient, and group activity. Two subgroups are similar if the overlap of their members exceeds a given threshold. A graph of subgroups is established to denote similarity between them. Nodes represent subgroups and links represent the weighted similarity between them.

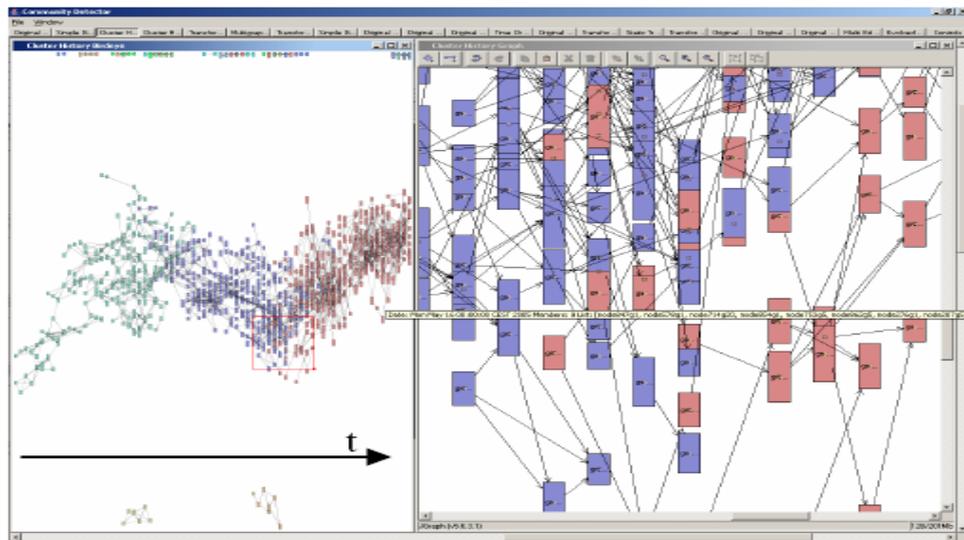


Figure 4-21: Community dynamics with changing members. History view. The revealed communities are displayed in different colors. Data source: Online international student community in the University of Magdeburg (1,000 members; 250,000 guestbook entries over a period of 18 months). Source: Falkowski et al. (2006)

In a second clustering step, the given graph of similar community instances is iteratively clustered to detect communities and breaks in their evolution over time (see Figure 4-21). The appearance of a community in a time period is called community instance. The obtained clustering results can be verified qualitatively in a way that “global events” can be related to the breaks between community clusters.

Falkowski et al. (2008) categorize the different changes in the relationships among actors (transitions) that may lead to community expansion or fusion or to new communities (positive changes) or to a community split or decay (negative changes): Positive changes are creation, absorption and merge, negative changes are removal, reduction, split and move.

Network abstraction: Besides identifying cohesive subgroups for further analysis of their relationships within and between other groups, the GN algorithm can also be used for network abstraction in large data sets. The identified clusters are represented as a single node in the subsequent analysis of the data and the inner structure of the cluster-node is not taken into account or analyzed separately (see e.g. Newman and Girvan 2004).

Community identification in Wikipedia data: Nazir et al. (2009) evaluate the performance of GN algorithm compared to a hierarchical clustering algorithm based on structural equivalence (Wasserman and Faust 1994) for two-mode affiliation network using the Wikipedia data. They conclude that the GN algorithm edge betweenness technique, when applied to two-mode affiliation network is better in terms of modularity value and identifies stronger social communities in terms of social ties. However, it is less time efficient.

5 Content-based Clustering for Knowledge Identification

As community structures influence the way information is shared and the way actors behave (Burt 1992; Burt 2001; Burt 2004), methods of community detection are an important means to understand the global structure of a network and the distribution of actors and activities (Scott 1991). However, these approaches only take the structure of the network into account and therefore the analysis can only access the results of people’s interaction but not their reasons. The reasons are determined by the context of the network which can be approximated by the content objects shared among its participants. In this chapter a new method for knowledge identification in social corpora using content-based clustering is proposed. This method is especially designed to identify clusters of shared experience and knowledge and their influence on the development of the overall network.

This chapter is organized into three parts: First, the method of content-based clustering for knowledge identification in social corpora is explained (see section 5.1). Afterwards, the prototype based on this method is presented (see section 5.2). Finally, this chapter concludes with a case study which applies the method to a corporate e-mail data set using the prototype (see section 5.3).

5.1 Method

In this section the research guideline of the new method for knowledge identification in social corpora using content-based clustering is proposed. Being part of the SNI framework presented in chapter 1 the method contributes to the analytical toolset for analyzing social networks that can be characterized by the three SNI dimensions (see section 2.2.2): level of detail, level of investigation and the temporal dimension.

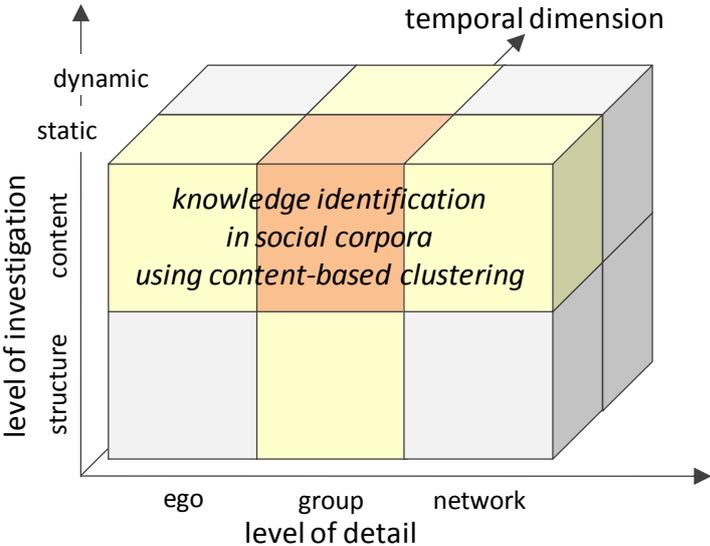


Figure 5-1: Content-based clustering for knowledge identification. Categorization using the SNI dimensions

As highlighted in Figure 5-1 this method is designed as a static content analysis on ego level

(i.e. key players), group level and network level whereas the main focus is on group level including also elements of structural analysis and network dynamics.

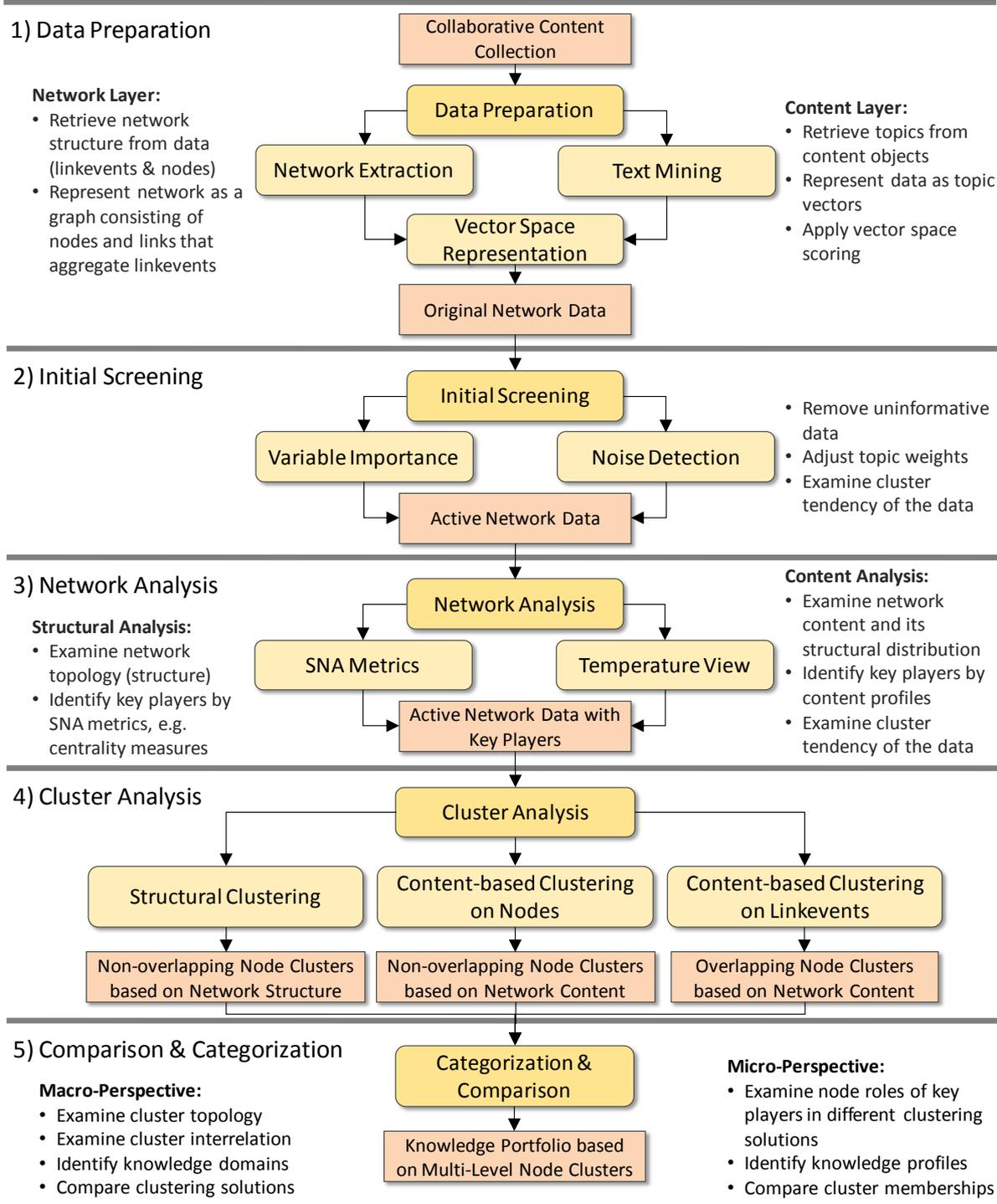


Figure 5-2: Content-based clustering for knowledge identification. Research guideline

This section is organized following the research guideline shown in Figure 5-2 which helps to structure and guide the analysis. It incorporates not only the cluster analysis itself (see section 5.1.4) and the subsequent categorization and comparison of clustering results (see section 5.1.5) but also some preliminary methods of data preparation (see section 5.1.1), initial screening (see section 5.1.2) and network analysis (see section 5.1.3).

5.1.1 Data Preparation

The initial step of the research guideline for content-based clustering on social corpora is to impose some network structure on the “raw” data and to retrieve meaningful topics from its content objects. As illustrated in Figure 2-25 on page 50, chapter 1 (SNI Framework), this method is designed to deal with any type of content objects that can be affiliated with a set of actors. For example, the content objects may be e-mails that are exchanged between a sender and one or more recipients, or presentation slides that are created by several authors, or the protocol of a business meeting containing the list of attendees and the subject of discussion, or a blog with references on other blogs or websites. In the context of this work this data is termed *collaborative content object*, or just *collaborative content* or *content object*. Collaborative content can then be defined as any kind of interaction (e.g. co-work, co-authorship, exchange, usage) on or with content objects. The whole set of collaborative content objects are called *collaborative content collection*. Sometimes, the collaborative content is not automatically recorded. Thus, one has to use interviews or questionnaires to capture the data (see chapter 1 (SNI Framework), section 2.2.3.3). The process of data preparation covers network extraction (see section 5.1.1.1), text mining (see section 5.1.1.2) and vector space representation (see section 5.1.1.3).

5.1.1.1 Network Extraction

The variety of collaborative content creation processes within an organization provide rich data sources for network as well as content analysis. As illustrated in Figure 5-3 there are two perspectives on a collaborative content collection which allow analyzing the data on two interrelated layers: The *content layer* contains the collaborative content objects whereas the *network layer* contains the network structure consisting of nodes and links that can be extracted from the data. The two layers are linked by assigning the collaborative content objects to the nodes that are affiliated with them.

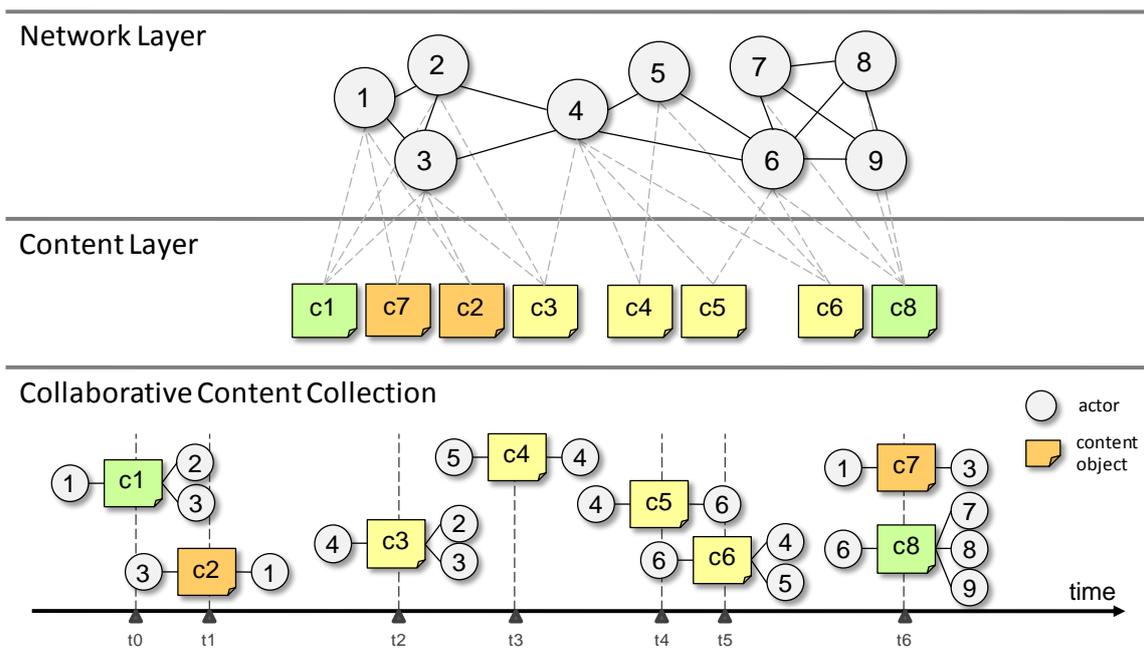


Figure 5-3: Network and content extraction from collaborative content creation processes

Using the SNI data model (see section 2.2.1 for more details) collaborative content objects are

regarded as timed linkevents and the nodes affiliated with them as actors. One or more linkevents that involve the same two nodes are aggregated as links. Links and nodes can be visually represented as an undirected graph. During the data preparation process such a network structure is imposed on the data.

5.1.1.2 Text Mining

In order to obtain meaningful topics for content-based clustering a text mining process is applied on the linkevents of the data set. In general, a text mining process includes linguistic preprocessing, text transformation for feature generation (tokenization) and feature selection (see chapter 3 (Text Mining), sections 3.2 to 3.3).

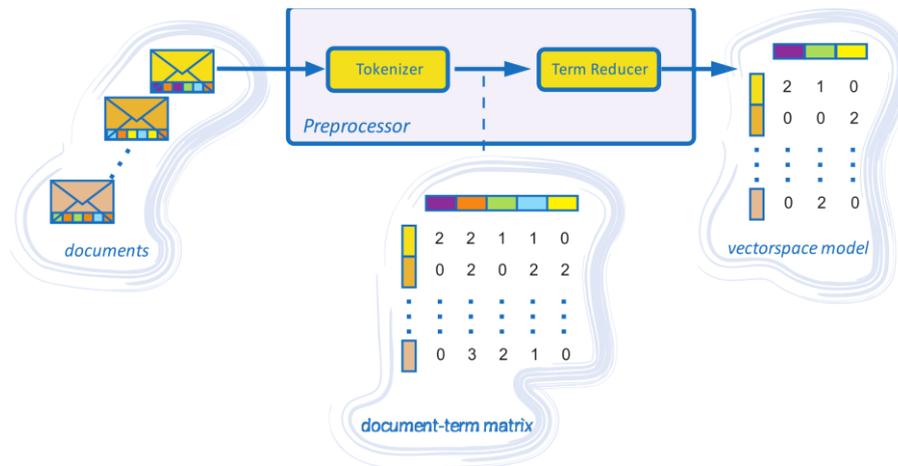


Figure 5-4: Components of the ContentMiner text mining software

The ContentMiner is a text mining software based on the SNI data model developed by the IKM Research group. The text mining process provided by the ContentMiner consists of automated linguistic preprocessing and feature generation by the Tokenizer component and automated feature selection by the TermReducer component. It transforms the documents, or linkevents, into topic vectors using the vector space model (see Figure 5-4).

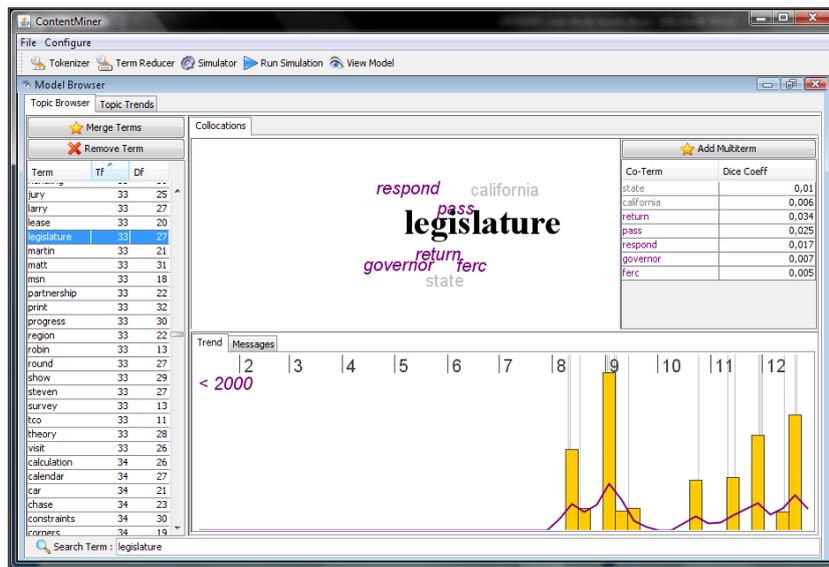


Figure 5-5: ContentMiner. Example of the manual data inspection by the ModelViewer component

The ModelViewer component allows further manual feature selection (term removal and

multi-term generation) by visual inspection of the topics, their cumulated term frequency and document frequencies, collocation analyses with co-occurring topics based on the Dice coefficient (see section 4.3.4) and their temporal distribution in the data set (see Figure 5-5). Several text mining methods are available for the Tokenizer and TermReducer that can be combined and configured according to the type of data and the problem at hand. For example, removing many low-frequency terms will improve the overall content similarity and tends to result in few but large content cluster whereas removing many high-frequency terms will reduce the overall content similarity and tends to result in many but small content clusters. Both variants and their combination might be desirable. The text mining process is optional as the data may already provide useful tags, e.g. documents annotated with keywords by the authors. These tags can be used as topics represented each content object after the text mining process.

5.1.1.3 Vector Space Representation

The text mining process assigns a list of meaningful topics to each object in the collaborative content collection, whereas the network extraction process transforms these collaborative content objects into linkevents. Both processes do not depend on each other. However, combining their results allows representing the different network elements as topic vectors using the vector space model described in section 3.3.2.

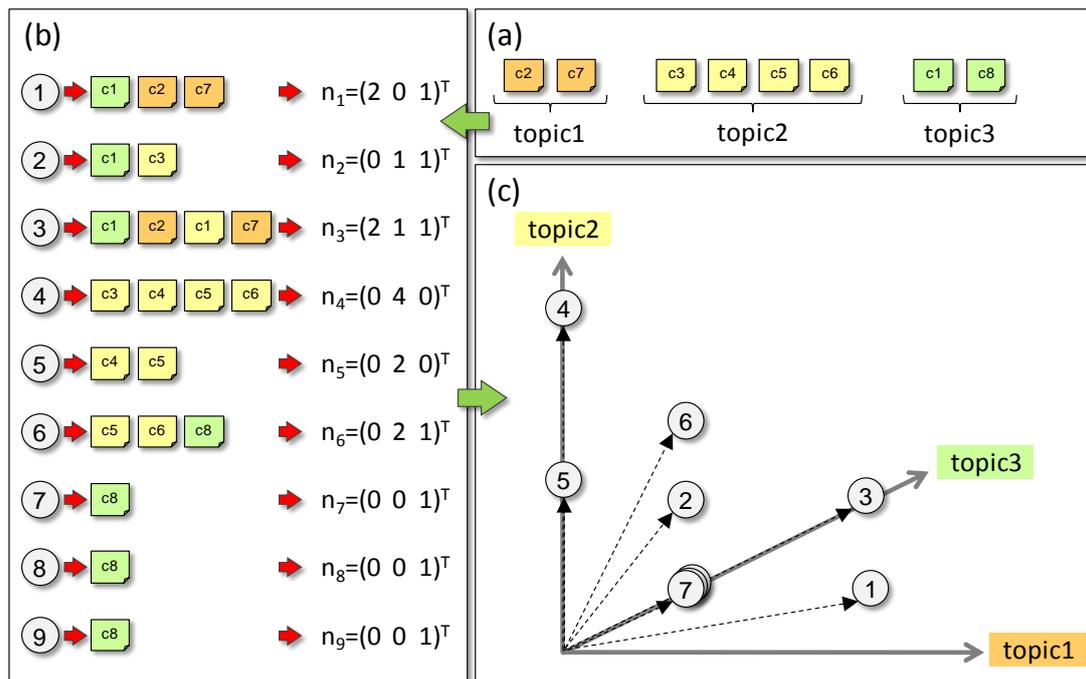


Figure 5-6: Representation of nodes in the vector space. Example: (a) extracting topics from collaborative content objects; (b) assigning content objects to nodes and retrieving topic vectors; (c) representing nodes in a three-dimensional vector space

Naturally, if the content objects are represented by linkevents in the network each linkevent can also be described by a topic vector corresponding to the list of topics that have been retrieved from its content. Different weighting schemes can be applied to the data to calculate the topic weights (see section 3.4.3). Given a representation of the topics in a vector space (see section 3.3.2) the weighting schemes are also called vector space scoring. Furthermore, the nodes and links in the network have an expression as topic vectors: The nodes are

affiliated with the linkevents as they represent the actors that are involved in the creation or exchange of the collaborative content objects and the links aggregate one or more linkevents. As several linkevents belong to one node or one link a function of topic weight aggregation has to be used to determine the weights of each topic in the topic vectors of these network elements. This function is also called weighting scheme. For example, for each topic in the topic vector of a certain node the average of its weights in each linkevent that belongs to this node can be used. Other node or link weighting schemes can be the number of linkevents it occurs in, the minimum or maximum weight, or the sum of all weights. In Figure 5-6 the process of representing a node a as topic vector is depicted based on the collaborative content collection already shown in Figure 5-3, section 5.1.1.1. In Figure 5-6 a) each content object is represented by only one topic, i.e. topic1, topic2 or topic3. In Figure 5-6 b) these content objects are assigned to the nodes that are affiliated with them. Each node can then be expressed by a topic vector using the sum of all linkevent weights as topic weights. As the dictionary of this exemplary data set consists of only three topics the nodes can be represented as vectors in a three-dimensional vector space.

5.1.2 Initial Screening

Initial screening is an important step in the cluster analysis process described in chapter 4 (Cluster Analysis), section 4.1.3. In general, in this step the raw data has to be adapted for formal analysis. Besides some normalization the data should be reviewed in a rough manner. Outliers and features with no contribution to the analysis can be removed. In the context of this work, the initial screening process covers several techniques for changing the weights of the topics (variable importance) associated with nodes and linkevents and detecting noise in the data set. Noise detection will result into reducing the data set by removing linkevents, links and nodes. Changing the importance of variables (see section 5.1.2.1) as well as detecting noise (see section 5.1.2.2) will affect the clustering tendency of the data and the clustering solutions obtained by the content-based clustering procedure.

5.1.2.1 Variable Importance

In cluster analysis, the importance of a variable is determined by its weight, i.e. a numerical value. Using topics as variables the importance of a topic may most simply be its number of occurrences in the content object. More advanced, the importance of the topic can be influenced by changing and adjusting this value using standardization or weighting techniques. Changing the variable importance may affect the results from outlier detection as linkevents and node might become more or less similar.

5.1.2.1.1 Standardization of Variables

Although the purpose and the requirements of the analysis and the type of the data may require a standardization of the data (see section 4.2.2.2), there is some controversy about the benefits and effect of standardization on the results of cluster analysis (Aldenderfer and Blashfield 1984: 20; Kaufman and Rousseeuw 1990: 10). Standardized data are unitless and thus independent from the choice of measurement units (Kaufman and Rousseeuw 1990: 6). If the variables have an absolute meaning standardization should not be used. Additionally, standardization will reduce the differences between objects and thus dampen the clustering structure inherent in the data (Everitt 1980; Jain and Dubes 1988: 24; Kaufman and

Rousseeuw 1990: 10). Using the new method of content-based clustering for knowledge identification, standardization techniques can only be applied to linkevents (see section 5.2.4.1).

5.1.2.1.2 Weighting of Variables

Weighting means to change the impact of a variable on the cluster analysis from what has been recorded by the values of this variable to what the researcher believes to be more appropriate due to some a priori knowledge (see e.g. Williams 1971; Aldenderfer and Blashfield 1984: 21). The concept of weighting variables is related to the concept of standardization and is equally controversially discussed (see section 4.2.2.3). There are two possible reasons for weighting variables (Kaufman and Rousseeuw 1990: 14). Either it turns out that a wrong measurement unit has been used and thus the weights of all variables measured with this unit have to be altered, or some variables are regarded as more important than other variables with the same unit. In the first case it is preferable to standardize these variables whereas in the second case different weights can be imposed on the data.

In text mining, some of the earliest applications of term weighting are reported by Luhn (1957; 1958) with special focus on the importance of medium-frequency terms (see section 4.2.2.3). Given a representation of the topics in a vector space (see section 3.3.2), the weighting schemes are also called vector space scoring. There are several variations from vector space scoring methods which apply different weights to term occurrences (Manning et al. 2008: 116). An overview of the principle weighting schemes in use for a document vector is presented in Table 3-5 in section 3.4.3.2 together with a mnemonic for representing specific combinations of weights. The weighting scheme can then be represented as a triplet of mnemonics. The first letter in each triplet refers to term frequency component of the weighting, the second letter the document frequency component and the third letter the form of normalization used.

In the context of this work weighting schemes can be applied to nodes and linkevents which will have an effect on the clustering results as well as links and the entire network which will only make a difference in data inspection (see section 5.3.3.1). Weighting schemes on nodes and links are also called *topic weight aggregation functions* (see section 5.1.1.3).

5.1.2.2 Noise Detection

In cluster analysis, noise refers to missing values and uninformative variables as well as objects that are too far removed from the others measured by some proximity index (see chapter 4 (Cluster Analysis), section 4.2.3).

5.1.2.2.1 Missing Data

If not all measurements are available in the recorded data set the data matrix contains missing values (Kaufman and Rousseeuw 1990: 14). There are several reasons for missing values (see section 4.2.3.1) but when dealing with topics derived from text or documents most often there is no value which could be measured (“inapplicable question”). That means the keyword simply does not occur in the text or it is already wrongly recorded in the original data, e.g. misspelled keywords. Missing values can be encoded in the data matrix by some indicator and recognized by the clustering procedure (Kaufman and Rousseeuw 1990: 14). In general, there

exist three approaches how to deal with missing data: eliminate part of the data, estimate the missing values, or compute an estimated distance between two data objects with missing values. Dixon (1979) recommends the third method as the best overall. However, when clustering topics one will be limited to apply some spelling correction techniques during the text mining process.

5.1.2.2.2 Outlier Detection

A pattern that is far removed from the rest of the data is regarded as outlier (Jain and Dubes 1988: 98). It might have been included by error in the measurement process or error in data coding. Forcing an outlier to belong to a cluster seriously disturbs the shape of that clusters and will have a strong influence on the whole clustering structure (see section 4.2.3.3). Outliers can be identified by using a threshold. If the similarities of an object to all other objects fall below this threshold the object can be regarded as an outlier and therefore removed from the data set. Some clustering algorithms also treat small clusters as outliers. Therefore, singleton clusters and too small clusters can also identified by a threshold of the minimum number of objects in a cluster. Outlier detection will result in removing objects from the data set. In the context of this work, this can affect linkevents, links and nodes (see section 5.2.4.3).

5.1.2.2.3 Dealing with Uninformative Variables

Uninformative variables are recorded in the data set but contain no relevant information for the task performed from the cluster analysis (see section 4.2.3.2). According to Kaufman and Rousseeuw (1990: 14) these variables are “worse than useless” as they flatten the effect of clustering tendencies in the data. In the worst case no meaningful clustering can be obtained. How to identify and deal with uninformative data is also an important task during the text mining process, e.g. feature generation and feature selection, e.g. removing stop words. When clustering keywords the simplest case of uninformative data are those data objects that do not contain any topic. Dealing with uninformative data will result in removing objects from the data set. In the context of this work, this can affect linkevents, links and nodes (see section 5.2.4.3).

5.1.3 Network Analysis

After the preparation of the data and the initial screening it is recommend to perform a preliminary network analysis before clustering the data to examine the network structure and identify important key players and their roles in the entire network. Therefore, SNA metrics can be calculated on node, link and network level (see section 5.1.3.1). Furthermore, the temperature view presented in this section helps to detect important key players based on content similarity and network structure and determine the clustering tendency of the data. This novel approach is described in section 5.1.3.2.

5.1.3.1 SNA Metrics

Calculating SNA metrics on node, link or network level helps to gain a first understanding of the network and to select important key players for further analysis. An overview of suitable metrics is given in chapter 1 (SNI Framework), section 2.1.4. For example, the three centrality metrics degree centrality, betweenness centrality and closeness centrality help to determine

the activity and popularity of a node, the independence and responsibility as well as the efficiency and influence. The choice of the metrics depends on the data set and the problem at hand.

5.1.3.2 Temperature View

The *temperature view*, or *temperature graphview*, allows a visual inspection of the distribution of content similarities between nodes. This approach can be used to gain a first understanding of the network, to determine the clustering tendency of its content and to select important key players for further analysis. It is also useful to analyze and compare the effect of the different methods of noise detection and variable importance on the content similarity of the network and its elements. The concept of the temperature view is based on the content profiles of the nodes in the network which are mapped to the network structure. Therefore, the temperature view combines content analysis with structural analysis.

The *content profile* of a node is the visualization of the pairwise content similarities calculated between the focal node and all other nodes in the network based on the node weighting scheme and the chosen proximity index. The overview of all content profiles in a network as illustrated in Figure 5-8 a) on page 195 provides a first understanding of the content similarities and therefore the clustering tendency of the data (see chapter 4 (Cluster Analysis), section 4.6.6). The concept of the temperature view based on content profiles of nodes can be illustrated using the exemplary data presented in Figure 5-3, section 5.1.1.1, and Figure 5-6, section 5.1.1.3. The pairwise content similarities are calculated using the cosine similarity (see chapter 4 (Cluster Analysis), section 4.3.2). The results for the example are given in Table 5-1. Based on these values the content profile of each node is established.

Table 5-1: Pairwise content similarities using the cosine similarity

		Node								
		1	2	3	4	5	6	7	8	9
Node	1	-	0.3	0.9	0.0	0.0	0.2	0.4	0.4	0.4
	2	0.3	-	0.6	0.7	0.7	0.9	0.7	0.7	0.7
	3	0.9	0.6	-	0.4	0.4	0.5	0.4	0.4	0.4
	4	0.0	0.7	0.4	-	1.0	0.9	0.0	0.0	0.0
	5	0.0	0.7	0.4	1.0	-	0.9	0.0	0.0	0.0
	6	0.2	0.9	0.5	0.9	0.9	-	0.4	0.4	0.4
	7	0.4	0.7	0.4	0.0	0.0	0.4	-	1.0	1.0
	8	0.4	0.7	0.4	0.0	0.0	0.4	1.0	-	1.0
	9	0.4	0.7	0.4	0.0	0.0	0.4	1.0	1.0	-

There are eleven categories of content similarities, ranging from 0% to 100% similarity. For each node the number of other nodes is calculated with a content similarity larger than the last category but not more than the present category. For example, for each node the 30% category contains the number of other nodes in the network with a pairwise content similarity $20\% < sim \leq 30\%$. These *temperature values* are the numeric expression of the content profiles. The resulting values of the example are given in Table 5-2. Each category is depicted by a color ranging from white over yellow to red representing increasing similarity values.

Table 5-2: Temperature values. Numeric expression of content profiles

	Content Similarity										
	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	2	0	1	1	3	0	0	0	0	1	0
2	0	0	0	1	0	0	1	5	0	1	0
3	0	0	0	0	5	1	1	0	0	1	0
4	4	0	0	0	1	0	0	1	0	1	1
Node 5	4	0	0	0	1	0	0	1	0	1	1
6	0	0	1	0	3	1	0	0	0	3	0
7	2	0	0	0	3	0	0	1	0	0	2
8	2	0	0	0	3	0	0	1	0	0	2
9	2	0	0	0	3	0	0	1	0	0	2

Based on the data in Table 5-2 the percentage of other nodes that belong to a category can be calculated for each node. Figure 5-7 provides the content profile of node 1 as an example.

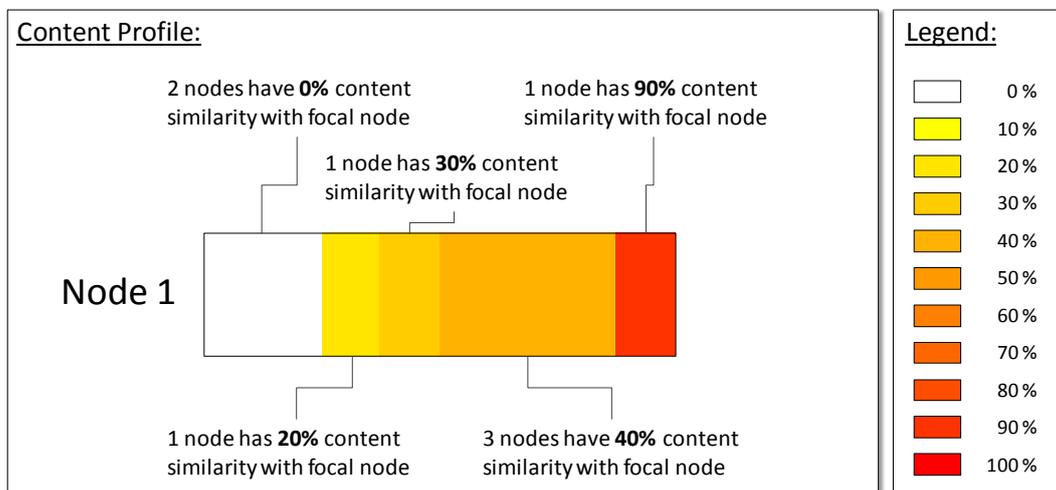


Figure 5-7: Content profile. Example

Table 5-12 in section 5.3.4.2 demonstrates the effect of different weighting schemes and proximity metrics on the content profiles of a network. Obviously, the choice of these parameters has a strong influence on the similarities. In general, the distribution of content similarities depends on the number of nodes, linkevents and topics as well as the distribution of linkevents and topics among nodes. For example, the average node weighting scheme will produce content similarities similar to those obtained by the sum node weighting scheme if only a few linkevents with rather different topics are assigned to the nodes. Although some of these parameters are highly depending on the data itself, it is recommended to use the cosine similarity as proximity index.

Although the temperature view of the content profiles allows examining the general distribution of content similarities among nodes it does not account for its structural distribution. Therefore, the *temperature graphview* maps the content profile of a selected node to the graph of the network. This visualization helps to identify whether a content profile accounts for a local clustering of similar nodes or an even distribution throughout the entire network. The combination of structural distribution of node similarities in the graph and the

corresponding content profile helps to gain a deep understanding of both the structure and the content of the data.

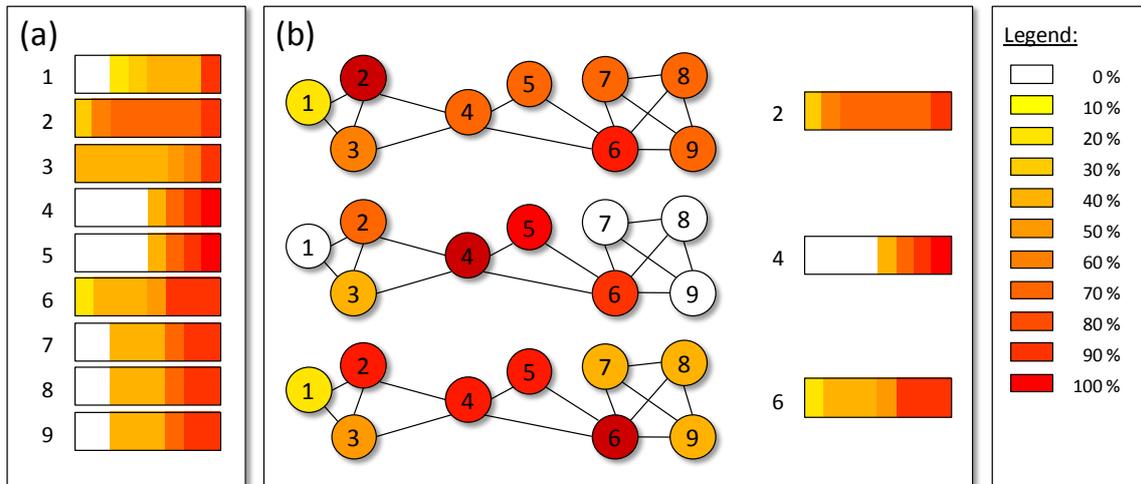


Figure 5-8: Visual representation of content profiles and temperature graphview: (a) overview of content profiles; (b) temperature graphviews for three selected nodes

Using the data from Table 5-2, the overview of all content profiles and the temperature graphviews of three selected nodes from the example are given in Figure 5-8. The focal node of a temperature graphview is depicted as dark red node. When inspecting the content profiles and the graphviews one can identify groups of nodes with similar content profiles (e.g. node 4 and node 5; node 7, node 8 and node 9) and examine the structural distribution of these similarities. For example, the most similar node of focal node 2 is node 6 which is not directly linked with the node but most of the other nodes have also a high similarity with the focal node, whereas the most similar nodes of node 4 are node 5 and node 6 which are directly linked with node 4 but the other nodes have zero or low similarity with the focal node. Furthermore, a large number of content profiles with many low similarities and a few high similarities (e.g. node 1) will indicate a good clustering tendency of the data whereas many content profiles with almost equal similarity values (e.g. node 2 or node 3) will indicate that the data should be regarded as only one large group of nodes.

5.1.4 Cluster Analysis

The core component of the new method for knowledge identification in social corpora proposed in this chapter is the content-based clustering method which can be applied on nodes as well as linkevents. In this section both variants of content-based clustering and their comparison to the commonly used structural clustering are explained. The content-based clustering method is not designed as a substitute to structural clustering but as an additional analytical device to investigate on the data and identify clusters of knowledge. The temperature view presented in section 5.1.3.2 combines content and structure analysis on ego level by mapping content similarities to the structure of the network. Content-based clustering extends the idea of the temperature view to group level analysis. In Figure 5-9 the three clustering strategies are illustrated using the exemplary data from Figure 5-3, section 5.1.1.1, and Figure 5-6. They are described in the following three sections 5.1.4.1, 5.1.4.2 and 5.1.4.3.

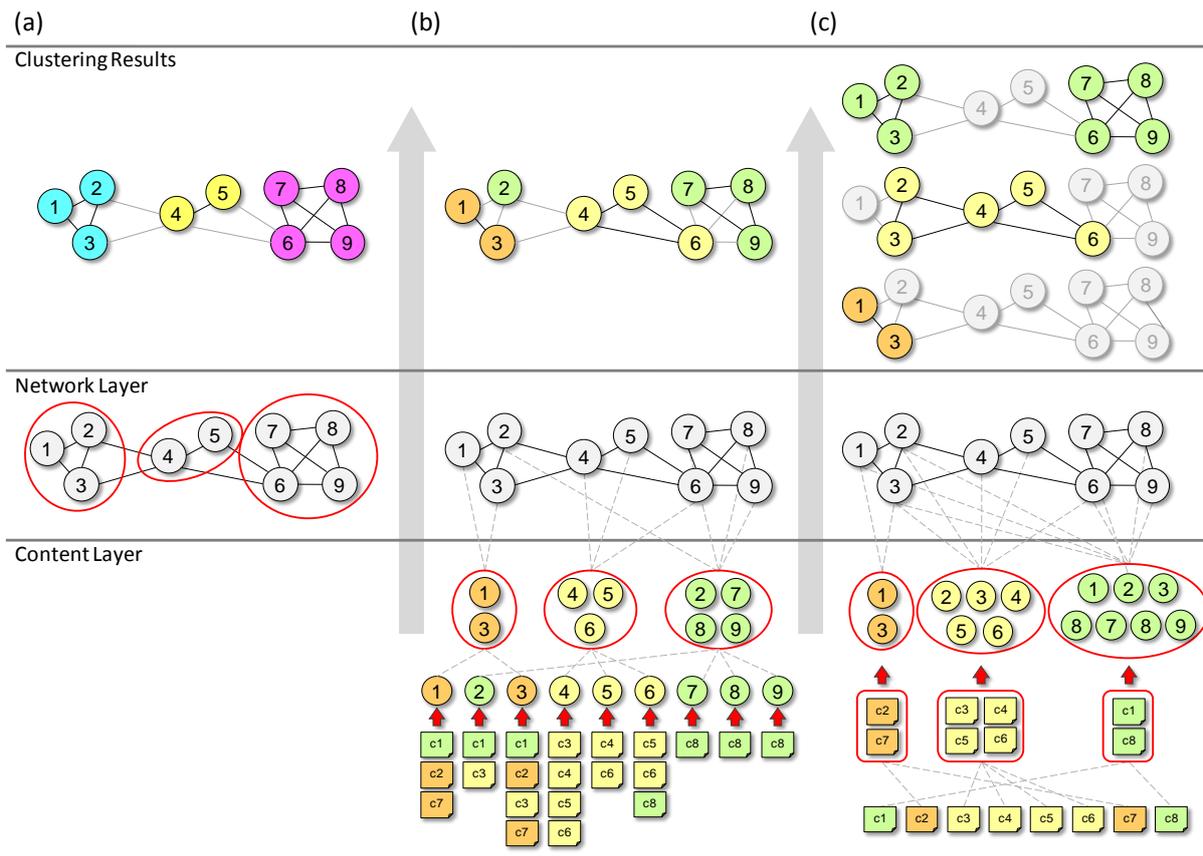


Figure 5-9: Comparison of clustering strategies: (a) structural clustering; (b) content-based clustering on nodes; (c) content-based clustering on linkevents

Finally, this section concludes by providing visual and numerical techniques to compare clusters from all three clustering methods for multi-level cluster analysis and validation (see section 5.1.5.3).

5.1.4.1 Structural Clustering

In general, structural clusters or community structures can be defined as densely connected groups of nodes, with only sparse connections between the different groups (Newman 2006: 8577). There are several approaches how to detect such community structures. An overview is given in chapter 4 (Cluster Analysis), section 4.7.1. In the context of this work the popular edge betweenness clustering algorithm by Girvan and Newman (2002) is used for structural clustering. It is a hierarchical divisive clustering algorithm which takes advantage of the structural relationships between nodes in a network graph to produce a nested series of partitions. The algorithm successively splits up the network structure using the edge betweenness centrality as clustering criterion. To identify the appropriate number of clusters a measure of the quality of a particular partition, called modularity, is calculated at each level. A high modularity value indicates a strong community structure. For more details, see chapter 4 (Cluster Analysis), section 4.7.2.

In Figure 5-9 a) the clustering results for structural clustering using the exemplary data from Figure 5-3, section 5.1.1.1, is illustrated. Obviously, this approach does not take any content information into account to establish the clusters. The resulting clustering solution consists of

non-overlapping node clusters. Each cluster is either a singleton cluster or a fully connected component of several nodes.

In summary, the structural clusters group together those nodes which have stronger relations among each other compared to the rest of the network. They can be compared to the results from the analysis of the entire network in the previous step of the research guideline to examine the difference between embeddedness and interaction of a node with the entire network versus embeddedness and interaction only with members of the densely connected subgroup.

5.1.4.2 Content-based Clustering on Nodes

In contrast to the structural clustering approach explained in the previous section, content-based clustering on nodes does not establish the clustering solution on the network layer but on the content layer taking the content similarities of the data into account. The algorithm was first published in Bobrik and Trier (2009). As illustrated in Figure 5-6 nodes can be represented as topic vectors in the vector space by aggregating the content objects, or linkevents, that are affiliated with them using a suitable node weighting scheme. Comparing the content similarities of the topic vectors a clustering solution can be obtained grouping together those nodes with high content similarities. The cluster analysis can be performed employing a hierarchical or partitioning clustering algorithm. The choice of the clustering parameters depends on the data itself and the problem at hand, i.e. number of clusters, proximity metrics etc. Some recommendations are given in section 5.2.6.

Table 5-3: General algorithm of content-based clustering on nodes. Based on: Bobrik and Trier (2009)

General Algorithm of Content-based Clustering on Nodes	
Prerequisite	Let the content of a linkevent be represented by a topic vector. Let the content collection of a node be determined by the linkevents that are affiliated with the node. Let the content collection of a node be represented by a topic vector using a node weighting scheme to aggregate the topic vectors of its content objects (linkevents). Let the similarity of two nodes be determined by calculating the similarity of their topic vectors.
Step 1	Apply a hierarchical or partitioning clustering algorithm from Table 4-6 or Table 4-9 on the nodes to obtain non-overlapping node clusters that group together similar nodes.
Step 2	Map the node clusters to the network structure to determine which links belong to the same cluster (intra-cluster links) or connect different clusters (inter-cluster links).
Step 3	Calculate node metrics on group level to determine the role of a node in its cluster.

This approach does not include any structural information into the clustering decision. In a second step, the clustering solution is mapped to the network layer to determine which link belongs to a cluster. For each node SNA metrics can be calculated on group level to examine its role in the content cluster. Here, the type of interaction of an actor in his cluster provides useful insights. For example, in an e-mail corpus there are two types of interaction on node and linkevent level: senders or linkevents sent and recipients or linkevents received. The type and share of linkevents of a node on all nodes of a cluster can be related to its influence and prominence in this cluster. A pseudo code description of the general algorithm of content-based clustering on nodes is given in Table 5-3.

In Figure 5-9 b) the clustering result for content-based clustering on nodes using the exemplary data from Figure 5-3, section 5.1.1.1, and Figure 5-6 is illustrated. The resulting clustering solution consists of non-overlapping node clusters. In contrast to the structural clustering approach the clusters can consist of more than one component, i.e. a node may have no interaction with the other members of the cluster (see Figure 5-10 b)).

In summary, the node-based content clusters group together those nodes with similar content objects representing a shared context or similar knowledge. They can be compared to the results from the analysis of the entire network in the previous step of the research guideline to examine the difference between embeddedness and interaction of a node with the entire network versus embeddedness and interaction only within its cluster expressing its domain of knowledge. In contrast to the structural clustering results the member of a cluster may be less connected or even unconnected comparing their roles and positions in the entire network. Thus, this approach helps to determine the flow of information and the distribution of knowledge within each knowledge domain. Examining the links between clusters one can also draw some conclusion about how these domain of knowledge are interrelated.

5.1.4.3 Content-based Clustering on Linkevents

In contrast to the structural clustering approach explained in section 5.1.4.1, content-based clustering on linkevents does not establish the clustering solution on the network layer but on the content layer taking the content similarities of the data into account. After the text mining process the content objects of the data, i.e. the linkevents in the network, can be represented as topic vectors in the vector space. Comparing the content similarities of the topic vectors a clustering solution can be obtained grouping together those linkevents with high content similarities. The cluster analysis can be performed employing a hierarchical or partitioning clustering algorithm. The choice of the clustering parameters depends on the data itself and the problem at hand, i.e. number of clusters, proximity metrics etc. Some recommendations are given in section 5.2.6. This approach does not include any structural information into the clustering decision. In contrast to content-based clustering on nodes, this approach is directly applied to the content objects of the data instead of their aggregation on node level. In a second step, the clustering solution of the linkevents is mapped to the nodes that are affiliated with them resulting into overlapping clusters of knowledge. Still, no structural information is taken into account. In a third step, the node clusters are mapped to the network layer to determine which link belongs to a cluster. Although overlapping node-clusters are obtained the clustering procedure itself identifies non-overlapping linkevent-clusters. For each node SNA metrics can be calculated on group level to examine its role in the multi-level content clusters. Here, the type of interaction of an actor in his clusters provides useful insights. For example, in an e-mail corpus there are two types of interaction on node and linkevent level: senders or linkevents sent and recipients or linkevents received. The relation between linkevents sent and received of a node can vary from cluster to cluster indicating his type of activity in each cluster. The share of linkevents of a node on all nodes of a cluster can be related to his influence and prominence in this cluster. Furthermore, the distribution of linkevents of an actor in the different clusters helps to determine the strength of his cluster memberships and how much he is involved in the domain of knowledge which is represented by each cluster. Thus, a detailed knowledge profile can be retrieved. A pseudo code

description of the general algorithm of content-based clustering on linkevents is given in Table 5-4.

Table 5-4: General algorithm of content-based clustering on linkevents

General Algorithm of Content-based Clustering on Linkevents	
Prerequisite	Let the content of a linkevent be represented by a topic vector. Let the similarity of two linkevents be determined by calculating the similarity of their topic vectors.
Step 1	Apply a hierarchical or partitioning clustering algorithm from Table 4-6 or Table 4-9 on the linkevents to obtain non-overlapping linkevent clusters that group together similar linkevents.
Step 2	Retrieve overlapping node clusters from linkevent clusters by grouping together nodes that are affiliated with linkevents from the same linkevent cluster.
Step 3	Map node clusters to network structure to determine which links belong to the same cluster (intra-cluster links) or connect different clusters (inter-cluster links).
Step4	Calculate node metrics on group level to determine the different roles of a node in its clusters.

In Figure 5-9 c) the clustering result for content-based clustering on linkevents using the exemplary data from Figure 5-3, section 5.1.1.1, and Figure 5-6 is illustrated. The resulting clustering solution consists of overlapping node clusters. In contrast to the structural clustering approach the clusters can consist of more than one component, i.e. a node may have no interaction with the other members of the cluster, and in contrast to content-based clustering on nodes each node can belong to more than one cluster (see Figure 5-10 c)). For example, in Figure 5-9 c) and Figure 5-10 c) 50% of the linkevents of node 2 belong to the first cluster (green) and the other 50% belong to the second cluster (yellow). In the first cluster node 2 is involved in half of the content objects whereas in the second clusters he is only involved in 25% of the content objects. In both cases, node 2 is the recipient of the linkevent. The first cluster consists of two rather small unconnected components. The second cluster is also rather small but fully connected. In both clusters the node has the same number of direct contacts but due to the shape of the cluster he has more indirect influence. Mostly due to the size of both clusters, the node takes no prominent role in terms of centrality. Thus, this node has an evenly distributed knowledge profile but more direct influence in the first cluster due to its share on the intra-cluster linkevents. He takes no active part in any of his knowledge clusters.

In summary, the linkevent-based content clusters group together those nodes with similar content objects representing a shared context or similar knowledge. They can be compared to the results from the analysis of the entire network in the previous step of the research guideline to examine the difference between embeddedness and interaction of a node with the entire network versus embeddedness and interaction only within its domain of knowledge. In contrast to the structural clustering results the member of a cluster may be less connected or even unconnected comparing their roles and positions in the entire network. In contrast to the results from content-based clustering on nodes each node can belong to more than one knowledge cluster taking different domains of knowledge and experience into account. Thus, this approach helps to determine the flow of information and the distribution of knowledge within each domain. Examining the links between clusters one can also draw some conclusion

about how these domain of knowledge are interrelated. Watts et al (2002) provide a study about searchability and multi-level social distances in social networks (see section 2.1.3.5). The overlapping clusters obtained by content-based clustering on linkevents can be related to their different categories of social interaction: Even if two actors are not directly connected by shared experience or knowledge they may be reachable through shared acquaintainships established over their different fields of knowledge expressed by their overlapping node-clusters.

5.1.5 Categorization & Comparison

The final step of the research guideline for knowledge identification in social corpora using content-based clustering is the thorough analysis of the clustering results by categorizing the different roles a node can obtain, identifying knowledge domains and knowledge profiles and comparing the cluster memberships of a node using different clustering strategies.

Figure 5-9 illustrates the cluster memberships of node 2 in different clustering solutions using the exemplary data from Figure 5-3, section 5.1.1.1, Figure 5-6 and Figure 5-9. For example, the node might be peripheral or even isolated in one cluster (e.g. Figure 5-9 b)) but well connected in another cluster (e.g. Figure 5-9 c), yellow cluster). Therefore, each node can obtain different roles in different contexts.

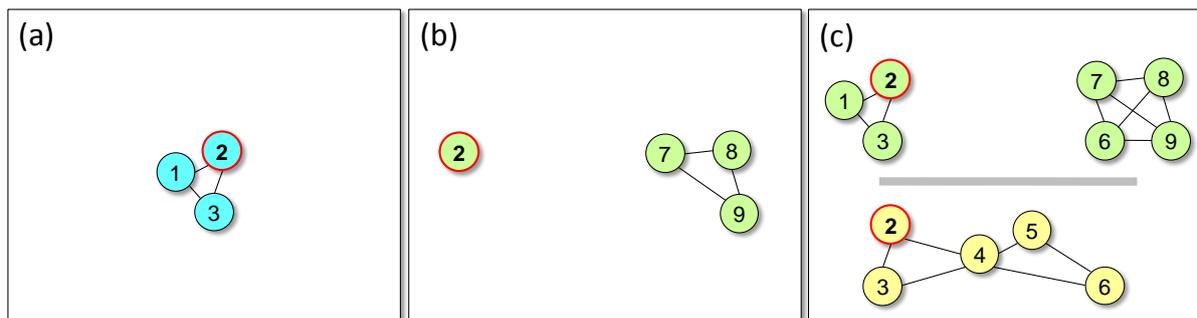


Figure 5-10: Node roles in different clustering solutions (focal node is marked with a red ring): (a) structural clustering; (b) content-based clustering on nodes; (c) content-based clustering on linkevents

Additionally, the size, shape and structure of the content clusters and their interrelation with other clusters can help to understand the development and current status of a knowledge domain and allow changing and improving the interaction within and between these domains. Comparing the clustering solutions with the temperature views one can further investigate on the context of the network on ego level as well as group level.

Section 5.1.5.1 deals with node role categorization based on degree and betweenness centrality. Section 5.1.5.2 presents two approaches how the knowledge portfolio can be categorized on group level (knowledge domains) and on actor level (knolwegde profiles). Finally, section 5.1.5.3 discusses some new metrics for cluster membership comparison to evaluate the differences and similarities between different clustering solutions.

5.1.5.1 Node Role Categorization

In this section a method for node role categorization is proposed. The categorization is based on the two prominent node metrics degree centrality and betweenness centrality (see section 2.1.4.2). Node roles can be calculated on network level as well as group level taking only

structural information into account. Using content clusters these roles can be related to the prominence and influence of a node in his knowledge domain.

Table 5-5: Node role categorization on degree and betweenness centrality

		Betweenness Centrality				
		0 %	Low	Medium	High	100 % ¹⁾
Degree Centrality	0 %	Isolated	X	X	X	X
	Low	Peripheral Specialist	Middleman between main part and small rest	Mediator	Bridge between main parts	Triple & many isolated nodes
	Medium	Specialist	Team Worker	Integrator	Broker	Number of isolated nodes decreases/ number of contacts increases
	High	(Specialist) only with very small networks	Coordinator	Information Spreader	Hot Spot	
	100 % ¹⁾	Dyad	Number of links between contacts increases/ number of contacts decreases			Triple or Star

1) Extreme centrality values provide no insights about the impact of the network structure as these networks are too small.

To identify these node roles, first degree centrality and betweenness centrality values have to be calculated for each node. On network level, these values are calculated based on the entire network structure, i.e. all nodes and links. They should be calculated separately for each unconnected component. On group level, these values are calculated on structure of the cluster each node belongs to, i.e. only cluster members and intra-cluster links. Using linkevent-based content clustering each node may belong to more than one cluster. Thus, different node roles can be identified depending on the context that is expressed by a content cluster. In a second step, the centrality values are assigned to the three categories “low”, “medium” and “high”. Both 0 % and 100 % are regarded as extreme centrality values that provide no insights about the impact of the network structure as these networks are too small. The node role can then be retrieved from Table 5-5.

The thresholds for low, medium and high values depend on the network size. With increasing number of nodes the impact of a node in terms of degree and betweenness centrality becomes less obvious. In smaller networks, a node can easily gain a prominent position having contacts to a large proportion of other nodes or connecting subgroups. However, with increasing network size the density of the network tends to decrease and high centrality values become less likely. This effect can be related to the effectivity of substructures in a larger context and the possible limits in the capacity of people to maintain stable relationships (see e.g. Dunbar’s number, section 2.1.3.2).

The necessary data for node role categorization is provided by the prototype introduced in section 5.2.

Examples for node role categorization on network level and on group level using the three different clustering approaches are given in the case study, chapter 5.3. Here, thresholds for

low, medium and high centrality values are given based on the size of the network and clusters as well as the experience of the author.

5.1.5.2 Knowledge Categorization

In this section two different approaches of knowledge categorization on different levels of investigation are presented. The categorization of knowledge domains on group level (see section 5.1.5.2.1) and the categorization of knowledge profiles on actor level (see section 5.1.5.2.2).

5.1.5.2.1 Knowledge Domains

Using content-based clustering on nodes or linkevents (see sections 5.1.4.2 and 5.1.4.3) the resulting non-overlapping or overlapping content clusters can be interpreted as knowledge domains. Each knowledge domain is then characterized by the set of affiliated topics and documents. In business context, these knowledge domains represent the overall knowledge portfolio of the enterprise. Analyzing the structure of this portfolio an overview of the demand and supply of knowledge can be gained on group level.

There are three categories to describe a knowledge domain: 1) homogeneity, 2) frequency and 3) activity type. Additionally, each knowledge domain can also be characterized by the distribution of different node roles in the cluster (see section 5.1.5).

Table 5-6: General algorithm of homogeneity categorization of knowledge domains

General Algorithm of Homogeneity Categorization of Knowledge Domains	
Prerequisite	Let C be a set of content clusters. Let N be a set of nodes. Let $\#N(c)$ be the number of nodes assigned to cluster $c \in C$. Let $\#LE_{intra}(c)$ be the number of intra-cluster linkevents per cluster $c \in C$. Let $\#LE_{intra}(c, n)$ be the number of intra-cluster linkevents of node $n \in N$ in cluster c . Let $\%LE_{intra}(c, n) = \#LE_{intra}(c, n) / \#LE_{intra}(c)$ be the percentage of intra-cluster linkevents of node n in cluster c .
Step 1	Calculate $\%LE_{intra}(c, n)$ for all nodes in each cluster. Calculate maximum value $max(c) = \max(\%LE_{intra}(c, n))$, mean value $\mu(c) = \mu(\%LE_{intra}(c, n))$ and standard deviation $\sigma(c) = \sigma(\%LE_{intra}(c, n))$ for all clusters with $\#N(c) > 1$.
Step 2	Calculate $f(c) = max(c) - (\mu(c) - \sigma(c))$ for all clusters with $\#N(c) > 1$.
Step 3	Calculate the 25 th percentile P_{25} of $f(c)$ on all clusters with $\#N(c) > 1$.
Step 4	Assign category "homogeneous" to all clusters with $f(c) \leq P_{25}$ and $\#N(c) > 1$. Assign category "inhomogeneous" to all clusters with $f(c) > P_{25}$ and $\#N(c) > 1$. Trivial clusters with $\#N(c) \leq 1$ are not categorized.

The homogeneity of a knowledge domain is calculated from the number of linkevents that have been exchanged among the members of the domain, i.e. intra-cluster linkevents. The homogeneity of a knowledge domain can be either "homogeneous" or "inhomogeneous". Inspecting the mean values, the maximum values and the standard deviations allows to establish this categorization. Similar mean and maximum values together with a low standard deviation indicate a participation pattern where the cluster contains nodes with approximately equal share of participation. Thus, this type of knowledge domain is characterized by all

cluster members having roughly the same impact on the generation, distribution and maintenance of the knowledge within this domain. No node obtains a prominent role in these clusters. Their patterns of participation can be classified as “homogeneous” in terms of node participation. In contrast, a maximum value that largely exceeds the mean value together with a high standard deviation indicates a participation pattern where the cluster has one (or more) dominant and several subsidiary nodes. Thus, this type of knowledge domain is dominated by a few cluster members in terms of activity and participation. This pattern of participation can be classified as “heterogeneous” in terms of node participation. A pseudo code description of the general algorithm of homogeneity categorization of knowledge domains is given in Table 5-6.

The frequency of a knowledge domain is calculated from the number of nodes that belong to the domain, i.e. cluster members. As a result, a cluster can either represent a special field of knowledge in which only a few knowledge worker are involved (“infrequent” field of knowledge) or a general field of knowledge with many knowledge workers (“frequent” field of knowledge). This may be due to work requirements, e.g. daily business versus special tasks, availability of knowledge workers, as well as the phase in its lifecycle, e.g. beginning, maturing, decaying (see e.g. stages of community development in Wenger et al. (2002: 69). A pseudo code description of the general algorithm of frequency categorization of knowledge domains is given in Table 5-7.

Table 5-7: General algorithm of frequency categorization of knowledge domains

General Algorithm of Frequency Categorization of Knowledge Domains	
Prerequisite	Let C be a set of content clusters. Let $\#N(c)$ be the number of nodes assigned to cluster $c \in C$.
Step 1	Calculate $\#N(c)$ for each cluster.
Step 2	Calculate the mean value μ of all $\#N(c) > 1$.
Step3	Assign category “infrequent” to all clusters with $\#N(c) > 1$ and $\#N(c) < \mu$. Assign category “frequent” to all clusters with $\#N(c) > 1$ and $\#N(c) \geq \mu$. Trivial clusters with $\#N(c) \leq 1$ are not categorized.

The activity type of a knowledge domain is calculated from the number of linkevents that have been exchanged among the members of the domain, i.e. intra-cluster linkevents. The activity type can be “low”, “medium” or “high”. A pseudo code description of the general algorithm of activity type categorization of knowledge domains is given in Table 5-8.

Table 5-8: General algorithm of activity type categorization of knowledge domains

General Algorithm of Activity Type Categorization of Knowledge Domains	
Prerequisite	Let C be a set of content clusters. Let $\#LE_{intra}(c)$ be the number of intra-cluster linkevents per cluster $c \in C$.
Step 1	Calculate $\#LE_{intra}(c)$ for each cluster.
Step 2	Calculate the 80 th percentile P_{80} of $\#LE_{intra}(c)$ on all clusters.
Step 3	Assign category “high” to all clusters with $\#LE_{intra}(c) > P_{80}$.
Step 4	Calculate the 80 th percentile P'_{80} of $\#LE_{intra}(c)$ on all clusters with $\#LE_{intra}(c) \leq P_{80}$.
Step 5	Assign category “medium” to all clusters with $\#LE_{intra}(c) > P'_{80}$, otherwise assign category “low”.

The thresholds used in this section are selected due to the experience of the author. They may depend on the network size, i.e. number of nodes as well as number of linkevents. The necessary data for knowledge domain categorization is provided by the prototype introduced in section 5.2.

Knowledge domains should only be categorized on group level using overlapping or non-overlapping content clusters. However, using this approach structural clusters can be categorized as well. As these clusters are based on direct interactions and not only similar context an overview of the topology of work groups or interaction patterns will be obtained.

In summary, there are twelve different types of knowledge domains that may be present in the data. The distribution of these domains will vary for each data set establishing a characteristic knowledge portfolio on group level. Examples of knowledge domain categorization are given in the case study, chapter 5.3.

5.1.5.2.2 Knowledge Profiles

Using content-based clustering on linkevents (see sections 5.1.4.3) overlapping content clusters are obtained. Each node will be affiliated with one or more content clusters that represent certain knowledge domains. Each knowledge profile is then characterized by the set of knowledge domains an individual belongs to. In business context, these knowledge profiles represent the individual knowledge portfolio of the enterprise. Analyzing the structure of this portfolio an overview of the demand and supply of individual knowledge can be gained on actor level.

There are three categories to describe a knowledge profile: 1) homogeneity, 2) diversification and 3) activity type. Additionally, each knowledge profile can also be characterized by the type of node roles the actor obtains in the different clusters (see section 5.1.5).

Each knowledge profile consists of the set of knowledge domains in which a certain node participates. The homogeneity of a knowledge profile is calculated from the number of linkevents the node as exchanged within each domain, i.e. intra-cluster linkevents of the node. The homogeneity of a knowledge profile can be either “homogeneous” or “inhomogeneous”.

Inspecting the mean values, the maximum values and the standard deviations allows to establish this categorization. Similar mean and maximum values together with a low standard deviation indicate a participation pattern where the node has a similar share in each cluster.

Thus, a homogeneous knowledge profile is characterized by an roughly equal distribution of the node’s activity and participation among all of his clusters. These nodes do not obtain a prominent field of knowledge. Their patterns of participation can be classified as “homogeneous” in terms of participation in different fields of knowledge. In contrast, a maximum value that largely exceeds the mean value together with a high standard deviation indicates a participation pattern where the node has one (or more) dominant and several subsidiary fields of knowledge. Therefore, inhomogeneous knowledge profiles are dominated by a few clusters in terms of activity and participation. This pattern of participation can be classified as “heterogeneous” in terms of participation in different fields of knowledge. A pseudo code description of the general algorithm of homogeneity categorization of knowledge profiles is given in Table 5-9.

Table 5-9: General algorithm of homogeneity categorization of knowledge profiles

General Algorithm of Homogeneity Categorization of Knowledge Profiles	
Prerequisite	Let C be a set of content clusters. Let N be a set of nodes. Let C_n be a subset of C containing the set of content clusters node n belongs to, $C_n \subseteq C$. Let $\#C(n)$ be the number of clusters node n belongs to, $n \in N$. Let $\#LE_{intra}(c, n)$ be the number of intra-cluster link events of node $n \in N$ in cluster $c \in C$. Let $\%LE_{intra}(c, n) = \#LE_{intra}(c, n) / \sum_{c \in C_n} \#LE_{intra}(c, n)$ be the percentage of intra-cluster link events of node n in cluster c .
Step 1	Calculate $\%LE_{intra}(c, n)$ for all nodes in each cluster. Calculate maximum value $max(c, n) = max(\%LE_{intra}(c, n))$, mean value $\mu(c, n) = \mu(\%LE_{intra}(c, n))$ and standard deviation $\sigma(c, n) = \sigma(\%LE_{intra}(c, n))$ for all nodes with $\#C(n) > 1$.
Step 2	Calculate $f(c, n) = max(c, n) - (\mu(c, n) - \sigma(c, n))$ for all nodes with $\#C(n) > 1$.
Step 3	Calculate the 25 th percentile P_{25} of $f(c)$ on all clusters with $\#N(c) > 1$.
Step 4	Assign category “homogeneous” to all nodes with $f(c, n) \leq P_{25}$ and $\#C(n) > 1$. Assign category “inhomogeneous” to all nodes with $f(c, n) > P_{25}$ and $\#C(n) > 1$. Trivial nodes with $\#C(n) \leq 1$ are not categorized.

The diversification of a knowledge profile is calculated from the number of clusters the node belongs to. As a result, a node can either concentrate thoroughly on a few fields of knowledge (“specialized”) or acquire diverse but not fundamental knowledge in many fields of knowledge (“diversified”). This may be due to work requirements, e.g. increasing number of supervisory tasks in management positions in contrast to elaborate and time-consuming involvement in a single task on operational level as well as personal characteristics. This can also be related to Granovetter’s concept of strong and weak ties (see section 2.1.3.3). Specialized knowledge profiles consist of only a few clusters, whereas diversified knowledge profiles consist of many clusters. A pseudo code description of the general algorithm of diversification categorization of knowledge profiles is given in Table 5-10.

Table 5-10: General algorithm of diversification categorization of knowledge profiles

General Algorithm of Diversification Categorization of Knowledge Profiles	
Prerequisite	Let $c \in C$ be a set of content clusters. Let N be a set of nodes. Let C_n be a subset of C containing the set of content clusters node n belongs to, $C_n \subseteq C$. Let $\#C(n)$ be the number of clusters node n belongs to, $n \in N$.
Step 1	Calculate $\#C(n)$ for each node.
Step 2	Calculate the mean value μ of all $\#C(n) > 1$.
Step 3	Assign category "specialized" to all nodes with $\#C(n) > 1$ and $\#C(n) < \mu$. Assign category "diversified" to all nodes with $\#C(n) > 1$ and $\#C(n) \geq \mu$. Trivial nodes with $\#C(n) \leq 1$ are not categorized.

The activity type of a knowledge profile is calculated from the number of linkevents the node as exchanged within each domain, i.e. intra-cluster linkevents of the node. The activity type can be "low", "medium" or "high". A pseudo code description of the general algorithm of activity type categorization of knowledge profiles is given in Table 5-11.

Table 5-11: General algorithm of activity type categorization of knowledge profiles

General Algorithm of Activity Type Categorization of Knowledge Profiles	
Prerequisite	Let $c \in C$ be a set of content clusters. Let N be a set of nodes. Let C_n be a subset of C containing the set of content clusters node n belongs to, $C_n \subseteq C$. Let $\#LE_{intra}(c, n)$ be the number of intra-cluster linkevents of node $n \in N$ in cluster $c \in C$.
Step 1	Calculate $sum(c, n) = \sum_{c \in C_n} \#LE_{intra}(c, n)$ for each node.
Step 2	Calculate the 80 th percentile P_{80} of $sum(c, n)$ for all nodes.
Step 3	Assign category "high" to all nodes with $sum(c, n) > P_{80}$.
Step 4	Calculate the 80 th percentile P'_{80} of $sum(c, n)$ on all nodes with $sum(c, n) \leq P_{80}$.
Step 5	Assign category "medium" to all nodes with $sum(c, n) > P'_{80}$, otherwise assign category "low".

The thresholds used in this section are selected due to the experience of the author. They may depend on the network size, i.e. number of nodes as well as number of linkevents.

The necessary data for knowledge profile categorization is provided by the prototype introduced in section 5.2.

Knowledge profiles can only be calculated on clustering results allowing a node to belong to more than one cluster, e.g. content-based clustering on linkevents.

In summary, there are twelve different types of knowledge profiles that may be present in the data. The distribution of these profiles will vary for each data set establishing a characteristic knowledge portfolio on actor level. Examples of knowledge profile categorization are given in the case study, chapter 5.3.

5.1.5.3 Comparison of Clustering Results

Besides the visual inspection of clustering solutions it is also possible to compare two clustering solutions from different types of clustering procedures to gain an understanding

about cluster membership stability and dissociation on ego level and group level. The two clustering solutions, or partitions, are denoted as P_1 and P_2 . For each node, $n \in N$, and for each pair of clusters the node belongs to, $n \in C_1$ and $n \in C_2$, with one cluster from each clustering solution, $C_1 \in P_1$ and $C_2 \in P_2$, the number of nodes that belong to the same cluster (“neighbors”) and the actor-centric cluster overlap is calculated. Therefore, the *neighbors* of a node in a cluster are defined as

$$neighbors(n, C_j) = \{n_i \in N: n_i \neq n \wedge n_i \in C_j\}$$

The *actor-centric cluster overlap* is then defined as follows:

$$overlap(n, C_1, C_2) = neighbors(n, C_1) \cap neighbors(n, C_2)$$

When comparing two clustering solutions there are two metrics of *actor-centric group membership stability* based on the neighbors of a node and the actor-centric cluster overlap which have been first proposed by Bobrik and Trier (2009). The first stability metric calculates the amount of neighbors of a node in one clustering solution that are also grouped together in the other clustering solution. This measure is defined as follows:

$$GMS1_1 = GMS1(n, C_1, C_2) = \frac{\#overlap(n, C_1, C_2)}{\#neighbors(n, C_1)} \quad \text{for partition } P_1 \text{ and}$$

$$GMS1_2 = GMS1(n, C_2, C_1) = \frac{\#overlap(n, C_1, C_2)}{\#neighbors(n, C_2)} \quad \text{for partition } P_2.$$

The second metric is based on the Dice association coefficient (see section 4.3.4). It measures how stable the group membership of a node is in terms of constant group neighbors (overlap) and group size. It can be formalized as follows:

$$GMS2 = GMS2(n, C_1, C_2) = \frac{2 * \#overlap(n, C_1, C_2)}{\#neighbors(n, C_1) + \#neighbors(n, C_2)}.$$

Based on these actor-centric group membership stabilities the cluster-centric group membership stability can be calculated for both clustering solutions. The *cluster-centric group membership stability* is defined for all pairs of clusters of the corresponding clustering solutions that share at least one node (cluster-centric overlap > 0). The set of nodes that belong to the same cluster is called *members*. It is defined as

$$members(C_j) = \{n_i \in N_a: n_i \in C_j\}.$$

The *cluster-centric overlap* between two clusters is then defined as

$$overlap(n, C_1, C_2) = members(C_1) \cap members(C_2).$$

When comparing two clustering solutions there are two aggregated metrics of cluster-centric group membership stability for each clustering solution based on the members of a node and the cluster-centric cluster overlap. For the first clustering solution the average group membership stabilities are defined as

$$AVG(GMS1) = AVG(GMS1(n, C_1, C_2)) = \frac{\sum_{n \in C_1} GMS1(n, C_1, C_2)}{\#members(C_1)} \quad \text{and}$$

$$AVG(GMS2) = AVG(GMS2(n, C_1, C_2)) = \frac{\sum_{n \in C_1} GMS2(n, C_1, C_2)}{\#members(C_1)}.$$

For the second clustering solution the average group membership stabilities are defined as

$$AVG(GMS1) = AVG(GMS1(n, C_2, C_1)) = \frac{\sum_{n \in C_2} GMS1(n, C_2, C_1)}{\#members(C_2)} \quad \text{and}$$

$$AVG(GMS2) = AVG(GMS2(n, C_2, C_1)) = \frac{\sum_{n \in C_2} GMS2(n, C_2, C_1)}{\#members(C_2)}.$$

These aggregated group membership stabilities indicate the average number of nodes per cluster that are grouped together in both clustering solutions.

All these metrics are implemented in the prototype. Figure 5-29 in section 5.2.7.4 provides some exemplary data by comparing a structural clustering solution with a content-based clustering solution on nodes.

5.2 Prototype

The design of the prototype is based on the general methodology of cluster analysis proposed in chapter 4.1.3. The data set used in this section to demonstrate the features of the prototype is a smaller subsample of the Enron e-mail corpus described in section 5.3.1. It consists of 48 nodes with one isolated node, 122 links and 791 linkevents from January to December 2000. After initial screening (see section 5.2.4), 34 linkevents, two links, and the isolated node have been removed from the data because they do not contain any topic. The topics of the linkevents are weighted using the *tf.idf* weighting scheme. The average weighting scheme is applied to nodes, links and the entire network. This reduced data set is used to demonstrate the features of the prototype.

This section is organized as follows: Section 5.2.1 gives an overview of the general architecture and design principles of the prototype. The data format for data access and exchange is explained in section 5.2.2. Sections 5.2.3 to 5.2.7 then explain and demonstrate the different features of the prototype available via the graphical user interface. These features are related to the research guideline depicted in Figure 5-2. The data view and the temperature view correspond to the step 3 (network analysis) and the clustering view corresponds to step 4 (cluster analysis) of this guideline.

5.2.1 Architecture

In this section the general architecture of the prototype is described. This covers the programming language (see section 5.2.1.1), an overview of the conceptual design of the prototype (see section 5.2.1.2) as well as a reduced class diagram (see section 5.2.1.3).

5.2.1.1 Programming Language

The prototype for content-based cluster analysis is programmed in Java²⁸ (version 6) using some additional software libraries. The representation of the data as a graph, e.g. the network graphview (see section 5.2.3.1) or the temperature graphview (see section 5.2.5), is developed using the software library Java Universal Network/Graph Framework (JUNG)²⁹ (version 2.0.1). The JUNG library provides different means of network visualization³⁰, e.g. changing the graph layout, the node size and color, or the link width. The different types of graph layouts³¹ can be selected via the “Layout” menu item in the main menu bar of the prototype. It is recommended to use the Fruchterman-Reingold Layout (“FR Layout”) (see Fruchterman and Reingold 1991). There are also a number of network metrics and algorithm available by the JUNG library. The structural clustering (see section 5.2.7.1) is based on the graph-based edge betweenness clustering algorithm by Girvan and Newman (see section 4.7.2) and implemented using the `EdgeBetweennessClusterer`³² of the JUNG library. The graph-based

²⁸ <http://java.com>

²⁹ <http://jung.sourceforge.net/>

³⁰ edu.uci.ics.jung.visualization

³¹ edu.uci.ics.jung.algorithms.layout

³² edu.uci.ics.jung.algorithms.cluster.EdgeBetweennessClusterer

metrics are calculated on the graph representation of the network using some JUNG packages, e.g. shortest paths³³ or centrality metrics³⁴.

The clustering view provides several charts (see section 5.2.7), e.g. dendrogram charts, network activity charts and cluster validation plots, that are based on the java chart library JFreeChart³⁵ (version 1.0.13).

The graphical user interface of the prototype is programmed by using the eclipse plug-in Jigloo SWT/Swing GUI Builder³⁶ (version 4.6.4).

Furthermore, the java JDOM³⁷ library is used which allows handling complex xml files and interoperates with existing standards such as the Simple API for XML (SAX) and the Document Object Model (DOM).

5.2.1.2 Conceptual Design

As illustrated in Figure 5-11 the conceptual design of the prototype is structured into five layers through which the data is processed from the external data source (“Data Access Layer”) to the graphical user interface (“GUI Layer”).

The data format for data access and exchange as well as the internal data format is based on the SNI data model described in chapter 1 (SNI Framework), section 2.2.1. In order to meet the requirements it has been extended by data objects for cluster analysis, i.e. “cluster” and “level”. As described in section 5.2.2.1 the original data is stored in a SQL database using the SNI data model and then extracted and converted to a xml file which is conform with the CLAN xsd schema. The Data Access Layer retrieves the data from a xml file and processes them to the Data Layer. This second layer converts the data into the internal data format. The Data Processing Layer then controls the parameters for initial screening and cluster analysis (“Cluster Strategy”) and the graphical representation of the network (“Graph Control”). It directly processes the data to the graphical user interface (“GUI Layer”) or to the intermediate Visualization Layer which converts the data into its representations as a graph. The GUI Layer provides various means to present the data and to handle the interaction with the user. The input by the user is then processed back to the layers below.

³³ edu.uci.ics.jung.algorithms.shortestpath

³⁴ edu.uci.ics.jung.algorithms.scoring

³⁵ <http://www.jfree.org/jfreechart/>

³⁶ <http://www.cloudgarden.com/jigloo/>

³⁷ <http://www.jdom.org/>

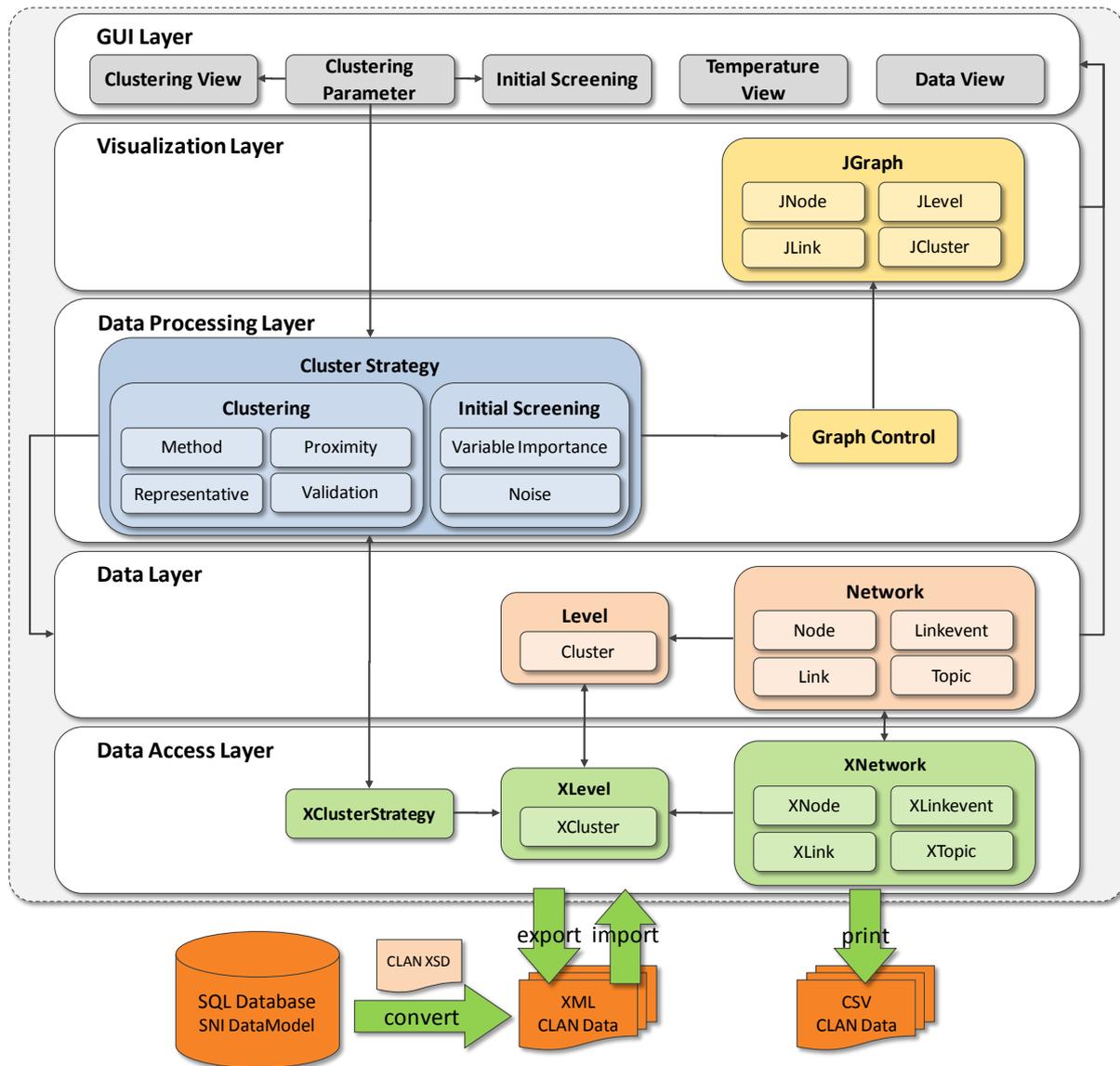


Figure 5-11: Architecture of the prototype. Conceptual design

5.2.1.3 Class Diagram

In Figure 5-12 the class diagram of the general architecture of the prototype is shown. It consists of the main packages and most important classes on a coarse grained level. The coloring of the classes corresponds to the different layers of the conceptual design of the prototype as depicted in Figure 5-11 in the previous section.

Besides the main class and the graphical user interface³⁸ (“ClusterAnalyzerMainGui”) there are five main packages: data, dataAccess, clustering, visualization and xnetwork. The clustering package contains all classes for data processing, i.e. the different parameters for initial screening and cluster analysis. The “ClusterStrategy” class is the center of this package which controls the program. The data package consists of the network elements of the internal data model whereas the xnetwork package consists of the network elements for data access and exchange via xml files controlled by the dataAccess package (“XMLImport” and

³⁸ There are further GUI classes in the gui package that are not included in this class diagram.

“XMLExport”) extending the JDOM library (see section 5.2.1.1). In this package the export of the data as csv files for further spreadsheet analysis is implemented as well (“CSVImport”). The data exchange is explained in more detail in the next section. The representation of the data as graph is implemented by the visualization package extending the JUNG library (see section 5.2.1.1)

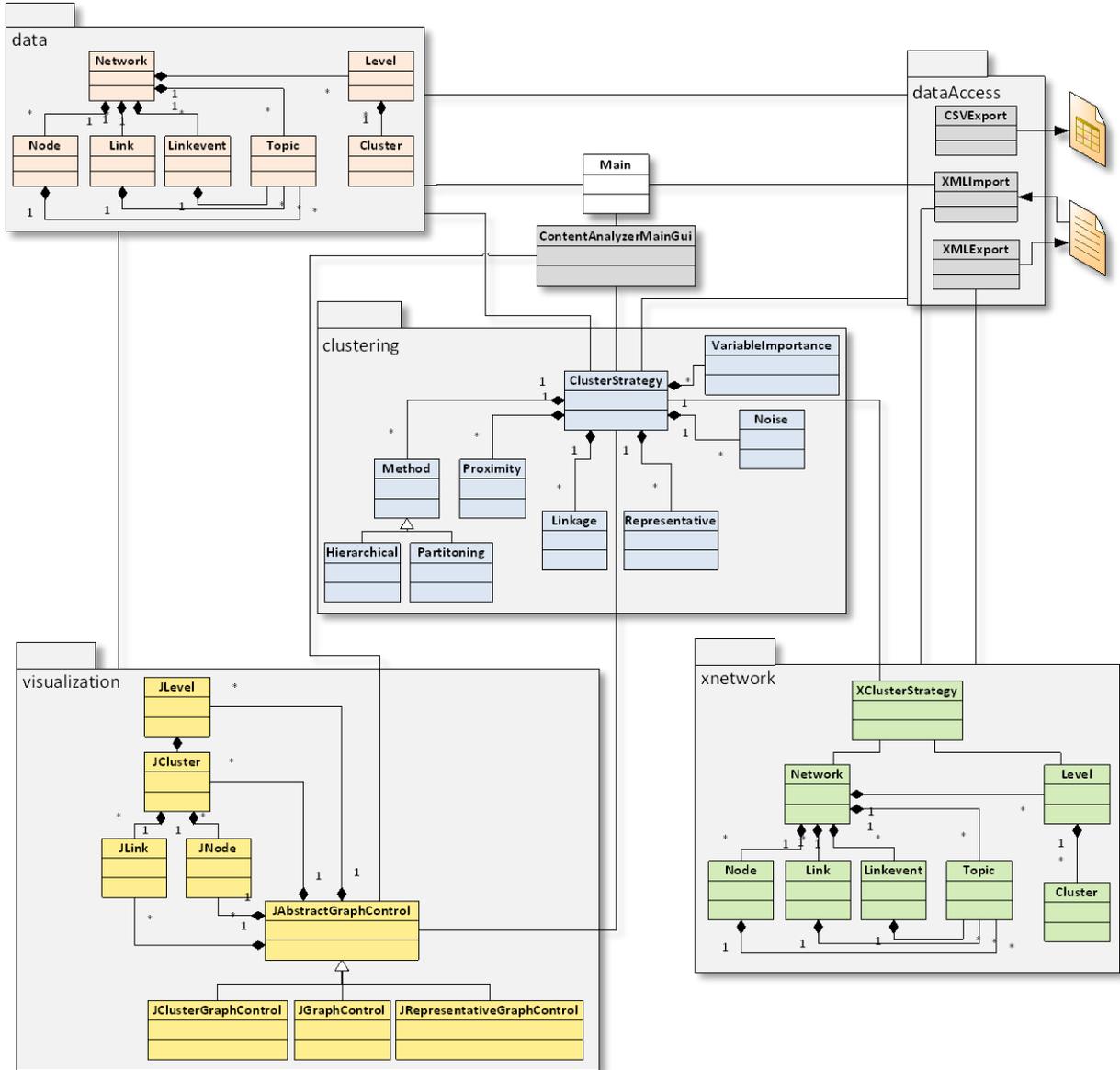


Figure 5-12: Architecture of the prototype. Class diagram

5.2.2 Data Access and Exchange

The Data Access Layer illustrated in Figure 5-11 covers those parts of the prototype that retrieve the data from an xml file (“import”) or store the results from the analysis in an external file for further reuse together with the network itself as an xml file (“export”) or for additional spreadsheet analysis as a csv file (“print”). The prototype can only process xml files that are conform to the xsd schema described in see section 5.2.2.1. The data access and exchange is controlled via the “File” menu item and the “Print” menu item in the main menu bar of the prototype (see section 5.2.2.2).

5.2.2.1 XSD Schema for Data Exchange

The data format for data access and exchange as well as the internal data format is based on the SNI data model described in chapter 1 (SNI Framework), section 2.2.1. As illustrated in Figure 5-11 the original data is stored in a SQL database using the SNI data model and then extracted and converted to an xml file. However, other data sources and data formats of the original data are possible if the resulting file is validated using the CLAN xsd schema.



Figure 5-13: Prototype data format. CLAN XSD schema. Network elements partly expanded

Although the SNI data format captures sender-recipient communication as well as thread-based communication, at the moment the prototype is only designed to handle sender-recipient communication. However, thread-based communication can easily be transformed to fit the data model of the prototype. The prototype can only process xml files that are conform to the CLAN xsd schema. This xsd schema is shown in Figure 5-13. Here, the basic elements of the network and its elements are partly expanded: The root element is called “xnetwork”. It covers all properties of the network (“xfilters”, “xnetworkpropertytypes” and “xproperties”), the lists of its elements (“xnodes”, “xlinkevents” and “xlinks”), the topics related to the

content of network (“xtopics”) and (optionally) the parameters of the network and cluster analysis (“xclusterstrategy”). The child node “xfilters” contains the information about which filters have been used when creating the xml file, e.g. time filter, node filters, linkevent filters or topic filter. The properties of the network and its elements (“xproperties” with corresponding property types “xnetworkpropertytypes”, “xnodepropertytypes”, “xlinkpropertytypes” and “xlinkeventpropertytypes”) provide static information about network properties and metrics that have been calculated on the original network data before creating the xml file. These properties are not updated when initial screening procedures are applied (see section 5.2.4).

Besides the “xproperties” child node, each network element consists of the “xtopicfrequencies” child node which provides the list of topics and their frequencies that can be found in the content objects assigned to the element. The elements “xlink” and “xlinkevent” also contain the information about the sender and recipient nodes that belong to them.

The element “xclusterstrategy” corresponds to the cluster strategy as well as initial screening parameters (see sections 5.2.4 and 5.2.6) and those elements contained in the clustering view (see section 5.2.7). It is included in the xml file if the data are exported from the prototype. It is not included if the xml file is generated from the SQL data source. In Figure B-1 in the appendix B, section B.1, the element “xclusterstrategy” is partly expanded.

5.2.2.2 Import, Export and Print

In order to analyze and cluster networking data a xml file has to be imported which is conform to the data format described in the previous section. This can either be a “raw” network without any previous clustering results or a network saved together with the results from some previous analysis using the prototype. For example, parts of the network might been set inactive after some initial screening procedures. Here, one can decide if the full network data or only the active network data should be exported to the xml file. Furthermore, a clustering solution might be used for further analysis. In this case, one can decide whether the whole set of clustering solutions obtained by either hierarchical or partitioning clustering shall be saved for later reuse or only one selected level. The import and export feature is available via the “File” menu item in the main menu bar and the “Import CLAN Data” button in the secondary menu bar.

Besides importing and exporting the data as xml files it is also possible to export the quantitative data of the different data tables as csv files. This feature allows additional spreadsheet analyses of the data, e.g. for node role or knowledge categorization (see section 5.1.5). One can either export all data or only those available in the different views of the prototype, e.g. initial data view, temperature view or clustering view. These options are available via the “Print” menu item in the main menu bar.

The charts programmed using the JFreeChart library (see section 5.2.1.1) also provide the options to save the chart as a png file or print it as a pdf file.

5.2.3 Data View

The data view consists of two components: the “Initial Data View” tabbed pane (see section 5.2.3.1) and “the Active Network Data” tabbed pane (see section 5.2.3.2). They allow a thorough quantitative and qualitative analysis of the original data as well as the data containing only active network elements after initial screening.

5.2.3.1 Initial Data View

The “Initial Data View” tabbed pane provides an overview over the network and the three different types of network elements: node, link and linkevent.

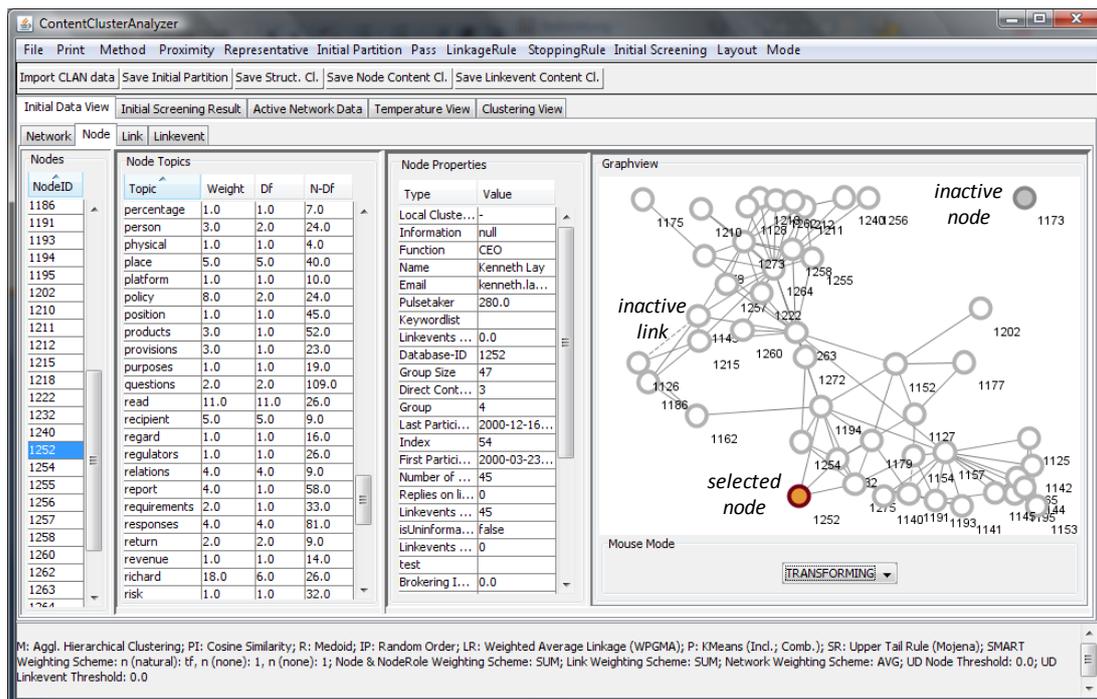


Figure 5-14: Initial data view. Node view. Selected node 1252

The “Network” data view contains a topics table with all topics occurring in the network, their current network weights depending on the network weighting scheme selected (see section 5.2.4.1) and their document frequencies. It further provides a table with data specific network properties that have been imported from the data file. The properties table does not change after initial screening procedures have been applied, but the topics table is updated. Weights and document frequencies are recalculated and topics that have no active linkevent ($df = 0$) are removed. They are set inactive until the data is reset to the original network configuration. Additionally, this data view contains a visualization of the network as a graph. If a node is set inactive by some initial screening procedure its color changes from orange to gray. If a link is set inactive it is painted as dotted line.

Using the node data view a node has to be selected from the node selection table containing all node ids. The data view is then updated with the data from the selected node (see Figure 5-14). Similar to the network data view the node data view contains a node topics table, a node properties table and a graphview. Here, only the selected node is highlighted as orange node with red border. All other active nodes are painted white with gray borders, all inactive nodes are painted gray, all inactive links are painted as dotted lines. In this data set 34

linkevents, two links (between nodes 1126 and 1143 and between nodes 1154 and 1193) and one node (node 1173) do not contain any topic (see section 5.2.4.3). Thus, they are set inactive.

The link and linkevent data view provide similar data as the node data view. In addition to link ids and linkevent ids the sender and recipient node ids of these elements are included as well.

The “Initial Data View” tabbed pane is especially helpful for gaining first insights of the data and validating the results from initial screening procedures, e.g. the effects on topic weights and the removal of network elements from further analysis.

5.2.3.2 Active Network Data

The “Active Network Data” contains the properties and metrics on network level and node level calculated from the active network elements. These data is updated after each initial screening procedure (see section 5.2.4). The data provided by these tables can be used to examine the results from the initial screening procedures and to identify key players for further analysis.

5.2.4 Initial Screening

The initial screening procedures available via the “Initial Screening” menu item in the main menu bar cover adjustment of variable importance (see section 5.2.4.1) and noise detection (see section 5.2.4.3). A further initial screening procedure is the mode of linkevent affiliation to nodes which can be changed using the “Mode” menu item in the main menu bar (see section 5.2.4.2). The results from the different initial screening procedures are presented in the “Initial Screening Result” tabbed pane.

5.2.4.1 Variable Importance

The “Variable Importance” menu item contains several options to change the topic weights within the network.

The use of standardized variables in cluster analysis is controversially discussed in the literature. Thus, no recommendation can be made at this issue. For more information about standardization of variables in cluster analysis see chapter 4, section 4.2.2.2.

For linkevent weighting the SMART weighting scheme has been implemented. It provides different configurations for term frequency, document frequency and normalization to calculate topic weights. Usually, the natural term frequency tf and inverted document frequency idf with no normalization factor are used when computing similarities between a query and a document vector in information retrieval systems. However, when comparing linkevents for content-based clustering there is no need to dampen the effect of a topic appearing often in the entire document collection. Thus, it is recommended to use natural term frequency tf and no document frequency or normalization. For more information about the SMART weighting scheme see chapter 3 (Text Mining), section 3.4.3.2. If the linkevent weighting scheme is changed it is not only applied to all linkevents but also the topic weights of the other network elements are changed as they depend on the linkevent weights.

There are four options for the node weighting schemes how to aggregate topic weights from the linkevents affiliated with a node: For each topic the sum, average or maximum of its weights or the number of linkevents it occurs in (“count”) can be calculated. The average topic weight is only calculated from those linkevents assigned to a node which actually contain the topic. The link and network weighting schemes are similar to the node weighting scheme providing the same options how the topic weights are calculated from the linkevents affiliated with each link or the entire network.

Table 5-12 in section 5.2.5 illustrates the effects of weighting schemes and linkevent standardization on content similarities.

5.2.4.2 Mode

There are two modes available via the “Mode” menu item in the main menu bar which determines how nodes are affiliated with linkevents when calculating content similarities. Choosing the “sent” mode linkevents are only assigned to the sender nodes, whereas choosing the “sent and received” mode linkevents are assigned to both sender nodes and recipient nodes. This choice has a strong influence on calculating and comparing content similarities and the results of content-based clustering. Temperature view (see section 5.2.5) and content-based clustering on nodes (see section 5.2.7.2) calculate the content similarities between nodes from their linkevent collections. Depending on the type of mode selected both methods use either only linkevents sent or linkevents sent and received as linkevent collection for each node. Usually, using the “sent” mode will yield less similar content profiles than using the “sent and received” mode. Content-based clustering on linkevents (see section 5.2.7.3) clusters single linkevents instead of linkevent collections but determines the cluster membership of nodes to linkevent clusters by the type of mode selected. Again, using the “sent” mode less nodes will usually be assigned to linkevent clusters than using the “sent and received” mode. The change in pairwise node similarities by using the different modes is illustrated in Table 5-12. It is recommended to use the “sent and received” mode.

5.2.4.3 Noise Detection

Noise detection can be applied to linkevents or nodes as network elements. There are two types of noise: uninformative data or outlier. A network element is regarded as uninformative data if the number of topics assigned to this element does not exceed a certain threshold. Comparing the content dissimilarities of a network element with all other elements of the same type the element is regarded as outlier if none of its content dissimilarities exceeds a certain threshold. Outliers are set inactive and therefore excluded from further analysis (until the data is reset to the original network configuration). Removing linkevents from the network might result into nodes or links that do not exceed the given thresholds. These elements are set inactive as well. Furthermore, removing nodes or linkevents from the network might result into nodes, linkevents or links that do not exceed the given thresholds anymore. In a second step, these elements are identified and set inactive as well. To visualize the results of noise detection inactive nodes are painted as gray nodes in the graphviews and inactive links are painted as dotted lines. After the removal of network elements the weighting schemes are updated (see section 5.2.4.1). For more information about noise detection see chapter 4 (Cluster Analysis), section 4.2.3.

5.2.4.4 Initial Screening Results

The “Initial Screening Results” tabbed pane contains four tables with the initial screening results for nodes, linkevents, links and topics. The initial screening result tables are initialized with the original network data and updated after each initial screening procedure.

The tables for nodes, linkevents and links show if one of these network elements is active in the current network configuration or not. It is active if it has not become an outlier or uninformative data due to noise detection on nodes or linkevents (see section 5.2.4.3). The tables for nodes and linkevents also contain the thresholds used for uninformative data and outlier detection. All three network element tables also contain the data specific properties of each network element which do not change (e.g. e-mail address, name, business unit of an employee), the date or activity time span (date of first and last linkevent) and some general network metrics like degree, betweenness and closeness centrality that are recalculated after the network changes. The topic table contains the topics together with the type of linkevent standardization used and the standardization factor calculated for each topic.

5.2.5 Temperature View

The “Temperature View” tabbed pane allows a visual inspection of the distribution of content similarities between nodes. This approach can be used for an initial screening of the data to gain a first understanding of the network or to select important key players for further analysis. It is also useful to analyze and compare the effect of the different methods of noise detection and variable importance on the content similarity of the network and its elements. The Temperature view is also available for each clustering method to visualize the content similarities within a cluster (see section 5.1.4).

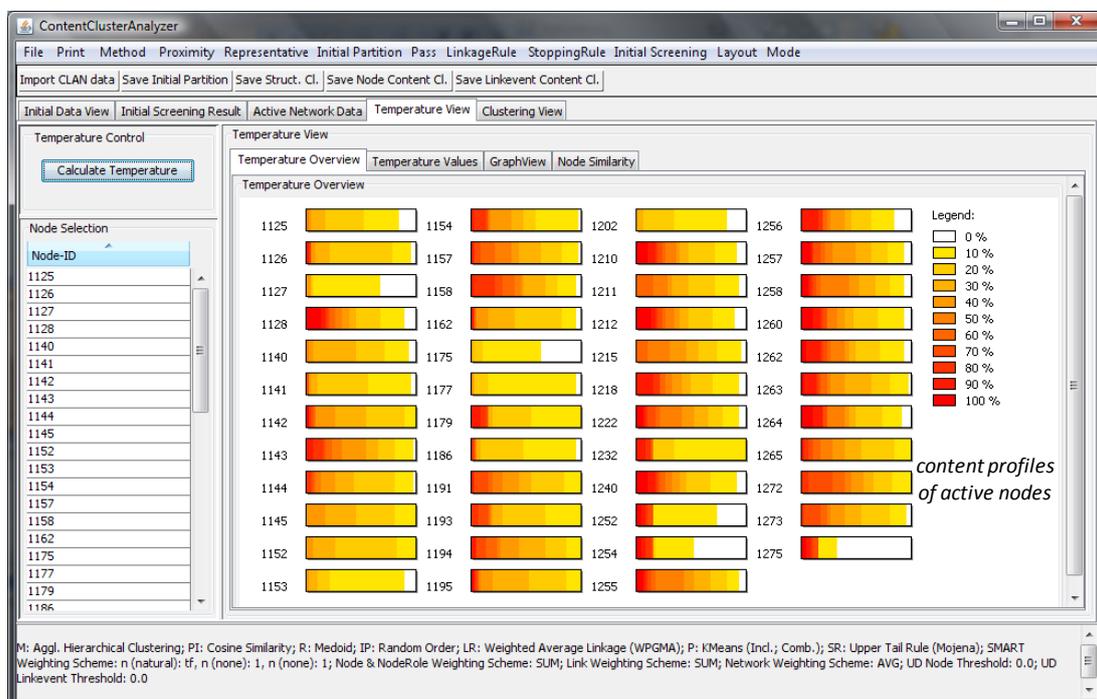


Figure 5-15: Temperature overview. Content profiles of active nodes

The “Temperature Overview” tabbed pane contains the content profiles of all active nodes in the network (see Figure 5-15). Content profiles are the visualization of the content similarities

calculated for all pairs of nodes in the network based on the weighting scheme and proximity selected (see section 5.1.3.2). The content profiles shown in Figure 5-15 are calculated using the cosine similarity, the sent and received mode, the average node weighting scheme and the *tf.idf* linkevent weighting scheme (see Table 5-12, configuration c)). Linkevents, links and nodes without any topic are removed by a preliminary initial screening procedure (see section 5.2.4).

In addition to the visual inspection of content profile via the temperature overview the “Temperature Values” tabbed pane contains a table with numeric expressions of the content profiles for each node (see Table 5-2 in section 5.1.3.2)

Table 5-12 demonstrates the effect of different initial screening and cluster strategy parameters on the node content similarities. The table contains ten content profiles with different parameter configurations. In general, the distribution of content similarities depends on the number of nodes, linkevents, and topics as well as the distribution of linkevents and topics among nodes. For example, the average node weighting scheme will produce content similarities similar to those obtained by the sum node weighting scheme if only a few linkevents with rather different topics are assigned to the nodes.

In configurations a) to f) the “sent and received” mode is used, whereas in configurations g) and h) only the linkevents sent are taken into account (see section 5.2.4.2). Changing from linkevents sent and received to only linkevents sent will dramatically reduce the pairwise content similarities.

In configurations e) and h) the content similarities are calculated by the Pearson coefficient whereas all other configurations use the cosine similarity (see section 5.2.6.2). When changing the proximity index from cosine similarity to Pearson coefficient the number of very similar nodes decreases but the number of nodes with medium and small content similarities increases.

The variable importance of nodes and linkevents is affected by the weighting schemes and the type of standardization used. For more details on the available weighting schemes and linkevent standardization see section 5.2.4.1. The node weighting schemes employed are either the sum of all term frequency (configuration a)) or the average of all term frequency (configurations b) to h)). When changing the node weighting scheme from sum to average the content similarities become less extreme: The number of very similar nodes decreases but the number of nodes with medium and small content similarities increases.

Linkevent weights are either calculated from term frequency only (*tf*, configurations a) and b)), term frequency and inverted document frequency (*tf.idf*, configurations c), e) to h)) or augmented term frequency and inverted document frequency (*augmented.idf*, configuration d)). Changing the linkevent weighting scheme from *tf* to *tf.idf* mainly affects the medium and especially small content similarities. Comparing configuration b) and c) one can see that the inverted document frequency reduces these similarities but has little effect on the strong similarities. Changing the linkevent weighting scheme from *tf.idf* to *augmented.idf* has a similar effect like changing the proximity index, i.e. the content similarities become more evenly distributed and less extreme.

Table 5-12: Comparison of content similarity profiles. The effect of initial screening and cluster strategy parameters on node similarities

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
	1175 1177 1179 1186 1240 1252 1254 1255 1256 1257							
	Cosine similarity	Pearson coefficient						
	Node weighting: sum	Node weighting: average						
	Linkevent weighting: tf	Linkevent weighting: tf.idf	Linkevent weighting: tf.idf	Linkevent weighting: augmented.idf	Linkevent weighting: tf.idf	Linkevent weighting: tf.idf	Linkevent weighting: tf.idf	Linkevent weighting: tf.idf
	No standardization	Linkevent standardization: Z-score1	No standardization	No standardization				
	Mode: sent & received	Mode: sent	Mode: sent & received	Mode: sent	Mode: sent			

Linkevent standardization is applied only in configuration f). Comparing configurations c) and f) linkevent standardization affects the medium and especially small content similarities. One can see that the standardization with mean value and standard variance (Z-score1) reduces these similarities but has little effect on the strong similarities.

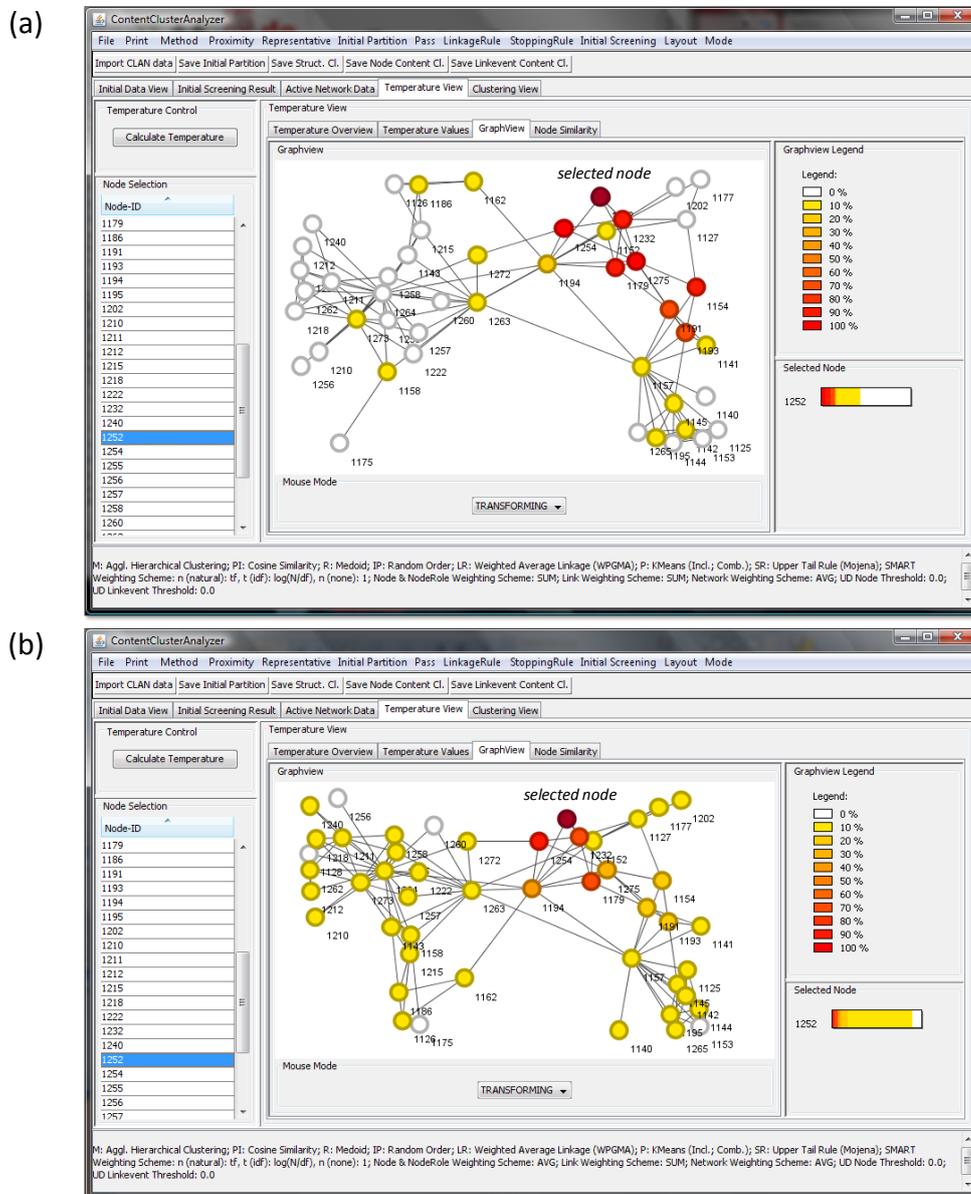


Figure 5-16: Temperature graphview. Structural distribution of content similarities of a selected node. Comparison of different node weighting schemes: (a) node weighting scheme: sum; (b) node weighting scheme: average

Although the temperature view of the content profiles allows examining the general distribution of content similarities among nodes it does not account for its structural distribution. Therefore, single nodes can be selected by the “Node Selection” table. The “Graphview” tabbed pane and the “Node Similarity” tabbed pane then provide further details of the content profile of the selected node. The node similarity table contains the pairwise node similarities of a content profile for a more quantitative analysis. The graphview maps the content profile of a selected node to the graph of the network (see section 5.1.3.2). This visualization helps to identify whether a content profile accounts for a local clustering of similar nodes or an even distribution throughout the network. The combination of structural

distribution of node similarities in the graph and the corresponding content profile helps to gain a deep understanding of structure and content of the data.

Figure 5-16 illustrates the effect of the node weighting scheme on the structural distribution of the content similarities of a node. In this figure the content profile and graphview of node 1252 is given a) using the sum node weighting scheme and b) using the average node weighting scheme as in configuration c) in Table 5-12. The focal node is depicted as dark red node. Comparing both configurations, the number of high similarities decreases but the number of medium and small similarities increases in Figure 5-16 b). The similarities are more evenly distributed in the network.

5.2.6 Cluster Strategy

The main menu bar provides several options to customize the parameters of the content-based clustering procedures. These options include the type of clustering method (see section 5.2.6.1), the proximity measure and the type of cluster representative (see section 5.2.6.3) as well as the type of initial partition (see section 5.2.6.4) and the type of pass for partitioning clustering (see section 5.2.6.5). For hierarchical agglomerative clustering the linkage rule (see section 5.2.6.6) and the stopping rule (see section 5.2.6.7) can be customized.

5.2.6.1 Method

There are two types of clustering methods available via the “Method” menu item: agglomerative hierarchical clustering and partitioning clustering. It is recommended to use the partitioning clustering method for clustering content objects represented as topic vectors. Agglomerative hierarchical clustering can be employed to calculate the initial partition to determine the number of clusters (see section 4.5.2.1). The subsequent partitioning clustering procedure will further refine this solution. For more information about clustering methods see chapter 4 (Cluster Analysis), section 4.4 (hierarchical clustering) and section 4.5 (partitioning clustering).

5.2.6.2 Proximity

There are three categories of proximity measures available via the “Proximity” menu item: association coefficients, correlation coefficients, and distances measures. Available association coefficients are the Dice coefficient, the Gower and Legendre coefficient, Jaccard coefficient, the Rogers and Tanimoto coefficient, the simple matching coefficient and the Sokal and Sneath coefficient. Available correlation coefficients are the cosine similarity and the Pearson coefficient. Available distance measures are the Euclidean distance, the Squared Euclidean distance, the Manhattan distance, the Canberra distance and the Chebychev distance. It is recommended to use the cosine correlation coefficient to calculate similarities between content objects represented as topic vectors. The temperature view (see section 5.2.5) is only defined for correlation coefficients and distance measures. For more information about proximity measures see chapter 4 (Cluster Analysis), section 4.3.

5.2.6.3 Representative

There are two different types of representatives available via the “Representative” menu item: centroid or medoid. It is recommended to use medoids as representatives as they have a direct

expression as nodes or linkevent, depending on the type of clustering. For more information about cluster representatives see chapter 4 (Cluster Analysis), section 4.2.1.3.

5.2.6.4 Initial Partition

There are two types of initial partitions for partitioning clustering available via the “Initial Partition” menu item: random partition and existing partition. Using the random partition one has to estimate the number of clusters whereas using an existing partition, e.g. from some preliminary hierarchical clustering, the number of clusters is given. For more information about cluster representatives see chapter 4 (Cluster Analysis), section 4.2.1.3.

5.2.6.5 Pass

When using the partitioning clustering method there are two passes available via the “Pass” menu item: the K -means pass as inclusive combinatorial pass or the forgy pass as inclusive non-combinatorial pass. It is recommended to use the K -means pass for clustering content objects represented as keyword vectors. For more information about the type of pass see chapter 4 (Cluster Analysis), section 4.5.2.4.

5.2.6.6 Linkage Rule

When using the agglomerative hierarchical clustering method there are seven different linkage rules available via the “Linkage Rule” menu item: single linkage, complete linkage, average linkage, weighted average linkage, centroid linkage, median linkage and Ward’s method. It is recommended to use the group or weighted average linkage rule to calculate the initial partition of a partitioning clustering solution when comparing topic vectors. For more information about linkage rules see chapter 4 (Cluster Analysis), section 4.4.1.

5.2.6.7 Stopping Rule

There are three stopping rules available via the “Stopping Rule” menu item: the Upper Tail rule, the Calinski and Harabasz stopping rule and the Hartigan stopping rule. It is recommended to use the Calinski and Harabasz stopping rule to determine the optimal clustering solution (“best cut”) when using hierarchical agglomerative clustering as it is based on within-cluster similarities as well as between-cluster similarities. For more information about stopping rules as relative indices to validate partitional structures see chapter 4 (Cluster Analysis), section 4.6.4.3.

5.2.7 Clustering View

The ”Clustering View” tabbed pane consists of the three clustering procedures structural clustering (see section 5.2.7.1), content-based clustering on nodes (see section 5.2.7.15.2.7.2) and content-based clustering on linkevents (see section 5.2.7.3) as well as the cluster comparison view which allows comparing the cluster memberships and stabilities of different clustering solutions (see section 5.2.7.4).

5.2.7.1 Structural Clustering

The “Structural Clustering” tabbed pane allows calculating non-overlapping node clusters based on the network structure. This clustering solution can be used as baseline solution for comparing and evaluating the results of the content-based clustering procedures. The

structural clustering algorithm is basically the edge betweenness clustering algorithm for community detection by Girvan and Newman. The implementation is provided by the Java JUNG library (see section 5.2.1.1). For more details on the theoretical foundation of the algorithm see chapter 4 (Cluster Analysis), section 4.7.2.

The “Structural Clustering” tabbed pane is divided into two parts. The left side of the tabbed pane provides the “Clustering Control” panel and the “Level Selection” table. The right side of the tabbed pane provides different perspectives on the clustering results. Via the “Clustering Control” panel the clustering procedure is started and the optimal level is displayed. The “Level Selection” table is updated with the results of the clustering procedure.

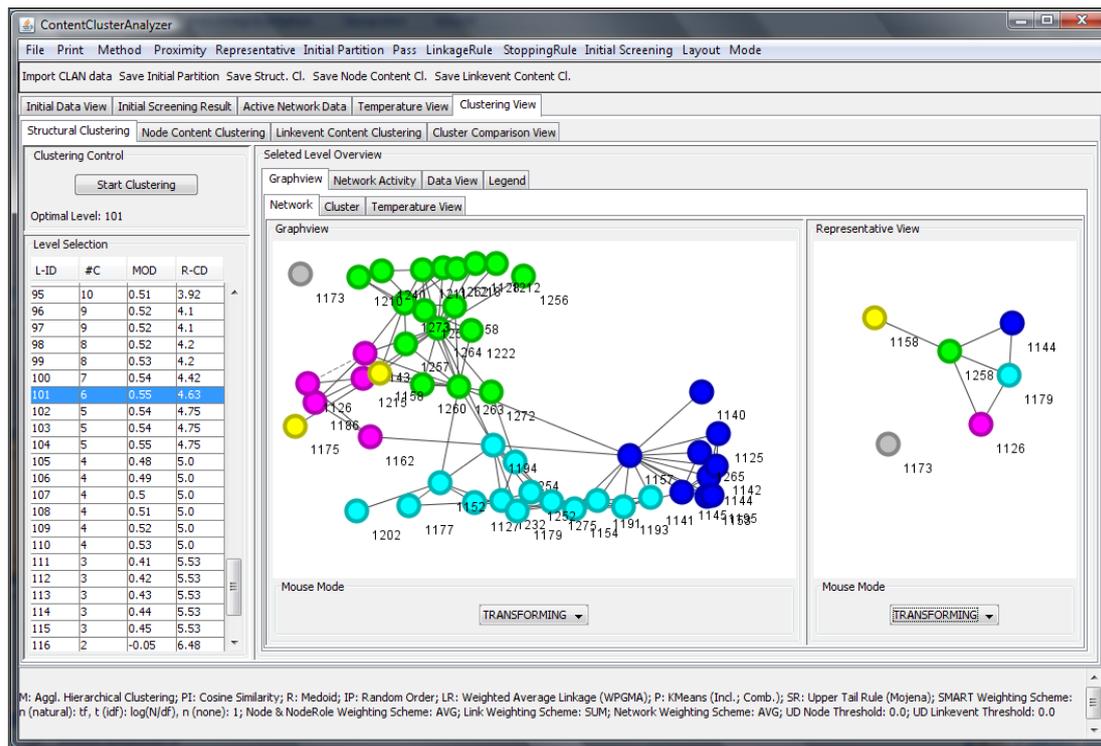


Figure 5-17: Structural clustering view

An example of structural clustering based on the edge betweenness clustering algorithm by Girvan and Newman is given in Figure 5-17. Linkevents, links and nodes without any topic have been removed during a preliminary initial screening step. Inactive network elements are painted as gray nodes or dotted lines. They are not part of the analysis. The optimal clustering solution is determined by comparing the modularity values of each partition (“level”). Level 101 has the highest modularity value, $mod = 0.55$. All levels can be selected by the “Level Selection” table. For each level, this table contains the level id (“L-ID”), the number of clusters (“#C”), the modularity value (“MOD”) and the average content dissimilarity of the cluster representatives to all other cluster members (“R-CD”).

The “Graphview” tabbed pane allows a visual inspection of the selected partition by three different tabbed panes: “Network”, “Cluster” and “Temperature View”. On network level, the components “Graphview” and “Representative View” are provided. The graphview contains the entire network with the node color indicating cluster memberships. The representative view illustrates the interaction of the clusters represented by their cluster representative. The cluster representative is calculated due to the content similarities of the cluster members and

the type of representative selected as part of the cluster strategy (see section 5.2.6). The “Graphview” tabbed pane also allows to inspect the patterns of interaction of a single cluster (“Cluster”) and its temperature view (“Temperature View”).

The data provided by the “Graphview”, “Network Activity” and “Data View” tabbed panes are the same as for content-based clustering on nodes. For more details see section 5.2.7.2.

5.2.7.2 Content-based Clustering on Nodes

Content-based clustering on nodes groups together those nodes with high similarities between the topics retrieved from the linkevent collections assigned to them (for more details see section 5.1.4.2). In this section the different components for selecting and analyzing a suitable solution are explained.

If not marked otherwise, the following cluster strategy parameters are used in this section: The clustering solutions are obtained from agglomerative hierarchical clustering using the cosine similarity and the weighted average linkage rule. The type of cluster representative is medoid. The validation index is calculated using the Calinski and Harabasz stopping rule. The average node weighting scheme and the *tf.idf* linkevent weighting scheme are applied to compute topic weights. Linkevents, links and nodes without any topic are set inactive. Similarities between nodes are calculated from both linkevents sent and received.

5.2.7.2.1 Clustering Control and Level Selection

The “Node Content Clustering” tabbed pane is divided into two parts (see Figure 5-18 on page 226). The left side of the panel provides the “Clustering Control” panel and the “Level Selection” table. The right side of the panel provides different perspectives on the data which are explained in some detail in the remainder of this section.

The “Clustering Control” panel allows starting the clustering procedure due to the current cluster strategy configuration and displays the optimal level. The “Level Selection” table is updated with the results of the clustering procedure, either the list of succeeding levels of a hierarchical clustering procedure or the rearranged partitions of a partitioning clustering procedure. In the second case, the last entry (row) refers to the optimal solution. Using hierarchical clustering the table includes the following information for each level: level id (“L.id”), number of clusters (“#C”), fusion coefficient (“FI”), validation index (“VI”) as well as the overall (“CD”), the minimum (“Min-CD”) and the maximum (“Min-CD”) content dissimilarity of the level. The same data is provided for partitioning clustering, except for the fusion coefficient and validation index which are only defined on hierarchies.

5.2.7.2.2 Graphview

The “Graphview” tabbed pane contains different visual approaches to investigate on the selected clustering solution in general (“Network” tabbed pane) and a selected cluster from this solution (“Cluster” tabbed pane and “Temperature View” tabbed pane).

The “Network” tabbed pane provides a graphview of the entire network with gray nodes representing inactive nodes. Based on the selected clustering solution the node color of the active nodes corresponds to the cluster they belong to (see Figure 5-18). Additionally, the representative view illustrates the interaction of the clusters represented by their cluster

representative. The cluster representative is calculated due to the content similarities of the cluster members and the type of representative selected as part of the cluster strategy (see section 5.2.6).

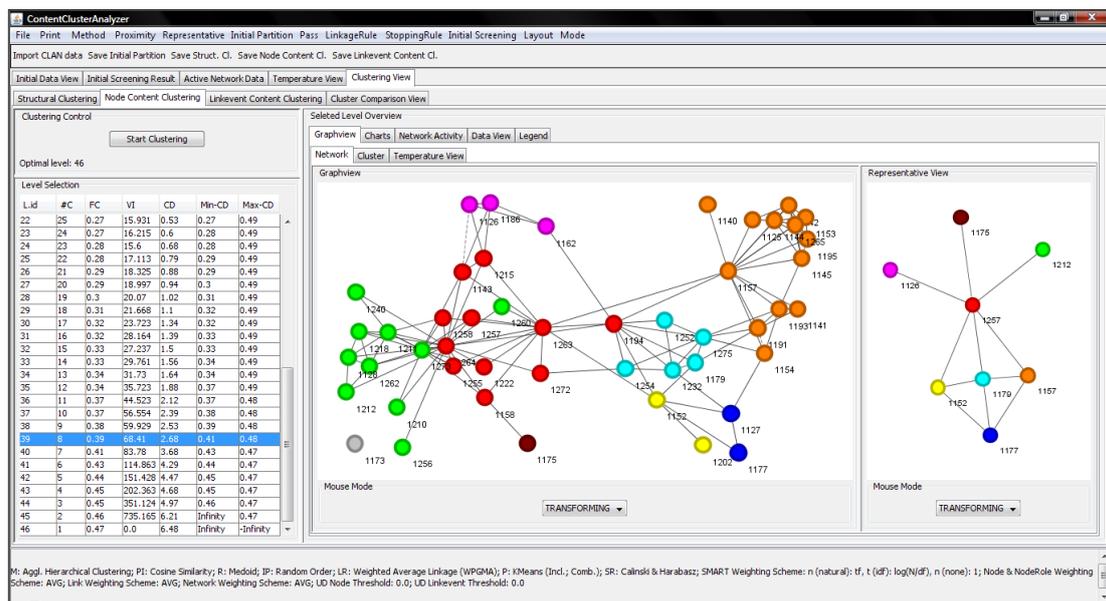


Figure 5-18: Content-based clustering on nodes. Graphview of the entire network and representative view. Hierarchical clustering

The network graphview helps to decide which clustering solution might be appropriate for the problem at hand. The “Cluster” tabbed pane provides a graphview of the selected cluster (see Figure 5-19 a)). Only the intra-cluster nodes and links are visualized in the graph. Green nodes represent nodes with only linkevents sent in the current cluster, blue nodes represent nodes with only linkevents received, and orange nodes represent nodes with linkevents sent and received. White nodes have no active linkevent in this cluster. The cluster graphview helps to investigate on the structure of a cluster as well as the position and behavior of its members.

Similar to the temperature view of the entire network (see section 5.2.5), the “Temperature View” tabbed pane provides the temperature view for the selected cluster. Figure 5-19 shows the overview of content profiles of all nodes within a selected cluster (Figure 5-19 b)) and the temperature graphview with node coloring according to the content profile of a selected node (Figure 5-19 c)). The temperature view of the selected cluster helps to investigate on the content similarities within a cluster. This allows validating the quality of the selected clustering solution. Rather dissimilar clusters will indicate that the current solution has a low quality whereas rather similar clusters will indicate a good solution.

The dendrogram of a hierarchical clustering solution helps to get an understanding how clusters have merged on subsequent levels. For more information about the use of dendrograms for comparing hierarchical clustering solutions see chapter 4.4.1.6.

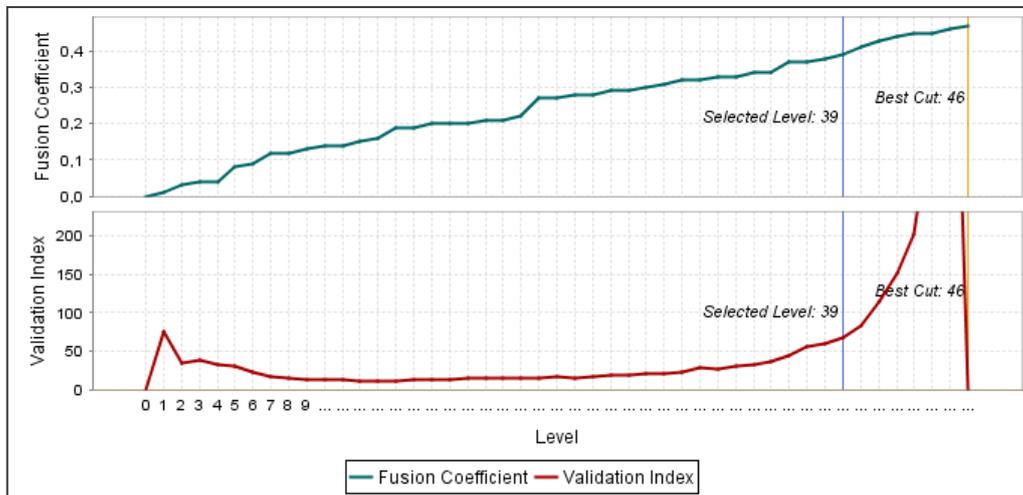


Figure 5-21: Content-based clustering on nodes. Level validation. Hierarchy overview with plots of fusion coefficient and validation index versus level

Further charts for clustering validation are the “Level CD” tabbed pane with a plot of the overall content dissimilarities (see Figure 5-22) and the “Min Max CD” tabbed pane with a plot for both minimum and maximum dissimilarities of the clustering solutions. These plots can be used for cluster validation. For example one can look for a significant “knee” in the plots which indicates a good clustering solution (see chapter 4.6.4.3). In Figure 5-22 such knee is marked with the blue “Selected Level” marker. This level is also marked in the dendrogram shown in Figure 5-20. Furthermore, the clustering solution illustrated in Figure 5-18 and Figure 5-19 corresponds to this level.

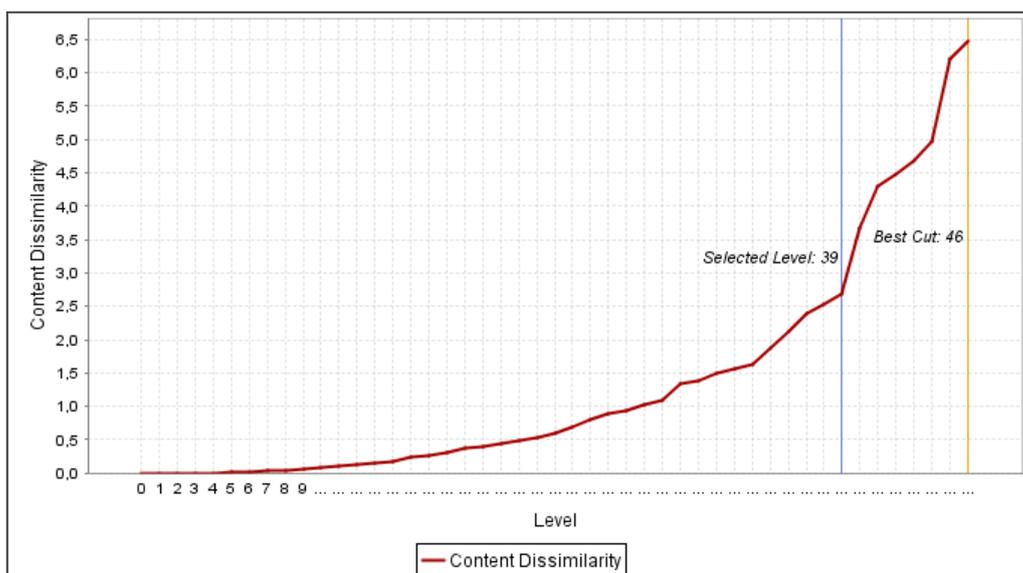


Figure 5-22: Content-based clustering on nodes. Charts for validation of hierarchical clustering solutions. Plot of average content dissimilarity

When using a partitioning clustering procedure the “Clustering Overview” tabbed pane contains a visual representation of the changes in cluster memberships after each iteration

(“level”³⁹) of the clustering algorithm. This diagram is depicted in Figure 5-23 a). In this example a random initial partition with 8 clusters is used (level 0). As a rather small data set is clustered there are only two iterations necessary to obtain a stable optimal clustering solution (level 2) where a change of cluster memberships does not improve the criterion function, i.e. the overall content dissimilarity.

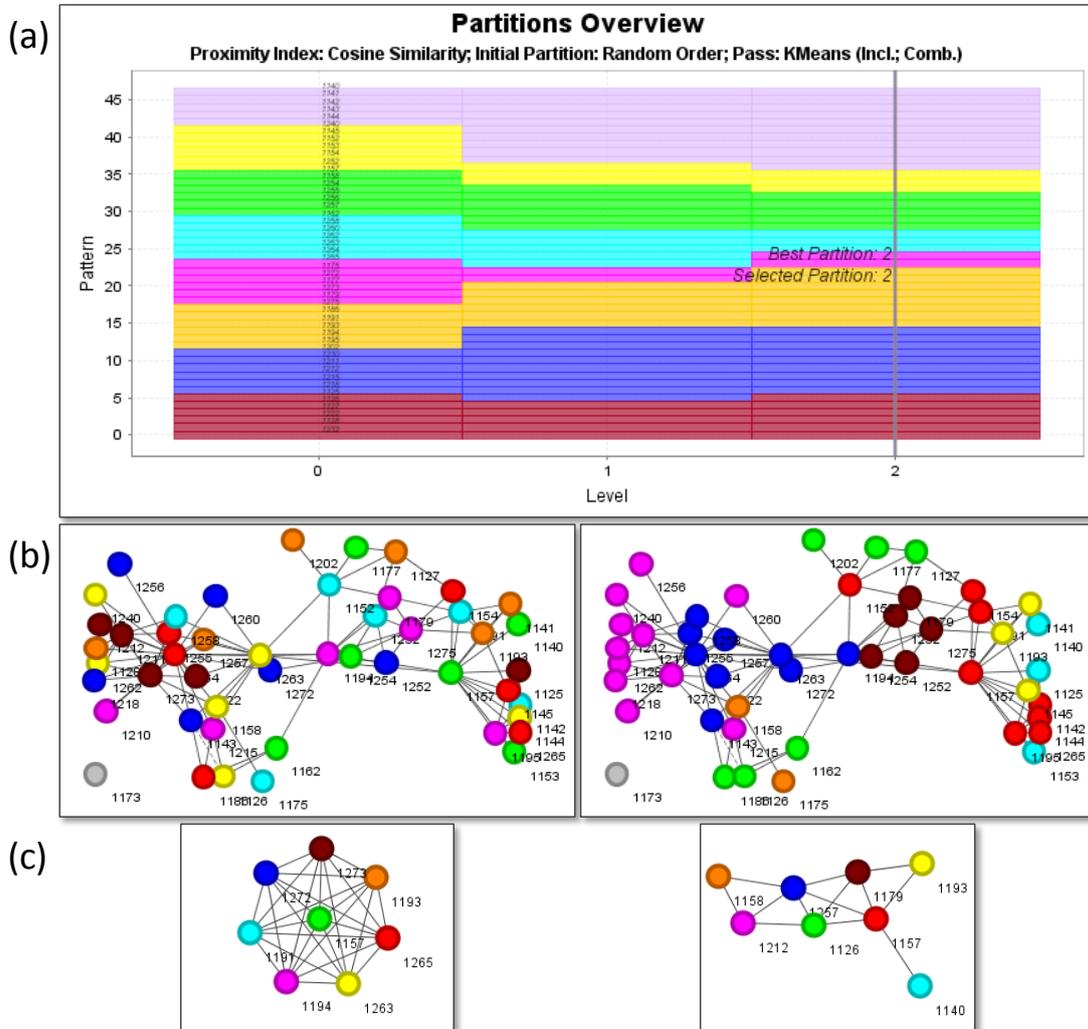


Figure 5-23: Content-based clustering on nodes. Partitioning clustering with random initial partition (level 0, left) and optimal solution (level 2, right): (a) clustering overview; (b) network graphview; (c) representative graphview

In Figure 5-23 b) the network graphviews and c) the representative graphviews of the initial (left) and final (right) clustering solutions are shown which map the clustering solution to the graph⁴⁰. The initial partition yields node memberships evenly distributed through the network whereas the optimal partition has reassigned those nodes to clusters whose content similarities are bound to the network structure. This solution is similar to the level obtained from hierarchical clustering with the same number of clusters (level 39, see Figure 5-18).

³⁹ With partitional clustering the prototype uses the terms level and pass as synonyms.

⁴⁰ The coloring of the clusters in the clustering overview plot does not correspond with the coloring of the clusters in both graphviews.

It is recommended to employ hierarchical clustering solutions as initial partitions which are further refined by a partitioning clustering procedure. This approach helps to overcome the problem of determining the right number of clusters and the problem of suboptimal cluster memberships established early in the hierarchy. For more information see chapters 4.5.2.1 and 4.5.2.5.

As fusion coefficient and validation index are only defined for hierarchies the “Level Validation” tabbed pane only the “Level CD” tabbed pane and the “Min Max CD” tabbed pane are available as further validations charts for partitioning clustering procedures.

5.2.7.2.4 Network Activity

The “Network Activity” tabbed pane helps to investigate on the temporal distribution of the activities within each cluster (“Intra-cluster View”) and between each cluster (“Inter-cluster View”). In Figure 5-24 the network activity view for a) intra-cluster activities and b) inter-cluster activities is shown for all clusters of the selected clustering solution. Additionally, in both plots the entry “all” refers to the unclustered network.

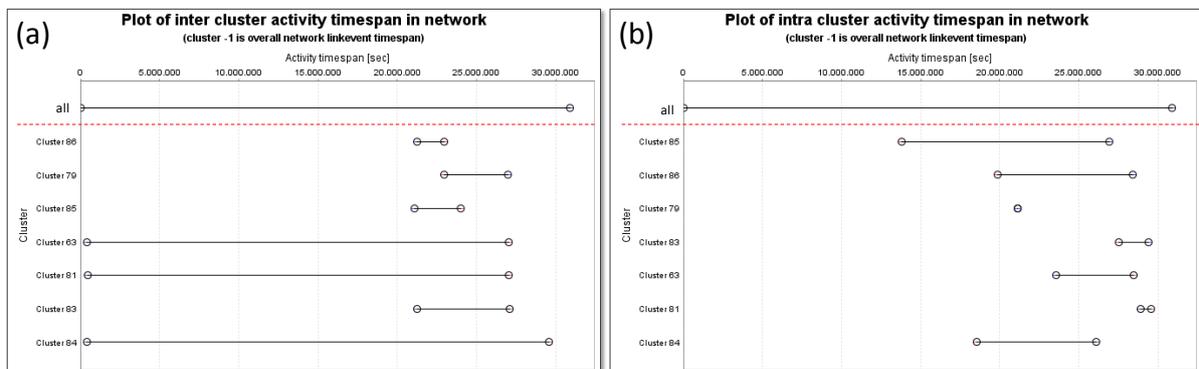


Figure 5-24: Content-based clustering on nodes. Network activity view: (a) inter-cluster activity; (b) intra-cluster activity

This view helps to gain an understanding of the temporal activities within and between the different clusters.

5.2.7.2.5 Data View

The “Data View” tabbed pane provides information about the different clustering solutions and their elements organized in six tabbed panes.

The “Level Overview” tabbed pane contains a table with general properties and metrics for each cluster of the selected level. The entire list is given in Table B-1 in the appendix B, section B.2.1.1.

The “Level Details” tabbed pane contains a table with properties and metrics for each node of the selected level. The entire list is given in Table B-2 in the appendix B, section B.2.1.2. If one entry is selected, the corresponding node is marked with a red ring in all graphviews and the “Node Overview” tabbed pane and “Node Details” tabbed pane are updated. The “Level Details” tabbed pane allows selecting a cluster for further inspection. Besides the visual inspection of within-cluster activities using the cluster graphview and the within-cluster content similarities using the temperature view (see section 5.2.7.2.2) the “Cluster Details”

tabbed pane can be employed to gain a deeper understanding of the selected cluster. It provides numeric and visual information about the selected cluster.

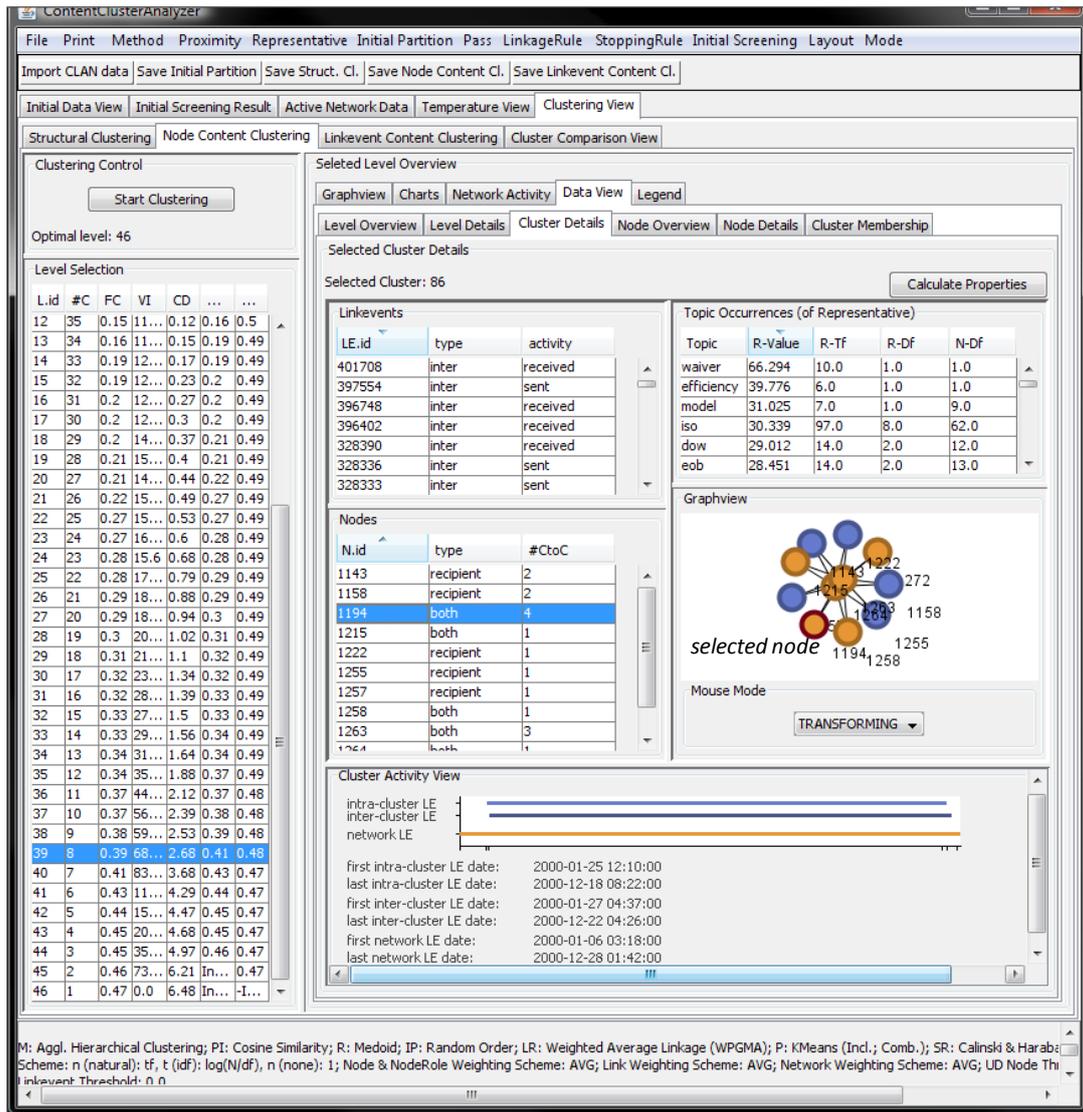


Figure 5-25: Content-based clustering on nodes. Cluster details

An example of the “Cluster Details” tabbed pane is depicted in Figure 5-25. It consists of the following elements: the linkevents table, the topic occurrences table, the nodes table, the graphview and the cluster activity view. The linkevents table contains a list of all linkevents of the cluster, their type (inter-cluster linkevent or intra-cluster linkevent) and their activity (sent or received). The topic occurrences table contains the topics of the cluster representative together with their current weight (“R-Value”), term frequency (“R-Tf”), document frequency (“R-Df”) and the overall document frequency in the network (“N-Df”) calculated from its linkevent collection. The nodes table contains a list of all nodes of the cluster, their type (sender, recipient or both) and the number of contacts to other clusters (“#CtoC”) each node has. The graphview provides the visual representation of the cluster as a graph with node coloring corresponding to the activity type of the node. This graphview is the same as depicted in the cluster graphview of the “Graphview” tabbed pane. If a single node is selected, it is marked with a red ring in all graphviews. The last element of the cluster details panel is the cluster activity view which shows a small plot with the time spans covered by the inter-

cluster linkevents, intra-cluster linkevents and all network linkevents. Additionally, the dates of the first and last linkevents of each category are provided as well.

To further investigate on a selected node the “Node Overview” tabbed pane and the “Node Details” tabbed pane provide detailed information about the node. The “Node Overview” tabbed pane contains a table with quantitative information about the selected node in its cluster and in the entire network. The entire list is given in Table B-3 in the appendix B, section B.2.1.3. The “Node Details” tabbed pane contains numeric and visual information about the selected node. In Figure 5-26 an example of this view is given. It consists of the following elements: the topics pie charts, the topic occurrences table, the node roles table and the node role activity view. Note that for content-based clustering on nodes there is no distinction between node (“N”) and node role (“NR”) as in content-based clustering on linkevents.

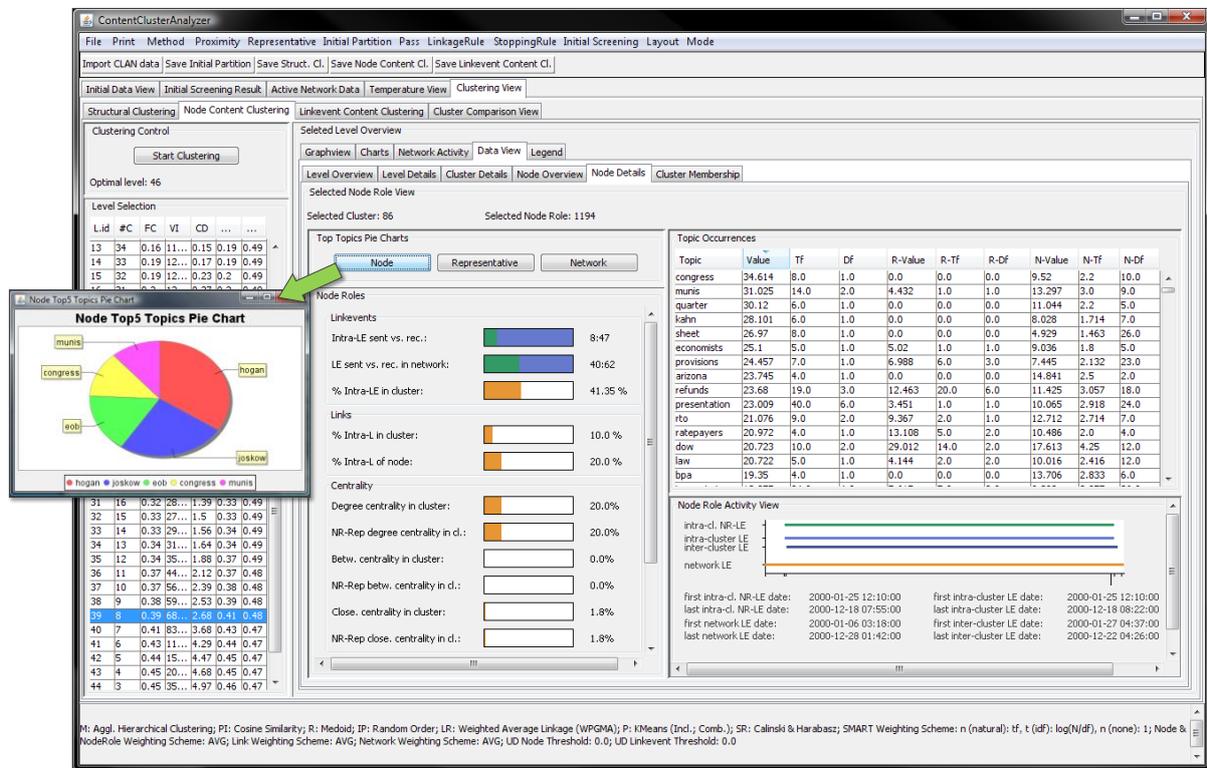


Figure 5-26: Content-based clustering on nodes. Node details

The top topics pie charts allow creating pie charts of the top 5 topics with the highest value for the selected node, the cluster representative and the entire network. The node roles panel provides information about the different roles of the selected node in its cluster and compares it to the entire network and the cluster representative. There are three categories: linkevents, links and centrality. For all these categories a visual and numeric expression of the node role is given. The linkevents category consists of the relation of intra-cluster linkevents sent versus received, the relation of linkevents sent versus received in the entire network and the percentage of all intra-cluster linkevents the selected node is involved in. The links category consists of the percentage of all active links of the node that are intra-cluster links and the percentage of all intra-cluster links the selected node is involved in. The centrality category compares all three centrality metrics of the selected node in its cluster with those of the cluster representative. The topic occurrences table contains the topics of the node together with their

current weight (“Value”), term frequency (“Tf”), and document frequency (“Df”) calculated from its linkevent collection. Additionally, these metrics are also provided for the cluster representative (“R-Value”, “R-Tf” and “R-Df”) and the entire network (“N-Value”, “N-Tf” and “N-Df”). The last element of the node details panel is the node role activity view which shows a small plot with the time spans covered by the inter-cluster linkevents, intra-cluster linkevents and all network linkevents. Additionally, the dates of the first and last linkevents of each category are provided as well.

The “Cluster Membership” tabbed pane provides an overview about which nodes are grouped together in the same cluster.

5.2.7.3 Content-based Clustering on Linkevents

Content-based clustering on linkevents groups together those linkevents with high similarities between their topics. The nodes are then assigned to these clusters of similar linkevents due to their linkevents sent or received. As a result, overlapping clusters of similar nodes are obtained. As each node can belong to more than one cluster its appearance in a cluster is called “node role”. For more details see section 5.1.4.3. In this section the different components for selecting and analyzing a suitable solution are explained.

If not marked otherwise, the following cluster strategy parameters are used in this section: The clustering solutions are obtained from agglomerative hierarchical clustering using the cosine similarity and the weighted average linkage rule. The type of cluster representative is medoid for both linkevent and node role cluster representatives. The validation index is calculated using the Calinski and Harabasz stopping rule. The average node weighting scheme and the *tf.idf* linkevent weighting scheme are applied to compute topic weights. Linkevents, links and nodes without any topic are set inactive. Similarities between nodes are calculated from both linkevents sent and received.

5.2.7.3.1 Clustering Control, Level Selection, Clusters and Nodes

The “Linkevent Content Clustering” tabbed pane is divided into two parts (see Figure 5-18). The left side of the tabbed pane provides the “Clustering Control” panel and the “Level Selection” table. The right side of the tabbed pane provides different perspectives on the data which are explained in some detail in the remainder of this section.

The “Clustering Control” panel allows starting the clustering procedure due to the current cluster strategy configuration and displays the optimal level. The “Level Selection” table is updated with the results of the clustering procedure, either the list of succeeding levels of a hierarchical clustering procedure as depicted in Figure 5-27 or the rearranged partitions of a partitioning clustering procedure. In the second case, the last entry (row) refers to the optimal solution. Using hierarchical clustering the table includes the following information for each level: level id (“L.id”), number of clusters (“#C”), fusion coefficient (“FI”), validation index (“VI”) as well as the overall (“CD”), the minimum (“Min-CD”) and the maximum (“Min-CD”) content dissimilarity of the level. The same data is provided for partitioning clustering, except for fusion coefficient and validation index which are only defined on hierarchies.

After selecting a level the “Clusters” table and “Nodes” table are updated (see Figure 5-27). The clusters table contains the list of clusters (“id”) of the selected level together with the

number of nodes in this cluster (“#N”). If a cluster is selected from this table, the “Graphview” tabbed pane as well as the “Cluster Details” tabbed pane are updated. The nodes table consists of the list of all active nodes (“id”) and the number of clusters they appear in (“#C”). If a node is selected from this table, the node is marked with a red ring in each graphview and the “Node Overview” tabbed pane is updated.

5.2.7.3.2 Graphview

The “Graphview” tabbed pane contains different visual approaches to investigate on the selected clustering solution in general (“Representative” tabbed pane) and a selected cluster from this solution (“Network” tabbed pane, “Cluster” tabbed pane and “Temperature View” tabbed pane). These components are slightly different from those available for content-based clustering on nodes (see section 5.2.7.2.2).

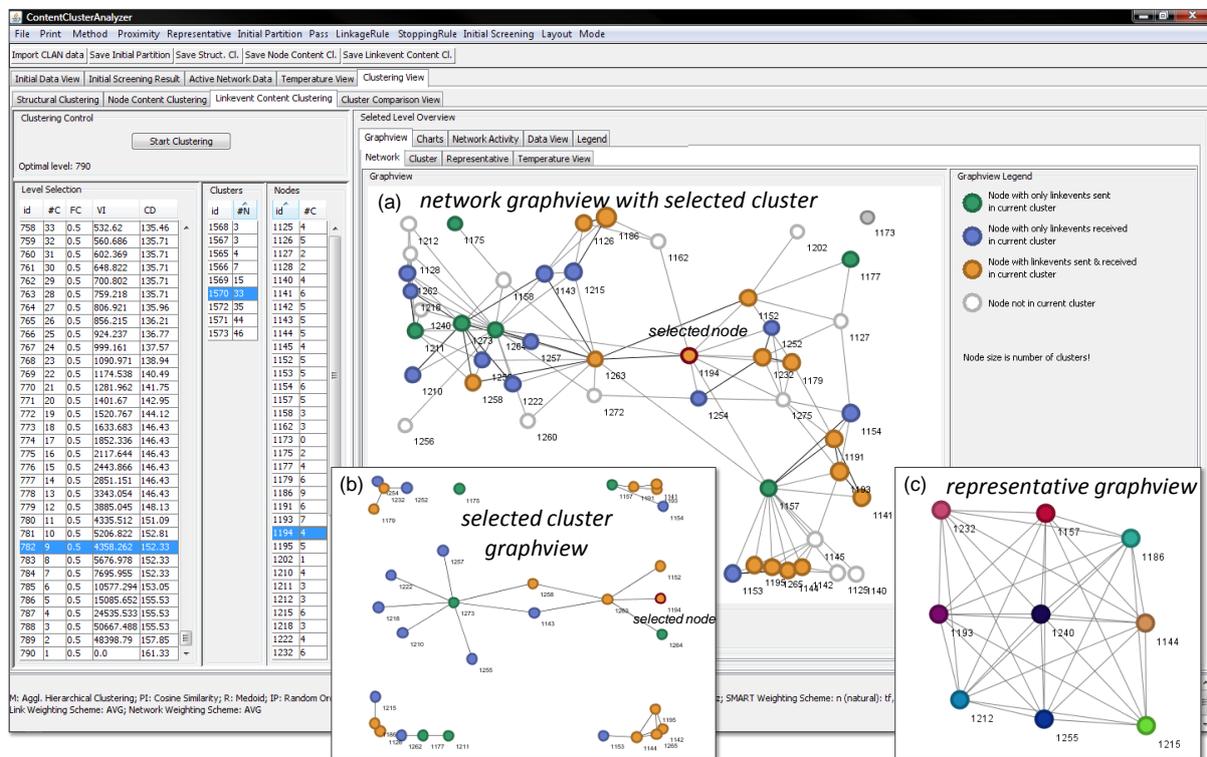


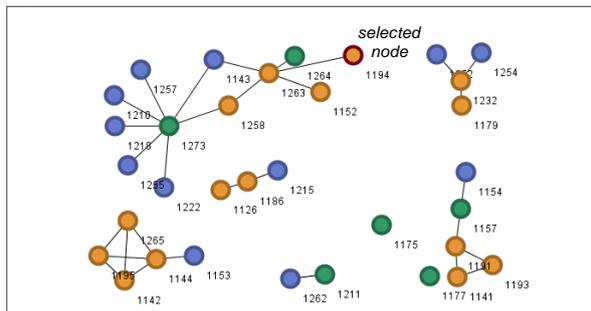
Figure 5-27: Content-based clustering on linkevents. Graphview: (a) network, (b) cluster and (c) representative

The “Network” tabbed pane provides a graphview of the entire network with gray nodes representing inactive nodes. Initially, all active nodes are painted as white nodes. The node size corresponds to the number of clusters the node appears in. If a cluster is selected from either the “Clusters” table or the “Level Overview” tabbed pane the graph is updated and all nodes within this cluster are painted according to the activity type in the cluster, i.e. green, blue or orange (see Figure 5-27 and section 5.2.7.2.2).

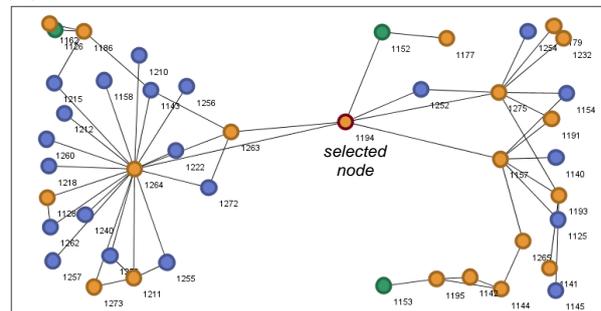
The “Cluster” tabbed pane provides a graphview of the selected cluster. In contrast to the network graphview only the intra-cluster nodes and links are visualized in the graph. As shown in Figure 5-27 the selected cluster consists of several unconnected components. The selected node 1194 is well-connected in the entire network but a peripheral node in the main component of the cluster. Figure 5-28 presents the graphviews of the four clusters this node is

assigned to. Here, one can see that the node obtains different roles in the clusters depending on its intra-cluster activity. With the exception of cluster 1570 (Figure 5-28 a)) all clusters are totally connected with only one component. In cluster 1570 (Figure 5-28 a)) node 1194 is peripheral with intra-cluster linkevents sent and received. In cluster 1571 (Figure 5-28 b)) node 1194 is a cutpoint between the two parts of the graph with intra-cluster linkevents sent and received. Removing this node will result into two larger components and an isolated dyad. In cluster 1572 (Figure 5-28 c)) node 1194 is peripheral with only intra-cluster linkevents received. In cluster 1573 (Figure 5-28 d)) node 1194 is well-connected in the center of the network with intra-cluster linkevents sent and received. However, the connectivity of this cluster does not solely depend on a single node as in cluster 1571.

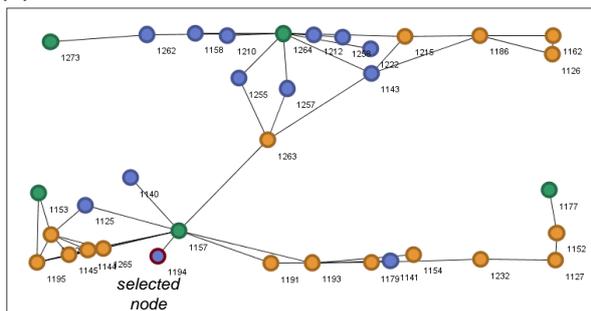
(a) cluster 1570



(b) cluster 1571



(c) cluster 1572



(d) cluster 1573

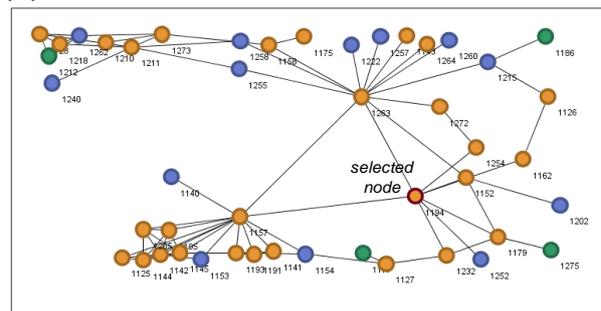


Figure 5-28: Content-based clustering on linkevents. Different cluster graphviews for selected node 1194

The network graphview and the cluster graphview help to investigate on the structure of a cluster, its embeddedness in the overall network structure as well as the position and behavior of its members.

As the content-based clustering on linkevents yields overlapping node clusters it is not possible to map all clusters of a level on a single graph of the entire network. Therefore, the “Representative” tabbed pane contains a graphview of all cluster representatives. The links represent the inter-cluster relationships between each cluster. An example is given in Figure 5-27. The representative view helps to understand the interconnection of the different clusters.

Similar to the temperature view of the entire network (see section 5.2.5) the “Temperature View” tabbed pane provides the temperature view for the selected cluster. The temperature view of the selected cluster helps to investigate on the content similarities within a cluster. This view offers the same component as for content-based clustering on nodes (see section 5.2.7.2.2). It helps to validate the quality of the selected clustering solution. Rather dissimilar clusters will indicate that the current solution has a low quality whereas rather similar clusters will indicate a good solution.

5.2.7.3.3 Charts

The “Charts” tabbed pane provides the same charts for hierarchical and partitioning content-based clustering on linkevents as for content-based clustering on nodes (see section 5.2.7.2.3).

5.2.7.3.4 Network Activity

The “Network Activity” tabbed pane provides the same plots of inter-cluster and intra-cluster network activity for content-based clustering on linkevents as for content-based clustering on nodes (see section 5.2.7.2.4).

5.2.7.3.5 Data View

The “Level Overview” tabbed pane contains a table with properties and metrics for each cluster of the selected level. The entire list is given in Table B-4 in the appendix B, section B.2.2. If one entry is selected, the corresponding graphviews and the “Cluster Details” tabbed pane are updated.

To further investigate on a selected cluster the “Cluster Overview” tabbed pane and the “Cluster Details” tabbed pane provide detailed information about the node. The “Cluster Overview” tabbed pane contains a table with properties and metrics for each node of the selected cluster. The data provided by this table is almost the same as in the “Level Details” tabbed pane for content-based clustering on nodes. The difference is that for content-based clustering on linkevents this table only contains the data for the selected cluster and not for all clusters of the selected clustering solution. The entire list is given in Table B-2 in the appendix B, section B.2.1.2. If one entry is selected, the corresponding node is marked with a red ring in all graphviews and the “Node Overview” tabbed is updated.

Besides the visual inspection of within-cluster activities using the cluster graphview and the within-cluster content similarities using the temperature view the “Cluster Details” tabbed pane can be employed to gain a deeper understanding of the selected cluster. It provides numeric and visual information about the selected cluster. This view contains the same components as for content-based clustering on nodes (see section 5.2.7.2.5 and Figure 5-25).

To further investigate on a selected node the “Node Overview” tabbed pane and the “Node Details” tabbed pane provide detailed information about the node. The “Node Overview” tabbed pane contains a table with quantitative information about the selected node in its clusters and in the entire network. Each node can belong to several linkevent-based clusters. The membership of a node (“N”) in a linkevent-based cluster is called node role (“NR”). The data provided by the node overview table for content-based clustering on linkevents are the same as for content-based clustering on nodes see section 5.2.7.2.5 and Table B-3 in the appendix B, section B.2.1.3. If a node, or more precisely a node role, is selected from this table the “Node Details” tabbed pane is updated. The “Node Details” tabbed pane contains numeric and visual information about the selected node. It consists of the following elements: the topics pie charts, the topic occurrences table, the node roles table and the node role activity view. This view contains the same components as for content-based clustering on nodes (see section 5.2.7.2.5 and Figure 5-26).

The “Cluster Membership” tabbed pane provides information about which nodes are grouped together in the same clusters.

5.2.7.4 Cluster Comparison View

The cluster comparison view allows to compare selected clustering solutions (“levels”) from the three different types of clustering procedures, i.e. structural clustering, content-based clustering on nodes and content-based clustering on linkevents. The metrics used for cluster comparison are explained in section 5.1.5.3. The metrics provided by the “Cluster Comparison View” tabbed pane are calculated only for active nodes, $n \in N_a$.

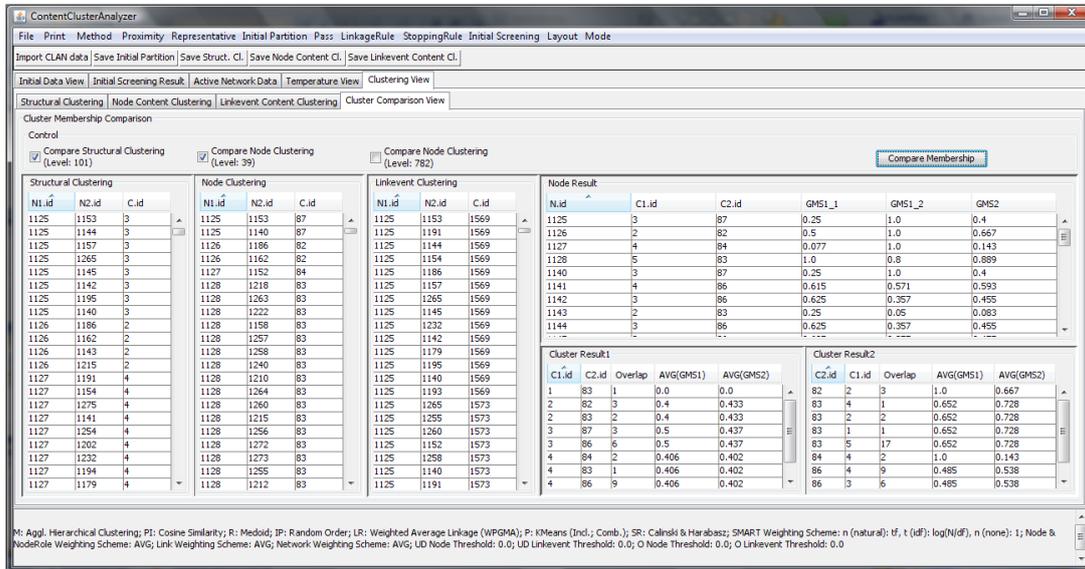


Figure 5-29: Cluster comparison view

For example, Figure 5-29 shows the “Cluster Comparison View” tabbed pane with the three selected levels presented in sections 5.3.5.1 to 5.3.5.3. The levels from structural clustering and content-based clustering on nodes are selected for cluster comparison. As a result, the “Node Result” table as well as the “Cluster Result1” and the “Cluster Result2” tables are updated. The labels “C1.id”, “GMS1_1” and “Cluster Result1” always refer to the left clustering solution in this view whereas the labels “C2.id”, “GMS1_2” and “Cluster Result2” always refer to the right clustering solution. In Figure 5-29 the labels containing the cipher “1” refer to the structural clustering solution and those containing the cipher “2” refer to the node content clustering solution.

The node result table contains the list of all active nodes together with one cluster from each of the selected clustering solutions (“C1.id” and “C2.id”) and some metrics of group membership stability (“GMS1_1”, “GMS1_2” and “GMS2”). Each cluster result table contains the list of all pairs of solutions (“C1.id” and “C2.id”) that share at least one node ($overlap > 0$) together with the cluster-centric overlap (“Overlap”) and some aggregated metrics of cluster-centric group membership stability (“AVG(GMS1)” and “AVG(GMS2)”).

The cluster comparison view helps to compare two clustering solutions from different types of clustering procedures to gain an understanding about cluster membership stability and dissociation on node level and cluster level.

5.3 Case Study “ Corporate E-Mail Exchange”

This case study provides an application of the method for knowledge identification in social corpora to a corporate e-mail data set to evaluate its benefits in business context. It is based on the research guideline proposed in section 5.1 using the prototype presented in section 5.2. This section is organized as follows: First, a brief introduction to the data set is given (see section 5.3.1). Afterwards, the data is prepared for network and cluster analysis including network transformation and text mining (see section 5.3.2). The subsequent initial screening of the network content covers the adjustment of variable importance, the selection of the mode of linkevent affiliation as well as noise detection (see section 5.3.3). In section 5.3.4 a preliminary network analysis is employed: SNA metrics are calculated to examine the network structure and identify important key players. Additionally, the temperature view is established on the data to investigate on the cluster tendency of the network content. Based on this preliminary network analysis in section 5.3.5 structural as well as content-based clustering is applied to the data to identify and categorize different types of node roles and clusters based on communicational interaction and shared experience. The results are compared with each other. Finally, in section 5.3.6 a summary of this case study is given that relates the results to the research questions of this work (see chapter 1, section 1.2).

5.3.1 Data Set

This case study is based on a subsample of the publicly available corporate Enron e-mail corpus. The Enron Corporation was an American energy company based in Houston, Texas, with business segments in electricity, natural gas, communications as well as pulp and paper industry (Fusaro and Miller 2002; Fox 2003). In 2000, Enron claimed revenues of over \$100 billion (Huntsman 2010). However, it filed for bankruptcy in late 2001 after it was uncovered that the reported financial condition was sustained substantially by institutionalized, systematic and creatively planned accounting fraud. This has become widely known as the "Enron scandal". In the United States, the downfall of Enron brought into questions the general accounting practices and business activities of many other companies. It therefore had strong influence on the creation of the Sarbanes-Oxley Act of 2002 which is “an act to protect investors by improving the accuracy and reliability of corporate disclosures made pursuant to the securities laws, and for other purposes” (Sarbanes-Oxley Act of 2002).

The Enron e-mail data set was made public by the Federal Energy Regulatory Commission during its investigation. It was later collected and prepared by Melinda Gervasio at the Stanford Research Institute for the CALO⁴¹ project. It contains the e-mails from former Enron senior executives and energy traders. The original data set suffers from several integrity problems. Most of these problems in the data set have been resolved by the CALO project. Some of the e-mails have been deleted due to requests from affected employees. William Cohen from the Carnegie Mellon University has made the data set available via internet for researchers⁴².

⁴¹ CALO:= A Cognitive Assistant that Learns and Organizes

⁴² Available at <http://www-2.cs.cmu.edu/~enron/>

This data set is a rare example of publicly available corporate communicational data. It therefore has been widely used for network and content analysis, e.g. analysis of language usage and semantic ambiguity arising with natural language (Kessler 2010), automatic classification and labeling of e-mails (Bekkerman et al. 2004; Klimt and Yang 2004), identification of communities and their evolution based on network structure (Chapanoud et al. 2005; Diesner et al. 2005; Falkowski et al. 2008), clustering of emails based on content analysis (Li et al. 2006; Bobrik and Trier 2009), topic and role discovery in social networks (McCallum et al. 2005), characteristics and patterns of communicative behavior (Diesner et al. 2005), or comparison of different strategies of searching for knowledge in digital social networks (Zhang and Ackerman 2005; Trier and Bobrik 2008b; VanBuren et al. 2009).

The subsample used in this case study consists of 4,150 internal e-mails exchanged between 112 non-isolated Enron employees on management level from September 1st, 2000, to March 30th, 2001. This subsample has been chosen for several reasons: On the one hand, this was done to prevent the analysis to be overcrowded with too many peripheral actors with minimal impact on the network and to focus on the internal collaborative relationships (focus on Enron employees). On the other hand preliminary studies on the Enron data set and its context have shown that within the selected period of time important communication threads can be found as it covers the California electricity crisis which led to the downfall of the Enron company (focus on selected period of time). Two additional reasons for the subsample selection had also to be considered. First, the selected period is more stable in the presence and activity of the network participants than earlier or later periods. Second, as the analysis of subgroups is more interested in larger collections of actors than in single individuals a window size has to be chosen where large and dense network structures can emerge. A much shorter period would yield networks with many unconnected, stringy components where the results from the structural as well as content-based clustering procedures are obvious but trivial. This subsample has already been used in Bobrik and Trier 2009 (2009). Several other publications of the IKM Research group are based on the Enron data set, e.g. Trier and Bobrik (2007a) or Trier and Bobrik (2008b).

5.3.2 Data Preparation

The first step in the analysis using the new method for knowledge identification in social corpora is the preparation of the data set. It covers the transformation of the data into a network representation (see section 5.3.2.1) and the application of text mining procedures (see section 5.3.2.2).

5.3.2.1 Network Transformation

Shetti and Adibi (2004) provide a download of a MySQL database containing a refined version of the corporate Enron e-mail data set. The original data set consists of 517,431 e-mails from 151 users distributed in 3,500 folders. These messages do not include attachments. Each message contains the e-mail address of senders and recipients, date and time, subject, body text and some other e-mail specific technical details. Some e-mail addresses were changed, e.g. invalid addresses. Other e-mails were entirely removed, e.g. messages returned by the email system as an email transaction failure or messages without any content. The cleaned Enron e-mail data set contains 252,759 messages from 151 employees.

As the data set already consists of messages it can be easily transferred to the SNI data model by modeling the messages as linkevents and their senders and recipients as nodes (see section 2.2.1). Subject, body text and message date are retrieved as linkevent properties, name (given name, last name) and e-mail address are retrieved as node properties. Shetti and Adibi also offer some information regarding the status of every employee in the organization hierarchy⁴³. This information is essential when studying the information flow in an organization. Thus, the organizational status is included as further node property. The list of Enron employees is given in Table C-1 and Table C-2 in the appendix C, section C.1. In the context of this work the nodeID is used to identify the Enron employees unless name and organizational status information are required. A time-based filter is applied to retrieve a subsample of the data covering only e-mails between September 2000 and March 2001. The generation of a time-based subsample has to be part of the data preparation step as the text mining process makes use of the size of the content collection to calculate topic weights. Limiting the number of linkevents to a subperiod will affect the results from the text mining process.

5.3.2.2 Text Mining

The content analysis is based on the subject and body text of each e-mail represented as linkevent. A semi-automatic text mining process is applied to the data to reduce each linkevent to a set of topics weighted by their importance (see section 3.2). The automated text mining methods presented in Table C-3 in the appendix C, section C.2, are employed in the given order to perform linguistic preprocessing, feature generation and feature selection using the ContentMiner software (see section 5.1.1.2). The methods, their specific order and their configurations are due to a thorough preliminary analysis of the data. For example, as the body texts often miss whitespaces long topics occur in the data that consist of two or more elements. Here, the WordSplitter tries to reconstruct the right topics and the TermLength filter will remove unusually long topics.

The effect of the subsequent steps of the automated feature selection process on the number of topics are shown in Figure 5-30. The originally retrieved topics are plotted as blue dots indicating their term frequency (x-axis) and document frequency (y-axis). Topics that will be removed by the selection step are marked as red dots and omitted in the following step. Topics that will be added by the selection step are marked as green dots and painted blue in the following step. The centroid of all topics is marked by a red circle. Its position is stable during the feature selection process but heavily affected by the final step of removing low-frequency terms.

⁴³ Available at <http://www.isi.edu/~adibi/Enron/Enron.htm>

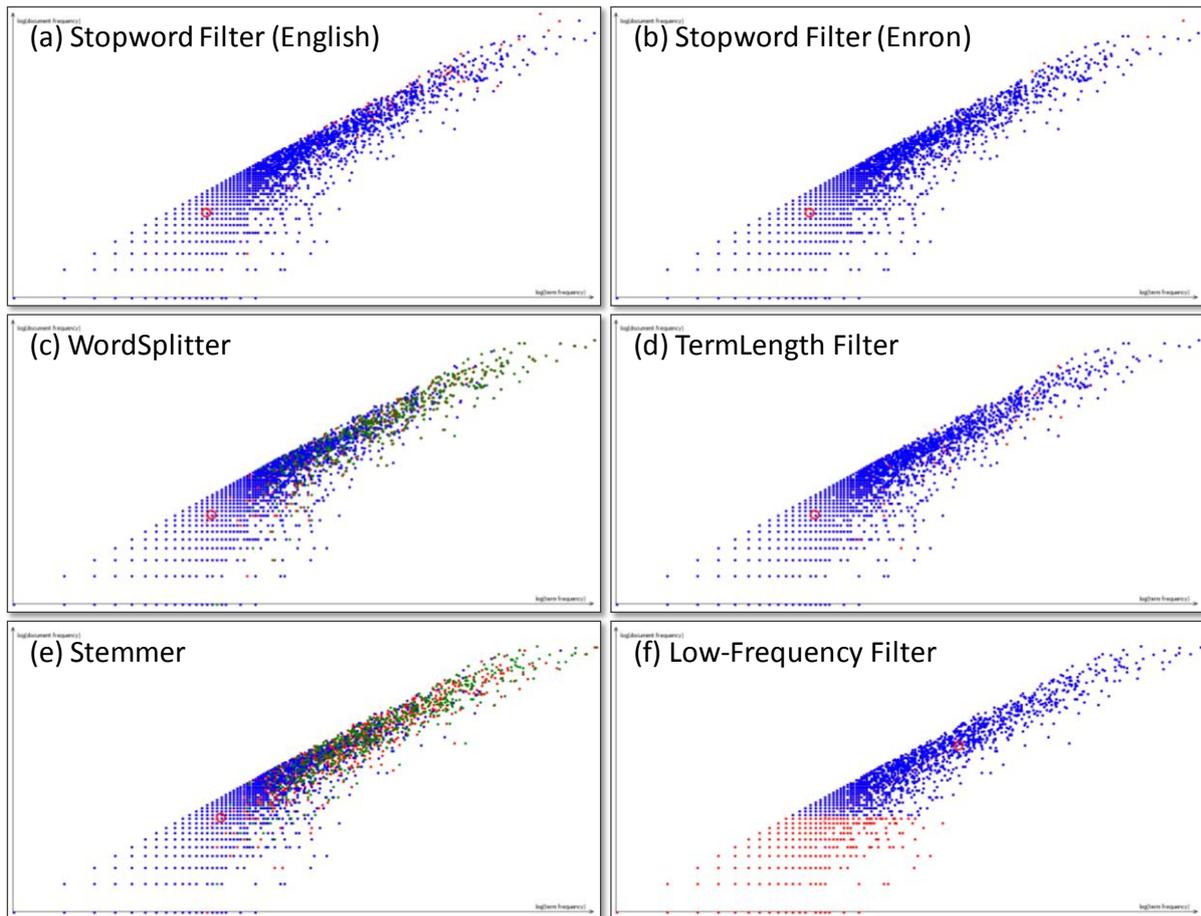


Figure 5-30: Case study. Results from automated feature selection. Axes: x-axis: $\log(\text{term frequency})$; y-axis: $\log(\text{document frequency})$. Topics plotted as blue dots; topics removed marked as red dot; topics added marked as green dot; centroid of all topics marked by a red circle

Table 5-13 gives an overview of term reduction by the different text mining methods that have been applied to the data.

Table 5-13: Case study. Term reduction by different text mining methods

Method	Topics	%
Tokenizer with POS Tagger	17,913	100%
(a) Stopword Filter (English)	-289	-1.61%
(b) Stopword Filter (Enron)	-19	-0.11%
(c) WordSplitter	-1,384	-7.73%
(d) TermLength Filter	-279	-1.56%
(e) Stemmer	-2,317	-12.93%
(f) Low-Frequency Filter	-11,702	-65.33%
Total	1,924	10.74%

Further topics were removed by a final manual inspection of the topic list, e.g. misspellings and wrong tags. Finally, the data set contains of 1,812 topics which is 10.12% of the original topics retrieved by the POS Tagger.

5.3.3 Initial Screening

The process of initial screening covers the determination of variable importance (see section 5.3.3.1), the selection of the mode (see section 5.3.3.2) and the detection and removal of noisy data (see section 5.3.3.3).

5.3.3.1 Variable Importance

Using the SMART weighting scheme on linkevents, the *tf.idf* weighting scheme is applied with natural term frequency (*tf*), inverted document frequency (*idf*), and no normalization. No standardization of linkevents is applied. Average topic occurrence is used as node, link and network weighting scheme.

5.3.3.2 Mode

The content collection of a node is based on the linkevent sent as well as received.

5.3.3.3 Noise Detection

The text mining process may result into linkevents where no meaningful topics have been retrieved. After screening the data for linkevents which do not contain any topic 70 linkevents are identified as uninformative data and removed from further analysis. As a result, two links have no active linkevents. Therefore, they are removed from the data as well. To improve the clustering tendency of the data, 461 linkevents with less than 5 topics are removed as well. As a result, 16 links have no more active linkevents and are removed from further analysis. There are two nodes (1161 and 1244) without any meaningful content which are removed as uninformative data.

In contrast to the removal of uninformative data which only takes the number of topics into account outlier detection is based on the content similarity of either linkevents or nodes regardless of the number of topics. To detect linkevent outliers an outlier threshold $\theta_{diss} \leq 0.4$ is used. Comparing the content dissimilarities of each linkevent with all other linkevents a linkevent is removed from the data if all of its content dissimilarities exceed this threshold. As the cosine similarity is used to calculate proximities the results have to be converted into dissimilarities using the first formula from section 4.3.2. A dissimilarity threshold $\theta_{diss} \leq 0.4$ is equivalent to a similarity threshold $\theta_{sim} > 0.2$. There is no linkevent outlier in the data. The same threshold is used to detect node outliers. As a result, node 1189 and 1209 can be regarded as outliers and are removed from further analysis. Moreover, eleven linkevents and three links have no active senders and are also excluded.

5.3.4 Network Analysis

After the preparation of the data and the initial screening a preliminary network analysis is employed to examine the network structure and identify important key players (see section 5.3.4.1). Furthermore, the content profiles and their distribution in the network provided by the temperature view help to detect important key players based on content similarity and network structure and determine the clustering tendency of the data (see section 5.3.4.2).

5.3.4.1 SNA Metrics

In this section SNA metrics are calculated on network level and node level to detect meaningful subgroups and important key players based on the structure of the network.

5.3.4.1.1 Network Level

After initial screening the network consists of 108 nodes and 3,616 linkevents distributed on 413 links. It covers the e-mail exchange of Enron employees between September 1st, 2000, and March 30th, 2001. For the entire network the density is 7.15% and the link strength is 8.76 e-mails per link on average. The network does not contain self-links.

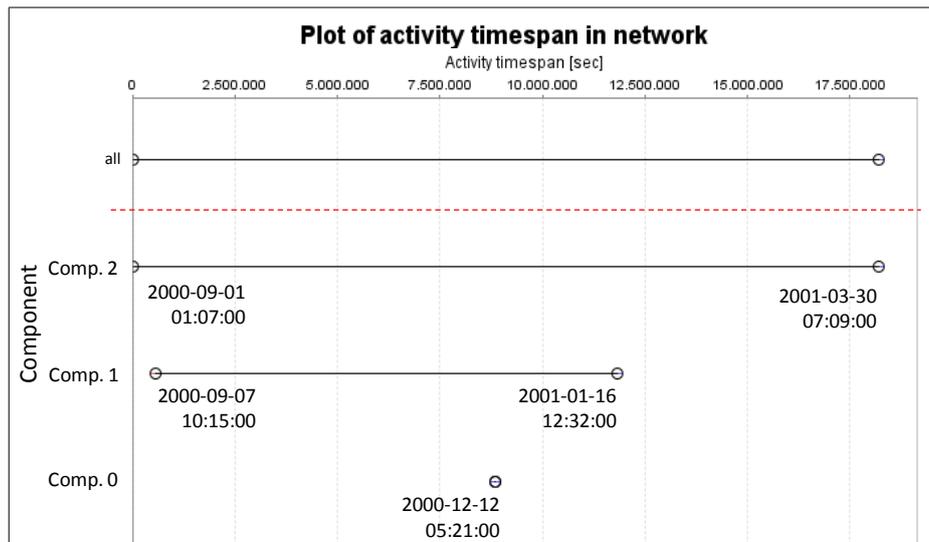


Figure 5-31: Case study. Network activity plot

There are three unconnected components (see Figure 5-32): two rather small components with two (nodes 1147 and 1243) and five nodes (nodes 1159, 1160, 1233, 1238 and 1249) and a main component with the majority of the nodes (101 nodes). The plot of the network activity of the entire network and the three components is shown in Figure 5-31. The smallest component (comp. 0) participates only on a single day. The activity of the other small component (comp. 1) covers about 62% of the entire network activity. The actors of this component are not active in the last third of the sample period. The main component (comp. 2) is active during the entire sample period.

An overview of the most important SNA metrics calculated for the entire network and the three components is given in Table 5-14. The main component consists of 101 nodes, 3,603 linkevents and 408 links. The density is 8.08% and the link strength is 8.83 e-mails per link on average. The diameter only amounts to a path length of 5 steps and the average path length between every pair of at least indirectly connected actors is 2.75 steps. The global clustering coefficient of the subsample is 51.46%. This indicates how many contacts of all actors are connected themselves.

Table 5-14: Case study. SNA metrics. Network level

Comp.	#N	#LE	#L	Density	ØLS	Diameter	ØDC	ØBC	ØCC	GCC	R.id
All	108	3616	413	7.15%	8.76	5 ¹⁾	11.26%	2.88%	2.67	48.12%	1198
0	2	2	1	100%	2.0	1	100%	0%	1.0	-	1243
1	5	11	4	40%	2.75	2	40%	26.67%	1.8	0%	1159
2	101	3603	408	8.08%	8.83	5	8.08%	1.76%	2.75	51.46%	1198

Comp. := component ØLS:= average link strength GCC:= global clustering coefficient
 #N:= number of nodes ØDC:= average degree centrality R.id:= Representative of component (due to content similarity)
 #LE:= number of linkevents ØBC:= average betweenness centrality
 #L: number of links ØCC:= average closeness centrality 1) maximum diameter of largest unconnected component

5.3.4.1.2 Node Level

The two small components of the network can be neglected in terms of size and activity. Thus, only the main component is taken into account.

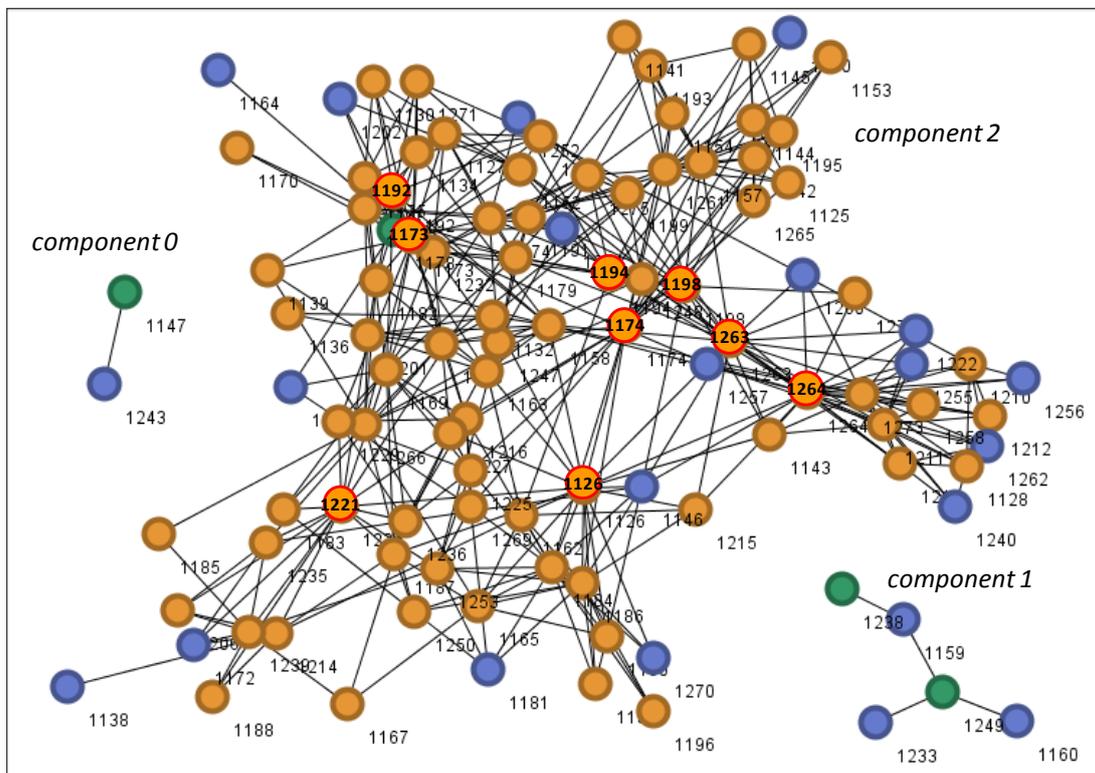


Figure 5-32: Case study. Graphview. Structural key players highlighted with red ring. Node color indicates type of interaction: green=only linkevents sent; blue=only linkevents received; orange=linkevents sent and received

Based on the top list of several node metrics some actors can be identified as key players in terms of information distribution (number of linkevents), activity and popularity (degree centrality), strength of interaction (average link strength), brokerage (betweenness centrality), access of resources and efficiency (closeness centrality) or connectivity (local clustering coefficient). An overview of these actors together with the most important metrics is provided in Table C-4, in the appendix C, section C.3.1. Additionally, each node can be assigned to one of the node role categories provided in Table 5-5, section 5.1.5.1. The node role

categorization is calculated per component. The data is given in Table C-5 and Table C-6 in the appendix C, section C.3.2.

Within the main component there are 15 peripheral specialists at the rim of the component only connected to a few other nodes, 82 middlemen that connect the center of the component with smaller outer parts and peripheral nodes, one mediator acting as an information distributor between larger subgroups, and three integrators that are central information provider and distributor within larger subgroups. There is no hot spot which indicates that the core of the main component is more distributed than centralized. However, as there is also no bridge or broker the main component is still well connected and its subgroups do not tend to drift apart from each other.

Taking a closer look on the top list of SNA metrics, there are some nodes which are among the top ten in several metrics indicating significant positions in the network. Most notable among these structural key players are nodes 1126, 1173, 1174, 1192, 1194, 1198, 1221, 1263, and 1264. These nodes are highlighted in the graphview of the entire network in Figure 5-32.

Nodes 1173, 1263 and 1264 act as integrators within the main component due to high degree and betweenness centrality values. Node 1173 is second best in number of contacts and among the top ten in terms of information distribution, strength of interaction, information brokerage and access of resources. Her clustering coefficient is rather low. Node 1263 has the best access of information and is second best in terms of activity and popularity and third best in terms of information brokerage. She is among the top ten in terms of information distribution covering almost the entire sample period. Node 1263 is Enron's house lawyer Mary Hain. Node 1264 has the highest number of contacts and is best in information brokerage and second best in access of resources. She has comparably few messages sent or received and therefore a low strength of interaction. She enters the network rather late and leaves it early.

Node 1126 is the only mediator of the main component. He has optimal access to the resources of the network and is second best in information brokerage and third best in activity and popularity. Node 1126 is Enron's manager Mike Grigsby.

When examining their position in the network (see Figure 5-32) the three integrators form an axis in the upper part of the main component. Mediator 1126 is the connector between the less connected upper and lower part of the main component. The other five structural key players act as middlemen between smaller subparts and the center of the main component.

Node 1174 is among the top ten in terms of information distribution, network activity information brokerage and access of resources. She contributes over almost the entire sample period. Node 1192 has the highest strength of interaction and third highest information distribution but is ranking only in the upper third in terms of activity and popularity as well as access to resources and is especially low in information brokerage. Node 1194 is second best in information distribution and strength of interaction covering almost the entire sample period. He is quite good but not top in terms of activity and popularity. He is among the top ten in terms of access of information but has a rather low clustering coefficient. Node 1194 is Enron's vice president for regulatory affairs Richard Shapiro.

Node 1198 has the highest number of linkevents covering almost the entire sample period with 759 e-mails sent and 226 e-mails received. He is third best in strength of interaction and access of information and among the top ten in terms of activity and popularity as well as information brokerage. He has a comparably low clustering coefficient. Node 1198 is Enron's government relation executive Jeff Dasovich.

The maximum degree is 24 contacts (node 1264). Interestingly, the most active actor and the most connected actor are not the same person. Moreover, the most connected node 1264 has a much smaller network activity with 47 e-mails sent and 53 e-mails received.

Node 1221 is among the top ten in terms of all three centrality measures but has comparably few messages sent or received covering a rather short period. Node 1221 is Enron's vice president Scott Neal.

In summary, high communicational activity is strongly related with a prominent structural position but high values in terms of connectedness, information access or brokerage can also be obtained with less communicational affords. Due to their prominent positions in the unclustered network the set of structural key players should be taken into account in the subsequent steps of the analysis. Their prominent position among is also emphasized by their organizational status as vice presidents or house lawyer responsible for regulatory affairs or governmental relations pointing at the Enron market manipulation and bankruptcy scandal.

5.3.4.2 Temperature View

In contrast to the use of SNA metrics to identify structural key players the temperature view makes use of the pairwise content similarities between all nodes. The temperature graphview maps the resulting content profiles to the network structure. This helps to further investigate on the structural key players and to detect content-based key players. Furthermore, some insights on the cluster tendency of the data can be gained. For more details see section 5.1.3.2.

When inspecting all the content profiles presented in the temperature overview in Figure C-1 in the appendix C, section C.3.3.1, different types of content profiles can be identified. This allows grouping together nodes with similar profiles. The different types of content profiles can be obtained by ordering the nodes by the average and maximum similarities of their content profiles. As a result, seven groups of similar content profiles can be identified. The temperature overview with ordered content profiles is depicted in Figure C-2 in the appendix C, section C.3.3.1.

Table C-7 in the appendix C, section C.3.3.2, provides an overview which nodes have a similar content profile. Additionally, the content profile and the temperature graphview of an exemplary node are given for each group. The structural key players identified in section 5.3.4.1 are marked bold (top 9) or underlined (top 21). Notably, the majority of the nine structural key players belong to the first and second group. Only one of them belongs to group 4. The nodes assigned to groups 1 show dichotomized content profiles with a few high similarities and many low or medium similarities. The nodes in group 2 have similar profiles but miss very high similarity values. Although the nodes in each group have similar content profiles, they can differ when mapping the content similarities to the network structure (temperature graphview). The nodes within group 1 and 2 all belong to the main component. They cannot be assigned to structural subgroups but are distributed through the entire

component. When comparing their graphviews some specific features within and between the seven groups of content profiles can be discovered.

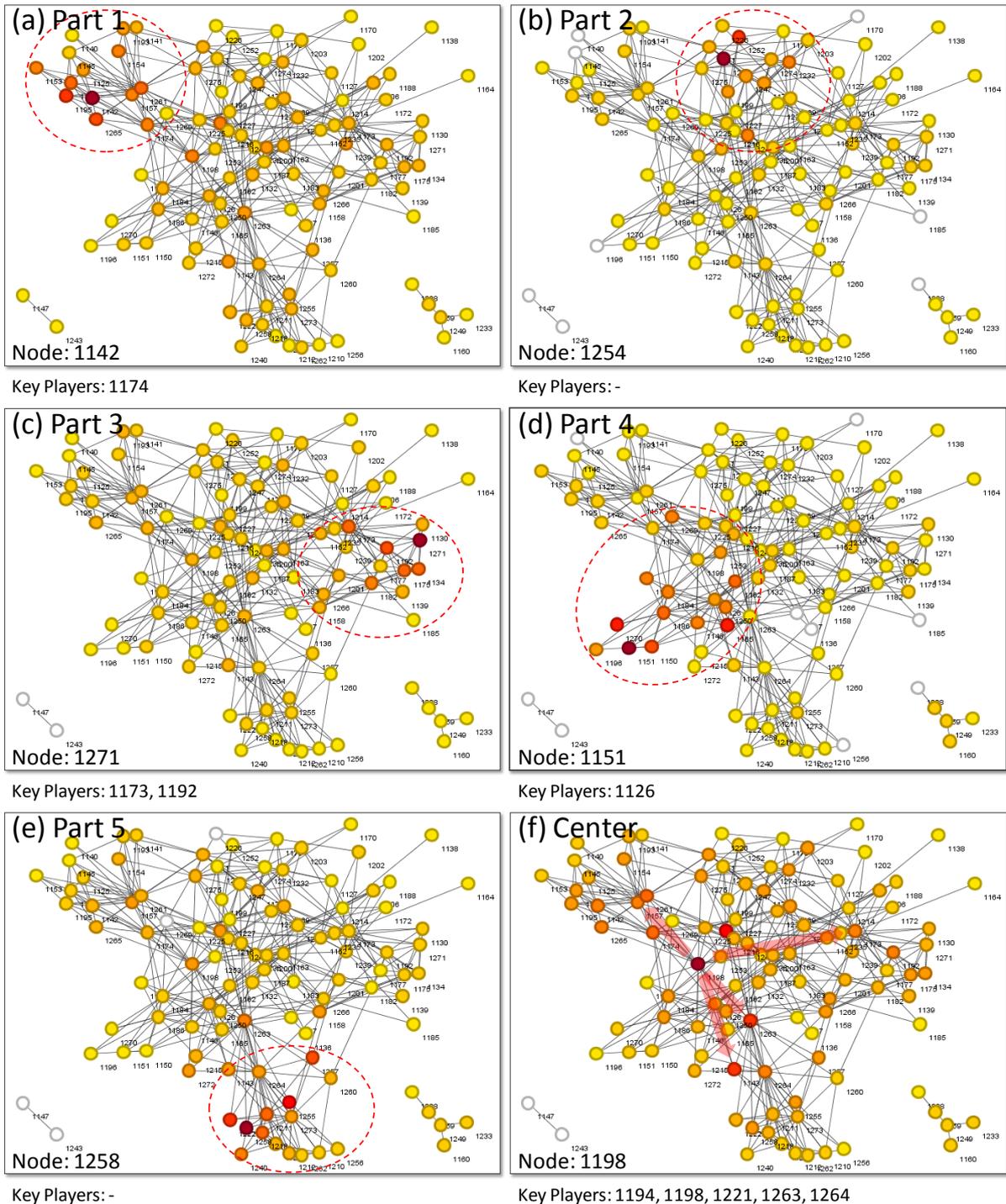


Figure 5-33: Case study. Mapping content to network structure. Parts of network with similar graphviews. Exemplary temperature graphviews

For each content profile, there is a corresponding graphview that maps the pairwise content similarities of the profile to the network structure. Similar content profiles can belong to very different structural distributions within the network but also groups of similar graphviews can be identified. When inspecting all temperature graphviews some areas of high content similarity can be identified. As shown in Figure 5-33 a) to e) there are five distinct areas (or parts) in the network that can be distinguished by mapping the content profiles to the network

structure. These areas indicate that the content-based clusters obtained by directly clustering nodes will be similar to the structural clusters and that nodes can be divided by their content similarities, merging those nodes with high similarities. As there are only few nodes with high pairwise similarities a rather fine grain clustering solution will be obtained.

There are several nodes that cannot be assigned to one of these network parts. The majority of these nodes are located in the “center” of the network (see Figure 5-33 f)). In contrast to the five parts of high content similarity, these central nodes have either rather homogenous content profile with medium similarities or the few nodes with high content similarities are spread through the entire network. The content of these nodes seems to integrate the distant parts of the network and the affiliated nodes gather and spread very different information acting as boundary spanners. Most of these nodes are structural key players, and both node 1198 (highest number of linkevents) and node 1264 (highest degree and betweenness centrality) belong to this group of nodes.

Two of the integrators (nodes 1263 and 1264) belong to this central part of the main component. This indicates that their content profiles and their temperature graphviews emphasize their global integrative role of people and content in the network as indicated by their structural node roles. In contrast, the third integrator, node 1173, provides a distribution of its content similarities that indicates that her influence on the network is more limited. Thus, her structural node role should better be interpreted as local integrator.

Mediator node 1126 belongs to part 4. He has no global distribution of his content similarities. However, he is able to establish and improve contacts and relationships in his local context and therefore has the potential to fill the gap between otherwise less connected subgroups as indicated by the analysis of his structural node role (see previous section 5.3.4.1.2).

The other structural key players act as middlemen at the rim or periphery of the main component connecting (peripheral) specialists and other middlemen with the center. Examining their content profiles these nodes mostly belong to parts 1 and 3 but also to the center. As a result they have different potentials to connect their contacts with the rest of the network. Those nodes belonging to the center can gain global access even to far different nodes whereas the other nodes help to establish local subgroups sharing a similar context.

In summary, analyzing the content profiles and temperature graphviews helps to determine whether the potentials indicated by the analysis of structural metrics and node roles can be due to the context of each node. As a result, some nodes have limited influence but nevertheless important influence in their local subgroups. Other nodes can gain global access and integrating distant parts of the network. Therefore, the slightly separated, less connected subgroups may be the result of the content and not only the structure itself.

5.3.5 Cluster Analysis

Based on the results of the preliminary network analysis the three clustering algorithms proposed in the research guideline in section 5.1 are applied to the data to identify and compare structural as well as content-based clusters (see sections 5.3.5.1 to 5.3.5.3). Based on the categorization of node roles, an overview and comparison of important key players in each of the three clustering solutions is given. Furthermore, the clustering solutions obtained by content-based clustering on nodes and linkevents are categorized as knowledge domains

(group level) and knowledge profiles (actor level). Additionally, the stability of the clustering solutions is examined in section 5.3.5.4.

5.3.5.1 Structural Clustering

The structural clustering procedure is applied to the data to obtain densely connected subgroups, i.e. nodes that are grouped together due to the intensity of relations among each other. This solution is used to examine if and how the actor join densely connected subgroups due to work interrelation. In order to gain the optimal clustering solution based on the network structure the hierarchical divisive edge betweenness clustering algorithm for graph-based community detection by Girvan and Newman (2002) is applied to the data (GN algorithm, see section 4.7.2). For each level, the modularity value quantifies the quality of the clustering solution. The optimal clustering solution of this data set splits the network into eight clusters and has a modularity value of 0.58 which indicates a strong clustering solution (see Newman and Girvan 2004: 8). The overall content dissimilarity of this solution is 9.59.

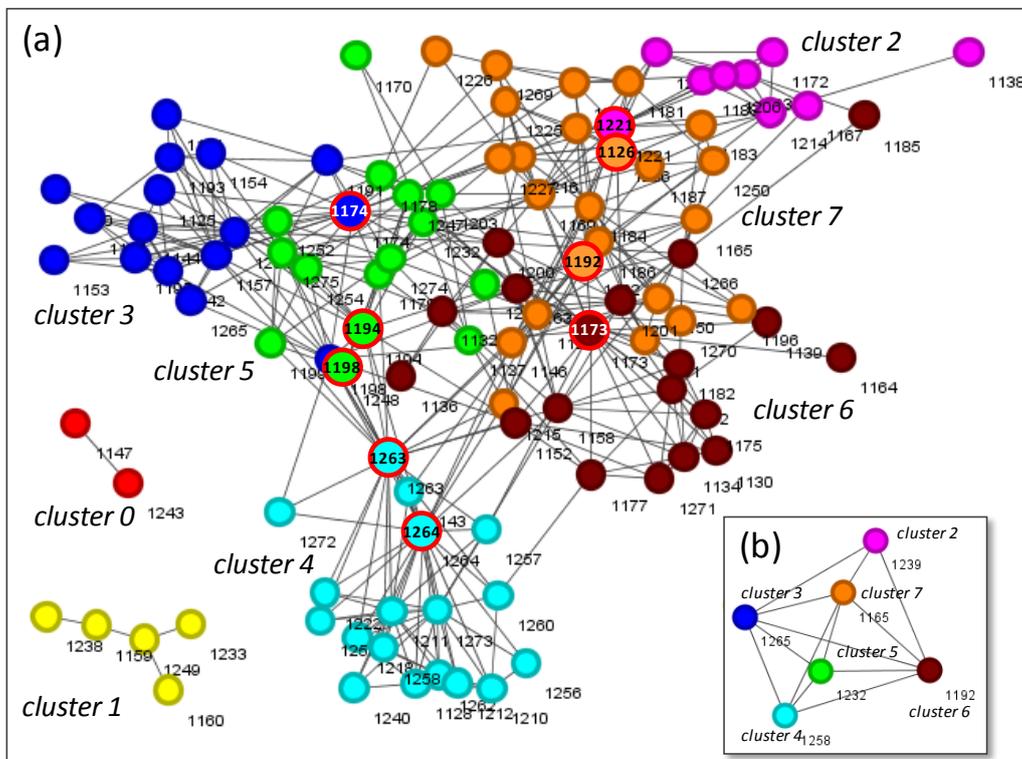


Figure 5-34: Case study. Structural clustering. Optimal solution: (a) graphview; (b) representative view (of main component). Structural key players (top 9) highlighted with red ring

In Figure 5-34 a) the graphview and in Figure 5-34 b) the representative view of the main component of this solution is given. The top 9 structural key players are highlighted with a red ring (see Figure 5-34 a)). The representatives of the clusters are those nodes that have on average the minimum content dissimilarity to all other nodes in the cluster. Regarding the representative view (see Figure 5-34 b)) there is a strong interconnection of the clusters belonging to the main component.

In Figure 5-35 the network activity time spans within each cluster and between each cluster are shown. As they are entirely separated from the main component cluster 0 and cluster 1 have no inter-cluster activity at all. Apart from these two clusters the activities of all other

clusters cover (almost) the entire sample period. Due to the small size in terms of nodes and linkevents and their separation from the main component cluster 0 and cluster 1 are neglected in the further analysis.

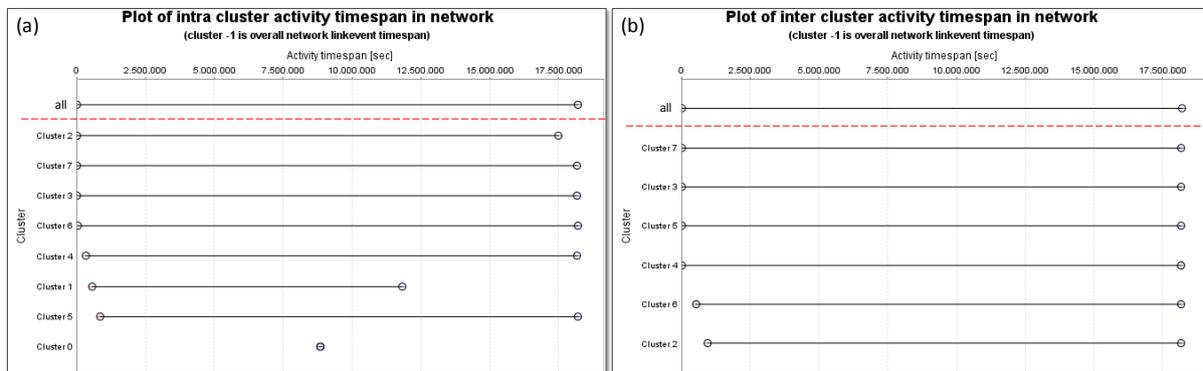


Figure 5-35: Case study. Structural clustering. Network activity view: (a) intra-cluster activity; (b) inter-cluster activity

In Table C-8 (appendix C, section C.4.1.1), an overview of the clustering solution of the main component is given. It includes the node memberships in each cluster with cluster representatives and structural key players being highlighted. Furthermore, some metrics are provided for each cluster. With more than 50 % the local clustering coefficient is high in all clusters of the main component, especially in cluster 3 (>80 %), and in cluster 4 (>70 %). Together with high density values these values indicate densely connected groups of nodes that account for a good clustering solution based on network structure. Nevertheless, as there is also a comparably high number of inter-cluster links this accounts for a strong interconnection between nodes from different clusters (see also the representative view in Figure 5-34 b)).

Comparing this clustering solution with the partitioning of the network obtained by mapping the content profiles to the graph (see section 5.3.4.2) there is a strong stability of the node memberships. Only the nodes of the central part of the network belong to several structural clusters.

Additionally, each node can be assigned to one of the node role categories provided in Table 5-5, section 5.1.5.1. The node role categorization is calculated per cluster. The data is given in Table C-9 and Table C-10, in the appendix C, section C.4.1.2.

The three integrators on network level obtain different node roles in their context indicated by the structural clusters. Node 1173 is now the central hot spot in cluster 6 with high degree and betweenness centrality values. In cluster 4, nodes 1263 has become a coordinator with increased degree centrality but decreased betweenness centrality. In the same cluster, all nodes group around node 1264 who is now the central star in a well connected subgroup. Due to a much increased degree centrality value, mediator node 1126 is now an information spreader in cluster 7. All five middlemen key players in the entire network have changed their node roles due to more or less increased degree centrality values, i.e. coordinators (node 1174 in cluster 3, node 1192 in cluster 6, and node 1221 in cluster 2) or team worker (node 1194 in cluster 5). Node 1198 has also a decreased betweenness centrality value and is now a specialist in cluster 3.

In general, due to the increased density within each cluster there are an increased number of coordinators and team workers. Some nodes gain prominent position in this local context which have been dispensable in the entire network, e.g. middleman node 1239 becomes the hot spot of cluster 2. Altogether, there are 36 team workers, 24 coordinators, ten information spreader, 23 (peripheral) specialist, two middlemen, two integrators, two hot spots and two stars in the structural clusters of the main component. As the betweenness centrality tends decrease there are no brokering nodes between less connected subgroups. However, as the edge betweenness clustering algorithm is designed to detect densely connected subgroups, most effects result from the clustering method itself.

In summary, the structural clustering solution provides an overview of densely connected subgroups in the network. The high modularity value indicates that these clusters fit the structure inherent in the data. The structural key players belong to all clusters in the main component. Their node roles based on their clustering context differ from those obtain in the entire network. Thus, densely connected subgroups provide no sufficient explanation for their prominent position in the network. There is a strong overlap between the areas of high content similarity identified by the temperature view (see section 5.3.4.2) and the structural clusters. However, there are several central nodes with wide spreading high content similarities throughout the entire network (see section 5.3.4.2). These nodes belong to very different structural cluster. Again, the explanatory power of structural clusters is limited. Therefore, content-based clustering might provide additional insights that can be related to different fields of knowledge.

5.3.5.2 Content-based Clustering on Nodes

To obtain node-based content clusters an agglomerative hierarchical clustering algorithm using the weighted average linkage rule and the cosine similarity is applied to the data (see section 5.1.4.2 and section 5.2.6). The stopping rule by Calinski and Harabasz (1974) is used as validation index for detecting an optimal clustering solution.

In Figure 5-36 a) to c) the dendrogram and excerpts of the plot of validation index versus level and the plot of overall content dissimilarity versus level are provided. In general, the data consists of two very different subgroups of knowledge as there are two sets of nodes which join rather late in the clustering process (indicated by the solid green line in Figure 5-36 a)). To gain a comparably small number of clusters a local minimum in the plot of the validation index has been chosen (see Figure 5-36 c)). This clustering solution consists of ten clusters. Inspecting the plot of overall content dissimilarities (see Figure 5-36 b)) this solution is the level just before a significant increase in the dissimilarity values. This indicates that there should be no less than ten content clusters.

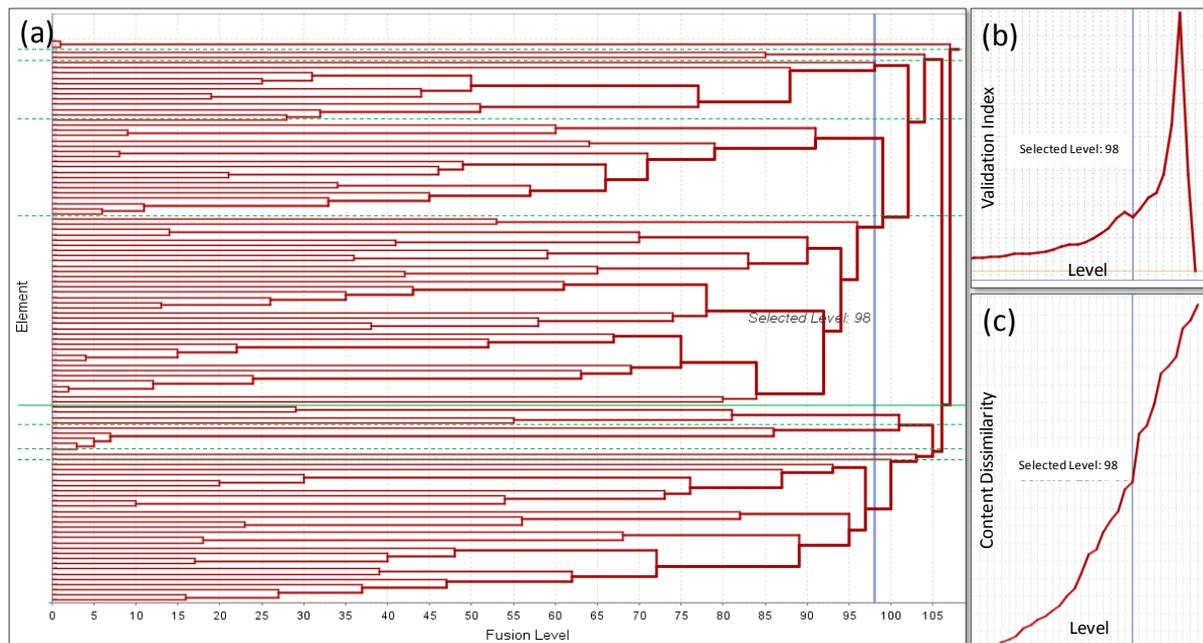


Figure 5-36: Case study. Content-based clustering on nodes. Hierarchical clustering overview (selected level marked with blue vertical line): (a) dendrogram (clusters marked with green horizontal lines); (b) plot of validation index versus level (excerpt); (c) plot of content dissimilarity versus level (excerpt)

This clustering solution is used as initial partition for a partitioning clustering procedure employing the K -means pass to improve the cluster memberships. Only one pass is necessary to gain an optimal solution indicating a good initial partition. The overall content dissimilarity is reduced from 9.24 to 9.04. This value is 5.74% smaller than the overall content dissimilarity obtained by the structural clustering procedure. This indicates that the content clusters better express groups of similar knowledge. The resulting content clusters are illustrated in Figure 5-37 a). The top 9 structural key players are highlighted with a red ring. The cluster representatives of the main component based on content similarity are presented in Figure 5-37 b). The graph of the content cluster representatives is less connected compared to the structural clustering. Thus, there is less interaction between clusters of different knowledge than between clusters of work relations.

The two small components are assigned to content clusters that are isolated from the main component. In this case, the structure of the network accounts for locally concentrated areas of similar knowledge that have neither a structural nor a content-based relation to the other nodes. In the main component there are two singletons (cluster 3 and cluster 4) and a two-node cluster (cluster 9). These nodes are well connected in the overall network structure. Examining the representative view in Figure 5-38 b) cluster 4 is peripheral whereas cluster 3 and cluster 9 are connected with the larger content clusters. Although the nodes of these clusters are separated by their knowledge they still have good indirect access to the network resources. Most notably, their roles in the overall network structure differ from those obtained by their knowledge.

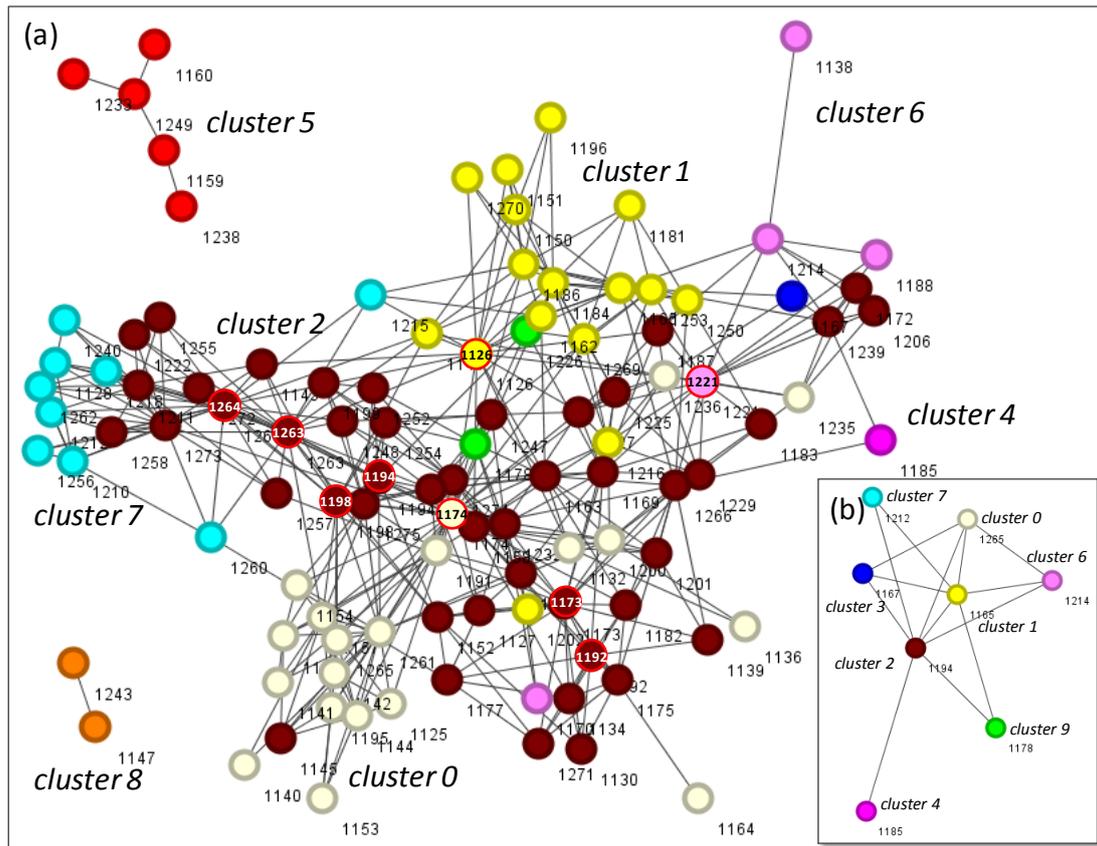


Figure 5-37: Case study. Content-based clustering on nodes. Optimal solution: (a) graphview; (b) representative view (main component). Structural key players (top 9) highlighted with red ring

There are three content clusters that are locally concentrated in the main component: cluster 0, cluster 1 and cluster 7. Cluster 0 is similar in structure and node memberships to the structural cluster 3 (see Figure 5-34 a)). However, there are also a number of node members that spread through the entire network. Here, the knowledge of this cluster integrates different parts of the network. Cluster 1 is similar in structure and node memberships to the structural cluster 7 but especially those nodes from the structural cluster located in the center of the main component belong to other content clusters. Cluster 7 is similar in structure and node memberships to the structural cluster 4. However, it is even more separated than the structural clusters as those nodes that link the peripheral nodes to the center of the main component do not share similar knowledge but belong to another cluster. The node members of cluster 2 are distributed through the entire main component of the network. The majority of structural key players belong to this content cluster. Although cluster 6 consists only of few nodes and most of them are peripheral its knowledge is connected to all three large content clusters. A single node (node 1170) is even located in the opposite part of the network.

Figure 5-38 contains the network activity view of all non-singleton clusters. With content clusters the intra-cluster activity (see Figure 5-38 a)) indicates the importance of the knowledge within a cluster. There are clusters covering only a single date, clusters with rather long activity time spans covering about 50% to 67% of the entire sample period and clusters whose members interact during the entire sample period. With content clusters the inter-cluster activity (see Figure 5-38 b)) indicates how much temporal influence the knowledge of a cluster has on the network activity. All non-singleton clusters of the main component

interact (almost) the entire sample period. This indicates that there is a strong interconnection of groups of knowledge and a good exchange of knowledge between these groups.

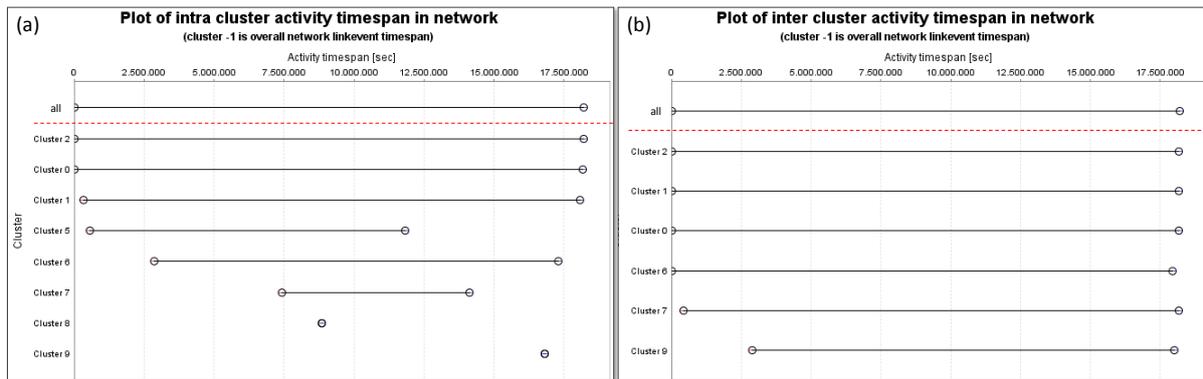


Figure 5-38: Case study. Content-based clustering on nodes. Network activity view: (a) intra-cluster activity; (b) inter-cluster activity

In Table C-11 (appendix C, section C.4.2.1) an overview of the clustering solution of the main component is given. It includes the node memberships in each cluster with cluster representatives and structural key players being highlighted. Furthermore, some metrics are provided for each cluster. The three largest 0, 1 and 2 have about equal numbers of intra-cluster and inter-cluster links. These three clusters contain all but one structural key player. The average local clustering coefficients within each cluster range between 48.81% and 75.21%. The locally centralized clusters 0 and 1 have comparably high clustering coefficients whereas the more distributed cluster 2 has a somewhat smaller coefficient. The density values are smaller than those obtained by structural clustering giving account for the structural distribution and scatter of similar knowledge in the network.

Based on the algorithms and thresholds provided in section 5.1.5.2.1, each content cluster can be categorized as knowledge domain (see Table C-14 in the appendix C, section C.4.2.3). Additionally, each node can be assigned to one of the node role categories provided in Table 5-5, section 5.1.5.1. The node role categorization is calculated per cluster. The data is given in Table C-12 and Table C-13 in the appendix C, section C.4.2.2. A summary of the node role and knowledge domain categorization is given in Table 5-15.

Clusters 3 and 4 are singletons and therefore can be neglected in the further analysis. Clusters 8 and 9 are dyads. In these trivial cases the knowledge domain has been categorized as homogeneous, infrequent with low activity. All other clusters are categorized as heterogeneous knowledge domains, i.e. there are a few knowledge leaders within these domains and several nodes that contribute less. Naturally, the three largest clusters have been categorized as frequent with many participants and medium or high activity, the other three clusters as infrequent with comparably few participants and low activity. In summary, there are three major knowledge domains mainly depending on a few knowledge workers.

The majority of nodes are middlemen (35), team workers (16) and (peripheral) specialists (26). There are six hot spots and six coordinators, four integrators and two information spreaders. Seven nodes are isolated in their knowledge domain. Compared to the structural clustering solution the comparably high number of middlemen and low number of integrators and coordinators indicates that the content clusters are less centralized. However, within this

more distributed structure there are several centers (hot spots) that are supported and connected by team workers and middlemen.

Table 5-15: Case study. Content-based clustering on nodes. Overview of knowledge domain & node role categorization

C.id	#N	#LE	Knowledge Domain Categorization			Node Role Categorization										
			Homogeneity	Frequency	Activity	(D)	(S)	C	HS	IS	I	ISO	MM	PS	S	TW
0	20	1472	het.	freq.	high			4	1		1	1	2	3	5	3
1	16	358	het.	freq.	medium			2	1	1		1		1	6	4
2	47	6959	het.	freq.	high						3	1	33	4		6
3	1	0	-	-	-							1				
4	1	0	-	-	-							1				
5	5	36	het.	infreq.	low				2						3	
6	5	35	het.	infreq.	low		2		1			1			1	
7	9	16	het.	infreq.	low				1	1		1		3		3
8	2	4	hom.	infreq.	low		2									
9	2	2	hom.	infreq.	low		2									

C.id:= cluster id
 #LE:= number of linkevents
 #N:= number of nodes

freq.:= frequent
 infreq.:= infrequent
 het.:= heterogeneous
 hom.:= homogeneous

(D):= (Dyad)
 (S):= (Specialist)
 C:= Coordinator
 HS:= Hot Spot
 IS:= Information Spreader
 I:= Integrator

ISO:= Isolated
 MM:= Middleman
 PS:= Peripheral Specialist
 S:= Specialist
 TW:= Team Worker

In Figure 5-39 the graphviews of the larger content clusters of the main component are given with important nodes highlighted. In cluster 0 in Figure 5-39 a) all but one node belong to a large component consisting of 19 nodes. Five nodes are structural key players highlighted with a red ring. There is a second smaller subgroup which is less connected. This smaller part is connected with the main part by integrator node 1200 who connects with the hot spot node 1261 and the supporting coordinator node 1174. Node 1157 acts as a further coordinator in the main part. In summary, this knowledge domain is characterized by its slightly distributed but still centralized structure.

In cluster 1 in Figure 5-39 b) all but one node belong to a large, densely connected component consisting of 15 nodes. Two nodes are structural key players highlighted with a red ring. These key players can be located in the center of the cluster which is marked with a red dashed ellipse. As hot spot (node 1184), information spreader (node 1186) and coordinators (nodes 1126 and 1158) they obtain the role of centralizing the knowledge in the cluster. They are supported by several team workers that connect and integrate the (peripheral) specialists.

Cluster 2 in Figure 5-39 c) provides some interesting structural properties. First, it consists of three unconnected components: a singleton, a triad and a large component. The large component consists of 43 nodes. 13 nodes are structural key players highlighted with a red ring. These key players can be located in two densely connected subgroups in the main component of cluster 2. Additionally, there is a third, smaller center in the main component. All three centers are marked with a red dashed ellipse in Figure 5-39 c). Each subgroup consists of a central integrator supported by some team workers and middlemen. Between these groups there are three middlemen (nodes 1127, 115 and 1163) and a team worker (node 1158) who help to connect these subgroups (marked with green arrows in Figure 5-39 c)). These four nodes obtain important roles in their content cluster as they connect the three

centers of knowledge. Only one of these nodes (node 1158) has been detected as structural key player based on metrics calculated on the entire network. However, in the entire network he is less prominent in terms of network influence. Again, the content-based clustering provides additional and sometimes contrasting insights into the network context.

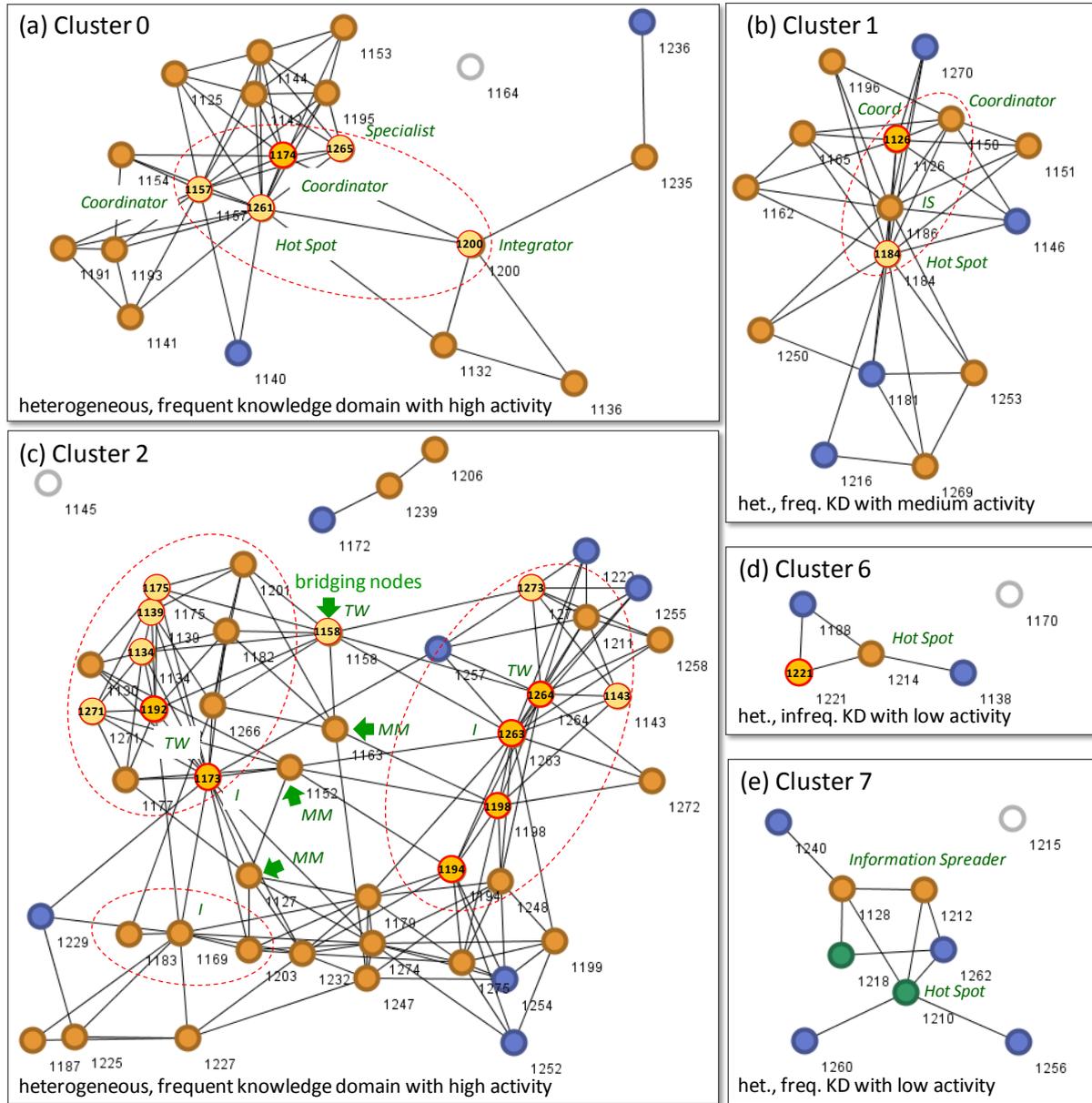


Figure 5-39: Case study. Content-based clustering on nodes. Level overview. Graphviews of selected clusters. Node color indicates type of interaction: green=only linkevents sent; blue=only linkevents received; orange=linkevents sent and received. Structural key players highlighted with thick red ring (top 9) or thin red ring and lighter node color (top 21)

Cluster 6 and cluster 7 both consist of a singleton and a main component of comparably small size. Each cluster has a central hot spot. They are located in the periphery of the main component of the entire network. Here, the structure of the network accounts for separated groups of knowledge.

All five node-based content clusters illustrate that in contrast to a structural clustering solution content clusters can consist of more than one component. Nodes that are well connected in the entire network and in their structural clusters can belong to small components or can even be

isolated in their cluster of knowledge. Comparing the results from structural clustering and content-based clustering on nodes (see appendix C, sections C.4.1.2 and C.4.2.2) (almost) all nodes obtain different roles in both clustering solutions. In some cases, the preliminary network analysis has identified key players that obtain also an important role in their cluster of knowledge but there is a significant shift in their ranking and their possibilities to realize their potentials. That means they cannot rely exclusively on their integration into the network structure when the focus is on generation, exchange and maintenance of knowledge.

In summary, content-based clustering on nodes helps to identify clusters of similar knowledge which can account for structural characteristics in the entire network. However, network structures do not always explain social interaction and fail to identify groups of similar knowledge which are distributed through the entire network. Here, content clustering on nodes allows examining these relations. It therefore complements conventional clustering approaches based on structural properties. Nevertheless, as people acquire multi-faceted knowledge due to different work and social contexts (see e.g. Watts et al. 2002) non-overlapping content clusters will sometimes fail to sufficiently explain this varied experience and diversified social identity. Here, overlapping clusters obtained by content-based clustering on linkevents can provide further insights.

5.3.5.3 Content-based Clustering on Linkevents

A first analysis revealed that the data set yields a large number of small or singleton content clusters when content-based clustering on linkevents is applied. Concentrating only one or two nodes these clusters do not provide meaningful insights into the complex nature of knowledge acquisition and distribution. In order to improve the clustering result all linkevents have been excluded from the analysis as outliers which have content dissimilarities to all other nodes that do not exceed a threshold value $\theta = 0.275$. The dissimilarity is calculated using the cosine similarity measure. As a result, 696 linkevents are regarded as outliers and therefore removed from the active network. In consequence 49 links and one node (node 1238) have no active linkevent and are therefore set inactive as well. This network consists of 2,920 linkevents on 364 links between 107 nodes with average link strength of 8.02 e-mails per link. The density is 6.42%, the global clustering coefficient amounts to 46.72%. These values are slightly smaller than those of the former network indicating a more disrupted structure. The diameter of the main components is still five paths.

To obtain linkevent-based content clusters an agglomerative hierarchical clustering algorithm using the weighted average linkage rule and the cosine similarity is applied to the data (see section 5.1.4.3 and section 5.2.6). The stopping rule by Calinski and Harabasz (1974) is used as validation index for detecting an optimal clustering solution. An appropriate partition can be chosen by examining the plot of the validation index versus level (see Figure 5-40). To gain a comparably small number of clusters the left side of the plot referring to many clusters with few linkevents is not taken into account. Instead, a significant knee in the curve is detected on the right side of the plot by calculating the inflection point (see section 4.6.3). This clustering solution consists of 87 clusters. It is used as initial partition for a partitioning clustering procedure employing the *K*-means pass to improve the cluster memberships. Only two passes are necessary to gain an optimal solution indicating a good initial partition. The

overall content dissimilarity is reduced from 385.26 to 320.35. Most clusters cover long periods of within and between cluster communication in the sample period.

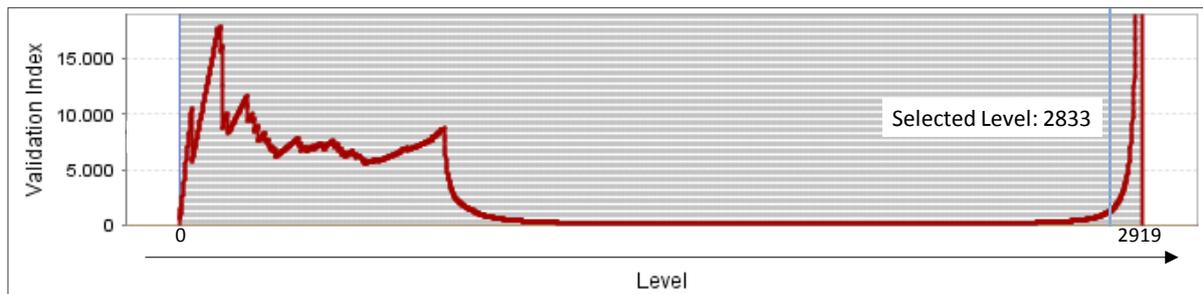


Figure 5-40: Case study. Content-based clustering on linkevents. Hierarchical clustering. Plot of validation index (excerpt); selected level marked with blue vertical line

An overview of the entire clustering solution is given in Table C-15 to Table C-21 in the appendix C, section C.4.3.1. For each node the list of its clusters and some average node metrics calculated from these clusters is given in Table C-22 to Table C-26 in the appendix C, section C.4.3.2. Table C-27 and Table C-28 provide an overview of cluster memberships of the 21 structural key players (see appendix C, section C.4.3.3).

Inspecting the linkevent-based content clusters there are some general findings that can be derived. The graphviews in this section include only those nodes from the main component. In contrast to the structural and node-based content clustering solutions the nodes from the two minor components and the main component are grouped together in three clusters: Cluster 17 contains both nodes from the smallest component and 20 other nodes distributed all over the main component. Cluster 44 contains node 1160 as recipient and node 1249 as sender from the second largest component and 23 other nodes from the main component as illustrated in Figure 5-43 b). Cluster 49 contains all nodes from the second largest component and 35 other nodes from the main component. This cluster is similar in shape and node memberships to cluster 73 (see Figure 5-44 c)). As the three components are unconnected structural clustering will not assign nodes from different components to the same cluster. Especially with very different numbers of linkevents per nodes content-based clustering on nodes fails to reveal these subtle content similarities as nodes with many linkevents dominate less active nodes. Here, content-based clustering on linkevents provides a suitable method for a fine grain analysis of similar knowledge.

5.3.5.3.1 Visual Inspection of Clustering Solution

The visual inspection of the clustering solution allows relating the results of the clustering procedure to the network structure. As overlapping cluster memberships are established the nodes of most clusters have contacts to (almost) all other clusters. As a result, the representative graphview does not provide any meaningful insights. Overlapping clusters cannot be represented in a single graphview. Therefore, there is a network graphview for each cluster which contains the whole network. Nodes that belong to the selected cluster are represented by their type of interaction, e.g. sending linkevents (green), receiving linkevents (blue) or both (orange). Nodes that do not belong to the cluster are depicted as light gray nodes. Intra-cluster links are painted black, inter-cluster links are painted gray. In order to provide some insights about the whole clustering solution and to encode overlapping cluster

memberships the node size relates to the number of clusters the node is assigned to. The top 9 structural key players identified in section 5.3.4.1.2 are highlighted.

Comparing the number of nodes, linkevents and structural key players per cluster three clusters with maximum values are detected: cluster 19 has maximum number of linkevents (136 linkevents), cluster 41 has maximum number of structural key players (16 key players) and cluster 63 has maximum number of nodes (42 nodes). The graphviews of cluster 19 and cluster 63 are provided in Figure 5-41 a) and b). Interestingly, the clusters with highest number of linkevents (clusters 19, 32, 66 and 85) have rather few node members. This mirrors the finding from the initial structural analysis that the most connected node is not the most active node (see section 5.3.4.1.2). Thus, there are certain fields of knowledge which are intensively discussed among few nodes. This relates to the concept of strong ties providing reliable contacts in daily work (see section 2.1.3.3). In contrast, large clusters in terms of node size often develop with only few linkevents involved (cluster 49, 63, 69 and 73). For example, the exchange and access of information necessary to solve unexpected problems or to include specialists in a project involves many nodes with only little interaction. This relates to the concept of Granovetter's strength of weak ties providing new information (see section 2.1.3.3). Furthermore, the set of nodes in the two largest clusters in terms of number of linkevents (see Figure 5-41 a)) and nodes (see Figure 5-41 b)) are rather disjoint covering separate parts of the network. Cluster 19 is locally centralized on ten nodes with a small diameter whereas cluster 63 is rather distributed consisting of several small components established by the comparably small number of 38 linkevents. This again relates to the concept of the different potentials and benefits of weak as well as strong ties.

Clusters 26, 41 and 58 have the highest number of structural key players, i.e. 15 key players in cluster 26 and cluster 58 and 16 key players in cluster 41. With 53 clusters more than half of all clusters have a structural key player as cluster representative. Nodes 1192 and 1173 have the highest number of cluster memberships of all structural key players, i.e. 50 and 52 clusters respectively. This gives account for the prominent position of the structural key players in the entire network and the two other clustering procedures discussed in the previous sections.

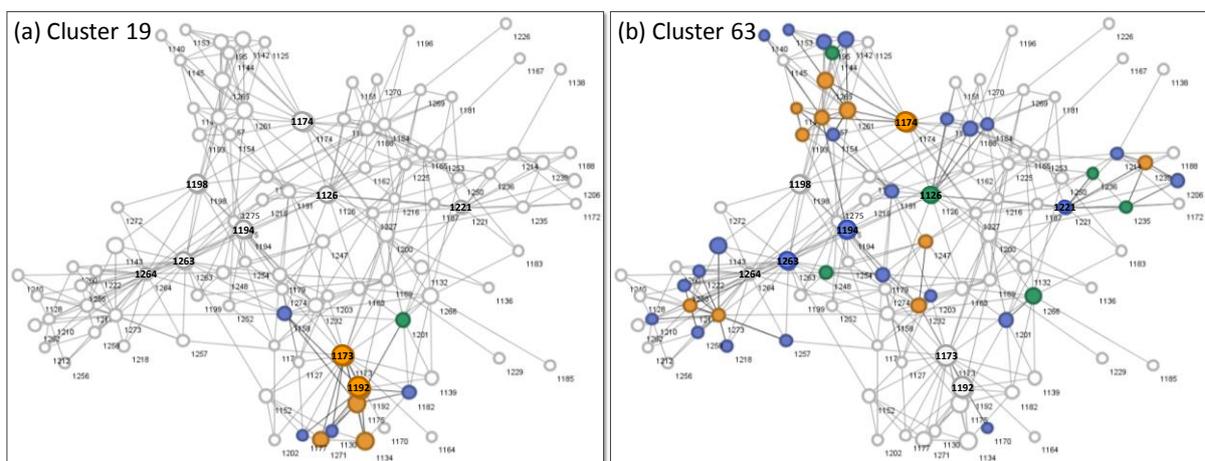


Figure 5-41: Case study. Content-based clustering on linkevents. Example of cluster size. Largest clusters by number of (a) linkevents (cluster 19) and (b) nodes (cluster 63)

The number of unconnected components within a cluster ranges from one to nine components. Although most of the clusters consist of more than one component there is only

one cluster with an isolated node (cluster 18, node 1232). Clusters with only one component (e.g. clusters 1, 19 and 67, see e.g. Figure 5-41 a)) relate to specialized knowledge which is acquired and exchanged only between a few well-connected nodes. Clusters with many components (e.g. clusters 17, 18, 41 and 63, see e.g. Figure 5-41 b)) relate to fields of knowledge which might be necessary in daily work but which might not be very specialized. Thus different unconnected subgroups emerge. These clusters consist of a large number of nodes.

Several groups of clusters can be identified which differ in shape and extension. There are some clusters which concentrate on some local parts (similar to the structural clusters and some of the node-based content clusters). These clusters have rather small diameters. Some of them consist of only one component giving account for the local concentration of a certain field of knowledge indicated by the linkevent of the clusters (see cluster 19 in Figure 5-41 a)). Other clusters consist of one or more main components and several smaller unconnected components distributed through the entire network (see cluster 46 and cluster 84 in Figure 5-42 a) and b)). This shows that there are certain fields of knowledge which integrate different parts of the network but have a strong concentration on a local area. A similar observation has been made examining the node-based content clusters identified in section 5.3.5.2.

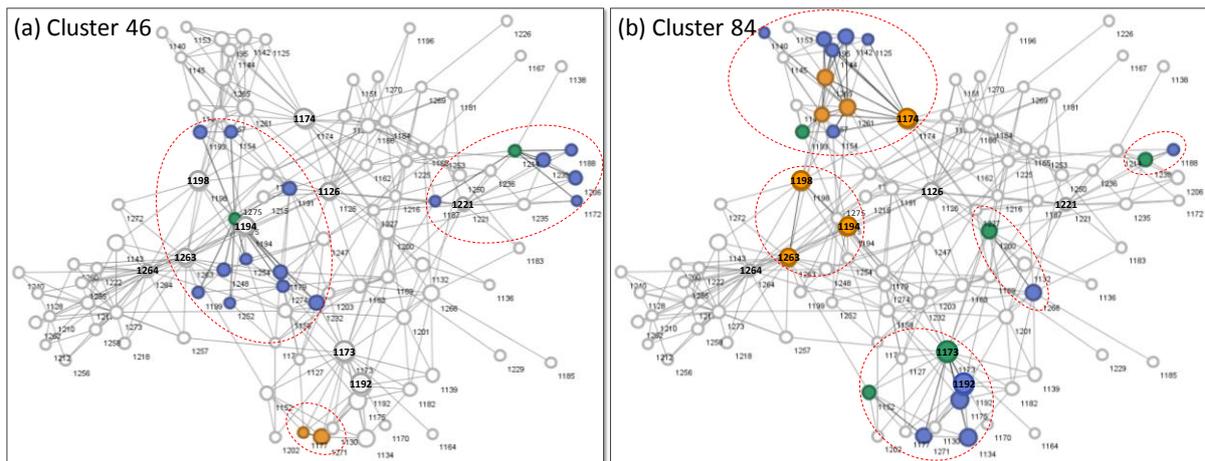


Figure 5-42: Case study. Content-based clustering on linkevents. Examples of local clusters: (a) cluster 46; (b) cluster 84

However, there are also a number of clusters which relate to the observation of central boundary spanners by involving different parts of the network without a clear concentration on a certain area. In Figure 5-43 a comparably small cluster (cluster 3, see Figure 5-43 a)) and a comparably large cluster (cluster 44, see Figure 5-43 b)) are illustrated which integrate different parts of the network. Obviously, there are some nodes that act as integrators or boundary spanners bridging different parts of the network which are assigned to separate clusters by structural as well as node-based content clustering. In cluster 3 node 1174 and in cluster 44 node 1173 obtain these roles. Both nodes are structural key players. Due to their node size they are also involved in many other clusters. With regard to Watts et al. (2002) one can assume that these nodes are highly valuable for their contacts (see section 2.1.3.5): Even if one of their contacts belongs only to a small, locally separated or disconnected cluster he gains indirect access to many other nodes and various fields of knowledge in different parts of the network within.

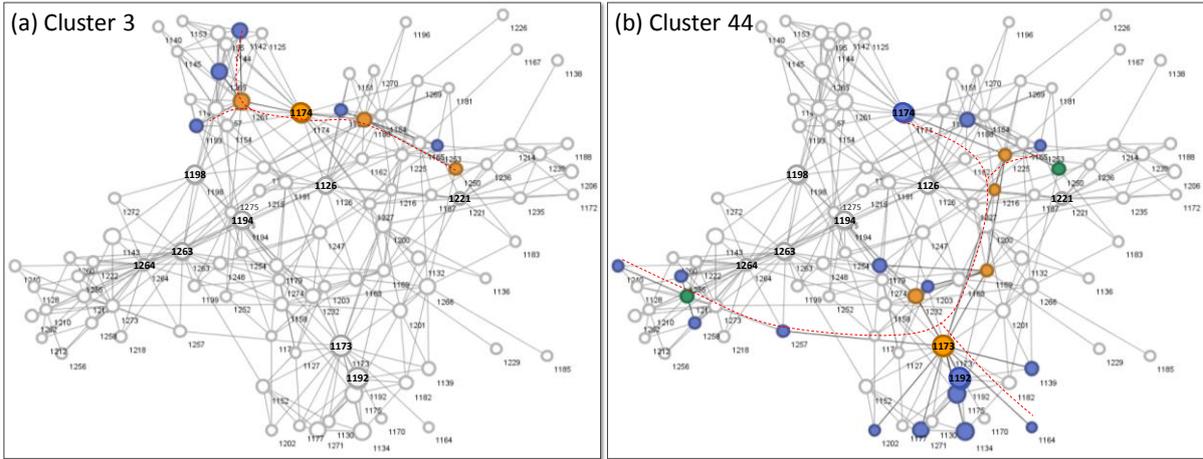


Figure 5-43: Case study. Content-based clustering on linkevents. Examples of distributed clusters: (a) cluster 3; (b) cluster 44

Figure 5-44 provides four different clusters which involve nodes from the peripheral subgroup identified as part 5 by the temperature view in section 5.3.4.2 (see Figure 5-33). This example shows how content-based clustering on linkevents discovers clusters related to structural and node-based content clusters as well as more elaborate clustering structures.

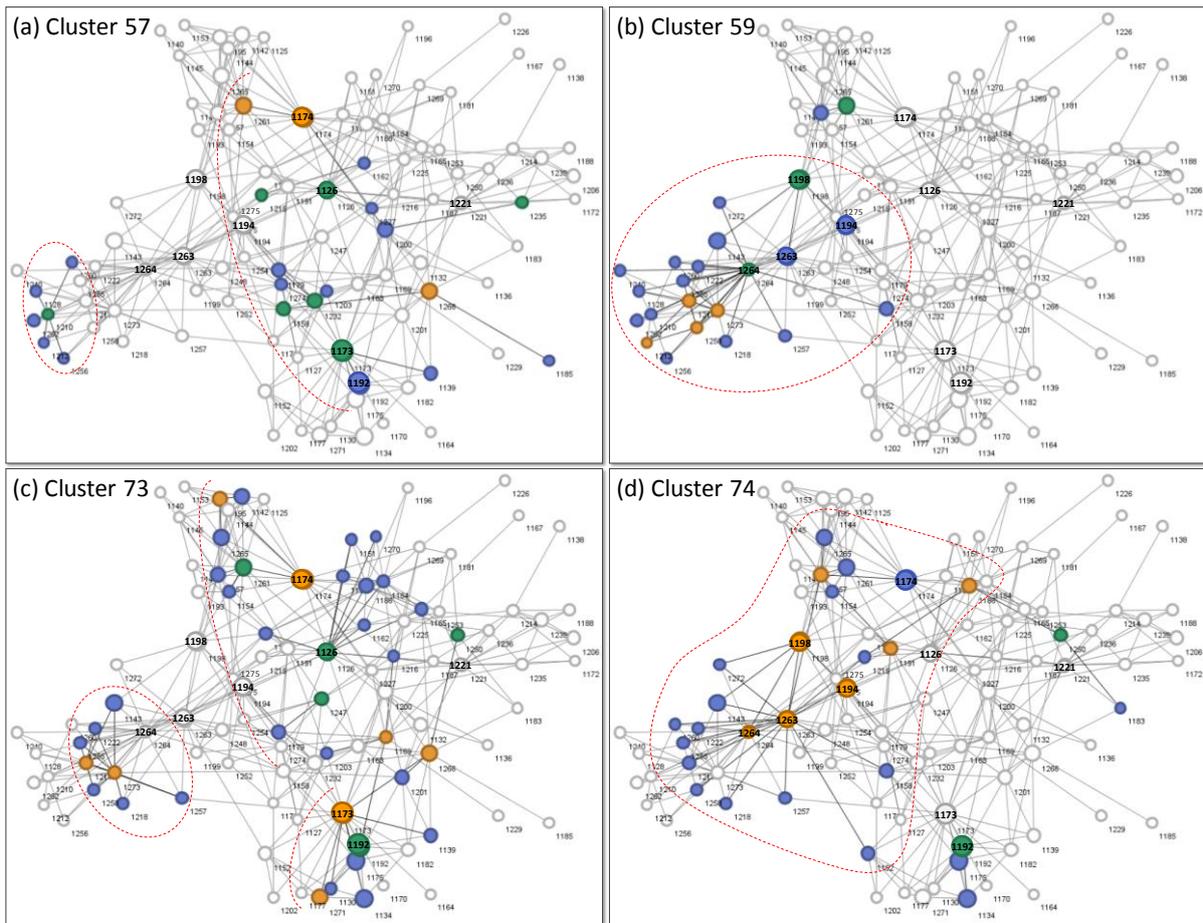


Figure 5-44: Case study. Content-based clustering on linkevents. Examples of different types of knowledge: (a) cluster 57; (b) cluster 59; (c) cluster 73; (d) cluster 74

In cluster 57 in Figure 5-44 a) only the peripheral nodes of this network part belong to the cluster. It is similar to the node-based content cluster 7 (see section 5.3.5.2). As shown in Figure 5-33 the nodes that connect the peripheral nodes to the center of the network have a

strong content similarity among each other but not with the peripheral nodes. Using content-based clustering on nodes, the connecting nodes are assigned to node-based content cluster 2. In the linkevent-based cluster only nodes from the opposite side of the network are similar to the peripheral nodes and assigned to cluster 57 as well. The connecting nodes belong to other clusters of knowledge.

Cluster 59 in Figure 5-44 b) is similar to the structural cluster 4 (see section 5.3.5.1). It covers the entire subgroup which is identified as part 5 by the temperature view in section 5.3.4.2. It is connected to the center of the main component by four structural key players, nodes 1194, 1198, 1263 and 1264. However, apart from an unconnected subgroup of two nodes, no node from the center of the network belongs to this cluster. Thus, the key players act as gatekeepers rather than as bridges to the center. Cluster 73 in Figure 5-44 c) is similar to cluster 57. It contains several separate clusters, one in the outer part and the others on the opposite side of the network. However, it does contain neither the peripheral nodes nor the connecting nodes from structural cluster 4. Similar to cluster 59, cluster 74 in Figure 5-44 d) covers the four structural key players 1194, 1198, 1263 and 1264 together with several surrounding nodes. However, it does not contain the peripheral nodes. In this cluster, the structural key players act as bridges to the center as well as to other outer parts of the network.

Although involving different parts of the network there is also some overlap between the cluster memberships. Several nodes, e.g. nodes 1126, 1173, 1174, 1192 and 1198, appear in more than one cluster. Examining the interconnection of these nodes with the other cluster members reveals that their role in a cluster often differs from their role in the entire network. Therefore, the structural position of a node in the entire network is not reliable for its access to network resources like information or knowledge. However, relating to the study of Watts et al. (2002) each node is at least indirectly connected with these resources through different fields of knowledge expressed by their cluster memberships.

There are several nodes in the clustering solution obtained by content-based clustering on nodes that have comparably high intra-cluster content dissimilarities. These nodes can be regarded as wrongly clustered (see section 5.3.5.2). Node 1170 is one of them. He is assigned to node-based content cluster 6. He is isolated in this cluster. The other four nodes belong to a separated part of the network. In the clustering solution obtained by content-based clustering on linkevents this node is assigned to two linkevent-based content clusters, i.e. cluster 62 and cluster 63. Only two of his node content cluster members are members in one of these linkevent content clusters as well. Although he is still peripheral in his content clusters he is better connected and his content dissimilarity to his cluster members is reduced by 16%.

Table C-22 to Table C-24 in the appendix C, section C.4.3.2 provide an overview of the number of clusters per node. About half of the nodes are assigned to more than 10 clusters. The average number of cluster is 14.10. Interestingly, all but one node among the 20% nodes with highest number of clusters are structural key players. This indicates that they might obtain their prominent structural position by communicational activity as well as multifaceted experience and activity in various fields of knowledge.

5.3.5.3.2 Node Role Categorization

Based on the algorithms and thresholds provided in section 5.1.5.2.1, the node role categorization is calculated per cluster using the node role categories provided in Table 5-5, section 5.1.5.1. As overlapping cluster memberships are obtained by content-based clustering on linkevents each node can be characterized by several node roles depending on his cluster memberships. An overview of the different node roles per node is given in Table C-29 and Table C-30 in the appendix C, section C.4.3.4.1. Additionally, Table C-31 and Table C-32, in the appendix C, section C.4.3.4.2 contains an overview of the different node roles per cluster.

Most nodes are (peripheral) specialists. There are a large number of middlemen and team workers integrating peripheral nodes and distributed parts of the clusters. Not all clusters have a hot spot not. This accounts for knowledge domains with decentralized, distributed shapes. There are only a few coordinators which may help to establish centralized structures. There are a number of mediators, integrators and information spreader with medium betweenness centrality and increasing degree centrality and even some brokers with high betweenness centrality and medium degree centrality. The largest cluster 63 – like several of the other larger clusters – consists only of two different node roles. The cluster members are either peripheral specialists or middlemen. This again indicates that the majority of clusters are distributed over the entire network with different subgroups that have to be integrated and supported.

Examining the distribution of node roles per cluster, the maximum number of different node roles per cluster is 6 node roles and the average number is 3 node roles with a standard deviation of 1 node role. In contrast, the maximum number of nodes per cluster is 42 and the average number 17 with a standard deviation of 9. As indicated by the total number of (peripheral) specialists the structure of each knowledge domain is established, supported and maintained only by few nodes and most cluster members obtain no central role. Although the maximum number of clusters a node belongs to is 52 and the average number is 24.79 with a standard deviation of 12.26 the average number of node roles per node is only 2.70 with a standard deviation of 2.02 and a maximum number of 10. Although the maximum number of different roles per node is 10 the average is only 2.79 a standard deviation of 2.02. Thus, most nodes obtain the same role in all clusters they belong to. Most often they are (peripheral) specialists and do not gain significant positions in their content clusters. Nevertheless there are also some nodes that obtain different roles in different clusters. Often, these nodes have been identified as structural key players in the preliminary network analysis. Here, fine-grain analysis of cluster affiliation using content-based clustering on linkevents helps to identify and explain their multi-faceted roles and their prominent influence on both their local clustering context and the global network structure. The different positions of a node in his different content clusters can be visually inspected in the graphviews depicted in Figure 5-41 to Figure 5-44 in section 5.3.5.3.1.

In Figure 5-45 an example of the node role categorization within a certain knowledge domain is given. Cluster 6 can be categorized as a heterogeneous, frequent knowledge domain with medium activity (see Figure 5-45 b)). In Figure 5-45 c) an overview of the node role categorization is provided. The prominent node roles are highlighted in the graphview

depicted in Figure 5-45 a) which shows a distributed content cluster integrating peripheral subgroup a with the center of the network

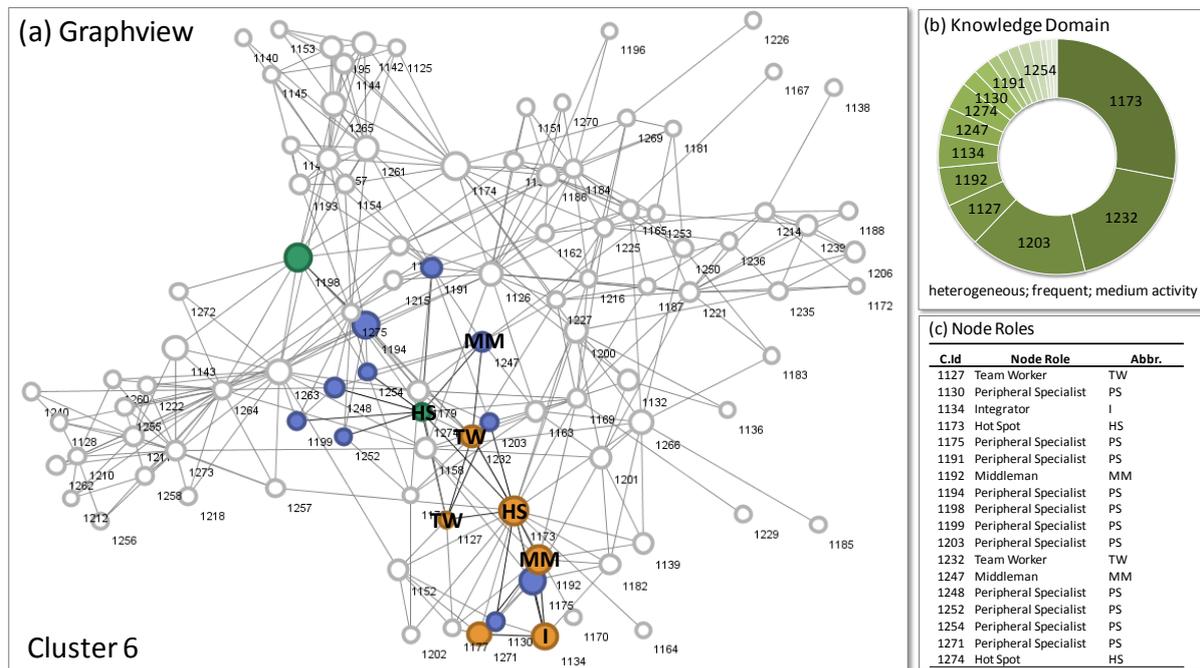


Figure 5-45: Case study. Content-based clustering on linkevents. Node role categorization. Cluster 6: (a) graphview; (b) knowledge domain categorization; (c) node roles (slice size indicates number of linkevents)

As indicated by the knowledge domain overview in Figure 5-45 b) high activity not always results into a prominent role in the cluster. For example, node 1203 has third best number of linkevents but is only a peripheral specialist.

There are two hot spots: Node 1173 who already gains a prominent position in the entire network as one of the three integrators and node 1274 who is only a middleman connecting (peripheral) specialist with the center of the network. Node 1173 has the maximum cluster membership of 52 clusters and obtains 8 different node roles. Node 1274 is only assigned to 9 clusters and obtains 3 different node roles. However, two times she can establish the position of a central hot spot. These two example illustrate how content-based clustering on linkevents not only helps to explain prominent nodes in the entire network but although allows to identify the hidden potentials of less prominent nodes.

5.3.5.3.3 Knowledge Domain Categorization

Based on the algorithms and thresholds provided in section 5.1.5.2.1, each content cluster can be categorized as knowledge domain. The data is given in Table C-33 and Table C-34 in the appendix C, section C.4.3.5.

In Figure 5-46 the portfolio of knowledge domains is provided by a bubble chart. On the x-axis the homogeneity values are plotted and categorized as either “homogeneous” or “heterogeneous”. On the y-axis the frequency values are plotted and categorized as either “infrequent” or “frequent”. Each bubble node represents a cluster. Its size relates to the exact activity value (number of intra-cluster linkevents) whereas its color relates to the three activity types (low, medium, high). Thus, by its size and position each bubble node characterizes a knowledge domain by the triplet of homogeneity, frequency and activity type. Additionally, a

summary about the clusters per knowledge domain category is given in Table C-35 in the appendix C, section C.4.3.5.

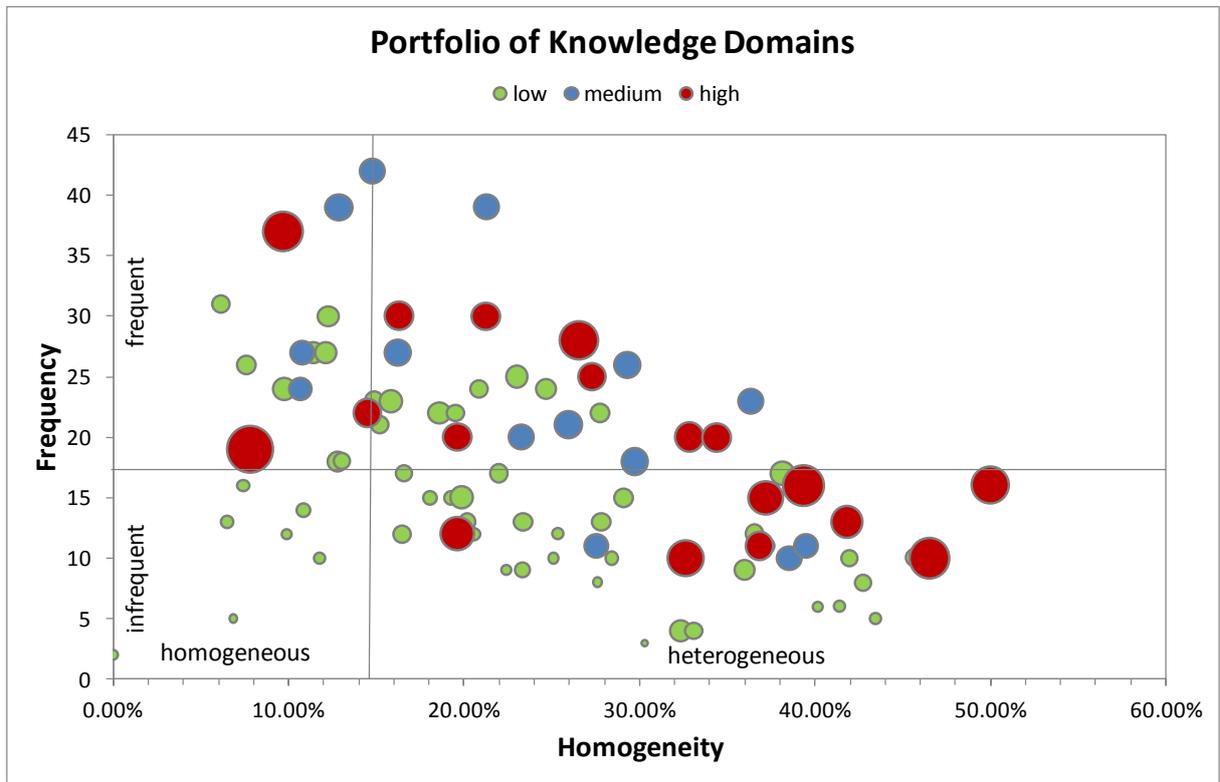


Figure 5-46: Case study. Content-based clustering on linkevents. Portfolio of knowledge domains. Node color indicates activity type (low, medium, high); node size indicates activity value (number of linkevents)

In Figure 5-46 there are no clusters that can be characterized as homogeneous, infrequent knowledge domains with either medium or high activity. All content clusters whose few participants have an (almost) equal share exchange only few linkevents (7). However, the majority of clusters are categorized as heterogeneous, infrequent knowledge domains (41). Most of them with low activity (30), but there are some clusters with medium activity (3) or high activity (8). This indicates that even among only few participants with low activity there will evolve prominent key players as main contributors and opinion leaders. This effect increases with increasing activity among the cluster members.

About half of the cluster can be characterized as infrequent with only few cluster members (48). The other clusters can be characterized as frequent knowledge domains where knowledge is acquired, generated, distributed and maintained among many cluster members (39). Here again, there may evolve prominent key players (heterogeneous knowledge domain) with low (8), medium (7) and high (7) activity. However, there is almost the same number of homogeneous, frequent knowledge domains (17). Although they have also medium activity (4) or high activity (3) the majority of them have only low activity (10). Again, with increasing activity community structures evolve.

Knowledge domains that are established by only few linkevents (low activity) can be interpreted as a knowledge resource that is very specialized, of temporary use, or in early or late stages within the knowledge lifecycle (see Wenger et al. 2002). This does not necessarily mean that this type of knowledge and its participants are less important for the enterprise but

rather should help to provide the necessary support for an evolving field of knowledge, prevent a further decay, or better integrate its knowledge workers in the overall context. If community structures and prominent key players exist, identifying them may provide further support. With increasing number of participants and activity these structures can help to stabilize and improve the communication and interaction within a knowledge domain.

5.3.5.3.4 Knowledge Profile Categorization

Knowledge profiles can only be identified when overlapping cluster memberships are available. Based on the algorithms and thresholds provided in section 5.1.5.2.2, the affiliation of each node to several content clusters can be categorized as knowledge profile. The data is given in Table C-36 and Table C-37 in the appendix C, section C.4.3.6. As shown in this table seven nodes have not been categorized as they only belong to one knowledge domain.

In Figure 5-47 the portfolio of knowledge profiles is provided by a bubble chart. On the x-axis the homogeneity values are plotted and categorized as either “homogeneous” or “heterogeneous”. On the y-axis the diversification values are plotted and categorized as either “specialized” or “diversified”. Each bubble represents a certain node. Its size relates to the exact activity value (number of intra-cluster linkevents) whereas its color relates to the three activity types (low, medium, high). Thus, by its size and position each bubble node characterizes a knowledge profile by the triplet of homogeneity, diversification and activity type. Additionally, a summary about the nodes per knowledge profile category is given in Table C-38 in the appendix C, section C.4.3.6.

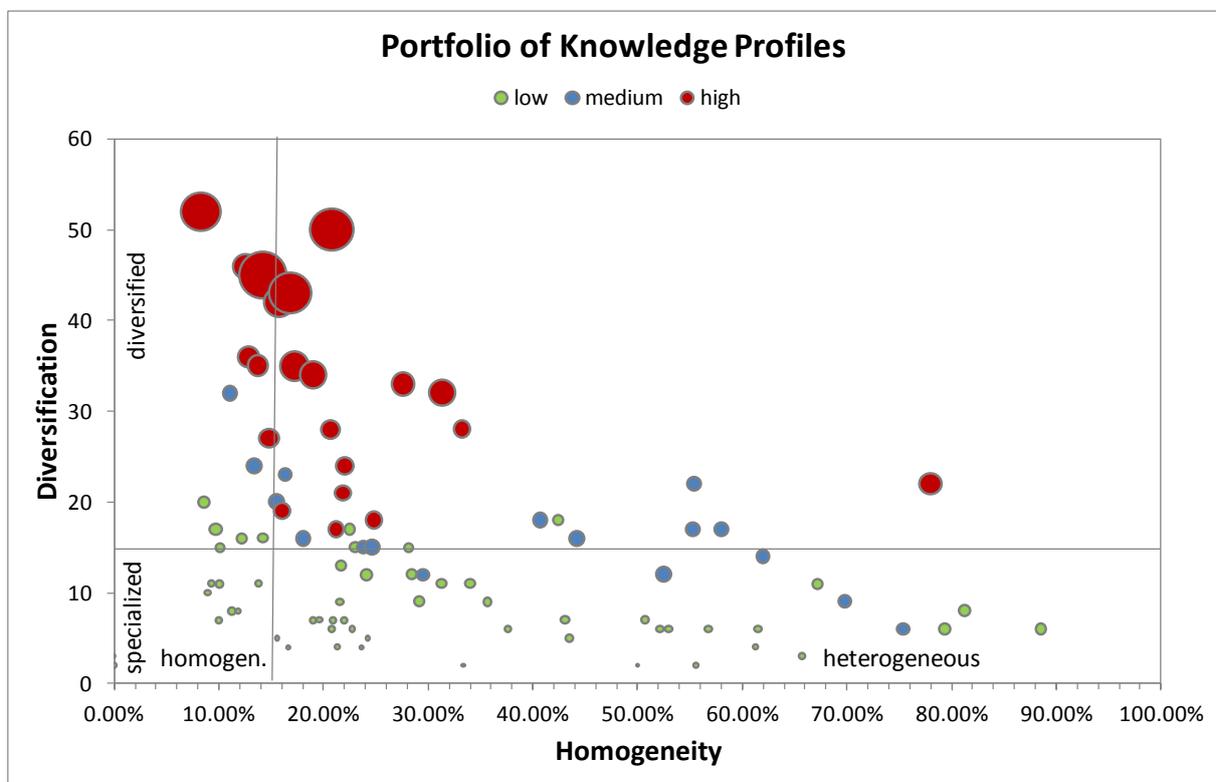


Figure 5-47: Case study. Content-based clustering on linkevents. Portfolio of knowledge profiles. Node color indicates activity type (low, medium, high); node size indicates activity value (number of linkevents) The majority of nodes (42) can be characterized with heterogeneous, specialized knowledge profiles with low activity. That means, they belong only to a few knowledge domains and

mainly contribute in only one or two of them with a small number of linkevents. This can be directly related to the large number of (peripheral) specialists. Some nodes show a heterogeneous, specialized knowledge profile with medium activity (7). There are also specialized nodes with low activity that uniformly contribute in all their knowledge domains (13), but no nodes with medium or high activity that have a homogeneous, specialized knowledge profile. This may indicate that increasing activity leads to favoring one or two knowledge domain due to work specialization.

The other 38 nodes have a diversified knowledge profile as they are involved in a larger number of knowledge domains. Although some of them have an (almost) equal contribution in all knowledge domains with low (4), medium (2) or high (6) activity, the majority of these nodes favor only a few knowledge domains and are less active in the others. Most of these latter nodes have high activity (16). There are also some with medium (8) or low (2) activity. This finding indicates that with increasing work load the number of fields of competence naturally increases but people tend to mainly concentrate on only a few of them. Therefore, within this multi-faceted context prominent roles evolve which have already been identified by the node role categorization in section 5.3.5.3.2.

5.3.5.3.5 Summary

In summary, the analysis of the clustering solution obtained by content-based clustering on linkevents shows how linkevent-based content clustering helps to investigate on multiple fields of knowledge cannot be identified by structural or node-based content clusters. The size, shape and extension of these clusters can be related to important concepts of generation, access and distribution of information and knowledge in social network analysis. Within his various clusters a node can adopt different roles like peripheral specialist, middleman, information broker or central hot spot depending on his position and the shape of the cluster. With regard to their number of clusters most of the structural key players identified in the preliminary network analysis obtain also a prominent role in at least some of their linkevent-based content clusters. This indicates that the structural position in the entire network is also a result of their diversity of experience within different groups of knowledge than only by structural relationships. Inspecting the participation patterns on group level and actor level a unique portfolio of knowledge domains and knowledge profiles can be identified. A thorough analysis of these portfolios will help to understand and support the enterprise's knowledge resources and affiliated knowledge workers.

5.3.5.4 Comparison of Clustering Results

The comparison of the clustering results from the three different clustering procedures includes the calculation of actor-centric and cluster-centric group membership stability values as introduced in section 5.1.5.3 as well as the calculation of centrality metrics to identify the various roles a node can obtain in different clustering solutions.

Structural as well as content-based clustering on nodes establishes non-overlapping cluster memberships. Applied to the Enron e-mail corpus both clustering solutions have almost the same number of clusters. Table C-39 in the appendix C, section C.4.4.1, provides an overview of the cluster-centric group membership stability values for cluster comparison. In both clustering solutions, the two unconnected components are assigned to identical clusters. Thus,

these clusters and their members have highest stability values. Singleton and very small clusters (e.g. node-based content cluster 3, 4 and 9) can be neglected due to their trivial size. The second and third largest content clusters 0 and 1 are very stable with $\emptyset(GMS2) > 55\%$. In contrast, the largest cluster 2 has only $\emptyset(GMS2) = 30\%$. Here it becomes obvious, that although the visual inspection accounts for a strong overlap of both clustering solutions especially the central cluster 2 groups together different nodes in terms of shared knowledge in contrast to strength of interaction.

Comparing content-based clustering on linkevents with the clustering solution obtained by structural clustering there are 276 pairs of clusters from both clustering solutions with at least one shared cluster members. Table C-41 and Table C-42 in the appendix C, section C.4.4.2 provide an overview of the two cluster centric group membership stability values. Only about 2.5% of all pairs of clusters provide a high overlap between both clustering solutions ($\emptyset GMS1 \geq 90\%$). With $\emptyset GMS1 = 100\%$ and $\emptyset GMS2 \geq 75.86\%$ structural cluster 1 and linkevent-based content cluster 45 have the highest value in terms of member overlap and a high stability value in overlap and size. However, inspecting all other pairs shows that the average group membership stability values $\emptyset GMS1$ and $\emptyset GMS2$ values are not correlated. Pairs of clusters with a high node member overlap can extremely differ in size. This indicates that the faceted fields of knowledge developed within the network are mostly not bound to the network structure and densely connected subgroups can only account for a small amount of generation and distribution of knowledge.

Comparing content-based clustering on linkevents with the clustering solution obtained by content-based clustering on nodes there are 173 pairs of clusters from both clustering solutions with at least one shared cluster members. Table C-46 and Table C-47 in the appendix C, section C.4.4.3 provide an overview of the two cluster centric group membership stability values. In about 10% of all cluster pairs there is a high overlap between both clustering solutions ($\emptyset GMS1 \geq 90\%$). With $\emptyset GMS1 = 100\%$ and $\emptyset GMS2 \geq 71.43\%$ node-based content cluster 0 and linkevent-based content cluster 1 have the highest values for both cluster-centric group membership stability metrics. However, inspecting all other cluster pairs shows that the $\emptyset GMS1$ and $\emptyset GMS2$ values are not correlated. Pairs of clusters with a high node member overlap can extremely differ in size. This indicates that the clustering solution obtained by content-based clustering on nodes does only account for a part of the multi-faceted fields of knowledge that can be developed among people. Comparing this solution to the structural clustering the clusters tend to be bound to the network structure whereas content-based clustering on linkevents identifies fields of knowledge that are dissociated from direct interaction expressed by network structure.

Comparing the actor-centric stabilities for all nodes some insights can be gained about the stability of their influence and social context in terms of constant group members and group size. Table C-40 in the appendix C, section C.4.4.1, provides an overview of the actor-centric group membership stability values for node comparison. In this section, the exemplary cluster membership comparisons for nodes 1174, 1198 and 1264 are provided.

Node 1174 belongs to the peripheral structural cluster 3 and node-based content cluster 0 which joins nodes from a peripheral subgroup with the center of the network. With $GMS2 = 76.5\%$ there is a large stability in terms of node members and cluster size. This is the highest

value for any pair of clusters from both clustering solutions. Thus, node 1174 obtains similar roles based on network structure and aggregated knowledge. However, comparing $GMS1_1 = 86.7\%$ (structural cluster) with $GMS1_2 = 68.4\%$ the node-based content cluster joins also several other nodes that are not bound to the network structure.

Node 1198 is the most active node in entire network, see Table C-4. He belongs to the peripheral structural cluster 3 and the central and largest node-based content cluster 2. There is only one stable cluster member. Thus, with $GMS2 = 3.3\%$ there is only a small stability in terms of node members and cluster size. This indicates that although the node is assigned to the periphery of the network due to his structural relationships he rather belongs to the central and most prominent field of knowledge in the network. Thus, the node obtained different roles in different contexts.

Node 1264 is the most central and most between node in entire network (see Table C-4). He belongs to the peripheral structural cluster 4 and the central and largest node-based cluster 2. With $GMS2 = 28.6\%$ there is a moderate stability in terms of node members and cluster size. With $GMS1_1 = 52.9\%$ about half of the structural nodes are still grouped together. With $GMS1_2 = 19.6\%$ a large number of new nodes are added to the knowledge cluster. Similar to node 1174, node 1264 connects the peripheral structural cluster to the center of the network. The notable difference between cluster members and cluster size indicates that node 1264 obtains an ambivalent role in the network context.

On node level there are 1,438 pairs of clusters for all nodes when comparing structural and content-based clustering on linkevents and 1,459 pairs of clusters for all nodes when comparing content-based clustering on nodes and linkevents. Therefore, only an exemplary analysis can be provided in this section. An exemplary overview of actor-centric group membership stability values for three nodes, i.e. nodes 1174, 1198 and 1264 is given in Table C-43 to Table C-45 and Table C-48 to Table C-49 in the appendix C, sections C.4.4.2 and C.4.4.3. For each node and each clustering comparison there are some clusters which are very stable expressed by uniformly high stability values. With structural clustering this indicates that the structural embeddedness of the node within the network is related to some special field of knowledge. Here, the position of the node in the network may account for a prominent role in a certain field of knowledge. With content-based clustering on nodes this indicates that there is a dominant field of knowledge within the variety of personal experience expressed by the different linkevent-based content clusters. Additionally, there can be uniformly low stability values indicating that the linkevent-based content clusters express personal experience which is less obvious and therefore dominated by structure and aggregated knowledge. There are some clustering combinations with highly correlated stability values. However, in general the three actor-centric group membership stability values are not correlated. Pairs of clusters often provide a high stability of group members in one clustering solution but only a low stability in the other. Here, one clustering solution is stable but joined with many other nodes.

An exemplary comparison of the different node roles an actor can obtain in the three different clustering solutions has already been given in the previous sections.

In summary, comparing the different clustering solutions on node and cluster level each node can obtain very different roles due to the size and topology of the clusters. The calculation of

actor-centric as well as cluster-centric group membership stability metrics helps to verify the results from the visual inspection and comparison of the three clustering solutions. Compared to the structural clusters the node-based content clusters tend to be bound to the network structure. However, especially the central content cluster groups together different nodes in terms of shared knowledge in contrast to the strength of interaction. Comparing content-based clustering on linkevents with the clustering solution obtained by structural clustering indicates that the faceted fields of knowledge developed within the network are mostly not bound to the network structure and densely connected subgroups can only account for a small amount of generation and distribution of knowledge. Comparing content-based clustering on linkevents with the clustering solution obtained by content-based clustering on nodes indicates that the non-overlapping node-based content clusters do only account for a part of the faceted fields of knowledge that can be developed. Often, these clusters represent the prominent cluster of a heterogeneous knowledge profile or an artificial, average field of knowledge retrieved from a homogeneous knowledge profile. Overlapping linkevent-based content clusters identify fields of knowledge that are dissociated from direct interaction expressed by network structure.

5.3.6 Summary

In section 5.1 of this chapter a new method for knowledge identification in social networks has been proposed which combines social network analysis, text mining and cluster analysis (research question 1). In section 5.2 a prototype is presented to support this method applied to social networking data. Current section 5.3 provides a detailed case study on a real-world corporate e-mail data set using the prototype to evaluate the benefits of this new approach in business context. The results can be related to the five research questions that have been formulated to guide this work (see chapter 1, section 1.2).

After the preparation and initial screening of the data set, two types of clustering approaches have been applied to the data: First, the state-of-the-art edge betweenness clustering algorithms by Girvan and Newman (2002) has been employed to detect densely connected subgroups within the network. The members of these subgroups are joined together due to their strong interaction expressed by the structure of the network. Second, the novel content-based clustering algorithm has been employed. This approach applies conventional clustering algorithms to the content of the network. Thus, groups of similar knowledge can be identified that are not necessarily bound to the network structure but can be spread through the entire network. This clustering approach can be assigned to the nodes generating non-overlapping groups of knowledge (content-based clustering on nodes). Here, a single field of knowledge is retrieved for each actor which expresses his dominant or arbitrary knowledge domain. As people usually are involved in various fields of knowledge due to different social and work contexts, content-based clustering can also be applied directly to the content objects (content-based clustering on linkevents). The clustering results are then mapped to the network structure. Here, overlapping clusters are obtained which relate to different knowledge domains. Furthermore, this approach also allows identifying knowledge profiles which describe an actor's activity within his different fields of knowledge.

These three different clustering procedures provide different perspectives on the data. Sometimes they verify each other, sometimes the results are contrasting (research question 2). Naturally, the structural clustering algorithm identifies clusters which are bound to the

network structure. In this case study, content-based clustering on nodes reveals a clustering solution that is more distributed through the network, especially regarding those nodes in the center of the network. However, most content clusters are similar to the structural clusters in the peripheral parts of the network that are more loosely connected to the center. When inspecting the clustering solution obtained by content-based clustering on linkevents, there are clusters whose structure is related to the network structure as well as clusters that are distributed through the network integrating nodes from different parts of the network. In general, one will obtain very different clustering solutions giving account for the multi-faceted knowledge which is acquired, maintained, exchanged in social networks representing informal work relations around knowledge-intensive business processes. The calculation of actor-centric as well as cluster-centric group membership stability metrics proposed in this work helps to verify the results from the visual inspection and comparison of the three clustering solutions.

When examining the roles and positions of each node within the entire network and the clusters from the different clustering procedures, most nodes obtain very different roles due to the size and topology of the clusters (research question 3). The size, shape and extension of these clusters can be related to important concepts of generation, access and distribution of information and knowledge in social network analysis. Within his clusters a node can adopt different roles like peripheral specialist, middleman, information broker or central hot spot depending on his position and the shape of the cluster. With regard to their number of clusters most of the structural key players identified in the preliminary network analysis obtain a prominent role as well. This indicates that the structural position in the entire network is also a result of their diversity of experience within different groups of knowledge than only by structural relationships.

The analysis of the clustering solution obtained by content-based clustering on linkevents shows how linkevent-based content clustering helps to investigate on multiple fields of knowledge that are missed by structural as well as node-based content clusters. Inspecting the participation patterns on node level and cluster level four categories of knowledge profiles and fields of knowledge can be identified (research question 4): On node level, the participation patterns of a node can be categorized into homogeneous or heterogeneous specialization and homogeneous or heterogeneous diversification. On cluster level, the participation patterns of different nodes within a cluster can be categorized into homogeneous or heterogeneous infrequent fields of knowledge and homogeneous or heterogeneous frequent fields of knowledge. Furthermore, the involvement of each node in a cluster can further classified by his prominence, activity and popularity.

The results from the cluster analysis can also be related to some prominent theories in knowledge management and social network analysis that try to explain the acquisition, maintenance and exchange of knowledge in the network (research question 5): The shape and structure of the identified fields of knowledge can be related to Granovetter's theory of strong and weak ties that both types are beneficial in different contexts of information access and sharing, e.g. unexpected problems versus well-established work relations (Granovetter 1973; Granovetter 1983). According to Belbin (1993) there are different roles an individual can obtain in a team which can be defined as the way an individual behaves in the team, contributes to its purpose and goals and interacts with others. He distinguishes nine

complementary team roles. Identifying and categorizing heterogeneous roles in different fields of knowledge supports the individual capabilities to participate in the knowledge lifecycle. Therefore, based on the prominent degree and betweenness centrality metrics are proposed in this work. As people acquire multi-faceted knowledge due to different work and social contexts (see e.g. Watts et al. 2002) the different knowledge profiles of an actor help explain this varied experience and diversified social identity. The categorization of knowledge domains can help to explain work requirements, e.g. daily business versus special tasks, availability of knowledge workers, as well as the phase in its lifecycle, e.g. beginning, maturing, decaying (see e.g. stages of community development in Wenger et al. (2002: 69).

Thus, this case study illustrates how the new method of knowledge identification can be employed to support and improve the effectiveness and efficiency of the knowledge lifecycle within an organization.

6 Conclusions

Chapter 5 has introduced a novel approach for identifying groups of similar knowledge in social corpora based on the theoretical foundation of chapters 1 to 4. In the first section of this final chapter a summary of the entire work is given. It relates the properties of the proposed method of knowledge identification in social corpora, its theoretical foundation and the results of the case study to the five research questions which have been formulated on the context of this work. In the second section future research to support and improve the effectiveness and efficiency of the knowledge lifecycle within an organization is discussed.

6.1 Summary of the Results

This work is motivated by the question how the analysis of knowledge networks can enhance workflow modeling and business process analysis (see chapter 1). In this context, a special focus is laid on those activities that cannot be described by a (complex) series of functions and events but by the knowledge required to achieve a certain goal. These activities are the basic elements of knowledge-intensive business processes which can be identified by a high degree of complexity, a low degree of structuredness, a high degree of communication, being hard to schedule, a high degree of exceptions from predefined business rules, a high degree of work autonomy and ongoing high degree of information need (see e.g. Davenport et al. 1996; Abecker et al. 2002; Heisig 2002). Knowledge worker transform their knowledge into productivity during their daily work. Therefore, analyzing these processes can help to identify sources and demand of knowledge. In today's large organizations it has become increasingly difficult to find people that have specific skills or knowledge or to explore and understand the overall picture of an organization's portfolio of topic expertise (Brunnert et al. 2007). Bringing together human actors with similar interests, skills or knowledge is a major challenge in community-based knowledge management. It is therefore essential to understand the characteristics and capabilities of each individual that are developed during formal as well as informal collaboration processes within the organization.

Due to a study by Fallows (2002) 98% of all employees with internet access collaborate via e-mail at their workplace to communicate, access information and improve their teamwork. such effects are not limited to the domain of e-mail communication: Today, electronic media complement formal work networks and provide more diverse, participative and less formally aligned relations (Bikson and Eveland 1990). All these interactions of people captured over time form a social network structure via communication (Krackhardt 1992). More recently, in the context of internet 2.0 or web 2.0 they are often called social software applications as they allow socializing via the internet. One of the primary methods for studying the resulting electronic communication, collaboration and interaction and their inherent communities is social network analysis (Garton et al. 1997). Research on these networks is directed at understanding the structures and properties of these complex systems to identify structural patterns (Kossinets and Watts 2006). One important interest is the location of groups in such networks based on their structure. For this objective SNA offers a series of clustering and grouping algorithms that find widespread application in practice. However, conventional SNA

is often limited to the analysis of static network structures. Although often content information can be directly linked with the relational data forming the social network the domain of content analysis with means of text mining and information retrieval has not yet been sufficiently accommodated in the methodological framework of SNA. Therefore, in chapter 1 the Social Network Intelligence framework has been introduced which is designed to enable IT-supported, network-oriented knowledge management on multiple integrated levels of analysis. Based on the theoretical foundation of state-of-the-art social network analysis it provides novel insights into network structure, content, behavior and context. The SNI framework consists of three components: the SNI data model of event-driven dynamic network analysis, the SNI dimensions and the SNI process. The SNI Process is the basic methodology of the SNI Framework. It is especially designed for identifying, analyzing and managing knowledge networks that evolve from knowledge-intensive business processes. The SNI process consists of six steps which are passed through iteratively and require the supervision of business and domain knowledge workers: 1) project initiation, 2) definition of scope, 3) data capturing & refinement, 4) visualization & analysis, 5) deriving concepts & action plan, 6) execution & implementation of actions. These steps correspond to the process model of process-oriented systems analysis. However, they are not especially designed for identifying, analyzing and managing knowledge networks that evolve from knowledge-intensive business processes. The essential part of this process is the visualization and analysis of the data (step 4). It is based on conventional SNA and contains of measures and recommendations for action on the three SNI dimensions: the level of detail dimension, the level of investigation dimension and the temporal dimension. By linking different perspective on the network and its components, these three SNI dimensions are the core of the analytical toolbox within the Social Network Intelligence framework.

The new method of content-based clustering for knowledge identification in social corpora is based on the SNI framework. It is designed as a static content analysis on ego level (i.e. key players), group level and network level whereas the main focus is on group level including also elements of structural analysis and network dynamics. Therefore, chapter 3 and chapter 4 present text mining and cluster analysis as theoretical foundations.

The way people distribute and gain information has drastically changed since the 1990s. Today, the amount of electronically stored documents and, more general, information items, increases dramatically. Besides publicly available information, huge amounts of information are private and are of interest to local communities, such as the records of customers of a business (Weiss et al. 2004). This information is mainly text. There are estimates that about 85% of business-relevant information originates in unstructured form (Grimes 2010). Methods for automatic text processing can be employed to deal with the information overload of modern communication systems and access the hidden patterns, e.g. by information extraction and document summarization (Weiss et al. 2004). Analogous to data mining on structured data text mining also finds patterns and trends in information samples that are far less structured but have greater immediate utility for users. Text data that can be retrieved from social media are often analyzed separately from the structure of the network. Few approaches exist where both types of data are combined (see e.g. Bobrik and Trier 2009; Diesner and Carley 2010). Neglecting the content of text data during network analysis can limit the understanding of the network structure, its evolution over time and the multiple roles

an actor can obtain in this context. Therefore conventional SNA will not be able to detect groups of similar knowledge that evolve within a multi-contextual network and where each node can obtain multiple roles. Such communities are not only structural phenomenon but have been defined by seminal literature as networks about something (Wenger et al. 2002). As part of the theoretical foundation of this work chapter 3 provides an overview of concepts and methods for automated mining of text samples in general and their application on social corpora as part of a SNA study.

One important interest in SNA is the location of groups within the networks. For example, Tyler et al. (2003) to automatically identify communities in a research department at Hewlett-Packard studying e-mail interactions. Examining the network structure actors tend to join in homogenous groups that are only loosely connected to other groups (Ravasz and Barabási 2003). Here, cluster analysis provides methods and techniques of classifying objects into meaningful sets (Aldenderfer and Blashfield 1984: 5). In contrast to discriminant analyses cluster analysis is about discovering groups inherent to the data (Jain and Dubes 1988; Everitt et al. 2001: 6). The basic elements of cluster analysis are the data consisting of objects described by a set of variables, or features. Using a specific clustering method these objects are assigned to groups according to the similarity of their feature values. Using a suitable visual representation clusters can easily be recognized by the human mind and one can give a functional definition of a cluster (Jain and Dubes 1988: 2). However, it is difficult to give an operational definition because data objects can be grouped into different clusters with different purposes in mind. Thus, clusters can vary in shape (clumsy or linear clusters), number and size (many small or few large clusters). Additionally, cluster membership can change over time and the number of clusters often depends of the aggregation level of the data. Chapter 4 is organized into two parts: First, a general overview of cluster analysis is given. Afterwards, this chapter presents applications of cluster analysis in social corpora with special focus on the popular graph-based edge-betweenness clustering algorithm.

Combining social network analysis, text mining and cluster analysis, chapter 5 presents a novel method for analyzing collaborative content and identifying the different groups of interest, knowledge, or skills within an organization is designed and implemented as a prototype. Therefore, a detailed research guideline how to conduct this analysis is provided covering four steps: 1) data preparation, 2) initial screening, 3) preliminary network analysis and 4) cluster analysis on network structure and content. As part of the analytical toolset of the SNI framework this method contributes to the analytical toolset for analyzing social networks that can be characterized by the three SNI dimensions: level of detail, level of investigation and the temporal dimension. It is designed as a static content analysis on ego level (i.e. key players), group level and network level whereas the main focus is on group level including also elements of structural analysis and network dynamics. The different steps involve and combine existing and well-established metrics and algorithms from text mining, cluster analysis and social network analysis. Beyond this several new algorithms and metrics have been developed. To complement the analysis of the network shape and structure with existing structural SNA metrics the temperature view has been developed. This new approach allows a visual inspection of the distribution of content similarities between nodes and can be used to gain a first understanding of the network, to determine the clustering tendency of the content of the network and to selected important key players for further analysis. There are

two new clustering approaches that have been invented in this work. Based on the content objects of the data content-based clustering on nodes generates a single, aggregated cluster of knowledge per node and content-based clustering on linkevents generates multiple clusters of knowledge per node. The network the popular graph-based community detection method by Newman and Girvan (2002) is employed as part of the cluster analysis to compare and evaluate the results from the content-based clustering procedures. A set of actor-centric and cluster-centric group membership stability metrics has been developed and implemented in the prototype to determine and compare the quality of the results from the different clustering solutions.

A case study on a real-world corporate e-mail data set is conducted to evaluate the benefits of this new approach of knowledge identification in business context. The results can be related to the five research questions that have been formulated to guide this work:

Research question 1 motivates the new method of content-based clustering for knowledge identification based on the combination of techniques and methods from different scientific disciplines. The case study illustrates how techniques from these three different fields of research can be combined to invent a new approach to access and support the knowledge lifecycle within an organization.

Research question 2 investigates in how far additional insights can be gained using the proposed method compared to existing approaches. As a result, applied to corporate e-mail data the method allows identifying groups of similar knowledge that are not necessarily bound to the network structure but can be spread through the entire network. The calculation of actor-centric as well as cluster-centric group membership stability metrics proposed in this work helps to verify the results from the visual inspection and comparison of the three clustering solutions.

Research question 3 refers to the identification of context-specific roles of an actor. When examining the roles and positions of each node within the entire network and the clusters from the different clustering procedures, most nodes adopt different roles like peripheral specialist, middleman, information broker or depending on his position and the shape of the component or cluster. If an actor is assigned to several clusters in one clustering solution he may obtain similar roles in each cluster but also varying roles like specialist, broker or even central hot spot.

Research question 4 deals with the categorization of knowledge to identify different fields of knowledge which allow deriving generalized characteristics of knowledge domains (group level) and to establish knowledge profiles (node level). On cluster level, the participation patterns of different nodes within a cluster can be categorized into homogeneous or heterogeneous infrequent knowledge domain and homogeneous or heterogeneous frequent knowledge domain with low, medium or high activity. On node level, the participation patterns of a node can be categorized into homogeneous or heterogeneous specialization and homogeneous or heterogeneous diversification with low, medium or high activity.

Research question 5 focuses on the benefits of the new approach for knowledge identification for social network analysis in general and business applications in particular. The results from the cluster analysis can be related to some prominent theories in knowledge management and

social network analysis that try to explain the acquisition, maintenance and exchange of knowledge in the network, e.g. Granovetter's (1973; 1983) theory of strong and weak ties (1973; 1983), the different team roles by Belbin (1993), searchability of networks due an individual's diverse social context (see Watts et al. 2002), the different stages of community development by Wenger et al. (2002). Thus, this case study illustrates how the new method of knowledge identification can be employed to support and improve the effectiveness and efficiency of the knowledge lifecycle within an organization.

Finally, this last chapter gives a detailed summary of the entire work covering the key aspects of all five chapters. It then concludes with a discussion of future research based on the proposed methods of knowledge identification in social corpora as part of the Social Network Intelligence framework for analyzing networks in business applications.

6.2 Future Research

In October 2010, Gartner (2010) has recommended ten strategic technologies with the potential for significant impact on the enterprise in the next three years: cloud computing, mobile applications and media tablets, social communications and collaboration, video, next generation analytics, social analytics, context-aware computing, storage class memory, ubiquitous computing, and fabric-based infrastructure and computers. IT managers should incorporate these strategies into their strategic planning process for developing and establishing emerging-technology portfolios to be able to meet their technological and organizational requirements.

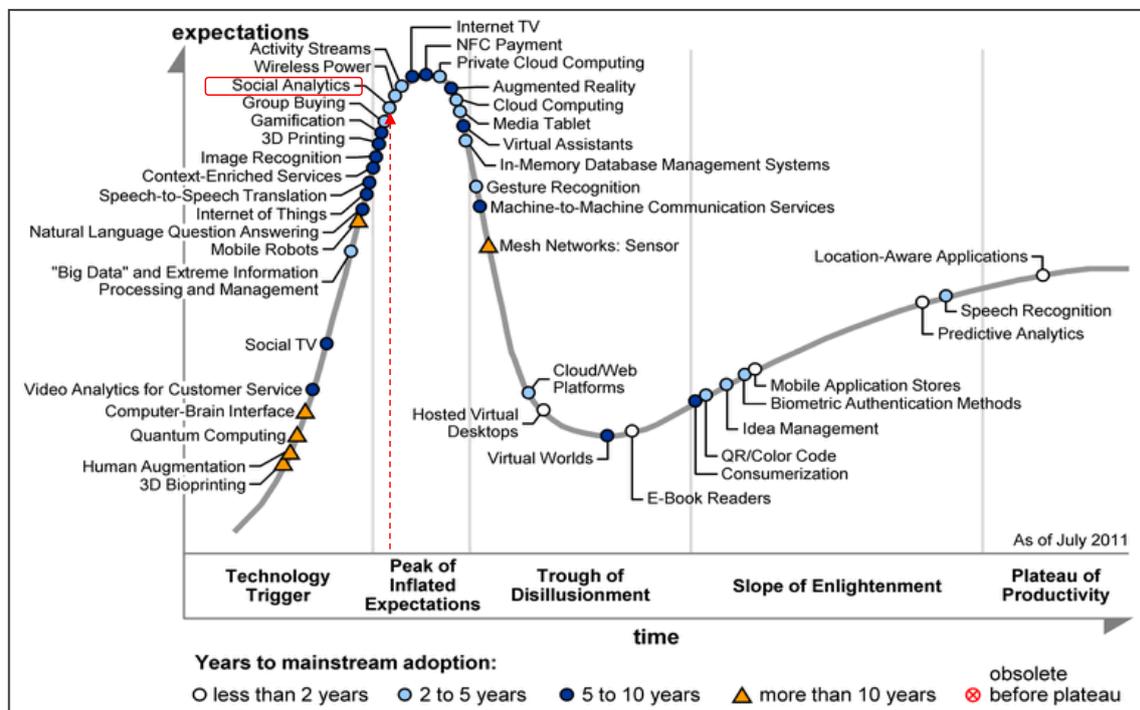


Figure 6-1: Gartner's hype cycle for emerging technologies (2011). Source: Gartner (2011)

In the context of this work the relevance of social communications and collaboration and social analytics is of paramount importance. Due to Gartner (2010) *social communications and collaboration* covers all kind of social media that can be divided into social networking, social collaboration, social publishing, and social feedback. Gartner (2010) predicts that by

2016, social technologies will be integrated with most business applications. Therefore, a coordinated strategy will involve the company's initiatives on social CRM, internal communications and collaboration, and public social. Social networking includes social profile management products, e.g. MySpace, Facebook, or LinkedIn, as well as SNA methods and algorithms to understand, evaluate and utilize human relationships for the discovery of people and knowledge. *Social analytics* then is defined as the "process of measuring, analyzing and interpreting the results of interactions and associations among people, topics and ideas (Gartner 2010)" that can be retrieved from social media applications. Social analytics include several different analysis techniques, e.g. social filtering, social network analysis, sentiment analysis and social media analytics. According to Gartner (2010) SNA tools can be employed to investigate on the structure and interdependencies of a network as well as the work patterns of individuals, groups or organizations. As illustrated in Figure 6-1 social analytics has reached the peak of inflated expectations of the Gartner's hype cycle for emerging technologies for 2011 with about two to five years to mainstream adoption (Gartner 2011). Trial and error are typical characteristics of this phase which will direct future research on knowledge management in general and social network analysis in particular to foster analytical advances. Here the proposed Social Network Intelligence framework as explained in chapter 1 can help to incorporate the analytical measures of social network analysis into a systems analysis to meet future challenges.

Although Gartner (2010) applies the term social analytics not only to the structure of a network but also to topics and ideas with a strong focus on discovering knowledge within an organization the main objective of SNA and social analytics is still on relationships and not on content objects. This work, however, has introduced a new approach for knowledge identification in social corpora that is based on the collaboratively created content and maps the result to the network structure. As illustrated by the case study in chapter 5, well-established SNA metrics and algorithm may be successfully employed in this context but there is also the necessity to develop new analytical measures that capture the properties of content objects associated with social networks. Here, the algorithms for content similarity and content-based clustering and metrics on cluster stability developed as part of the new method for knowledge identification in social corpora will provide a starting point for future research.

In chapter 5, section 5.3 a detailed case study has been conducted. The interpretation of the results serve to illustrate the usage of the method for knowledge identification in social corpora there are some limitations to the empirical insights that can be derived. According to Hevner et al. (2004: 80) the refinement and reassessment of an artifact due to its identified limitations is typically can direct future. Although there are approaches of fully automated cluster analysis, interpreting the clusters is still a non-standardized process that requires advanced knowledge and leaves ample room for reading meaning into the results. Therefore, the application of the method presented in this work to a public, real-world data set is mainly of illustrative nature. A thorough, domain oriented analysis of the results would require the involvement of subject matter experts. Further case studies should be conducted to decide if and how the findings of this work on knowledge profiles in business applications can be extended to different types of content objects. Here, analyzing similar data from social media where actors are interrelated by content objects, such as e-mails, newsgroup postings, or co-

authored documents, will help to verify and generalize the categories and thresholds of node roles, knowledge profiles (actor level) and knowledge domains (group level) that have been identified in the course of this work. Furthermore, analyzing different types of data, such as product recommendations from online retailers or wiki links, will allow identifying and analyzing groups of nodes with shared context that may not relate to individual knowledge but to e.g. customer profiles or knowledge domains. The the method for knowledge identification in social corpora has been originally designed as a static analysis. Similar to the approach by Falkowski et al. (2006; 2008) it can be extended by integrating network dynamics to analyze the evolution and stability of knowledge profiles and certain fields of knowledge over time.

Altogether the proposed method for knowledge identification in social corpora as part of the analytical measures that of the Social Network Intelligence framework and the results from the case study provide promising results for business applications and future research.

Appendix

A Text Mining

Table A-1: Comparison of different tag sets. CLAWS5, Brown and Penn Treebank tag set. Source: Manning and Schütze (1999: 141-142)

Category	Example	CLAWS c5	Brown	Penn
Adjective	happy, bad	AJ0	JJ	JJ
Adjective, ordinal number	sixth, 72nd, last	ORD	OD	JJ
Adjective, comparative	happier, worse	AJC	JJR	JJR
Adjective, superlative	happiest, worst	AJS	JJR	JJS
Adjective, superlative, semantically	chief, top	AJ0	JJS	JJ
Adjective, cardinal number	2, fifteen	CRD	CD	CD
Adjective, cardinal number one	One	PNI	CD	CD
Adverb	often, particularly	AV0	RB	RB
Adverb, negative	Not	XX0	*	RB
Adverb, comparative	Faster	AV0	RBR	RBR
Adverb, superlative	fastest	AV0	RBT	RBS
Adverb, particle	uo, off, out	AVP	RP	RP
Adverb, question	when, how, why	AVQ	WRB	WRB
Adverb, degree & question	how, however	AVQ	WQL	WRB
Adverb, degree	very, so, too	AV0	QL	RB
Adverb, degree postposed	enough, indeed	AV0	QLP	RB
Adverb, nominal	here, there, now	AV0	RN	RB
Conjunction, coordination	and, or	CJC	CC	CC
Conjunction, subordinating	although, when	CJS	CS	IN
Conjunction, completizer <i>that</i>	that	CJT	CS	IN
Determiner	this, each, another	DT0	DT	DT
Determiner, pronoun	any, some	DT0	DTI	DT
Determiner, pronoun, plural	these, those	DT0	DTI	DT
Determiner, prequalifier	quite	DT0	ABL	PDT
Determiner, prequantifier	all, half	DT0	ABN	PDT
Determiner, pronoun or double conj.	both	DT0	ABX	DT (CC)
Determiner, pronoun or double conj.	either, neither	DT0	DTX	DT (CC)
Determiner, article	the, a, an	AT0	AT	DT
Determiner, postdeterminer	many, same	DT0	AP	JJ
Determiner, possessive	their, your	DPS	PP\$	PRP\$
Determiner, possessive, second	mine, yours	DPS	PP\$\$	PRP
Determiner, question	which, whatever	DTQ	WDT	WDT
Determiner, possessive & question	whose	DTQ	WP\$	WP\$
Noun	aircraft, data	NN0	NN	NN
Noun, singular	woman, book	NN1	NN	NN
Noun, plural	women, books	NN2	NNS	NNS
Noun, proper, singular	London, Michael	NP0	NP	NNP
Noun, proper, plural	Australians, Methodists	NP0	NPS	NNPS
Noun, adverbial	tomorrow, home	NN0	NR	NN
Noun, adverbial, plural	Sundays, weekdays	NN2	NRS	NNS
Pronoun, nominal (indefinite)	none, everything, one	PN1	PN	NN
Pronoun, personal, subject	you, we	PNP	PPSS	PRP
Pronoun, personal, subject, 3SG	she, he, it	PNP	PPS	PRP
Pronoun, personal, object	you, them, me	PNP	PPO	PRP

Table A-2: Comparison of different tag sets (Ctd.). Source: Manning and Schütze (1999: 142)

Category	Example	CLAWS c5	Brown	Penn
Pronoun, reflexive	herself, myself	PNX	PPL	PRP
Pronoun, reflexive, plural	themselves, ourselves	PNX	PPLS	PRP
Pronoun, question, subject	who, whoever	PNQ	WPS	WP
Pronoun, question, object	who, whoever	PNQ	WPO	WP
Pronoun, existential <i>there</i>	there	EXO	EX	EX
Verb, base present form (not infinitive)	take, live	VVB	VB	VBP
Verb, infinitive	take, live	VVI	VB	VB
Verb, past tense	took, lived	VVD	VBD	VBD
Verb, present participle	taking, living	VVG	VBG	VBG
Verb, past/passive participle	taken, lived	VVN	VBN	VBN
Verb, present 3SG -s form	takes, lives	VVZ	VBZ	VBZ
Verb, auxiliary <i>do</i> , base	do	VDB	DO	VBP
Verb, auxiliary <i>do</i> , infinitive	do	VDB	DO	VB
Verb, auxiliary <i>do</i> , past	did	VDD	DOD	VBD
Verb, auxiliary <i>do</i> , present part.	doing	VDG	VBG	VBG
Verb, auxiliary <i>do</i> , past part.	done	VDN	VBN	VBN
Verb, auxiliary <i>do</i> , present 3SG	does	VDZ	DOZ	VBZ
Verb, auxiliary <i>have</i> , base	have	VHB	HV	VBP
Verb, auxiliary <i>have</i> , infinitive	have	VHI	HV	VB
Verb, auxiliary <i>have</i> , past	had	VHD	HVD	VBD
Verb, auxiliary <i>have</i> , present part.	having	VHG	HVG	VBG
Verb, auxiliary <i>have</i> , past part.	had	VHN	HVN	VBN
Verb, auxiliary <i>have</i> , present 3SG	has	VHZ	HVZ	VBZ
Verb, auxiliary <i>be</i> , infinitive	be	VBI	BE	VB
Verb, auxiliary <i>be</i> , past	were	VBD	BED	VBD
Verb, auxiliary <i>be</i> , past, 3SG	was	VBD	BEDZ	VBD
Verb, auxiliary <i>be</i> , present part.	being	VBG	BEG	VBG
Verb, auxiliary <i>be</i> , past part.	been	VBN	BEN	VBN
Verb, auxiliary <i>be</i> , present, 3SG	is, 's	VBZ	BEZ	VBZ
Verb, auxiliary <i>be</i> , present 1SG	am, 'm	VBB	BEM	VBP
Verb, auxiliary <i>be</i> , present	are, 're	VBB	BER	VBP
Verb, modal	can, could, 'll	VM0	MD	MD
Infinitive marker	to	TOO	TO	TO
Preposition, to	to	PRP	IN	TO
Preposition	for, above	PRP	IN	IN
Preposition, of	of	PRF	IN	IN
Possessive	's, '	POS	\$	POS
Interjection (or other isolate)	oh, yes, mmm	ITJ	UH	UH
Punctuation, sentence ender	. ! ?	PUN	.	.
Punctuation, semicolon	;	PUN	.	:
Punctuation, colon or ellipsis	: ...	PUN	:	:
Punctuation, comma	,	PUN	,	,
Punctuation, dash	-	PUN	-	-
Punctuation, left bracket	([{	PUL	((
Punctuation, right bracket)] }	PUR))
Punctuation, quotation mark, left	' "	PUQ	<i>not</i> ¹⁾	"
Punctuation, quotation mark, right	' "	PUQ	<i>not</i> ¹⁾	"
Foreign words (not in English lexicon)		UNC	(FW-)	FW
Symbol	[fj] *		<i>not</i> ¹⁾	SYM
Symbol, alphabetical	A, B, c, d	ZZO		
Symbol. List item	A A, First			LS

¹⁾ An entry of *not* means an item was ignored in tagging, or was separated off as a separate token

B Prototype

B.1 Data Format

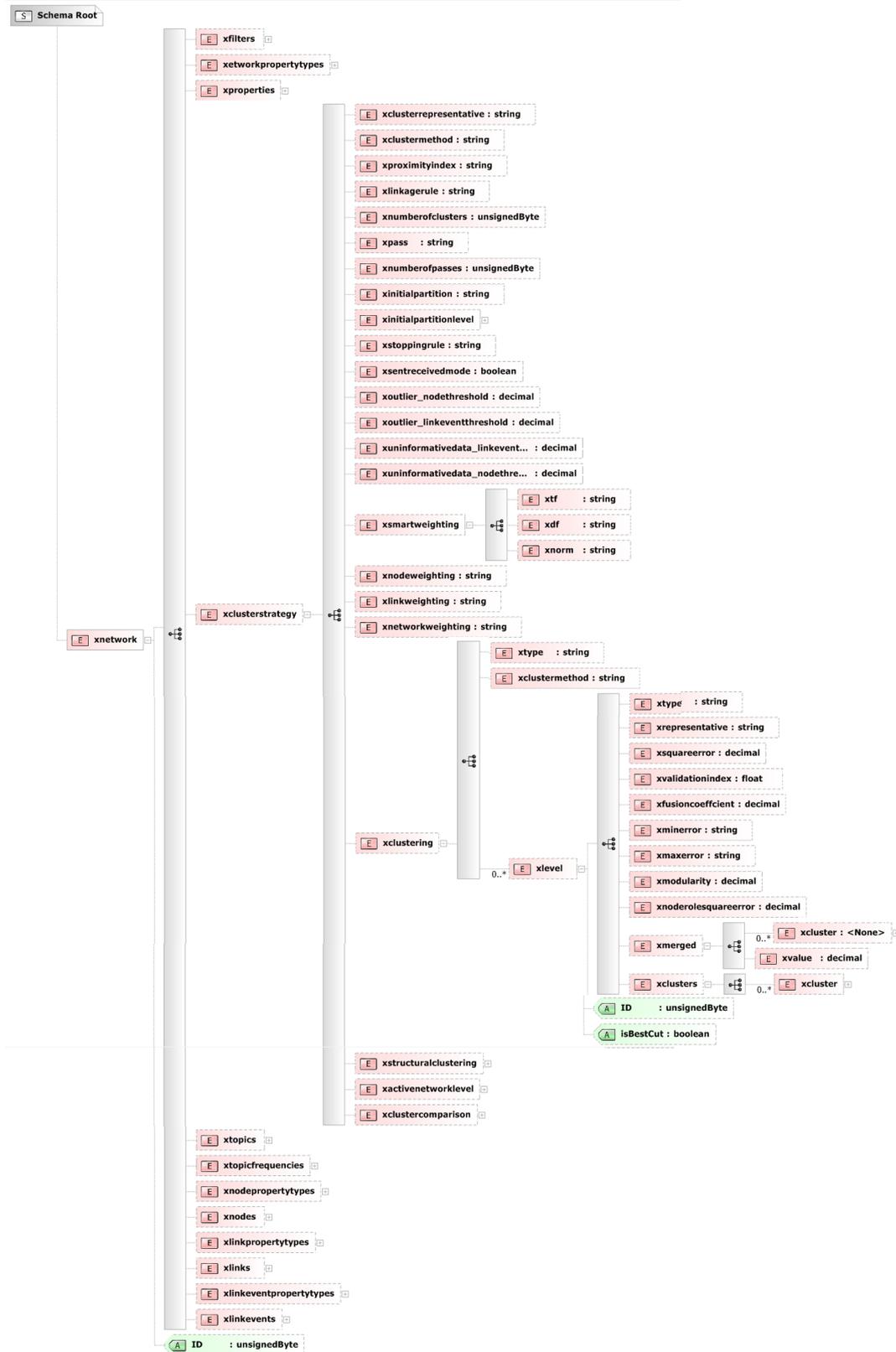


Figure B-1: Prototype data format. XSD schema. Element “xclusterstrategy” partly expanded

B.2 Cluster Analysis

B.2.1 Content-based Clustering on Nodes

B.2.1.1 Level Overview

Table B-1: Content-based clustering on nodes. Level overview

Element	Description	Element	Description
C.id	ID of the selected cluster	R-BC	Betweenness centrality of the cluster representative in the representative graph
Size	Size (number of nodes) of the cluster	R-CC	Closeness centrality of the cluster representative in the representative graph
Nodes	List of cluster members (node IDs)	R-DC	Degree centrality of the cluster representative in the representative graph
R.id	Node ID of cluster representative (when using medoids)	Diameter	Diameter of the cluster
#Comp	Number of unconnected components in the cluster	#Intra-LE	Number of linkevents within the cluster
AVG Comp Size	Average size of each component in the cluster	#Inter-LE	Number of linkevents between the cluster and other clusters
R-CD	Average content dissimilarity of cluster representative to all cluster members	#Inter-LE sent	Number of linkevents sent to other clusters
R-Min-CD	Minimum content dissimilarity of cluster representative to all cluster members	#Inter-LE received	Number of linkevents received from other clusters
R-Max-CD	Maximum content dissimilarity of cluster representative to all cluster members	MOD	Modularity of cluster (see structural clustering, section 5.2.7.1)
#Intra-L	Number of links within the cluster	First Intra-LE Date	Date of first linkevent exchanged in the cluster
#Inter-L	Number of links between the cluster and other clusters	Last Intra-LE Date	Date of last linkevent exchanged in the cluster
%Intra-L	Percentage of all active links that are links within the cluster	%Intra-Timespan	Percentage of entire network activity time span between the cluster and other clusters
%Inter-L	Percentage of all active links that are links between the cluster and other clusters	First Inter-LE Date	Date of first linkevent exchanged with other clusters
AVG Intra-Link Strength	Average number of linkevents per intra-cluster link	Last Inter-LE Date	Date of last linkevent exchanged with other clusters
AVG Inter-Link Strength	Average Number of linkevents per inter-cluster link	%Inter-Timespan	Percentage of entire network activity time span within the cluster
#CtoC	Number of other clusters that are connected to this cluster	First Inter-LE Date sent	Date of first linkevent sent to other clusters
%CtoC	Percentage of all other clusters that are connected to this cluster	Last Inter-LE Date sent	Date of last linkevent sent to other clusters
Density	Density of the cluster	First Inter-LE Date received	Date of first linkevent received from other clusters
#Intra-Iso	Number of isolated nodes in the cluster	Last Inter-LE Date received	Date of first linkevent received from other clusters
%Nodes	Percentage of all active nodes that are members in this cluster	Max Comp Diameter	Maximum diameter of the components within a component of the cluster
AVG Intra-LCC	Average local clustering coefficient in the cluster	#Intra-SelfL	Number of self-links in the cluster

B.2.1.2 Level Details

Table B-2: Content-based clustering on nodes. Level details

Element	Description	Element	Description
N.id	Id of the selected Node	BC	Betweenness centrality of the node in the entire network
C.id	ID of the cluster the node belongs to	Intra-CC	Degree centrality of the node within the cluster
Comp.id	ID of the component within the cluster the node belongs to	CC	Degree centrality of the node in the entire network
isR	Boolean value if the node is the representative of its cluster	Intra-LCC	Local clustering coefficient of the node within the cluster
isIntra-Iso	Boolean value if the node is isolated in the cluster	Max Intra-Dist	Degree Centrality of the node in the entire network
#Intra-L	Number of links between the nodes and other cluster members	#Intra-SelfL	Number of self-links of the node within the cluster
#Inter-L	Number of links between the node and other clusters	First Intra-LE Date	Date of first linkevent exchanged in the cluster
Intra-CD(R)	Content dissimilarity between node and cluster representative	Last Intra-LE Date	Date of last linkevent exchanged in the cluster
AVG Inter-CD(R)	Average content similarity between node and cluster representatives of other clusters	%Intra-Timespan	Percentage of entire network activity time span between the cluster and other clusters
AVG Intra-Link Strength	Average number of linkevents per intra-cluster link of the node	#CtoC	Number of other clusters the node is connected to
Intra-DC	Degree centrality of the node within the cluster	%CtoC	Percentage of all other clusters the node is connected to
DC	Degree Centrality of the node in the entire network	%LE	Percentage of all linkevents in the network that are exchanged by nodes in the cluster
Intra-BC	Betweenness centrality of the node within the network		

B.2.1.3 Node Overview

Note that for content-based clustering on nodes, node (“N”) and node role (“NR”) are the same.

Table B-3: Content-based clustering on nodes. Node overview

Element	Description	Element	Description
NR.id	Id of the selected Node	%CtoC	Percentage of all other clusters the node is connected to
C.id	ID of the cluster the node belongs to	Intra-LCC	Local clustering coefficient of the node within the cluster
#LE(C)	Number of linkevents in the cluster	DC	Degree centrality of the node within the cluster
#N(C)	Number of nodes in the cluster	BC	Betweenness centrality of the node within the network
Comp.id	ID of the component within the cluster the node belongs to	CC	Degree centrality of the node within the cluster
Comp-Size	Number of nodes of the component the node belongs to	#Intra-LE	Number of linkevents exchanged in the cluster
isR	Boolean value if the node is the representative of its cluster	#Intra-LE sent	Number of linkevents sent in the cluster
isIso	Boolean value if the node is isolated in the cluster	#Intra-LE rec	Number of linkevents received in the cluster
#Intra-L	Number of links between the nodes and other cluster members	Max Intra-Dist	Degree Centrality of the node in the entire network
#Inter-L	Number of links between the node and other clusters	#Intra-SelfL	Number of self-links of the node within the cluster
AVG Intra-Link Strength	Average number of linkevents per intra-cluster link of the node	%LE	Percentage of all linkevents in the cluster exchanged by the node
Intra-CD(NR-R)	Content dissimilarity between node and cluster representative	First Intra-LE Date	Date of first linkevent exchanged in the cluster
AVG Inter-CD(NR-R)	Average content similarity between node and cluster representatives of other clusters	Last Intra-LE Date	Date of last linkevent exchanged in the cluster
#CtoC	Number of other clusters the node is connected to	%Intra-Timespan	Percentage of entire network activity time span between the cluster and other clusters

B.2.2 Content-based Clustering on Linkevents

Table B-4: Content-based clustering on linkevents. Level overview

Element	Description	Element	Description
C.id	ID of the selected cluster	AVG Intra-LCC	Average local clustering coefficient in the cluster
Size	Size (number of linkevents) of the cluster	NR-R-BC	Betweenness centrality of the node role cluster representative in the representative graph
Les	List of cluster members (linkevent IDs)	NR-R-CC	Closeness centrality of the node role cluster representative in the representative graph
R.id	Linkevent ID of cluster representative (when using medoids)	NR-R-DC	Degree centrality of the node role cluster representative in the representative graph
#Nodes	Number of nodes assigned to the linkevent-based cluster according to their intra-cluster linkevents	Diameter	Diameter of the cluster
Nodes	List of cluster members (node IDs)	#Intra-LE	Number of linkevents within the cluster
NR-R.id	Node ID of node role cluster representative	#Inter-LE	Number of linkevents between the cluster and other clusters
#Comp	Number of unconnected components in the cluster	#Inter-LE sent	Number of linkevents sent to other clusters
AVG Comp Size	Average size (#nodes) of each component in the cluster	#Inter-LE received	Number of linkevents received from other clusters
NR-R-CD	Average content dissimilarity of node role cluster representative to all node cluster members	MOD	Modularity of cluster (see structural clustering, section 5.2.7.1)
NR-R-Min-CD	Minimum content dissimilarity of node role cluster representative to all node cluster members	First Intra-LE Date	Date of first linkevent exchanged in the cluster
NR-R-Max-CD	Maximum content dissimilarity of node role cluster representative to all node cluster members	Last Intra-LE Date	Date of last linkevent exchanged in the cluster
#Intra-L	Number of links within the cluster	%Intra-Timespan	Percentage of entire network activity time span between the cluster and other clusters
#Inter-L	Number of links between the cluster and other clusters	First Inter-LE Date	Date of first linkevent exchanged with other clusters
%Intra-L	Percentage of all active links that are links within the cluster	Last Inter-LE Date	Date of last linkevent exchanged with other clusters
%Inter-L	Percentage of all active links that are links between the cluster and other clusters	%Inter-Timespan	Percentage of entire network activity time span within the cluster
AVG Intra-Link Strength	Average number of linkevents per intra-cluster link	First Inter-LE Date sent	Date of first linkevent sent to other clusters
AVG Inter-Link Strength	Average Number of linkevents per inter-cluster link	Last Inter-LE Date sent	Date of last linkevent sent to other clusters
#CtoC	Number of other clusters that are connected to this cluster	First Inter-LE Date received	Date of first linkevent received from other clusters
%CtoC	Percentage of all other clusters that are connected to this cluster	Last Inter-LE Date received	Date of first linkevent received from other clusters
Density	Density of the cluster	Max Comp Diameter	Maximum diameter of the components within a component of the cluster
#Intra-Iso	Number of isolated nodes in the cluster	#Intra-SelfL	Number of self-links in the cluster
%Nodes	Percentage of all active nodes that are members in this cluster		

C Case Study

C.1 Data Preparation

Table C-1: List of Enron employees. Nodes 1125-1191

NodeID	E-Mail Address	Name	Function (Information)
1125	lynn.blair@enron.com	Lynn Blair	N/A
1126	mike.grigsby@enron.com	Mike Grigsby	Manager
1127	michelle.cash@enron.com	Michelle Cash	N/A
1128	monika.causholli@enron.com	Monika Causholli	Employee (Analyst Risk Management)
1130	marie.heard@enron.com	Marie Heard	N/A
1132	mark.whitt@enron.com	Mark Whitt	N/A
1134	stephanie.panus@enron.com	Stephanie Panus	Employee
1136	theresa.staab@enron.com	Theresa Staab	Employee
1138	richard.ring@enron.com	Richard Ring	Employee
1139	stacy.dickson@enron.com	Stacy Dickson	Employee
1140	teb.lokey@enron.com	Teb Lokey	Manager (Regulatory Affairs)
1141	tracy.geaccone@enron.com	Tracy Geaccone	N/A
1142	kevin.hyatt@enron.com	Kevin Hyatt	N/A (Pipeline Business)
1143	robert.badeer@enron.com	Robert Badeer	Director
1144	lindy.donoho@enron.com	Lindy Donoho	Employee
1145	kimberly.watson@enron.com	Kimberly Watson	N/A
1146	keith.holst@enron.com	Keith Holst	Director
1147	joe.quenet@enron.com	Joe Quenet	Trader
1150	jay.reitmeyer@enron.com	Jay Reitmeyer	Employee
1151	frank.ermis@enron.com	Frank Ermis	Director
1152	elizabeth.sager@enron.com	Elizabeth Sager	Employee
1153	darrell.schoolcraft@enron.com	Darrell Schoolcraft	N/A
1154	danny.mccarty@enron.com	Danny McCarty	Vice President
1157	shelley.corman@enron.com	Shelley Corman	Vice President (Regulatory Affairs)
1158	kim.ward@enron.com	Kim Ward	N/A
1159	juan.hernandez@enron.com	Juan Hernandez	N/A
1160	joe.stepenovitch@enron.com	Joe Stepenovitch	Vice President
1162	jane.tholt@enron.com	Jane Tholt	Vice President
1163	barry.tycholiz@enron.com	Barry Tycholiz	Vice President
1164	dana.davis@enron.com	Dana Davis	Vice President
1165	tori.kuykendall@enron.com	Tori Kuykendall	Trader
1167	martin.cuilla@enron.com	Martin Cuilla	Manager
1169	john.arnold@enron.com	Arnold John	N/A
1170	harry.arora@enron.com	Harry Arora	Vice President
1172	brad.mckay@enron.com	Brad Mckay	N/A
1173	tana.jones@enron.com	Tana Jones	N/A
1174	susan.scott@enron.com	Susan Scott	N/A
1175	susan.bailey@enron.com	Susan Bailey	N/A
1177	kay.mann@enron.com	Kay Mann	Employee
1178	lavorato@enron.com	John Lavorato	0
1179	greg.whalley@enron.com	Greg Whalley	President
1181	jason.wolfe@enron.com	Jason Wolfe	N/A
1182	jason.williams@enron.com	Williams Jason	N/A
1183	jim.schwieger@enron.com	Jim Schwieger	Trader
1184	monique.sanchez@enron.com	Monique Sanchez	N/A
1185	kevin.ruscitti@enron.com	Kevin Ruscitti	N/A
1186	matthew.lenhart@enron.com	Matthew Lenhart	Employee
1187	peter.keavey@enron.com	Peter Keavey	Employee
1188	scott.hendrickson@enron.com	Scott Hendrickson	N/A
1189	tom.donohoe@enron.com	Tom Donohoe	N/A
1191	stanley.horton@enron.com	Stanley Horton	President (Enron Gas Pipeline)

Table C-2: List of Enron employees. Nodes 1192-1275

NodeID	E-Mail Address	Name	Function (Information)
1192	sara.shackleton@enron.com	Sara Shackleton	N/A
1193	rod.hayslett@enron.com	Rod Hayslett	Vice President, Chief Financial Officer and Treasurer
1194	richard.shapiro@enron.com	Richard Shapiro	Vice President (Regulatory Affairs)
1195	michelle.lokay@enron.com	Michelle Lokay	Employee (Administrative Asisstant)
1196	matt.smith@enron.com	Smith Matt	N/A
1198	jeff.dasovich@enron.com	Jeff Dasovich	Employee (Government Relation Executive)
1199	james.derrick@enron.com	James Derrick	House Lawyer
1200	gerald.nemec@enron.com	Gerald Nemec	N/A
1201	debra.perlingiere@enron.com	Debra Perlingiere	N/A
1202	john.zufferli@enron.com	John Zufferli	Employee
1203	andy.zipper@enron.com	Andy Zipper	Vice President (Enron Online)
1206	judy.townsend@enron.com	Judy Townsend	Employee
1209	geoff.storey@enron.com	Geoff Storey	Director
1210	geir.solberg@enron.com	Geir Solberg	Employee
1211	cara.semperger@enron.com	Cara Semperger	Employee (Senior Analyst Cash)
1212	holden.salisbury@enron.com	Holden Salisbury	N/A (Cash Analyst)
1214	andrea.ring@enron.com	Andrea Ring	N/A
1215	cooper.richey@enron.com	Cooper Richey	Manager
1216	dutch.quigley@enron.com	Dutch Quigley	N/A
1218	phillip.platter@enron.com	Phillip Platter	Employee (Sr. Specialist)
1221	scott.neal@enron.com	Scott Neal	Vice President
1222	matt.motley@enron.com	Matt Motley	Director
1225	errol.mclaughlin@enron.com	Errol McLaughlin	Employee
1226	jonathan.mckay@enron.com	Jonathan Mckay	Director
1227	larry.may@enron.com	Larry May	Director
1229	mike.maggi@enron.com	Mike Maggi	Director
1232	louise.kitchen@enron.com	Louise Kitchen	President (Enron Online)
1233	jeff.king@enron.com	Jeff King	Manager
1235	john.hodge@enron.com	John Hodge	Managing Director
1236	john.griffith@enron.com	John Griffith	N/A
1238	doug.gilbert-smith@enron.com	Doug Gilbert-smith	Manager
1239	chris.germany@enron.com	Chris Germany	Employee
1240	lisa.gang@enron.com	Lisa Gang	N/A
1243	chris.dorland@enron.com	Chris Dorland	Employee
1247	sally.beck@enron.com	Sally Beck	Employee (Chief Operating Officer)
1248	rick.buy@enron.com	Rick Buy	Manager (Chief Risk Management Officer)
1249	don.baughman@enron.com	Don Baughman	Trader
1250	eric.bass@enron.com	Eric Bass	Trader
1252	kenneth.lay@enron.com	Kenneth Lay	CEO
1253	kam.keiser@enron.com	Kam Keiser	Employee
1254	jeff.skilling@enron.com	Jeffrey Skilling	CEO
1255	sean.crandall@enron.com	Sean Crandall	Director
1256	ryan.slinger@enron.com	Ryan Slinger	Trader
1257	mike.swerzbin@enron.com	Mike Swerzbin	Trader
1258	diana.scholtes@enron.com	Diana Scholtes	Trader
1260	bill.williams@enron.com	Bill Williams	N/A
1261	drew.fossum@enron.com	Drew Fossum	Vice President
1262	mark.guzman@enron.com	Mark Guzman	Trader
1263	mary.hain@enron.com	Mary Hain	House Lawyer
1264	lysa.akin@enron.com	Lysa Akin	Employee
1265	steven.harris@enron.com	Steven Harris	N/A
1266	dan.hyvl@enron.com	Dan Hyvl	Employee
1269	robin.rodrique@enron.com	Robin Rodrigue	N/A
1270	steven.south@enron.com	Steven South	N/A
1271	carol.clair@enron.com	Carol Clair	House Lawyer
1272	chris.stokley@enron.com	Chris Stokley	N/A
1273	kate.symes@enron.com	Kate Symes	Employee
1274	liz.taylor@enron.com	Liz Taylor	N/A
1275	rosalee.fleming@enron.com	Rosalee Fleming	Employee

C.2 Text Mining

Table C-3: Case study. Text mining methods. Configurations

Component	Method	Configuration
Tokenizer	Replacement Filter	The messages contain artifacts that should not be part of the analysis, e.g. e-mail headers, documents attached, links to websites, e-mail addresses, original messages attached to replies, html tags, etc. Wildcards are employed as search patterns. Matches to the search patterns are replaced by whitespaces or empty strings.
	Brill Tagger (POS Tagging)	Part-of-speech tagging is applied to the data. Only nouns indicated by the tags "NN", "NNS", "NNP" and "NNPS" are kept for further analysis. The ContentMiner implements the Brill Tagger using the Penn Treebank tag set.
TermReducer	Splitter	The text is split at whitespaces.
	Stopword Filter (English)	A list of alphabetic characters and common English stop words, e.g. "a", "always", "and", "often", etc.
	Stopword Filter (Enron)	An Enron specific stop word list including topics that are meaningless in this context (e.g. "Enron") or misspellings that will not be identified by the WordSplitter (e.g. "inthe").
	WordSplitter	Topics that do not fulfill the following criteria are split if the identified elements occur more often in the data set than the topic itself: Minimum cumulated term frequency = 5 Minimum term length = 3
	TermLength Filter	Topics which do not fulfill the following criteria are removed: Minimum term length = 3 Maximum term length = 30
	Stemmer	Stemming collapses derivationally related words to a single topic. The ContentMiner implements the Porter Stemmer (Porter 1980)
	Low-Frequency Filter	Topics which do not fulfill the following criteria are removed: Minimum cumulated term frequency = 10 Minimum document frequency = 10

C.3 Network Analysis

C.3.1 SNA Metrics on Node Level

Table C-4: Case study. Network analysis. Node level. SNA Metrics. Structural key players: top 9 marked bold, top 21 underlined

N.id	#LE	ØLS	DC [%]	BC [%]	CC	LCC [%]	First LE Date	Last LE Date	Time span [%]
1126	107 (21)	5.63 (49)	19 (3)	12.63 (2)	2.04 (1)	22.81 (47)	05.09.2000 11:52	29.03.2001 11:57	97.04 (19)
<u>1134</u>	392 (6)	49.0 (5)	8 (13)	0.22 (58)	2.84 (42)	64.29 (12)	05.09.2000 02:02	30.03.2001 02:42	97.76 (12)
<u>1139</u>	85 (29)	21.25 (10)	4 (17)	0.06 (66)	2.96 (52)	50.0 (18)	07.09.2000 03:29	29.03.2001 03:31	96.32 (23)
<u>1143</u>	317 (8)	52.83 (4)	6 (15)	0.42 (51)	2.54 (22)	60.0 (15)	05.09.2000 12:49	23.03.2001 08:48	94.6 (25)
<u>1157</u>	116 (19)	7.25 (37)	16 (5)	5.15 (12)	2.36 (10)	35.0 (29)	01.09.2000 07:24	30.03.2001 09:11	99.68 (4)
<u>1158</u>	93 (25)	7.15 (39)	13 (8)	6.45 (8)	2.26 (6)	32.05 (33)	31.10.2000 05:47	30.03.2001 07:09	71.42 (63)
1173	623 (4)	29.67 (8)	21 (2)	11.2 (4)	2.22 (4)	18.57 (50)	05.09.2000 02:51	30.03.2001 08:08	97.85 (10)
1174	340 (7)	20.0 (13)	17 (4)	7.97 (6)	2.26 (6)	28.68 (39)	01.09.2000 01:09	30.03.2001 09:11	99.8 (1)
<u>1175</u>	280 (9)	35 (7)	8 (13)	0.32 (53)	2.77 (36)	60.71 (14)	06.09.2000 07:51	30.03.2001 08:04	97.28 (14)
<u>1184</u>	64 (35)	4.0 (59)	16 (5)	4.13 (14)	2.37 (11)	29.17 (38)	01.09.2000 01:09	27.03.2001 06:52	98.57 (8)
1192	734 (3)	73.4 (1)	10 (11)	0.96 (40)	2.66 (29)	44.44 (20)	05.09.2000 02:02	30.03.2001 07:15	97.85 (10)
1194	793 (2)	66.08 (2)	12 (9)	4.7 (13)	2.22 (4)	27.27 (43)	01.09.2000 06:22	30.03.2001 10:47	99.73 (2)
1198	985 (1)	65.67 (3)	15 (6)	5.91 (9)	2.2 (3)	26.67 (44)	01.09.2000 06:22	30.03.2001 10:47	99.73 (2)
<u>1200</u>	142 (15)	12.91 (24)	11 (10)	5.34 (10)	2.24 (5)	27.27 (43)	01.09.2000 10:59	30.03.2001 06:20	99.55 (5)
1221	50 (44)	3.13 (67)	16 (5)	9.75 (5)	2.26 (6)	20.0 (49)	12.09.2000 05:35	27.03.2001 06:32	93.26 (32)
<u>1261</u>	226 (11)	11.89 (27)	19 (3)	6.94 (7)	2.33 (9)	29.24 (37)	05.09.2000 06:17	30.03.2001 09:11	97.8 (11)
1263	393 (5)	18.71 (14)	21 (2)	11.6 (3)	2.04 (1)	17.62 (51)	01.09.2000 06:56	30.03.2001 09:11	99.69 (3)
1264	100 (23)	4.17 (57)	24 (1)	13.72 (1)	2.18 (2)	21.74 (48)	07.09.2000 12:56	23.03.2001 10:47	93.69 (27)
<u>1265</u>	192 (12)	27.43 (9)	7 (14)	0.09 (63)	2.75 (34)	90.48 (2)	01.09.2000 07:24	30.03.2001 09:11	99.68 (4)
<u>1271</u>	270 (10)	45.0 (6)	6 (15)	0.05 (67)	2.96 (52)	73.33 (7)	06.09.2000 09:05	30.03.2001 07:15	97.23 (17)
<u>1273</u>	115 (20)	7.19 (38)	16 (5)	2.18 (26)	2.74 (33)	30.0 (36)	31.10.2000 05:47	30.03.2001 05:50	71.16 (66)

N.id:= node id
 ØLS:= average link strength

DC:= degree centrality
 BC:= betweenness centrality

CC: closeness centrality
 LE:= linkevent

C.3.2 Node Role Categorization

Table C-5: Case study. Network analysis. Node level. Node role categorization. Nodes 1125-1195

N.id	Comp.id	Intra-DC	Intra-BC	Type	DC Type	BC Type	Node Role
1125	2	5	0	both	low	0	Peripheral Specialist
1126	2	19	12.63	both	low	medium	Mediator
1127	2	7	0.27	both	low	low	Middleman
1128	2	7	0.03	both	low	low	Middleman
1130	2	5	0.04	both	low	low	Middleman
1132	2	7	1.36	both	low	low	Middleman
1134	2	8	0.22	both	low	low	Middleman
1136	2	2	0	both	low	0	Peripheral Specialist
1138	2	1	0	recipient	low	0	Peripheral Specialist
1139	2	4	0.06	both	low	low	Middleman
1140	2	2	0	recipient	low	0	Peripheral Specialist
1141	2	4	0	both	low	0	Peripheral Specialist
1142	2	10	0.62	both	low	low	Middleman
1143	2	6	0.42	both	low	low	Middleman
1144	2	9	0.23	both	low	low	Middleman
1145	2	6	0.02	both	low	low	Middleman
1146	2	8	2.53	recipient	low	low	Middleman
1147	0	100	0	sender	100	0	(Dyad)
1150	2	7	0.29	both	low	low	Middleman
1151	2	4	0	both	low	0	Peripheral Specialist
1152	2	9	2.97	both	low	low	Middleman
1153	2	4	0	both	low	0	Peripheral Specialist
1154	2	5	0.14	both	low	low	Middleman
1157	2	16	5.15	both	low	low	Middleman
1158	2	13	6.45	both	low	low	Middleman
1159	1	50	50	recipient	high	high	Hot Spot
1160	1	25	0	recipient	medium	0	Team Worker
1162	2	8	2.5	both	low	low	Middleman
1163	2	11	2.68	both	low	low	Middleman
1164	2	1	0	recipient	low	0	Peripheral Specialist
1165	2	8	1.28	both	low	low	Middleman
1167	2	3	0.08	both	low	low	Middleman
1169	2	15	5.3	both	low	low	Middleman
1170	2	2	0	both	low	0	Peripheral Specialist
1172	2	4	0.02	recipient	low	low	Middleman
1173	2	21	11.2	both	medium	medium	Integrator
1174	2	17	7.97	both	low	low	Middleman
1175	2	8	0.32	both	low	low	Middleman
1177	2	7	0.55	both	low	low	Middleman
1178	2	7	0.99	sender	low	low	Middleman
1179	2	12	2.38	both	low	low	Middleman
1181	2	5	0.06	recipient	low	low	Middleman
1182	2	8	0.31	both	low	low	Middleman
1183	2	4	0.21	both	low	low	Middleman
1184	2	16	4.13	both	low	low	Middleman
1185	2	2	0.12	both	low	low	Middleman
1186	2	14	2.81	both	low	low	Middleman
1187	2	8	1.24	both	low	low	Middleman
1188	2	3	0	both	low	0	Peripheral Specialist
1191	2	10	3.67	both	low	low	Middleman
1192	2	10	0.96	both	low	low	Middleman
1193	2	7	0.21	both	low	low	Middleman
1194	2	12	4.7	both	low	low	Middleman
1195	2	8	0.21	both	low	low	Middleman

N.id:= node id

Comp.id:= component id

Thresholds:

BC:= betweenness centrality

DC:= degree centrality

DC-low <20 %; DC-medium<40 %

Intra-BC:= intra-cluster degree centrality

Intra-DC:= intra-cluster degree centrality

BC-low<10 %; BC-medium<30 %

Table C-6: Case study. Network analysis. Node level. Node role categorization. Nodes 1196-1275

N.id	Comp.id	Intra-DC	Intra-BC	Type	DC Type	BC Type	Node Role
1196	2	3	0	both	low	0	Peripheral Specialist
1198	2	15	5.91	both	low	low	Middleman
1199	2	6	0.23	both	low	low	Middleman
1200	2	11	5.34	both	low	low	Middleman
1201	2	9	1.46	both	low	low	Middleman
1202	2	3	0.12	recipient	low	low	Middleman
1203	2	8	0.91	both	low	low	Middleman
1206	2	4	0.02	both	low	low	Middleman
1210	2	7	0.09	both	low	low	Middleman
1211	2	13	0.67	both	low	low	Middleman
1212	2	7	0.02	both	low	low	Middleman
1214	2	8	2.5	both	low	low	Middleman
1215	2	5	0.6	both	low	low	Middleman
1216	2	10	2.37	both	low	low	Middleman
1218	2	6	0.13	both	low	low	Middleman
1221	2	16	9.75	both	low	low	Middleman
1222	2	4	0.02	recipient	low	low	Middleman
1225	2	9	0.78	both	low	low	Middleman
1226	2	3	0.22	recipient	low	low	Middleman
1227	2	10	1.82	both	low	low	Middleman
1229	2	6	0.57	both	low	low	Middleman
1232	2	12	2.64	both	low	low	Middleman
1235	2	6	0.79	both	low	low	Middleman
1236	2	10	1.72	both	low	low	Middleman
1239	2	9	1.05	both	low	low	Middleman
1240	2	4	0	recipient	low	0	Peripheral Specialist
1247	2	9	1.5	both	low	low	Middleman
1248	2	7	0.97	both	low	low	Middleman
1250	2	6	0.6	both	low	low	Middleman
1252	2	5	0.12	recipient	low	low	Middleman
1253	2	8	1.66	both	low	low	Middleman
1254	2	7	0.37	recipient	low	low	Middleman
1255	2	4	0.02	recipient	low	low	Middleman
1256	2	3	0	recipient	low	0	Peripheral Specialist
1257	2	5	1.38	recipient	low	low	Middleman
1258	2	5	0.07	both	low	low	Middleman
1260	2	5	0.59	recipient	low	low	Middleman
1261	2	19	6.94	both	low	low	Middleman
1262	2	6	0.01	recipient	low	low	Middleman
1263	2	21	11.6	both	medium	medium	Integrator
1264	2	24	13.72	both	medium	medium	Integrator
1265	2	7	0.09	both	low	low	Middleman
1266	2	11	3.97	both	low	low	Middleman
1269	2	7	0.93	both	low	low	Middleman
1270	2	3	0	recipient	low	0	Peripheral Specialist
1271	2	6	0.05	both	low	low	Middleman
1272	2	3	0	both	low	0	Peripheral Specialist
1273	2	16	2.18	both	low	low	Middleman
1274	2	13	2.44	both	low	low	Middleman
1275	2	13	1.93	both	low	low	Middleman

C.3.3 Temperature View

C.3.3.1 Temperature Overview

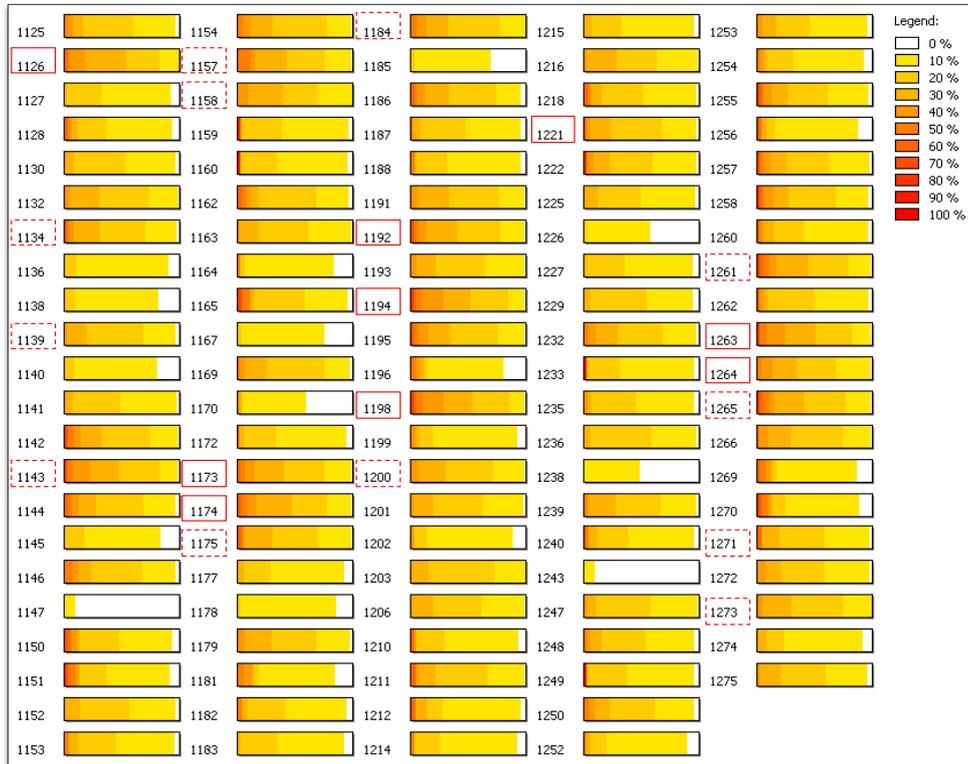


Figure C-1: Case study. Temperature view. Overview of content profiles. Similarity values ranging from 0% (white) to 100% (red). Structural key players marked with red solid line (top 9) or dashed line (top 21)

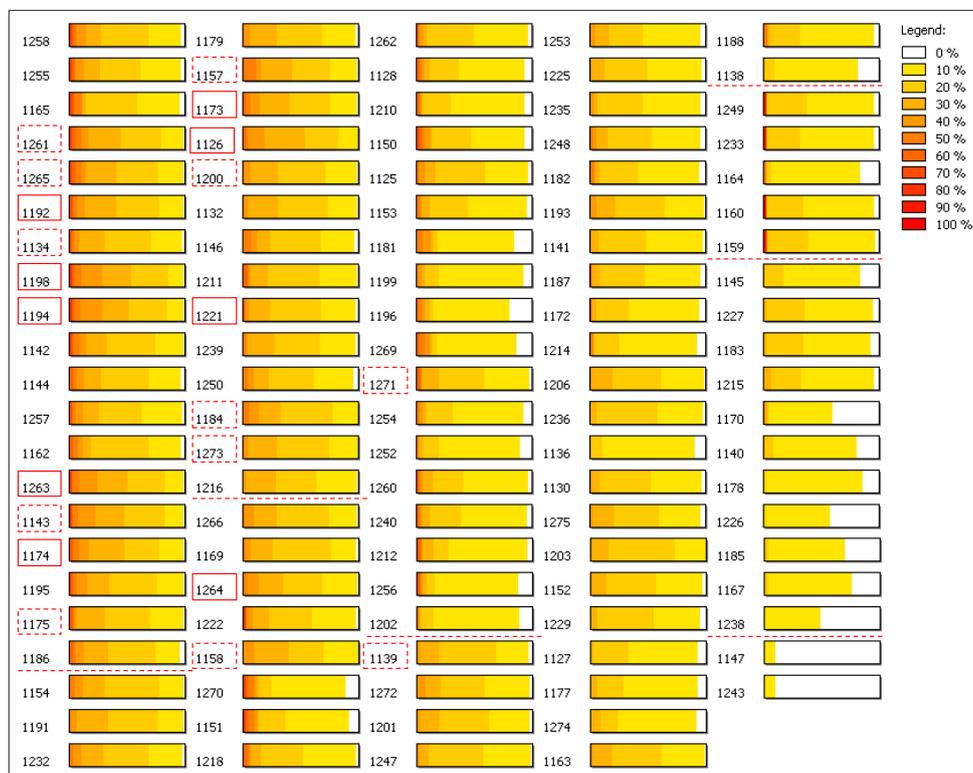
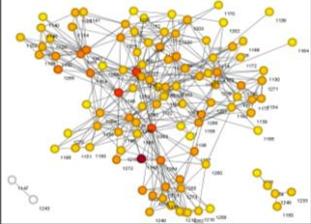
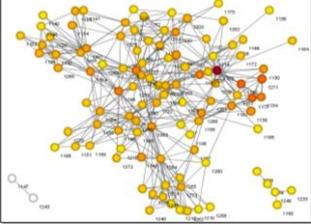
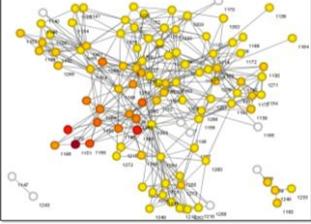
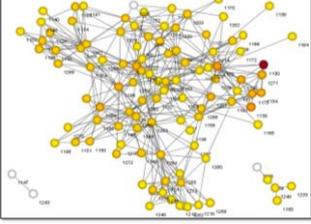
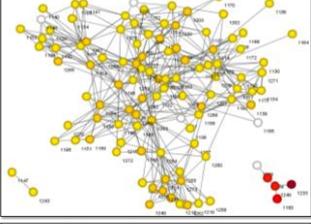
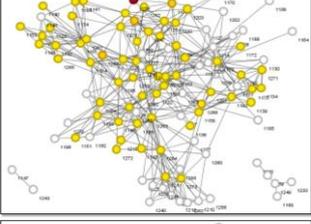
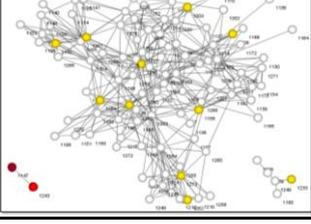


Figure C-2: Case study. Temperature view. Comparison of content profiles. Nodes ordered by profile similarity. Groups of similar profiles separated by red dashed lines. Similarity values ranging from 0% (white) to 100% (red). Structural key players marked with red solid line (top 9) or dashed line (top 21)

C.3.3.2 Groups of Similar Content Profiles

Table C-7: Case study. Groups of similar content profiles. Structural key players marked bold (top 9) or underlined (top 21)

Group	Content Profile (Example)	Temperature Graphview (Ex.)	Nodes
(1)	1143 		<u>1134</u> , 1142, <u>1143</u> , 1144, 1162, 1165, 1174 , <u>1175</u> , 1186, 1192 , 1194 , 1195, 1198 , 1255, 1257, 1258, <u>1261</u> , 1263 , <u>1265</u>
(2)	1173 		1126 , 1132, 1146, 1154, <u>1157</u> , 1173 , 1179, <u>1184</u> , 1191, <u>1200</u> , 1211, 1216, 1221 , 1232, 1239, 1250, <u>1273</u>
(3)	1151 		1125, 1128, 1150, 1151, 1153, <u>1158</u> , 1169, 1181, 1196, 1199, 1210, 1212, 1218, 1222, 1240, 1252, 1254, 1256, 1260, 1262, 1264 , 1266, 1269, 1270, <u>1271</u>
(4)	1130 		1127, 1130, 1136, 1138, <u>1139</u> , 1141, 1152, 1163, 1172, 1177, 1182, 1187, 1188, 1193, 1201, 1203, 1206, 1214, 1225, 1229, 1235, 1236, 1247, 1248, 1253, 1272, 1274, 1275
(5)	1233 		1159, 1160, 1164, 1233, 1249
(6)	1226 		1140, 1145, 1167, 1170, 1178, 1183, 1185, 1215, 1226, 1227, 1238,
(7)	1147 		1147, 1243

C.4 Cluster Analysis

C.4.1 Structural Clustering

C.4.1.1 Cluster Overview

Table C-8: Case study. Structural clustering. Overview of clustering solution (main component). Cluster representatives are marked with asterix, structural key players marked bold (top 9) or underlined (top 21)

ID	Size	Nodes	#IAL	#IEL	ØIALS	ØIELS	%IALE	%IELE	Den	ØIA-LCC	Dia
2	9	1138, 1167, 1172, 1188, 1206, 1214, 1221 , 1235, *1239	18	18	5.44	2.5	2.71	1.24	50	56.77	3
3	16	1125, 1140, 1141, 1142, 1144, 1145, 1153, 1154, <u>1157</u> , 1174 , 1191, 1193, 1198 , 1195, <u>1261</u> , * 1265	56	32	8.21	33.78	12.72	29.89	46.67	80.51	2
4	18	1128, <u>1143</u> , 1210, 1211, 1212, 1218, 1222, 1240, 1255, 1256, 1257, *1258, 1260, 1262, 1263 , 1264 , 1272, <u>1273</u>	61	24	4.33	23.67	7.30	15.71	39.87	71.41	2
5	15	1127, 1170, 1178, 1179, 1194 , 1199, 1202, 1203, *1232, 1247, 1248, 1252, 1254, 1274, 1275	42	39	3.57	24.03	4.15	25.91	40	52.31	3
6	19	1130, 1132, <u>1134</u> , 1136, <u>1139</u> , 1152, <u>1158</u> , 1163, 1164, 1173 , <u>1175</u> , 1177, 1182, 1185, * 1192 , <u>1200</u> , 1201, 1266, <u>1271</u>	58	37	23.03	3.78	36.95	3.87	33.92	56.16	3
7	24	1126 , 1146, 1150, 1151, 1162, *1165, 1169, 1181, 1183, <u>1184</u> , 1186, 1187, 1196, 1215, 1216, 1225, 1226, 1227, 1229, 1236, 1250, 1253, 1269, 1270	73	50	3.04	4.26	6.14	5.89	26.45	57.59	4

#IAL:= number of intra-cluster links

#IEL:= number of inter-cluster links

%IALE:= percentage of intra-cluster linkevents on all linkevents

%IELE:= percentage of inter-cluster linkevents on all linkevents

ØIALS:= average intra-cluster link strength

ØIELS:= average inter-cluster link strength

ØIA-LCC:= average intra-cluster local clustering coefficient

Den:= density

Dia:= diameter

C.4.1.2 Node Role Categorization

Table C-9: Case study. Structural clustering. Node role categorization. Clusters 0-4

N.id	C.id	Intra-DC	Intra-BC	Type	DC Type	BC Type	Role
1147	0	100	0	sender	100	0	(Dyad)
1243	0	100	0	recipient	100	0	(Dyad)
1159	1	50	50	recipient	high	high	Hot Spot
1160	1	25	0	recipient	medium	0	Specialist
1233	1	25	0	recipient	medium	0	Specialist
1238	1	25	0	sender	medium	0	Specialist
1249	1	75	83.33	sender	high	high	Hot Spot
1138	2	12.5	0	recipient	low	0	Peripheral Specialist
1167	2	12.5	0	recipient	low	0	Peripheral Specialist
1172	2	50	1.79	recipient	high	low	Coordinator
1188	2	37.5	0	both	medium	0	Specialist
1206	2	50	1.79	both	high	low	Coordinator
1214	2	75	28.27	both	high	medium	Information Spreader
1221	2	75	6.85	both	high	low	Coordinator
1235	2	50	0.89	both	high	low	Coordinator
1239	2	87.5	31.85	both	high	high	Hot Spot
1125	3	33.33	0	both	medium	0	Specialist
1140	3	13.33	0	recipient	low	0	Peripheral Specialist
1141	3	26.67	0	both	medium	0	Specialist
1142	3	66.67	3.68	both	high	low	Coordinator
1144	3	60	2.27	both	high	low	Coordinator
1145	3	40	0.95	both	high	low	Coordinator
1153	3	26.67	0	both	medium	0	Specialist
1154	3	26.67	0.32	both	medium	low	Team Worker
1157	3	93.33	20.51	both	high	medium	Information Spreader
1174	3	60	2.9	both	high	low	Coordinator
1191	3	26.67	0	both	medium	0	Specialist
1193	3	40	1.59	both	high	low	Coordinator
1195	3	53.33	1.33	both	high	low	Coordinator
1198	3	33.33	0	both	medium	0	Specialist
1261	3	100	27.02	both	100	medium	(Star)
1265	3	46.67	0.38	both	high	low	Coordinator
1128	4	41.18	1.09	both	high	low	Coordinator
1143	4	23.53	0.08	recipient	medium	low	Team Worker
1210	4	41.18	1.84	both	High	low	Coordinator
1211	4	76.47	9.36	both	High	low	Coordinator
1212	4	41.18	0.88	both	High	low	Coordinator
1218	4	35.29	0.69	both	medium	low	Team Worker
1222	4	23.53	0.08	recipient	medium	low	Team Worker
1240	4	23.53	0	recipient	medium	0	Specialist
1255	4	23.53	0.08	recipient	medium	low	Team Worker
1256	4	17.65	0	recipient	Low	0	Peripheral Specialist
1257	4	23.53	0.08	recipient	medium	low	Team Worker
1258	4	29.41	0.33	both	medium	low	Team Worker
1260	4	23.53	0.45	recipient	medium	low	Team Worker
1262	4	35.29	0.48	recipient	medium	low	Team Worker
1263	4	58.82	7.41	both	High	low	Coordinator
1264	4	100	27.58	both	100	medium	(Star)
1272	4	11.76	0	both	Low	0	Peripheral Specialist
1273	4	88.24	17.21	both	High	medium	Information Spreader

N.id:= node id BC:= betweenness centrality Intra-BC:= intra-cluster degree centrality
 Comp.id:= component id DC:= degree centrality Intra-DC:= intra-cluster degree centrality
 Thresholds: DC-low <20 %; DC-medium<40 % BC-low<10 %; BC-medium<30 %

Table C-10: Case study. Structural clustering. Node role categorization. Clusters 5-7

N.id	C.id	Intra-DC	Intra-BC	Type	DC Type	BC Type	Role
1127	5	28.57	0	both	medium	0	Specialist
1170	5	14.29	0	both	low	0	Peripheral Specialist
1178	5	35.71	15.9	sender	medium	medium	Integrator
1179	5	64.29	12.44	both	high	medium	Information Spreader
1194	5	35.71	1.61	both	medium	low	Team Worker
1199	5	35.71	0.86	both	medium	low	Team Worker
1202	5	7.14	0	recipient	low	0	Peripheral Specialist
1203	5	35.71	2.78	both	medium	low	Team Worker
1232	5	64.29	18.35	both	high	medium	Information Spreader
1247	5	35.71	1.36	both	medium	low	Team Worker
1248	5	35.71	0.95	both	medium	low	Team Worker
1252	5	35.71	3.33	recipient	medium	low	Team Worker
1254	5	42.86	5.02	recipient	high	low	Coordinator
1274	5	71.43	15.79	both	high	medium	Information Spreader
1275	5	57.14	5.13	both	high	low	Coordinator
1130	6	27.78	0.78	both	medium	low	Team Worker
1132	6	27.78	2.09	both	medium	low	Team Worker
1134	6	44.44	3.25	both	high	low	Coordinator
1136	6	11.11	0	both	low	0	Peripheral Specialist
1139	6	22.22	0.95	both	medium	low	Team Worker
1152	6	22.22	0.29	both	medium	low	Team Worker
1158	6	50	8.72	both	high	low	Coordinator
1163	6	33.33	0.84	both	medium	low	Team Worker
1164	6	5.56	0	recipient	low	0	Peripheral Specialist
1173	6	66.67	32.69	both	high	high	Hot Spot
1175	6	44.44	4.6	both	high	low	Coordinator
1177	6	27.78	0.42	both	medium	low	Team Worker
1182	6	44.44	4.57	both	high	low	Coordinator
1185	6	5.56	0	both	low	0	Peripheral Specialist
1192	6	50	5.2	both	high	low	Coordinator
1200	6	38.89	12.69	both	medium	medium	Integrator
1201	6	44.44	4.71	both	high	low	Coordinator
1266	6	44.44	13.96	both	high	medium	Information Spreader
1271	6	33.33	0.99	both	medium	low	Team Worker
1126	7	47.83	10.68	both	high	medium	Information Spreader
1146	7	21.74	2.54	recipient	medium	low	Team Worker
1150	7	30.43	1.13	both	medium	low	Team Worker
1151	7	17.39	0	both	low	0	Peripheral Specialist
1162	7	21.74	0.63	both	medium	low	Team Worker
1165	7	26.09	1.26	both	medium	low	Team Worker
1169	7	34.78	7.34	both	medium	low	Team Worker
1181	7	21.74	1.08	recipient	medium	low	Team Worker
1183	7	8.7	0.47	both	low	low	Middleman
1184	7	65.22	28.68	both	high	medium	Information Spreader
1186	7	56.52	12.96	both	high	medium	Information Spreader
1187	7	26.09	5.17	both	medium	low	Team Worker
1196	7	13.04	0	both	low	0	Peripheral Specialist
1215	7	13.04	0	both	low	0	Peripheral Specialist
1216	7	30.43	4.41	both	medium	low	Team Worker
1225	7	34.78	6.74	both	medium	low	Team Worker
1226	7	8.7	0.3	recipient	low	low	Middleman
1227	7	26.09	2.66	both	medium	low	Team Worker
1229	7	17.39	0	both	low	0	Peripheral Specialist
1236	7	30.43	3.34	both	medium	low	Team Worker
1250	7	21.74	6.24	both	medium	low	Team Worker
1253	7	21.74	1.2	both	medium	low	Team Worker
1269	7	26.09	5.92	both	medium	low	Team Worker
1270	7	13.04	0	recipient	low	0	Peripheral Specialist

C.4.2 Content-based Clustering on Nodes

C.4.2.1 Cluster Overview

Table C-11: Case study. Content-based clustering on nodes. Overview of clustering solution. Cluster representatives are marked with asterix, structural key players marked bold (top 9) or underlined (top 21)

ID	Size	Nodes	#IAL	#IEL	ØIALS	ØIELS	%IALE	%IELE	Den	ØIA-LCC	Dia
0	20	1125, 1132, 1136, 1140, 1141, 1142, 1144, 1153, 1154, <u>1157</u> , 1164, 1174 , 1191, 1193, 1195, <u>1200</u> , 1235, 1236, <u>1261</u> , *1265	53	54	9.51	5.54	13.94	8.27	27.89	67.01	4
1	16	<u>1126</u> , 1146, 1150, 1151, 1162, *1165, 1181, <u>1184</u> , 1186, 1196, 1202, 1216, 1250, 1253, 1269, 1270	41	47	3.17	3.96	3.60	5.14	34.17	75.21	2
2	47	1127, 1130, <u>1134</u> , <u>1139</u> , <u>1143</u> , 1145, 1152, <u>1158</u> , 1163, 1169, 1172, 1173 , <u>1175</u> , 1177, 1179, 1182, 1183, 1187, 1192 , * 1194 , 1198 , 1199, 1201, 1203, 1206, 1211, 1222, 1225, 1227, 1229, 1232, 1239, 1247, 1248, 1252, 1254, 1255, 1257, 1258, 1263 , 1264 , 1266, <u>1271</u> , <u>1272</u> , <u>1273</u> , 1274, 1275	155	122	16.74	3.81	71.76	12.86	14.34	48.81	3
3	1	*1167	0	3	0	1.67	0.00	0.14	0	0	0
4	1	*1185	0	2	0	5	0.00	0.28	0	0	0
6	5	1138, 1170, 1188, *1214, 1221	4	22	4.25	3.14	0.47	1.91	40	46.67	1
7	9	1128, 1210, *1212, 1215, 1218, 1240, 1256, 1260, 1262	10	30	0.5	3	0.14	2.49	27.78	15.19	3
9	2	*1178, 1226	1	8	1	1.38	0.03	0.30	100	0	1

#IAL:= number of intra-cluster links

#IEL:= number of inter-cluster links

%IALE:= percentage of intra-cluster linkevents on all linkevents

%IELE:= percentage of inter-cluster linkevents on all linkevents

ØIALS:= average intra-cluster link strength

ØIELS:= average inter-cluster link strength

ØIA-LCC:= average intra-cluster local clustering coefficient

Den:= density

Dia:= diameter

C.4.2.2 Node Role Categorization

Table C-12: Case study. Content-based clustering on nodes. Node role categorization. Clusters 0-2

N.id	C.id	Intra-DC	Intra-BC	Type	DC Type	BC Type	Role
1125	0	26.32	0	both	medium	0	Specialist
1132	0	15.79	3.12	both	low	low	Middleman
1136	0	10.53	0	both	low	0	Peripheral Specialist
1140	0	10.53	0	recipient	low	0	Peripheral Specialist
1141	0	21.05	0	both	medium	0	Specialist
1142	0	42.11	0.87	recipient	high	low	Coordinator
1144	0	42.11	0.87	both	high	low	Coordinator
1153	0	21.05	0	both	medium	0	Specialist
1154	0	21.05	0.19	recipient	medium	low	Team Worker
1157	0	63.16	9.69	recipient	high	low	Coordinator
1164	0	0	0	both	0	0	Isolated
1174	0	47.37	8.71	sender	high	low	Coordinator
1191	0	21.05	0	sender	medium	0	Specialist
1193	0	26.32	0.39	sender	medium	low	Team Worker
1195	0	36.84	0.44	both	medium	low	Team Worker
1200	0	26.32	24.07	both	medium	medium	Integrator
1235	0	10.53	9.94	both	low	low	Middleman
1236	0	5.26	0	both	low	0	Peripheral Specialist
1261	0	78.95	41.12	sender	high	high	Hot Spot
1265	0	31.58	0	both	medium	0	Specialist
1126	1	53.33	3.17	both	high	low	Coordinator
1146	1	26.67	0	both	medium	0	Specialist
1150	1	46.67	1.98	both	high	low	Coordinator
1151	1	26.67	0	both	medium	0	Specialist
1162	1	26.67	0	both	medium	0	Specialist
1165	1	33.33	0.24	both	medium	low	Team Worker
1181	1	33.33	1.11	recipient	medium	low	Team Worker
1184	1	93.33	36.03	both	high	high	Hot Spot
1186	1	80	17.14	recipient	high	medium	Information Spreader
1196	1	20	0	both	medium	0	Specialist
1202	1	0	0	both	0	0	Isolated
1216	1	13.33	0	both	low	0	Peripheral Specialist
1250	1	20	0	both	medium	0	Specialist
1253	1	26.67	0.32	recipient	medium	low	Team Worker
1269	1	26.67	0.95	both	medium	low	Team Worker
1270	1	20	0	both	medium	0	Specialist
1127	2	15.22	0.87	both	low	low	Middleman
1130	2	10.87	0.17	both	low	low	Middleman
1134	2	17.39	0.56	both	low	low	Middleman
1139	2	8.7	0.3	recipient	low	low	Middleman
1143	2	10.87	0.23	both	low	low	Middleman
1145	2	0	0	both	0	0	Isolated
1152	2	17.39	5.16	recipient	low	low	Middleman
1158	2	21.74	8.77	both	medium	low	Team Worker
1163	2	13.04	2.57	both	low	low	Middleman
1169	2	23.91	10.69	sender	medium	medium	Integrator
1172	2	2.17	0	both	low	0	Peripheral Specialist
1173	2	36.96	17.71	both	medium	medium	Integrator
1175	2	17.39	0.88	both	low	low	Middleman
1177	2	13.04	0.27	both	low	low	Middleman
1179	2	21.74	5.62	both	medium	low	Team Worker
1182	2	17.39	1.01	both	low	low	Middleman

N.id:= node id

BC:= betweenness centrality

Intra-BC:= intra-cluster degree centrality

Comp.id:= component id

DC:= degree centrality

Intra-DC:= intra-cluster degree centrality

Thresholds:

DC-low <20 %; DC-medium <40 %

BC-low <10 %; BC-medium <30 %

Table C-13: Case study. Content-based clustering on nodes. Node role categorization. Cluster 2 (ctd.)-9

N.id	C.id	Intra-DC	Intra-BC	Type	DC Type	BC Type	Role
1183	2	4.35	0.2	recipient	low	low	Middleman
1187	2	4.35	0	recipient	low	0	Peripheral Specialist
1192	2	21.74	3.03	recipient	medium	low	Team Worker
1194	2	19.57	3.42	both	low	low	Middleman
1198	2	19.57	3.98	both	low	low	Middleman
1199	2	13.04	0.75	both	low	low	Middleman
1201	2	13.04	0.49	both	low	low	Middleman
1203	2	13.04	0.17	both	low	low	Middleman
1206	2	2.17	0	both	low	0	Peripheral Specialist
1211	2	17.39	0.44	both	low	low	Middleman
1222	2	8.7	0.04	both	low	low	Middleman
1225	2	6.52	0.06	both	low	low	Middleman
1227	2	10.87	1.05	both	low	low	Middleman
1229	2	6.52	0.53	both	low	low	Middleman
1232	2	21.74	3.49	sender	medium	low	Team Worker
1239	2	4.35	0.1	recipient	low	low	Middleman
1247	2	15.22	2.55	both	low	low	Middleman
1248	2	15.22	1.8	both	low	low	Middleman
1252	2	8.7	0.04	recipient	low	low	Middleman
1254	2	13.04	0.48	both	low	low	Middleman
1255	2	8.7	0.04	both	low	low	Middleman
1257	2	10.87	2.91	both	low	low	Middleman
1258	2	8.7	0.04	both	low	low	Middleman
1263	2	32.61	14.59	both	medium	medium	Integrator
1264	2	28.26	6.01	recipient	medium	low	Team Worker
1266	2	15.22	2.51	both	low	low	Middleman
1271	2	13.04	0.2	both	low	low	Middleman
1272	2	6.52	0	both	low	0	Peripheral Specialist
1273	2	17.39	1.79	both	low	low	Middleman
1274	2	26.09	6.93	recipient	medium	low	Team Worker
1275	2	19.57	0.97	both	low	low	Middleman
1167	3	0	0	recipient	0	0	Isolated
1185	4	0	0	recipient	0	0	Isolated
1159	5	50	50	both	high	high	Hot Spot
1160	5	25	0	recipient	medium	0	Specialist
1233	5	25	0	both	medium	0	Specialist
1238	5	25	0	recipient	medium	0	Specialist
1249	5	75	83.33	both	high	high	Hot Spot
1138	6	25	0	both	medium	0	Specialist
1170	6	0	0	recipient	0	0	Isolated
1188	6	50	0	both	high	0	(Specialist)
1214	6	75	33.33	recipient	high	high	Hot Spot
1221	6	50	0	both	high	0	(Specialist)
1128	7	50	28.57	both	high	medium	Information Spreader
1210	7	62.5	41.67	recipient	high	high	Hot Spot
1212	7	37.5	2.38	both	medium	low	Team Worker
1215	7	0	0	no activity	0	0	Isolated
1218	7	25	2.38	both	medium	low	Team Worker
1240	7	12.5	0	no activity	low	0	Peripheral Specialist
1256	7	12.5	0	no activity	low	0	Peripheral Specialist
1260	7	12.5	0	recipient	low	0	Peripheral Specialist
1262	7	37.5	7.14	both	medium	low	Team Worker
1147	8	100	0	both	100	0	(Dyad)
1243	8	100	0	both	100	0	(Dyad)
1178	9	100	0	no activity	100	0	(Dyad)
1226	9	100	0	no activity	100	0	(Dyad)

C.4.2.3 Knowledge Domain Categorization

Table C-14: Case study. Content-based clustering on nodes. Knowledge domain categorization

C.id	#N	#LE	Max(#LE)	Ø(#LE)	SD(#LE)	Max(%LE)	(%LE)	SD(%LE)	Knowledge Domain Categorization		
									Homogeneity	Frequency	Activity
0	20	1472	221	73.60	72.41	15.01%	5.00%	4.92%	het.	freq.	high
1	16	358	81	22.38	23.88	22.63%	6.25%	6.67%	het.	freq.	medium
2	47	6959	933	148.06	215.53	13.41%	2.13%	3.10%	het.	freq.	high
3	1	0	0	0.00	-	-	-	-	-	-	-
4	1	0	0	0.00	-	-	-	-	-	-	-
5	5	36	10	7.20	3.11	27.78%	20.00%	8.65%	het.	infreq.	low
6	5	35	17	7.00	6.67	48.57%	20.00%	19.06%	het.	infreq.	low
7	9	16	4	1.78	1.39	25.00%	11.11%	8.72%	het.	infreq.	low
8	2	4	2	2.00	0.00	50.00%	50.00%	0.00%	hom.	infreq.	low
9	2	2	1	1.00	0.00	50.00%	50.00%	0.00%	hom.	infreq.	low

C.id:= cluster id
 #LE:= number of linkevents
 #N:= number of nodes
 SD:= standard deviation
 freq.:= frequent
 infreq.:= infrequent
 het.:= heterogeneous
 hom.:= homogeneous

C.4.3 Content-based Clustering on Linkevents

C.4.3.1 Cluster Overview

Table C-15: Case study. Content-based clustering on linkevents. Level overview. Part 1. Clusters 0-39

C.id	#LE	#N	%N	Nodes	R.id
0	47	26	24.3	1126 , 1132 , <u>1143</u> , 1154 , <u>1157</u> , <u>1158</u> , 1162 , 1163 , 1165 , 1174 , 1187 , 1191 , 1194 , 1198 , 1206, 1222, 1226, 1239, 1255, 1257, 1258, <u>1261</u> , 1263 , 1264 , <u>1265</u> , 1269	1174
1	32	11	10.28	1125, 1142, 1144, 1154, <u>1157</u> , 1174 , 1191, 1195, <u>1200</u> , <u>1261</u> , <u>1265</u>	1174
2	23	8	7.48	1126 , 1132, 1136, 1150, <u>1184</u> , 1194 , 1198 , <u>1200</u>	<u>1200</u>
3	15	9	8.41	1142, 1150, 1174 , 1186, 1193, 1250, 1253, <u>1261</u> , <u>1265</u>	1174
4	56	11	10.28	<u>1134</u> , 1173 , <u>1175</u> , 1182, 1192 , 1198 , 1262, 1263 , 1266, <u>1271</u> , <u>1273</u>	<u>1175</u>
5	46	11	10.28	<u>1134</u> , 1152, 1173 , <u>1175</u> , 1182, 1192 , <u>1200</u> , 1201, 1235, 1266, <u>1271</u>	<u>1271</u>
6	45	18	16.82	1127, 1130, <u>1134</u> , 1173 , <u>1175</u> , 1191, 1192 , 1194 , 1198 , 1199, 1203, 1232, 1247, 1248, 1252, 1254, <u>1271</u> , 1274	1191
7	11	6	5.61	1127, 1152, 1173 , <u>1175</u> , 1214, 1221	1152
8	41	27	25.23	1132, <u>1134</u> , <u>1143</u> , 1145, 1154, <u>1158</u> , 1162, 1163, 1173 , <u>1175</u> , 1179, 1182, 1192 , 1193, <u>1200</u> , 1215, 1218, 1222, 1232, 1247, 1255, 1257, 1258, <u>1261</u> , 1266, <u>1271</u> , <u>1273</u>	<u>1143</u>
9	46	10	9.35	1130, <u>1134</u> , 1152, 1173 , <u>1175</u> , 1192 , 1198 , 1203, 1232, 1263	<u>1175</u>
10	63	20	18.69	<u>1134</u> , 1150, <u>1158</u> , 1165, 1173 , <u>1175</u> , 1186, 1191, 1192 , 1199, 1211, 1215, 1248, 1250, 1252, 1254, 1262, <u>1271</u> , <u>1273</u> , 1274	1252
11	13	15	14.02	1126 , 1142, <u>1143</u> , 1153, <u>1157</u> , 1163, 1174 , <u>1184</u> , 1194 , 1195, 1198 , 1232, 1247, <u>1261</u> , <u>1265</u>	1174
12	41	22	20.56	1132, <u>1134</u> , 1142, <u>1158</u> , 1169, 1173 , <u>1174</u> , <u>1175</u> , 1182, 1192 , 1194 , 1195, 1198 , <u>1200</u> , 1211, 1225, <u>1261</u> , 1262, <u>1265</u> , 1266, <u>1271</u> , <u>1273</u>	1182
13	50	25	23.36	1126 , <u>1134</u> , <u>1139</u> , 1146, 1151, 1152, 1162, 1163, 1165, 1173 , 1174 , <u>1175</u> , 1177, 1178, 1179, <u>1184</u> , 1186, 1192 , 1226, 1227, 1232, 1252, 1254, 1269, <u>1271</u>	1126
14	21	17	15.89	1126 , <u>1134</u> , 1152, 1173 , <u>1175</u> , <u>1184</u> , 1192 , 1194 , 1198 , 1201, 1232, 1247, 1254, 1263 , 1264 , 1266, <u>1271</u>	<u>1175</u>
15	23	27	25.23	1126 , 1146, 1150, 1151, 1162, 1165, 1169, 1173 , 1174 , 1179, 1181, <u>1184</u> , 1186, 1192 , 1196, 1215, 1216, 1225, 1227, 1229, 1232, 1236, 1250, 1253, 1269, 1270, 1274	1174
16	23	10	9.35	<u>1134</u> , 1173 , <u>1175</u> , 1179, 1182, 1192 , 1216, 1225, 1232, <u>1271</u>	1173
17	25	22	20.56	1130, 1138, <u>1139</u> , 1147, 1150, 1152, 1169, 1173 , 1174 , <u>1184</u> , 1186, 1211, 1214, 1216, 1229, 1240, 1243, 1250, 1262, 1263 , 1266, <u>1273</u>	1174
18	19	16	14.95	1126 , <u>1134</u> , 1165, 1172, 1173 , 1174 , <u>1175</u> , 1192 , 1201, 1206, 1221 , 1227, 1232, 1235, 1239, 1266	<u>1175</u>
19	136	10	9.35	1130, <u>1134</u> , <u>1158</u> , 1173 , <u>1175</u> , 1177, 1182, 1192 , 1201, <u>1271</u>	1192
20	10	10	9.35	1132, <u>1134</u> , 1142, <u>1175</u> , 1192 , 1195, <u>1200</u> , 1206, 1239, <u>1265</u>	<u>1134</u>
21	3	3	2.8	1186, 1215, 1250	1186
22	25	26	24.3	<u>1134</u> , 1141, 1142, 1144, 1145, 1152, 1169, 1173 , 1174 , <u>1175</u> , 1177, 1179, 1187, 1192 , 1193, 1195, 1203, 1221 , 1225, 1232, 1247, 1248, <u>1261</u> , <u>1265</u> , 1269, <u>1271</u>	<u>1175</u>
23	15	14	13.08	1132, <u>1134</u> , 1136, 1144, <u>1158</u> , 1173 , 1174 , 1182, 1192 , 1195, <u>1200</u> , 1201, 1235, 1266	1195
24	51	30	28.04	1125, 1126 , 1130, 1142, <u>1143</u> , 1144, 1146, 1154, 1165, 1173 , 1174 , <u>1184</u> , 1186, 1194 , 1195, 1198 , <u>1200</u> , 1201, 1206, 1211, 1221 , 1235, 1239, 1248, <u>1261</u> , 1262, 1263 , 1264 , <u>1265</u> , 1266	1263
25	10	12	11.21	1132, <u>1139</u> , 1173 , 1179, 1186, 1194 , 1198 , <u>1200</u> , 1203, 1215, 1232, 1247	1194
26	52	30	28.04	1125, 1126 , <u>1139</u> , 1140, 1142, <u>1143</u> , 1145, 1150, <u>1157</u> , <u>1158</u> , 1173 , 1174 , <u>1175</u> , 1182, 1186, 1191, 1192 , 1193, 1194 , 1198 , 1201, 1221 , 1232, 1236, 1247, <u>1261</u> , 1263 , <u>1265</u> , 1266, 1272	1126
27	16	12	11.21	1126 , <u>1134</u> , <u>1157</u> , 1162, 1165, 1173 , 1174 , <u>1175</u> , 1182, 1186, 1192 , <u>1271</u>	<u>1271</u>
28	27	10	9.35	<u>1134</u> , <u>1139</u> , 1173 , <u>1175</u> , 1183, 1186, 1192 , 1250, 1266, <u>1271</u>	<u>1175</u>
29	7	10	9.35	1130, <u>1134</u> , 1138, <u>1139</u> , 1173 , <u>1175</u> , 1182, 1192 , 1214, <u>1271</u>	1182
30	46	21	19.63	1126 , 1128, 1142, <u>1143</u> , 1144, 1146, 1173 , 1174 , 1192 , 1193, 1194 , 1198 , 1211, 1248, 1260, <u>1261</u> , 1263 , 1264 , <u>1265</u> , 1272, <u>1273</u>	1264
31	22	13	12.15	1132, <u>1134</u> , <u>1158</u> , 1163, 1173 , <u>1175</u> , 1182, 1192 , <u>1200</u> , 1201, 1250, 1266, <u>1271</u>	1192
32	86	10	9.35	1130, <u>1134</u> , 1141, 1152, 1173 , <u>1175</u> , 1177, 1192 , 1193, <u>1271</u>	<u>1271</u>
33	13	10	9.35	1132, 1142, 1144, 1153, 1195, <u>1200</u> , 1206, 1239, <u>1265</u> , 1266	1195
34	31	24	22.43	1126 , 1142, <u>1143</u> , 1144, 1146, 1151, 1154, <u>1157</u> , 1162, 1165, 1174 , <u>1184</u> , 1186, 1188, 1191, 1194 , 1195, 1198 , 1206, 1235, 1239, <u>1261</u> , 1263 , <u>1265</u>	<u>1265</u>
35	28	12	11.21	<u>1134</u> , 1142, <u>1143</u> , <u>1157</u> , 1173 , 1174 , 1192 , 1194 , 1198 , 1206, 1239, <u>1261</u>	1198
36	7	9	8.41	1126 , 1154, 1174 , 1191, 1193, 1198 , 1227, <u>1261</u> , <u>1265</u>	<u>1261</u>
37	22	13	12.15	1130, 1132, <u>1134</u> , <u>1139</u> , 1163, 1173 , 1174 , <u>1175</u> , 1192 , <u>1200</u> , 1232, <u>1261</u> , <u>1271</u>	1173
38	34	27	25.23	1126 , 1141, 1142, <u>1143</u> , 1144, 1151, 1153, 1173 , 1174 , <u>1175</u> , 1179, 1186, 1192 , 1193, 1194 , 1195, 1198 , 1210, 1232, 1247, 1248, 1250, 1253, 1254, <u>1261</u> , 1262, <u>1265</u>	1179
39	35	4	3.74	<u>1143</u> , 1194 , 1198 , 1263	1198

#LE:= number of linkevents %N:= percentage of all nodes in network
 #N:= number of nodes R.id:= node representative of cluster
 Structural key players marked bold (top 9) or underlined (top 21)

Table C-16: Case study. Content-based clustering on linkevents. Level overview. Part 1. Clusters 40-69

C.id	#LE	#N	%N	Nodes	R.id
40	31	27	25.23	1126 , 1141, 1163, 1169, 1174 , 1179, 1181, 1186, 1191, 1193, 1194 , 1198 , 1203, 1211, 1216, 1221 , 1225, 1227, 1232, 1236, 1247, 1250, 1253, 1263 , 1269, <u>1273</u> , 1274	1247
41	31	30	28.04	1126 , 1132, <u>1134</u> , 1142, <u>1143</u> , 1146, 1151, 1154, <u>1157</u> , <u>1158</u> , 1163, 1173 , 1174 , 1175, 1188, 1191, 1192 , 1194 , 1195, 1198 , <u>1200</u> , 1201, 1206, 1239, <u>1261</u> , 1263 , <u>1265</u> , 1266, <u>1271</u> , 1275	1191
42	8	8	7.48	1126 , 1132, 1169, 1179, 1232, 1247, 1253, 1274	1274
43	9	6	5.61	<u>1157</u> , 1194 , 1198 , <u>1261</u> , 1263 , <u>1265</u>	<u>1157</u>
44	31	25	23.36	<u>1134</u> , <u>1139</u> , 1160, 1164, 1169, 1173 , 1174 , <u>1175</u> , 1179, 1186, 1192 , 1202, 1203, 1211, 1216, 1225, 1232, 1240, 1249, 1250, 1253, 1255, 1257, 1258, <u>1271</u>	1173
45	48	12	11.21	1130, 1132, <u>1134</u> , 1136, <u>1139</u> , 1173 , <u>1175</u> , 1182, 1192 , <u>1200</u> , 1201, <u>1271</u>	<u>1175</u>
46	38	19	17.76	1154, 1172, 1177, 1179, 1187, 1188, 1191, 1193, 1199, 1206, 1214, 1232, 1239, 1248, 1252, 1254, <u>1271</u> , 1274, 1275	1191
47	45	17	15.89	<u>1134</u> , <u>1139</u> , 1142, 1144, <u>1158</u> , 1173 , <u>1175</u> , 1177, 1182, 1192 , 1195, 1201, 1214, 1239, <u>1265</u> , 1266, <u>1271</u>	<u>1265</u>
48	22	15	14.02	<u>1134</u> , <u>1143</u> , <u>1158</u> , 1163, 1182, 1192 , 1201, 1211, 1218, 1222, 1255, 1257, 1258, 1266, <u>1273</u>	1258
49	46	39	36.45	1128, <u>1134</u> , 1142, <u>1143</u> , <u>1158</u> , 1159, 1160, 1169, 1173 , <u>1175</u> , 1192 , 1193, 1194 , 1195, 1198 , 1201, 1206, 1210, 1211, 1214, 1216, 1218, 1221 , 1222, 1225, 1233, 1235, 1236, 1239, 1249, 1253, 1255, 1257, 1258, <u>1261</u> , <u>1265</u> , 1266, <u>1271</u> , <u>1273</u>	1216
50	25	24	22.43	1126 , <u>1134</u> , <u>1143</u> , 1163, 1169, 1173 , 1174 , <u>1175</u> , 1182, <u>1184</u> , 1187, 1192 , 1218, 1221 , 1222, 1227, 1232, 1247, 1255, 1257, 1258, 1266, <u>1271</u> , <u>1273</u>	1221
51	14	12	11.21	1127, <u>1139</u> , 1142, <u>1143</u> , 1173 , 1174 , <u>1175</u> , 1194 , 1195, 1232, 1263 , 1266	1194
52	22	18	16.82	1130, <u>1134</u> , 1142, 1165, 1167, 1169, 1173 , 1174 , <u>1175</u> , 1181, 1186, 1195, 1225, 1250, 1262, <u>1265</u> , <u>1273</u> , 1274	1250
53	14	15	14.02	1126 , <u>1139</u> , 1146, 1150, 1151, 1165, 1169, 1173 , <u>1184</u> , 1186, 1199, 1214, 1221 , 1229, 1248	1150
54	23	13	12.15	<u>1134</u> , 1173 , <u>1175</u> , 1179, 1192 , 1194 , 1203, 1232, 1247, 1248, 1252, 1254, <u>1271</u>	1232
55	10	18	16.82	1126, 1130, <u>1143</u> , 1146, 1173 , 1174 , 1182, 1194 , 1198 , 1201, 1206, 1239, 1248, 1256, 1263 , 1264 , 1266, <u>1273</u>	1198
56	32	18	16.82	1132, 1136, <u>1143</u> , <u>1158</u> , 1173 , <u>1175</u> , 1185, 1192 , 1194 , 1198 , <u>1200</u> , 1201, 1211, 1232, 1240, 1247, 1263 , 1266	<u>1175</u>
57	24	24	22.43	1126 , 1128, <u>1139</u> , <u>1158</u> , 1162, 1173 , 1174 , 1179, 1185, 1192 , <u>1200</u> , 1203, 1210, 1212, 1215, 1227, 1232, 1235, 1256, 1260, <u>1261</u> , 1262, 1266, 1274	1262
58	33	22	20.56	1125, <u>1134</u> , <u>1139</u> , <u>1143</u> , <u>1157</u> , 1163, 1173 , <u>1175</u> , 1191, 1192 , 1194 , 1198 , <u>1200</u> , 1201, 1215, <u>1261</u> , 1263 , 1264 , <u>1265</u> , 1266, <u>1271</u> , 1272	<u>1175</u>
59	22	23	21.5	1128, <u>1143</u> , <u>1157</u> , <u>1158</u> , 1194 , 1198 , 1210, 1211, 1212, 1218, 1222, 1240, 1255, 1256, 1257, 1258, 1260, <u>1261</u> , 1262, 1263 , 1264 , 1272, <u>1273</u>	1218
60	73	13	12.15	1142, <u>1143</u> , <u>1157</u> , 1174 , 1186, 1194 , 1195, 1198 , 1250, <u>1261</u> , 1263 , 1264 , <u>1265</u>	1174
61	26	22	20.56	<u>1134</u> , 1141, 1142, <u>1143</u> , 1144, 1152, 1173 , 1174 , <u>1175</u> , 1178, 1191, 1192 , 1193, 1194 , 1195, 1198 , 1202, 1206, 1239, 1263 , <u>1265</u> , <u>1271</u>	1195
62	51	20	18.69	1126 , 1142, <u>1143</u> , 1144, <u>1157</u> , 1170, 1174 , 1179, 1186, 1194 , 1195, 1198 , 1199, 1203, 1232, 1247, 1248, <u>1261</u> , 1263 , <u>1265</u>	1195
63	38	42	39.25	1126 , 1140, 1141, 1142, <u>1143</u> , 1144, 1150, 1153, 1154, <u>1157</u> , 1170, 1174 , 1179, <u>1184</u> , 1186, 1191, 1193, 1194 , 1195, 1201, 1203, 1206, 1210, 1211, 1214, 1218, 1221 , 1222, 1232, 1235, 1236, 1239, 1247, 1248, 1255, 1257, 1258, <u>1261</u> , 1263 , <u>1265</u> , 1266, <u>1273</u>	1255
64	21	4	3.74	1194 , 1198 , 1263 , 1264	1198
65	17	17	15.89	<u>1134</u> , 1141, 1152, <u>1157</u> , 1174 , 1179, 1182, 1191, 1192 , 1193, <u>1200</u> , 1201, 1235, <u>1261</u> , <u>1265</u> , 1266, <u>1271</u>	<u>1265</u>
66	113	16	14.95	1126 , <u>1143</u> , 1146, <u>1157</u> , 1163, 1169, 1178, 1179, 1194 , 1198 , 1248, 1260, <u>1261</u> , 1263 , <u>1265</u> , 1275	1198
67	75	15	14.02	<u>1143</u> , 1154, 1179, 1191, 1193, 1194 , 1198 , 1199, 1232, 1248, 1252, 1254, 1263 , 1274, 1275	1232
68	5	5	4.67	1142, 1144, 1174 , 1195, 1198	1142
69	49	37	34.58	1128, 1132, 1142, <u>1143</u> , 1144, 1152, <u>1157</u> , <u>1158</u> , 1174 , 1188, 1194 , 1198 , <u>1200</u> , 1202, 1210, 1211, 1212, 1215, 1218, 1222, 1232, 1235, 1239, 1240, 1247, 1255, 1256, 1257, 1258, 1260, <u>1261</u> , 1262, 1263 , 1264 , <u>1265</u> , 1272, <u>1273</u>	1240

Table C-17: Case study. Content-based clustering on linkevents. Level overview. Part 1. Cluster 70-86

C.id	#LE	#N	%N	Nodes	R.id
70	49	23	21.5	1125, <u>1134</u> , 1140, <u>1143</u> , 1152, 1154, <u>1157</u> , 1173 , 1174 , 1188, 1191, 1192 , 1193, 1194 , 1198 , 1206, 1214, 1221 , 1239, <u>1261</u> , 1263 , 1264 , <u>1265</u>	<u>1157</u>
71	8	13	12.15	1126 , <u>1143</u> , 1169, 1172, 1174 , 1179, 1183, <u>1184</u> , 1187, 1194 , 1198 , 1221 , 1236	1169
72	10	5	4.67	<u>1139</u> , 1194 , 1198 , 1263 , 1266	1198
73	46	39	36.45	1126 , 1130, <u>1134</u> , <u>1139</u> , 1142, <u>1143</u> , 1146, 1150, 1151, 1154, <u>1157</u> , 1162, 1165, 1169, 1173 , 1174, <u>1175</u> , 1179, <u>1184</u> , 1186, 1192 , 1195, 1201, 1203, 1211, 1216, 1218, 1222, 1247, 1250, 1255, 1257, 1258, <u>1261</u> , <u>1265</u> , 1266, 1270, <u>1271</u> , <u>1273</u>	<u>1184</u>
74	77	28	26.17	<u>1134</u> , <u>1143</u> , 1152, 1154, <u>1157</u> , <u>1158</u> , 1174 , <u>1175</u> , 1183, 1186, 1191, 1192 , 1194 , 1198 , 1211, 1215, 1218, 1222, 1250, 1255, 1257, 1258, 1260, <u>1261</u> , 1263 , 1264 , <u>1265</u> , 1272	1257
75	10	12	11.21	1126 , 1138, 1146, 1150, 1165, 1169, 1181, 1183, 1186, 1214, 1229, 1250	1181
76	26	21	19.63	1126 , 1127, <u>1134</u> , <u>1139</u> , 1162, 1173 , 1174 , <u>1175</u> , 1177, 1179, <u>1184</u> , 1192 , 1194 , 1198 , 1221 , 1239, <u>1261</u> , 1263 , 1266, 1269, 1275	1177
77	9	2	1.87	1194 , 1198	1198
78	45	20	18.69	1126 , 1127, <u>1143</u> , 1146, 1152, <u>1158</u> , 1165, 1173 , 1179, 1192 , 1194 , 1198 , <u>1200</u> , 1232, 1248, 1263 , 1264 , 1266, 1272, 1275	<u>1143</u>
79	37	15	14.02	1126 , 1132, <u>1134</u> , 1152, <u>1158</u> , 1163, 1165, 1173 , 1175, 1182, 1192 , <u>1200</u> , 1201, 1266, <u>1271</u>	1201
80	22	23	21.5	1132, 1142, <u>1143</u> , 1144, 1145, 1152, <u>1157</u> , 1163, 1172, 1174 , 1177, 1188, 1194 , 1195, 1198 , <u>1200</u> , 1206, 1221 , 1235, 1239, <u>1261</u> , 1263 , <u>1265</u>	1195
81	35	20	18.69	1125, 1126 , 1132, 1142, 1144, 1153, 1162, 1163, 1165, 1173 , 1174 , <u>1175</u> , <u>1184</u> , 1187, 1192 , 1195, <u>1200</u> , 1221 , 1253, <u>1265</u>	1195
82	31	9	8.41	<u>1143</u> , <u>1157</u> , 1174 , 1194 , 1198 , <u>1261</u> , 1263 , 1264 , <u>1265</u>	<u>1265</u>
83	21	31	28.97	1126 , 1128, 1132, 1138, <u>1143</u> , 1146, <u>1157</u> , 1162, 1163, 1174 , 1175, 1192 , 1194 , 1198 , <u>1200</u> , 1206, 1211, 1214, 1222, 1225, 1239, 1240, 1248, 1255, 1257, 1258, 1263 , 1264 , <u>1265</u> , <u>1271</u> , <u>1273</u>	1143
84	33	24	22.43	1125, <u>1134</u> , 1140, 1142, 1144, 1152, 1154, <u>1157</u> , 1173 , 1174 , <u>1175</u> , 1188, 1192 , 1193, 1194 , 1195, 1198 , <u>1200</u> , 1239, <u>1261</u> , 1263 , <u>1265</u> , 1266, <u>1271</u>	1157
85	124	16	14.95	1132, <u>1134</u> , <u>1139</u> , <u>1158</u> , 1163, 1174 , 1182, 1192 , 1198 , <u>1200</u> , 1201, 1221 , <u>1261</u> , 1262, 1266, <u>1273</u>	1201
86	25	11	10.28	1132, <u>1134</u> , 1145, 1163, 1173 , <u>1175</u> , 1192 , <u>1200</u> , 1201, <u>1261</u> , 1266	1192

Table C-18: Case study. Content-based clustering on linkevents. Level overview. Part 2. Cluster 0-59

C.id	#Comp	Max Comp Diameter	Density	#IAL	#IEL	ØIALS	ØIELS	ØIA-LCC	#IALE	#IELE	#IELEs	#IELEr
0	5	1	8	26	161	1.81	5.59	16.04	47	900	604	707
1	1	2	27.27	15	63	2.13	6.49	30.41	32	409	229	303
2	3	1	21.43	6	64	3.83	9.95	37.5	23	637	407	232
3	1	2	22.22	8	69	1.88	5.42	0	15	374	174	249
4	3	1	21.82	12	88	4.67	15.08	36.97	56	1327	1035	377
5	3	1	20	11	62	4.18	9.87	34.55	46	612	459	334
6	2	4	15.03	23	90	1.96	11.02	24.67	45	992	723	498
7	3	1	20	3	57	3.67	11.95	0	11	681	363	330
8	8	1	6.84	24	146	1.71	7.61	15.43	41	1111	496	798
9	2	1	24.44	11	79	4.18	19.04	31.67	46	1504	1211	720
10	4	2	8.95	17	124	3.71	6.06	7.5	63	751	491	511
11	4	2	10.48	11	124	1.18	6.86	0	13	851	492	603
12	8	1	8.23	19	154	2.16	11.68	22.73	41	1798	1364	1146
13	5	2	8	24	145	2.08	5.63	14.67	50	817	407	611
14	5	1	11.03	15	159	1.4	6.42	22.35	21	1021	781	476
15	2	3	10.26	36	126	0.64	8.86	29.41	23	1116	633	615
16	3	2	20	9	68	2.56	9.43	28.67	23	641	455	514
17	9	1	5.63	13	150	1.92	9.81	0	25	1472	739	846
18	6	2	9.17	11	124	1.73	10.91	14.58	19	1353	927	873
19	1	2	33.33	15	43	9.07	12.19	50.71	136	524	401	333
20	4	1	13.33	6	57	1.67	14.33	0	10	817	335	539
21	1	2	66.67	2	20	1.5	2.3	0	3	46	18	28
22	5	4	7.69	25	152	1	7.53	17.31	25	1145	923	909
23	4	2	13.19	12	90	1.25	12.12	32.14	15	1091	750	637
24	6	2	7.36	32	212	1.59	9.2	29.11	51	1950	1439	1304
25	5	1	10.61	7	96	1.43	13.98	0	10	1342	732	647
26	5	3	7.82	34	194	1.53	8.65	20.81	52	1679	1195	1225
27	3	1	22.73	15	98	1.07	7.77	41.94	16	761	445	478
28	3	2	17.78	8	61	3.38	11.25	21.67	27	686	508	460
29	2	1	17.78	8	45	0.88	15.87	0	7	714	612	629
30	4	2	9.05	19	170	2.42	7.56	12.96	46	1285	872	644
31	2	5	15.38	12	66	1.83	10.92	16.67	22	721	607	523
32	2	2	26.67	12	44	7.17	10.36	36.67	86	456	370	358
33	3	1	17.78	8	51	1.63	9.18	16.67	13	468	198	303
34	2	5	10.87	30	144	1.03	4.7	17.92	31	677	499	571
35	4	1	12.12	8	107	3.5	11.72	0	28	1254	819	709
36	2	3	22.22	8	79	0.88	14.06	23.33	7	1111	779	400
37	3	2	14.1	11	92	2	8.15	9.74	22	750	473	477
38	7	1	6.84	24	176	1.42	8.05	19.51	34	1416	864	828
39	1	2	50	3	40	11.67	6.13	0	35	245	165	174
40	5	4	7.98	28	183	1.11	5.28	7.16	31	967	639	515
41	8	2	5.06	22	193	1.41	7.3	0	31	1408	1171	1071
42	2	3	21.43	6	69	1.33	5.13	0	8	354	146	241
43	2	1	26.67	4	68	2.25	9.06	0	9	616	468	258
44	4	3	7.67	23	128	1.35	8.53	4.47	31	1092	678	847
45	3	1	30.3	20	41	2.4	8.73	50.83	48	358	265	198
46	3	2	9.36	16	85	2.38	6.31	0	38	536	196	401
47	4	1	12.5	17	86	2.65	9.37	20.98	45	806	563	613
48	3	1	13.33	14	72	1.57	14.14	28.89	22	1018	421	674
49	9	2	4.45	33	190	1.39	8.26	13.82	46	1570	1121	972
50	5	1	7.25	20	163	1.25	10.09	8.61	25	1645	976	1285
51	5	2	10.61	7	111	2	17.51	0	14	1944	713	1324
52	6	2	7.84	12	125	1.83	10.14	0	22	1267	651	803
53	5	1	9.52	10	109	1.4	7.97	0	14	869	432	481
54	3	3	16.67	13	78	1.77	14.76	23.08	23	1151	424	973
55	5	2	8.5	13	156	0.77	12.14	0	10	1894	1370	1254
56	6	2	8.5	13	125	2.46	10.16	12.96	32	1270	857	751
57	7	3	6.16	17	159	1.41	9.13	0	24	1451	797	750
58	4	1	9.52	22	154	1.5	8.09	19.39	33	1246	934	901
59	2	4	11.86	30	95	0.73	5.19	22.84	22	493	384	339

#Comp:= number of comp.
 ØIA-LCC:= average intra-cluster local clustering coefficient

#IAL:= number of intra-cluster links
 #IEL:= number of inter-cluster links
 ØIALS:= average intra-cluster link strength
 ØIELS:= average inter-cluster link strength

#IALE:= number of intra-cluster linkevents
 #IELE:= number of inter-cluster linkevents
 #IELE:= number of inter-cluster LE received
 #IELE:= number of inter-cluster LE sent

Table C-19: Case study. Content-based clustering on linkevents. Level overview. Part 2. Clusters 60-86

C.id	#Comp	Max Comp Diameter	Density	#IAL	#IEL	ØIALS	ØIELS	ØIA-LCC	#IALE	#IELE	#IELEs	#IELEr
60	3	1	14.1	11	121	6.64	4	11.54	73	484	395	379
61	6	2	6.93	16	141	1.63	7.09	0	26	999	669	742
62	3	3	13.68	26	136	1.96	4.48	37.93	51	609	380	436
63	7	3	4.88	42	222	0.9	6.9	19.06	38	1531	692	1173
64	1	2	50	3	58	7	6.45	0	21	374	343	133
65	5	2	8.82	12	117	1.42	10.42	0	17	1219	691	771
66	4	2	12.5	15	128	7.53	4.85	21.29	113	621	370	381
67	1	3	13.33	14	87	5.36	4.62	0	75	402	227	307
68	2	2	30	3	44	1.67	23.66	0	5	1041	717	404
69	6	2	5.71	38	165	1.29	4.71	16.11	49	777	480	599
70	5	2	8.3	21	166	2.33	7.54	20.45	49	1251	954	874
71	3	1	14.1	11	105	0.73	7.43	10.77	8	780	354	470
72	2	1	30	3	51	3.33	12.57	0	10	641	433	274
73	6	2	5.13	38	210	1.21	8.32	14.29	46	1747	1244	1452
74	2	6	7.94	30	159	2.57	7.36	15.83	77	1170	614	831
75	3	1	13.64	9	64	1.11	2.38	0	10	152	75	97
76	7	2	6.67	14	188	1.86	9.19	0	26	1728	1333	941
77	1	1	100	1	20	9	25.95	0	9	519	352	167
78	3	2	13.68	26	150	1.73	8.27	34.67	45	1240	757	626
79	2	6	15.24	16	84	2.31	7.33	32.22	37	616	461	419
80	4	6	9.09	23	129	0.96	5.89	17.39	22	760	578	621
81	5	2	11.58	22	133	1.59	9.53	22	35	1268	787	881
82	2	2	27.78	10	98	3.1	5.46	59.26	31	535	435	399
83	6	2	5.59	26	179	0.81	10.88	7.53	21	1947	1316	1480
84	5	3	9.78	27	141	1.22	11.25	23.47	33	1586	1231	1003
85	4	2	12.5	15	112	8.27	16.45	29.84	124	1842	1209	801
86	3	2	18.18	10	78	2.5	10.85	24.24	25	846	606	423

Table C-20: Case study. Content-based clustering on linkevents. Level overview. Part 3. Clusters 0-59

C.id	First IALE Date	Last IALE Date	%Intra-Timespan	First IELE Date	Last IELE Date	%Inter-Timespan
0	08.09.2000 06:03	09.03.2001 08:59	86.44	01.09.2000 01:07	30.03.2001 07:09	100
1	12.10.2000 04:55	09.03.2001 08:39	70.32	01.09.2000 07:24	30.03.2001 09:11	99.68
2	02.10.2000 02:34	26.03.2001 03:03	83.28	01.09.2000 06:56	30.03.2001 09:11	99.69
3	24.10.2000 03:33	20.02.2001 01:29	56.44	01.09.2000 07:24	30.03.2001 09:11	99.68
4	05.09.2000 02:51	27.02.2001 01:26	83.03	01.09.2000 06:30	30.03.2001 10:47	99.73
5	06.09.2000 01:28	29.03.2001 02:31	96.82	01.09.2000 10:59	30.03.2001 07:09	99.8
6	10.10.2000 10:14	26.03.2001 07:42	79.19	01.09.2000 06:56	30.03.2001 09:11	99.69
7	18.09.2000 05:05	23.02.2001 08:18	75.05	05.09.2000 02:51	30.03.2001 08:08	97.85
8	18.09.2000 04:17	29.03.2001 11:59	91.25	01.09.2000 07:24	30.03.2001 07:09	99.88
9	06.09.2000 09:58	29.03.2001 01:13	96.62	01.09.2000 06:30	30.03.2001 06:34	99.88
10	13.09.2000 01:51	14.03.2001 10:40	86.55	05.09.2000 11:52	30.03.2001 07:09	97.65
11	13.09.2000 09:08	28.03.2001 01:52	93.09	01.09.2000 06:56	30.03.2001 09:11	99.69
12	27.09.2000 05:34	30.03.2001 10:43	87.41	01.09.2000 06:56	30.03.2001 07:09	99.89
13	06.09.2000 07:51	28.03.2001 06:48	96.3	01.09.2000 07:24	30.03.2001 06:34	99.86
14	13.09.2000 11:48	09.03.2001 10:14	83.97	05.09.2000 12:49	30.03.2001 07:09	98.11
15	05.09.2000 11:52	20.03.2001 10:20	92.75	01.09.2000 07:24	30.03.2001 06:34	99.86
16	07.09.2000 04:36	30.03.2001 02:42	96.76	06.09.2000 09:05	30.03.2001 06:34	97.46
17	11.09.2000 08:11	30.03.2001 09:04	94.92	01.09.2000 06:56	30.03.2001 09:11	99.69
18	21.09.2000 01:19	27.03.2001 01:23	88.73	01.09.2000 01:07	30.03.2001 07:09	100
19	20.09.2000 08:04	30.03.2001 08:04	90.63	06.09.2000 07:51	30.03.2001 07:09	97.49
20	08.11.2000 08:01	06.03.2001 01:05	55.85	01.09.2000 01:07	30.03.2001 09:11	99.8
21	07.11.2000 04:40	06.12.2000 07:31	13.82	05.09.2000 11:52	21.03.2001 01:47	93.06
22	02.10.2000 10:39	30.03.2001 04:38	85.05	01.09.2000 07:24	30.03.2001 09:11	99.68
23	11.09.2000 12:17	26.03.2001 03:30	93.06	01.09.2000 07:24	30.03.2001 07:09	99.88
24	11.09.2000 07:01	19.03.2001 01:38	89.83	01.09.2000 01:07	30.03.2001 07:09	100
25	13.09.2000 05:58	09.03.2001 08:20	84.29	01.09.2000 06:56	30.03.2001 09:11	99.69
26	01.09.2000 06:30	30.03.2001 09:11	99.7	01.09.2000 07:24	30.03.2001 07:09	99.88
27	05.09.2000 02:02	09.03.2001 07:34	87.91	01.09.2000 07:24	30.03.2001 09:11	99.68
28	17.10.2000 08:50	30.03.2001 12:25	77.65	06.09.2000 07:51	30.03.2001 08:08	97.28
29	04.10.2000 08:14	14.02.2001 07:26	63.11	05.09.2000 02:51	30.03.2001 08:08	97.85
30	06.09.2000 01:58	14.03.2001 08:58	89.84	01.09.2000 07:24	30.03.2001 09:11	99.68
31	11.09.2000 09:35	22.03.2001 08:36	91.1	05.09.2000 01:15	30.03.2001 07:09	98.1
32	06.09.2000 09:05	26.03.2001 04:48	95.29	05.09.2000 02:02	30.03.2001 09:11	97.89
33	03.11.2000 06:09	22.02.2001 01:12	52.57	01.09.2000 01:07	30.03.2001 09:11	99.8
34	05.09.2000 05:16	20.03.2001 11:39	93.38	01.09.2000 07:24	30.03.2001 09:11	99.68
35	08.09.2000 06:09	23.03.2001 03:58	92.98	01.09.2000 01:07	30.03.2001 09:11	99.8
36	13.12.2000 08:54	05.03.2001 03:35	38.8	01.09.2000 06:30	30.03.2001 10:47	99.73
37	19.09.2000 01:59	14.03.2001 06:59	83.63	01.09.2000 07:24	30.03.2001 06:34	99.86
38	06.10.2000 10:47	27.02.2001 11:25	68.6	01.09.2000 06:56	30.03.2001 09:11	99.69
39	05.09.2000 11:10	23.03.2001 03:07	94.28	05.09.2000 12:49	30.03.2001 09:11	97.91
40	16.10.2000 04:40	29.03.2001 07:58	77.88	01.09.2000 07:24	30.03.2001 09:11	99.68
41	05.09.2000 02:19	29.03.2001 11:57	97.46	01.09.2000 07:24	30.03.2001 07:09	99.88
42	22.01.2001 05:11	06.03.2001 04:05	20.38	01.09.2000 10:59	30.03.2001 06:34	99.79
43	07.11.2000 10:21	25.03.2001 02:32	65.54	01.09.2000 07:24	30.03.2001 09:11	99.68
44	05.09.2000 09:03	30.03.2001 08:08	97.73	01.09.2000 07:24	30.03.2001 06:34	99.86
45	12.09.2000 04:13	28.03.2001 05:02	93.49	06.09.2000 09:05	30.03.2001 07:09	97.47
46	22.11.2000 04:01	14.12.2000 07:39	10.51	01.09.2000 01:07	30.03.2001 06:34	99.99
47	22.09.2000 01:16	23.03.2001 10:12	86.55	01.09.2000 01:07	30.03.2001 07:09	100
48	28.11.2000 08:32	29.03.2001 12:16	57.23	05.09.2000 12:49	30.03.2001 07:09	98.11
49	06.10.2000 08:11	30.03.2001 05:50	82.99	01.09.2000 01:07	30.03.2001 07:09	100
50	03.11.2000 06:15	09.03.2001 06:01	59.78	01.09.2000 07:24	30.03.2001 09:11	99.68
51	17.10.2000 03:03	22.03.2001 06:34	74.35	01.09.2000 06:30	30.03.2001 06:34	99.88
52	05.09.2000 01:15	21.02.2001 07:48	80.34	01.09.2000 07:24	30.03.2001 09:11	99.68
53	25.10.2000 12:34	19.03.2001 11:20	69.27	05.09.2000 01:15	30.03.2001 08:08	97.88
54	01.11.2000 12:51	20.02.2001 01:06	52.67	01.09.2000 06:30	30.03.2001 06:34	99.88
55	19.09.2000 10:34	27.03.2001 12:49	89.49	01.09.2000 01:07	30.03.2001 07:09	100
56	13.10.2000 08:39	30.03.2001 06:34	79.91	05.09.2000 12:49	30.03.2001 07:09	98.11
57	12.09.2000 01:25	29.03.2001 06:43	94.29	01.09.2000 07:24	30.03.2001 07:09	99.88
58	11.10.2000 06:17	29.03.2001 03:38	80.14	01.09.2000 07:24	30.03.2001 07:09	99.88
59	13.09.2000 03:36	26.03.2001 09:09	91.92	01.09.2000 07:24	30.03.2001 07:09	99.88

IALE:= intra-cluster linkevents

IELE:=inter-cluster linkevents

Table C-21: Case study. Content-based clustering on linkevents. Level overview. Part 3. Clusters 60-86

C.id	First IALE Date	Last IALE Date	%Intra-Timespan	First IELE Date	Last IELE Date	%Inter-Timespan
60	05.09.2000 09:44	30.03.2001 03:23	97.62	01.09.2000 07:24	30.03.2001 09:11	99.68
61	21.09.2000 10:04	21.03.2001 02:43	85.76	01.09.2000 01:07	30.03.2001 09:11	99.8
62	05.09.2000 06:17	23.03.2001 10:49	94.53	01.09.2000 07:24	30.03.2001 09:11	99.68
63	26.10.2000 09:35	29.03.2001 12:58	72.9	01.09.2000 01:07	30.03.2001 07:09	100
64	08.09.2000 04:43	02.02.2001 12:15	69.68	05.09.2000 12:49	30.03.2001 09:11	97.91
65	27.09.2000 09:30	22.03.2001 05:55	83.46	01.09.2000 07:24	30.03.2001 07:09	99.88
66	01.09.2000 06:56	30.03.2001 10:42	99.72	01.09.2000 07:24	30.03.2001 09:11	99.68
67	10.09.2000 11:23	30.03.2001 10:47	95.12	01.09.2000 07:24	30.03.2001 06:34	99.86
68	26.09.2000 03:01	04.12.2000 05:36	32.81	01.09.2000 06:30	30.03.2001 10:47	99.73
69	01.09.2000 01:07	14.03.2001 06:19	92.17	01.09.2000 07:24	30.03.2001 07:09	99.88
70	05.09.2000 12:49	01.03.2001 11:19	84.45	01.09.2000 01:07	30.03.2001 09:11	99.8
71	04.10.2000 06:53	08.02.2001 02:15	60.19	01.09.2000 06:56	30.03.2001 09:11	99.69
72	30.10.2000 01:21	29.03.2001 03:27	71.2	05.09.2000 12:49	30.03.2001 09:11	97.91
73	08.09.2000 02:43	29.03.2001 05:45	95.91	01.09.2000 07:24	30.03.2001 07:09	99.88
74	05.09.2000 09:38	23.03.2001 07:37	94.4	01.09.2000 07:24	30.03.2001 07:09	99.88
75	25.09.2000 09:50	01.12.2000 05:51	31.73	05.09.2000 11:52	30.03.2001 08:08	97.43
76	11.09.2000 04:21	01.03.2001 07:33	81.22	01.09.2000 01:07	30.03.2001 09:11	99.8
77	01.12.2000 04:11	21.03.2001 08:21	52.28	01.09.2000 06:56	30.03.2001 09:11	99.69
78	14.09.2000 03:29	16.03.2001 09:52	87.22	01.09.2000 10:59	30.03.2001 07:09	99.8
79	31.10.2000 05:16	29.03.2001 12:16	70.58	05.09.2000 11:52	30.03.2001 07:09	97.65
80	06.09.2000 08:03	27.03.2001 11:20	95.91	01.09.2000 07:24	30.03.2001 09:11	99.68
81	10.09.2000 01:34	30.03.2001 09:10	95.29	01.09.2000 07:24	30.03.2001 09:11	99.68
82	14.09.2000 08:07	22.03.2001 01:12	89.56	01.09.2000 07:24	30.03.2001 09:11	99.68
83	01.09.2000 10:59	22.03.2001 05:32	95.76	01.09.2000 01:07	30.03.2001 10:47	99.83
84	01.09.2000 07:24	28.03.2001 08:13	98.71	01.09.2000 10:59	30.03.2001 09:11	99.61
85	07.09.2000 04:08	30.03.2001 07:09	97.09	01.09.2000 06:30	30.03.2001 10:47	99.73
86	19.09.2000 10:13	28.03.2001 08:58	90.13	05.09.2000 06:17	30.03.2001 07:09	98

C.4.3.2 Node Overview

Table C-22: Case study. Content-based clustering on linkevents. Node membership in clusters. Overview. Nodes 1125-1175

N.id	#C	Clusters
1125	7	70, 24, 81, 1, 26, 58, 84
1126	32	0, 2, 11, 13, 14, 15, 18, 24, 26, 27, 30, 34, 36, 38, 40, 41, 42, 50, 53, 55, 57, 62, 63, 66, 71, 73, 75, 76, 78, 79, 81, 83
1127	5	6, 7, 51, 76, 78
1128	6	30, 49, 57, 59, 69, 83
1130	12	6, 9, 17, 19, 24, 29, 32, 37, 45, 52, 55, 73
1132	21	0, 2, 8, 12, 20, 23, 25, 31, 33, 37, 41, 42, 45, 56, 69, 79, 80, 81, 83, 85, 86
1134	42	4, 5, 6, 8, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 27, 28, 29, 31, 32, 35, 37, 41, 44, 45, 47, 48, 49, 50, 52, 54, 58, 61, 65, 70, 73, 74, 76, 79, 84, 85, 86
1136	4	2, 23, 45, 56
1138	4	17, 29, 75, 83
1139	18	13, 17, 25, 26, 28, 29, 37, 44, 45, 47, 51, 53, 57, 58, 72, 73, 76, 85
1140	4	26, 63, 70, 84
1141	7	22, 32, 38, 40, 61, 63, 65
1142	28	1, 3, 11, 12, 20, 22, 24, 26, 30, 33, 34, 35, 38, 41, 47, 49, 51, 52, 60, 61, 62, 63, 68, 69, 73, 80, 81, 84
1143	34	0, 8, 11, 24, 26, 30, 34, 35, 38, 39, 41, 48, 49, 50, 51, 55, 56, 58, 59, 60, 61, 62, 63, 66, 67, 69, 70, 71, 73, 74, 78, 80, 82, 83
1144	17	1, 22, 23, 24, 30, 33, 34, 38, 47, 61, 62, 63, 68, 69, 80, 81, 84
1145	5	8, 22, 26, 80, 86
1146	13	13, 15, 24, 30, 34, 41, 53, 55, 66, 73, 75, 78, 83
1147	1	17
1150	10	2, 3, 10, 15, 17, 26, 53, 63, 73, 75
1151	7	13, 15, 34, 38, 41, 53, 73
1152	17	5, 7, 9, 13, 14, 17, 22, 32, 61, 65, 69, 70, 74, 78, 79, 80, 84
1153	5	11, 33, 38, 63, 81
1154	14	0, 1, 8, 24, 34, 36, 41, 46, 63, 67, 70, 73, 74, 84
1157	24	0, 1, 11, 26, 27, 34, 35, 41, 43, 58, 59, 60, 62, 63, 65, 66, 69, 70, 73, 74, 80, 82, 83, 84
1158	20	0, 8, 10, 12, 19, 23, 26, 31, 41, 47, 48, 49, 56, 57, 59, 69, 74, 78, 79, 85
1159	1	49
1160	2	44, 49
1162	11	0, 8, 13, 15, 27, 34, 57, 73, 76, 81, 83
1163	18	0, 8, 11, 13, 31, 37, 40, 41, 48, 50, 58, 66, 79, 80, 81, 83, 85, 86
1164	1	44
1165	15	0, 10, 13, 15, 18, 24, 27, 34, 52, 53, 73, 75, 78, 79, 81
1167	1	52
1169	15	12, 15, 17, 22, 40, 42, 44, 49, 50, 52, 53, 66, 71, 73, 75
1170	2	62, 63
1172	4	18, 46, 71, 80
1173	52	4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 35, 37, 38, 41, 44, 45, 47, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 61, 70, 73, 76, 78, 79, 81, 84, 86
1174	46	0, 1, 3, 11, 12, 13, 15, 17, 18, 22, 23, 24, 26, 27, 30, 34, 35, 36, 37, 38, 40, 41, 44, 50, 51, 52, 55, 57, 60, 61, 62, 63, 65, 68, 69, 70, 71, 73, 74, 76, 80, 81, 82, 83, 84, 85
1175	43	4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 26, 27, 28, 29, 31, 32, 37, 38, 41, 44, 45, 47, 49, 50, 51, 52, 54, 56, 58, 61, 73, 74, 76, 79, 81, 83, 84, 86

Appendix A: Case Study

Table C-23: Case study. Content-based clustering on linkevents. Node membership in clusters. Overview. Nodes 1177-1229

N.id	#C	Clusters
1177	8	13, 19, 22, 32, 46, 47, 76, 80
1178	3	13, 61, 66
1179	22	8, 13, 15, 16, 22, 25, 38, 40, 42, 44, 46, 54, 57, 62, 63, 65, 66, 67, 71, 73, 76, 78
1181	4	15, 40, 52, 75
1182	19	4, 5, 8, 12, 16, 19, 23, 26, 27, 29, 31, 45, 47, 48, 50, 55, 65, 79, 85
1183	4	28, 71, 74, 75
1184	15	2, 11, 13, 14, 15, 17, 24, 34, 50, 53, 63, 71, 73, 76, 81
1185	2	56, 57
1186	23	56, 57, 3, 10, 13, 15, 17, 21, 24, 25, 26, 27, 28, 34, 38, 40, 44, 52, 53, 60, 62, 63, 73, 74, 75
1187	6	0, 22, 46, 50, 71, 81
1188	7	34, 41, 46, 69, 70, 80, 84
1191	17	0, 1, 6, 10, 26, 34, 36, 40, 41, 46, 58, 61, 63, 65, 67, 70, 74
1192	50	4, 5, 6, 8, 9, 10, 12, 13, 14, 15, 16, 18, 19, 20, 22, 23, 26, 27, 28, 29, 30, 31, 32, 35, 37, 38, 41, 44, 45, 47, 48, 49, 50, 54, 56, 57, 58, 61, 65, 70, 73, 74, 76, 78, 79, 81, 83, 84, 85, 86
1193	17	3, 8, 22, 26, 30, 32, 36, 38, 40, 46, 49, 61, 63, 65, 67, 70, 84
1194	43	0, 2, 6, 11, 12, 14, 24, 25, 26, 30, 34, 35, 38, 39, 40, 41, 43, 49, 51, 54, 55, 56, 58, 59, 60, 61, 62, 63, 64, 66, 67, 69, 70, 71, 72, 74, 76, 77, 78, 80, 82, 83, 84
1195	24	1, 11, 12, 20, 22, 23, 24, 33, 34, 38, 41, 47, 49, 51, 52, 60, 61, 62, 63, 68, 73, 80, 81, 84
1196	1	15
1198	45	70, 35, 76, 61, 83, 0, 24, 55, 49, 4, 85, 68, 36, 9, 25, 11, 2, 71, 38, 6, 77, 12, 66, 34, 62, 60, 30, 80, 82, 40, 43, 67, 69, 26, 41, 74, 59, 58, 84, 78, 39, 64, 72, 14, 56
1199	6	6, 10, 46, 53, 62, 67
1200	27	1, 2, 5, 8, 12, 20, 23, 24, 25, 31, 33, 37, 41, 45, 56, 57, 58, 65, 69, 78, 79, 80, 81, 83, 84, 85, 86
1201	22	5, 14, 18, 19, 23, 24, 26, 31, 41, 45, 47, 48, 49, 55, 56, 58, 63, 65, 73, 79, 85, 86
1202	3	5, 14, 18, 19, 23, 24, 26, 31, 41, 45, 47, 48, 49, 55, 56, 58, 63, 65, 73, 79, 85, 86, 44, 61, 69
1203	11	63, 9, 54, 25, 6, 62, 22, 40, 44, 73, 57
1206	16	0, 18, 20, 24, 33, 34, 35, 41, 46, 49, 55, 61, 63, 70, 80, 83
1210	6	38, 49, 57, 59, 63, 69
1211	16	10, 12, 17, 24, 30, 40, 44, 48, 49, 56, 59, 63, 69, 73, 74, 83
1212	3	10, 12, 17, 24, 30, 40, 44, 48, 49, 56, 59, 63, 69, 73, 74, 83, 57, 59, 69
1214	11	7, 17, 29, 46, 47, 49, 53, 63, 70, 75, 83
1215	9	8, 10, 15, 21, 25, 57, 58, 69, 74
1216	7	15, 16, 17, 40, 44, 49, 73
1218	9	8, 48, 49, 50, 59, 63, 69, 73, 74
1221	16	7, 18, 22, 24, 26, 40, 49, 50, 53, 63, 70, 71, 76, 80, 81, 85
1222	11	0, 8, 48, 49, 50, 59, 63, 69, 73, 74, 83
1225	9	12, 15, 16, 22, 40, 44, 49, 52, 83
1226	2	0, 13
1227	7	13, 15, 18, 36, 40, 50, 57
1229	4	15, 17, 53, 75

Table C-24: Case study. Content-based clustering on linkevents. Node membership in clusters. Overview. Nodes 1232-1274

N.id	#C	Clusters
1232	28	6, 8, 9, 11, 13, 14, 15, 16, 18, 22, 25, 26, 37, 38, 40, 42, 44, 46, 50, 51, 54, 56, 57, 62, 63, 67, 69, 78
1233	1	49
1235	11	5, 18, 23, 24, 34, 49, 57, 63, 65, 69, 80
1236	6	15, 26, 40, 49, 63, 71
1239	20	0, 18, 20, 24, 33, 34, 35, 41, 46, 47, 49, 55, 61, 63, 69, 70, 76, 80, 83, 84
1240	6	17, 44, 56, 59, 69, 83
1243	1	17
1247	17	6, 8, 11, 14, 22, 25, 26, 38, 40, 42, 50, 54, 56, 62, 63, 69, 73
1248	16	6, 10, 22, 24, 30, 38, 46, 53, 54, 55, 62, 63, 66, 67, 78, 83
1249	2	44, 49
1250	15	3, 10, 15, 17, 21, 28, 31, 38, 40, 44, 52, 60, 73, 74, 75
1252	6	6, 10, 13, 46, 54, 67
1253	8	3, 15, 38, 40, 42, 44, 49, 81
1254	8	6, 10, 13, 14, 38, 46, 54, 67
1255	12	0, 8, 44, 48, 49, 50, 59, 63, 69, 73, 74, 83
1256	4	55, 57, 59, 69
1257	12	0, 8, 44, 48, 49, 50, 59, 63, 69, 73, 74, 83
1258	12	0, 8, 44, 48, 49, 50, 59, 63, 69, 73, 74, 83
1260	6	30, 57, 59, 66, 69, 74
1261	36	0, 1, 3, 8, 11, 12, 22, 24, 26, 30, 34, 35, 36, 37, 38, 41, 43, 49, 57, 58, 59, 60, 62, 63, 65, 66, 69, 70, 73, 74, 76, 80, 82, 84, 85, 86
1262	11	4, 10, 12, 17, 24, 38, 52, 57, 59, 69, 85
1263	35	0, 4, 9, 14, 17, 24, 26, 30, 34, 39, 40, 41, 43, 51, 55, 56, 58, 59, 60, 61, 62, 63, 64, 66, 67, 69, 70, 72, 74, 76, 78, 80, 82, 83, 84
1264	15	0, 14, 24, 30, 55, 58, 59, 60, 64, 69, 70, 74, 78, 82, 83
1265	35	0, 1, 3, 11, 12, 20, 22, 24, 26, 30, 33, 34, 36, 38, 41, 43, 47, 49, 52, 58, 60, 61, 62, 63, 65, 66, 69, 70, 73, 74, 80, 81, 82, 83, 84
1266	33	4, 5, 8, 12, 14, 17, 18, 23, 24, 26, 28, 31, 33, 41, 47, 48, 49, 50, 51, 55, 56, 57, 58, 63, 65, 72, 73, 76, 78, 79, 84, 85, 86
1269	6	0, 13, 15, 22, 40, 76
1270	2	15, 73
1271	32	4, 5, 6, 8, 10, 12, 13, 14, 16, 19, 22, 27, 28, 29, 31, 32, 37, 41, 44, 45, 46, 47, 49, 50, 54, 58, 61, 65, 73, 79, 83, 84
1272	7	26, 30, 58, 59, 69, 74, 78
1273	18	4, 8, 10, 12, 17, 30, 40, 48, 49, 50, 52, 55, 59, 63, 69, 73, 83, 85
1274	9	6, 10, 15, 40, 42, 46, 52, 57, 67

Table C-25: Case study. Content-based clustering on linkevents. Node membership in clusters. Metrics. Nodes 1125-1192

N.id	#C	Average									#R	%R
		#IAL	IACD	IALS	IADC	IABC	IACC	IALCC	%IATime span	#CtoC		
1125	7	1.29	0.11	1.76	6.10	0.00	1.82	14.29	0.63	50.86	0	0.00
1126	32	2.09	0.11	1.37	10.68	2.55	1.52	4.38	18.40	76.97	2	6.25
1127	5	1.80	0.11	2.70	13.63	0.85	1.17	13.33	2.93	57.40	0	0.00
1128	6	1.33	0.08	2.50	5.09	0.00	1.86	33.33	10.13	26.50	0	0.00
1130	12	1.58	0.16	3.53	13.51	2.08	1.75	10.83	30.33	46.33	0	0.00
1132	21	1.43	0.17	2.79	11.11	0.61	1.46	26.98	23.53	43.00	0	0.00
1134	42	2.36	0.12	3.52	17.03	2.73	1.59	39.37	39.40	37.50	1	2.38
1136	4	1.75	0.16	1.13	16.20	0.00	1.46	75.00	5.52	4.50	0	0.00
1138	4	1.00	0.15	1.25	7.07	0.00	1.13	0.00	0.00	0.00	0	0.00
1139	18	1.17	0.16	3.03	8.56	0.21	1.70	5.56	15.56	54.50	0	0.00
1140	4	1.00	0.11	2.00	3.70	0.00	2.23	0.00	2.25	31.50	0	0.00
1141	7	1.29	0.18	1.81	5.88	0.21	1.69	4.76	1.19	41.29	0	0.00
1142	28	1.82	0.12	2.18	11.54	1.06	1.79	31.19	16.56	48.21	1	3.57
1143	34	1.32	0.12	5.63	7.49	0.22	2.07	23.53	22.22	34.91	3	8.82
1144	17	2.24	0.08	2.18	12.50	0.92	2.00	34.90	17.24	46.29	0	0.00
1145	5	1.40	0.18	1.40	6.88	0.87	1.85	20.00	6.11	43.80	0	0.00
1146	13	1.31	0.06	2.12	6.49	2.37	2.07	23.08	11.85	61.15	0	0.00
1147	1	1.00	0.19	2.00	4.76	0.00	1.00	0.00	0.00	0.00	0	0.00
1150	10	1.40	0.08	1.14	9.40	1.54	2.00	0.00	0.00	33.80	1	10.00
1151	7	1.29	0.08	1.79	5.30	0.09	2.12	14.29	15.14	29.43	0	0.00
1152	17	1.00	0.16	2.29	6.75	0.00	2.07	0.00	2.23	63.24	1	5.88
1153	5	1.00	0.14	1.40	5.96	0.00	1.96	0.00	0.69	41.80	0	0.00
1154	14	1.29	0.09	4.00	6.88	1.28	2.20	21.43	8.76	51.00	0	0.00
1157	24	3.13	0.07	1.25	15.55	4.30	1.82	20.00	18.85	66.63	3	12.50
1158	20	1.20	0.16	3.43	6.69	0.05	1.91	10.00	11.42	77.80	0	0.00
1159	1	1.00	0.20	3.00	2.63	0.00	1.67	0.00	29.85	0.00	0	0.00
1160	2	1.00	0.21	1.50	3.40	0.00	1.34	0.00	14.93	0.00	0	0.00
1162	11	1.27	0.12	1.41	5.76	0.08	1.91	18.18	7.66	63.00	0	0.00
1163	18	1.28	0.19	2.03	7.91	0.84	1.76	22.22	2.23	62.94	0	0.00
1164	1	1.00	0.02	1.00	4.17	0.00	2.38	0.00	0.00	0.00	0	0.00
1165	15	1.33	0.13	1.50	7.66	0.38	1.86	13.33	14.66	54.80	0	0.00
1167	1	1.00	0.17	2.00	5.88	0.00	1.00	0.00	0.00	0.00	0	0.00
1169	15	2.27	0.12	1.88	11.91	5.14	1.43	9.56	28.91	67.40	1	6.67
1170	2	1.00	0.21	1.00	3.85	0.00	1.67	0.00	0.00	0.00	0	0.00
1172	4	1.25	0.16	1.25	7.94	0.00	1.94	25.00	19.10	16.50	0	0.00
1173	52	3.00	0.12	3.13	20.51	6.36	1.31	18.31	41.60	72.48	3	5.77
1174	46	2.43	0.10	2.16	13.86	4.07	1.67	22.39	30.38	75.76	7	15.22
1175	43	1.81	0.11	2.38	13.48	1.09	1.69	28.18	27.08	52.72	9	20.93
1177	8	1.13	0.14	2.38	6.97	0.21	1.66	0.00	2.45	56.88	1	12.50
1178	3	2.33	0.22	0.73	10.75	1.21	1.00	0.00	0.00	26.33	0	0.00
1179	22	1.05	0.15	3.14	6.26	0.10	1.80	0.00	0.59	63.14	1	4.55
1181	4	1.25	0.04	1.00	6.63	0.58	2.34	0.00	2.62	31.00	1	25.00
1182	19	1.63	0.13	3.11	12.80	3.18	1.74	33.33	21.25	59.16	2	10.53
1183	4	1.00	0.10	1.25	8.06	0.00	2.75	0.00	0.00	36.75	0	0.00
1184	15	2.07	0.11	1.61	9.74	4.03	1.86	0.50	13.65	51.47	1	6.67
1185	2	1.00	0.18	4.50	5.12	0.00	1.50	0.00	0.70	0.00	0	0.00
1186	23	2.09	0.11	1.47	16.18	9.90	1.63	0.87	13.88	52.70	1	4.35
1187	6	1.50	0.13	1.25	8.19	0.26	1.63	16.67	0.08	38.00	0	0.00
1188	7	1.00	0.18	1.57	4.23	0.00	1.60	0.00	9.02	21.43	0	0.00
1191	17	1.18	0.08	3.90	6.11	0.49	2.28	5.88	0.34	57.35	3	17.65
1192	50	2.42	0.13	4.32	17.14	3.05	1.48	26.84	39.43	54.86	3	6.00

IAL:= intra-cluster links
 IALS: intra-cluster link strength
 IACD:= intra-cluster content diss.
 IADC:= intra-cluster degree centrality

IABC:= intra-cluster betweenness centrality
 IACC:= intra-cluster closeness centrality
 IALCC:= intra-cluster local clustering coefficient
 #CtoC:= contacts to other clusters

#R:= number of clusters the node is a repr.
 %R:=percentage of clusters is a repr.

Table C-26: Case study. Content-based clustering on linkevents. Node membership in clusters. Metrics. Nodes 1193-1274

N.id	#C	Average									#R	%R
		#IAL	IACD	IALS	IADC	IABC	IACC	IALCC	%IATime span	#CtoC		
1193	17	1.24	0.13	3.65	6.65	0.20	1.84	7.84	3.48	37.65	0	0.00
1194	43	1.49	0.12	9.82	12.68	1.33	1.77	22.48	38.98	56.86	2	4.65
1195	24	1.83	0.10	2.23	11.43	0.99	1.84	31.94	23.57	47.33	6	25.00
1196	1	2.00	0.00	1.50	7.69	0.00	2.42	100.00	25.74	0.00	0	0.00
1198	45	2.84	0.13	5.11	23.92	9.85	1.36	14.33	49.51	58.84	7	15.56
1199	6	1.00	0.06	8.17	6.04	0.00	1.98	0.00	0.13	38.83	0	0.00
1200	27	1.74	0.17	2.64	12.14	1.80	1.37	11.73	29.80	75.37	1	3.70
1201	22	1.59	0.16	2.32	10.46	2.80	1.50	8.06	15.18	65.45	2	9.09
1202	3	1.00	0.14	1.00	3.90	0.00	1.46	0.00	0.00	41.00	0	0.00
1203	11	1.27	0.16	1.86	7.38	0.07	2.06	9.09	11.81	60.64	0	0.00
1206	16	1.13	0.19	1.72	5.80	0.00	1.39	12.50	8.03	25.94	0	0.00
1210	6	1.67	0.07	2.87	6.33	0.66	1.68	0.00	3.80	23.33	0	0.00
1211	16	2.31	0.18	2.29	8.45	0.38	1.45	17.92	12.58	46.69	0	0.00
1212	3	1.67	0.00	4.00	6.92	0.00	1.92	33.33	10.67	17.00	0	0.00
1214	11	1.45	0.17	1.02	9.31	0.62	1.20	0.00	1.03	28.73	0	0.00
1215	9	1.11	0.16	1.56	10.15	1.99	1.80	0.00	0.75	50.22	0	0.00
1216	7	2.00	0.06	1.60	8.43	3.24	1.75	20.00	23.30	66.43	1	14.29
1218	9	1.00	0.08	3.00	3.79	0.00	1.98	0.00	10.80	34.22	1	11.11
1221	16	1.81	0.15	1.44	10.29	2.11	1.53	15.00	10.73	62.25	1	6.25
1222	11	1.09	0.06	2.64	4.00	0.00	2.05	9.09	12.21	36.18	0	0.00
1225	9	2.00	0.12	2.03	8.89	3.58	1.96	18.52	30.89	40.67	0	0.00
1226	2	1.00	0.24	1.00	4.09	0.00	1.00	0.00	0.00	0.00	0	0.00
1227	7	2.43	0.13	1.04	11.53	3.80	1.36	3.81	7.65	52.86	0	0.00
1229	4	1.00	0.18	2.75	6.21	0.00	1.57	0.00	20.36	0.00	0	0.00
1232	28	1.54	0.15	2.77	8.72	0.73	1.51	8.93	10.40	74.43	2	7.14
1233	1	1.00	0.20	2.00	2.63	0.00	1.67	0.00	29.85	0.00	0	0.00
1235	11	1.55	0.20	1.15	7.50	0.63	1.51	9.09	6.45	35.09	0	0.00
1236	6	1.17	0.14	1.42	4.53	0.02	2.31	0.00	1.79	20.67	0	0.00
1239	20	1.70	0.19	1.42	7.77	0.45	1.10	3.33	20.39	23.35	0	0.00
1240	6	1.17	0.14	3.00	4.80	0.00	1.51	16.67	3.63	15.00	1	16.67
1243	1	1.00	0.19	2.00	4.76	0.00	1.00	0.00	0.00	0.00	0	0.00
1247	17	1.59	0.15	2.08	8.51	2.63	1.37	5.88	14.04	49.47	1	5.88
1248	16	1.06	0.09	5.03	5.61	0.00	1.81	6.25	0.66	48.00	0	0.00
1249	2	2.00	0.21	1.00	6.03	0.22	1.00	0.00	14.93	0.00	0	0.00
1250	15	1.27	0.13	2.33	10.66	1.21	1.88	6.67	11.09	38.67	1	6.67
1252	6	1.00	0.04	7.50	6.06	0.00	2.05	0.00	0.09	39.50	1	16.67
1253	8	1.13	0.10	1.13	6.78	0.26	2.32	0.00	3.97	49.75	0	0.00
1254	8	1.00	0.06	6.13	5.81	0.00	1.98	0.00	0.07	54.63	0	0.00
1255	12	1.42	0.07	2.76	5.20	0.00	1.98	33.33	19.51	33.17	1	8.33
1256	4	1.00	0.06	1.50	4.39	0.00	1.71	0.00	0.00	20.50	0	0.00
1257	12	1.08	0.05	2.83	4.02	0.00	2.08	8.33	16.13	66.67	1	8.33
1258	12	1.50	0.07	2.85	5.58	0.01	1.98	31.94	21.33	34.08	1	8.33
1260	6	1.00	0.10	3.17	4.51	0.00	1.96	0.00	3.48	37.17	0	0.00
1261	36	3.33	0.12	1.48	19.58	7.98	1.45	17.86	18.62	66.83	1	2.78
1262	11	1.00	0.16	2.64	5.12	0.00	1.35	0.00	3.03	23.82	1	9.09
1263	35	3.71	0.13	2.78	20.89	7.84	1.59	6.80	38.66	67.86	1	2.86
1264	15	4.13	0.09	2.48	19.82	7.45	1.80	20.86	21.14	65.47	1	6.67
1265	35	1.86	0.09	2.17	10.44	0.32	1.95	37.05	19.01	60.63	4	11.43
1266	33	1.67	0.17	3.68	10.84	1.84	1.33	5.05	22.58	68.70	0	0.00
1269	6	1.67	0.15	1.38	6.73	1.29	1.83	5.56	10.43	33.67	0	0.00
1270	2	1.50	0.00	2.25	5.16	0.00	2.15	50.00	48.02	30.00	0	0.00
1271	32	1.91	0.11	3.03	13.77	1.59	1.79	37.29	21.29	47.84	3	9.38
1272	7	1.14	0.08	2.64	4.74	0.00	2.22	14.29	7.38	41.00	0	0.00
1273	18	3.39	0.15	2.14	14.01	2.57	1.15	6.66	18.88	38.94	0	0.00
1274	9	2.44	0.09	5.01	16.32	10.07	2.10	0.40	3.33	63.33	1	11.11

C.4.3.3 Structural Key Players

Table C-27: Case study. Content-based clustering on linkevents. Share of structural key players in cluster. Overview

C.id	#KP	RiskP	#N	%KP/N	%KP	C.id	#KP	RiskP	#N	%KP/N	%KP	C.id	#KP	RiskP	#N	%KP/N	%KP
0	11	true	26	42.31%	52.38%	29	6	false	10	60.00%	28.57%	58	15	true	22	68.18%	71.43%
1	5	true	11	45.45%	23.81%	30	12	true	21	57.14%	57.14%	59	9	false	23	39.13%	42.86%
2	5	true	8	62.50%	23.81%	31	7	true	13	53.85%	33.33%	60	9	true	13	69.23%	42.86%
3	3	true	9	33.33%	14.29%	32	5	true	10	50.00%	23.81%	61	11	false	22	50.00%	52.38%
4	8	true	11	72.73%	38.10%	33	2	false	10	20.00%	9.52%	62	9	false	20	45.00%	42.86%
5	6	true	11	54.55%	28.57%	34	10	true	24	41.67%	47.62%	63	11	false	42	26.19%	52.38%
6	7	false	18	38.89%	33.33%	35	9	true	12	75.00%	42.86%	64	4	true	4	100.00%	19.05%
7	3	false	6	50.00%	14.29%	36	5	true	9	55.56%	23.81%	65	8	true	17	47.06%	38.10%
8	10	true	27	37.04%	47.62%	37	9	true	13	69.23%	42.86%	66	8	true	16	50.00%	38.10%
9	6	true	10	60.00%	28.57%	38	10	false	27	37.04%	47.62%	67	4	false	15	26.67%	19.05%
10	7	false	20	35.00%	33.33%	39	4	true	4	100.00%	19.05%	68	2	false	5	40.00%	9.52%
11	9	true	15	60.00%	42.86%	40	7	false	27	25.93%	33.33%	69	12	false	37	32.43%	57.14%
12	13	false	22	59.09%	61.90%	41	16	false	30	53.33%	76.19%	70	13	true	23	56.52%	61.90%
13	9	true	25	36.00%	42.86%	42	1	false	8	12.50%	4.76%	71	7	false	13	53.85%	33.33%
14	11	true	17	64.71%	52.38%	43	6	true	6	100.00%	28.57%	72	4	true	5	80.00%	19.05%
15	5	true	27	18.52%	23.81%	44	7	true	25	28.00%	33.33%	73	14	true	39	35.90%	66.67%
16	5	true	10	50.00%	23.81%	45	7	true	12	58.33%	33.33%	74	13	false	28	46.43%	61.90%
17	6	true	22	27.27%	28.57%	46	1	false	19	5.26%	4.76%	75	1	false	12	8.33%	4.76%
18	7	true	16	43.75%	33.33%	47	8	true	17	47.06%	38.10%	76	13	false	21	61.90%	61.90%
19	6	true	10	60.00%	28.57%	48	5	false	15	33.33%	23.81%	77	2	true	2	100.00%	9.52%
20	5	true	10	50.00%	23.81%	49	13	false	39	33.33%	61.90%	78	10	true	20	50.00%	47.62%
21	0	false	3	0.00%	0.00%	50	11	true	24	45.83%	52.38%	79	8	false	15	53.33%	38.10%
22	9	true	26	34.62%	42.86%	51	7	true	12	58.33%	33.33%	80	10	false	23	43.48%	47.62%
23	6	false	14	42.86%	28.57%	52	6	false	18	33.33%	28.57%	81	9	false	20	45.00%	42.86%
24	13	true	30	43.33%	61.90%	53	5	false	15	33.33%	23.81%	82	9	true	9	100.00%	42.86%
25	5	true	12	41.67%	23.81%	54	6	false	13	46.15%	28.57%	83	14	true	31	45.16%	66.67%
26	15	true	30	50.00%	71.43%	55	9	true	18	50.00%	42.86%	84	13	true	24	54.17%	61.90%
27	8	true	12	66.67%	38.10%	56	9	true	18	50.00%	42.86%	85	10	false	16	62.50%	47.62%
28	6	true	10	60.00%	28.57%	57	8	false	24	33.33%	38.10%	86	6	true	11	54.54%	28.57%

#KP:= number of structural key players
 RiskP:= node representative is structural key players
 #N:= number of nodes
 %KP/N:= percentage of nodes in cluster that are structural key players
 %KP:= percentage of all structural key players that are member of this cluster

Table C-28: Case study. Content-based clustering on linkevents. Cluster memberships of structural key players

N.id	#R	#C	Nodes			Linkevents		
			$\emptyset N$	Max	Min	$\emptyset LE$	Max	Min
1126	2	32	7.84	42	8	11.56	113	7
1134	1	42	8.71	39	10	18.74	136	7
1139	0	18	3.79	39	5	6.99	124	7
1143	3	34	8.83	42	4	15.37	113	8
1157	3	24	6.13	42	6	10.74	113	9
1158	0	20	5.10	39	10	11.16	136	15
1173	3	52	11.20	39	6	20.77	136	7
1174	7	46	11.17	42	5	17.37	124	5
1175	9	43	9.25	39	6	18.10	136	7
1184	1	15	3.93	42	8	4.93	51	8
1192	3	50	11.01	39	10	22.08	136	7
1194	2	43	9.68	42	2	17.30	113	8
1198	7	45	9.49	39	2	19.17	124	5
1200	1	27	5.76	37	8	10.37	124	10
1221	1	16	4.26	42	6	6.62	124	8
1261	1	36	9.02	42	6	16.37	124	7
1263	1	35	8.09	42	4	15.54	113	9
1264	1	15	3.70	37	4	6.85	77	10
1265	4	35	8.80	42	6	14.76	113	7
1271	3	32	7.06	39	10	14.10	136	7
1273	0	18	5.20	42	11	8.37	124	10

Structural key players marked bold (top 9) or underlined (top 21)

C.4.3.4 Node Role Categorization

C.4.3.4.1 Node Overview

Table C-29: Case study. Content-based clustering on linkevents. Node role categorization. Node overview. Nodes 1125-1188

N.id	#C	#NR	(D)	(S)	B	C	HS	I	IS	ISO	M	MM	PS	S	ST	TW
1125	7	1											7			
1126	32	5			1						1	7	19			4
1127	5	4										1	2	1		1
1128	6	1											6			
1130	12	3							1				10	1		
1132	21	4										2	17	1		1
1134	42	8				2	1	1	1			4	24	5		4
1136	4	2											3	1		
1138	4	1											4			
1139	18	3										2	15	1		
1140	4	1											4			
1141	7	2										1	6			
1142	28	5							1			6	18	2		1
1143	34	3										1	32	1		
1144	17	4										3	10	2		2
1145	5	2										1	4			
1146	13	2										1	12			
1147	1	1											1			
1150	10	2						1					9			
1151	7	2										1	6			
1152	17	2											16	1		
1153	5	1											5			
1154	14	2						1					13			
1157	24	4						5				4	12	3		
1158	20	3										2	17	1		
1159	1	1											1			
1160	2	1											2			
1162	11	2										1	10			
1163	18	3									1		16	1		
1164	1	1											1			
1165	15	3										1	13			1
1167	1	1											1			
1169	15	4						3				4	7			1
1170	2	1											2			
1172	4	1											4			
1173	52	8			2	2	3		5			14	17	2		7
1174	46	7			1		1	3				12	21	4		4
1175	43	6				2	1					2	32	5		1
1177	8	2										1	7			
1178	3	2											2			1
1179	22	2										1	21			
1181	4	2										1	3			
1182	19	5			1			1				2	12	3		
1183	4	1											4			
1184	15	3					1					1	13			
1185	2	1											2			
1186	23	6						2			1	3	14		1	2
1187	6	2										2	4			
1188	7	1											7			

N.id:= node id
 #C:= number of clusters (per node)
 #NR:= number of different nodes roles
 (D):= (Dyad)
 (S):= (Specialist)

B:= Broker
 C:= Coordinator
 HS:= Hot Spot
 I:= Integrator
 IS:= Information Spreader

ISO:= Isolated
 M:= Mediator
 MM:= Middleman
 PS:= Peripheral
 S:= Specialist

ST:= Star
 TW:= Team Worker

Table C-30: Case study. Content-based clustering on linkevents. Node role categorization. Node overview. Nodes 1191-1275

N.id	#C	#NR	(D)	(S)	B	C	HS	I	IS	ISO	M	MM	PS	S	ST	TW
1191	17	2										1	16			
1192	50	9			1	2	1	1	1			12	21	4		7
1193	17	2										2	15			
1194	43	6	1					1				3	32	5		1
1195	24	4										5	15	2		2
1196	1	1											1			
1198	45	10	1		1			3	3		1	16	9	2	2	7
1199	6	1											6			
1200	27	5						2				8	13	1		3
1201	22	5						1	1			1	18			1
1202	3	1											3			
1203	11	2										1	10			
1206	16	1											16			
1210	6	2											5			1
1211	16	3										5	10			1
1212	3	1											3			
1214	11	4										1	8	1		1
1215	9	3		1							1		7			
1216	7	3									1	2	4			
1218	9	1											9			
1221	16	4							1			4	10	1		
1222	11	1											11			
1225	9	3									2	2	5			
1226	2	1											2			
1227	7	4									1	1	4			1
1229	4	1											4			
1232	28	4								1		7	19			1
1233	1	1											1			
1235	11	3										2	8			1
1236	6	2										1	5			
1239	20	2										8	12			
1240	6	1											6			
1243	1	1											1			
1247	17	4						1			1	2	13			
1248	16	1											16			
1249	2	2										1	1			
1250	15	4		1							1	2	11			
1252	6	1											6			
1253	8	2										1	7			
1254	8	1											8			
1255	12	1											12			
1256	4	1											4			
1257	12	1											12			
1258	12	2										1	11			
1260	6	1											6			
1261	36	6					3	2	1			7	14			9
1262	11	1											11			
1263	35	8				1	1	3	5			8	12	4		1
1264	15	5					2					1	9	2		1
1265	35	5				1						4	27	2		1
1266	33	5						2				7	19	1		4
1269	6	2										2	4			
1270	2	1											2			
1271	32	6					1				1	2	22	5		1
1272	7	1											7			
1273	18	4							1			6	8			3
1274	9	3					2						6			1
1275	6	3					1		1				4			

C.4.3.4.2 Cluster Overview

Table C-31: Case study. Content-based clustering on linkevents. Node role categorization. Cluster overview. Clusters 0-40

C.id	#N	#NR	(S)	(D)	B	C	HS	I	IS	ISO	M	MM	PS	S	ST	TW
all			2	2	7	10	20	31	22	1	12	207	1049	65	3	78
0	26	3						3				4	19			
1	11	4				1	2						6	2		
2	8	3											4	3		1
3	9	3			1		2						6			
4	11	5				1		1	1				5	3		
5	11	5				1			1				5	3		1
6	18	5					2	1				2	11			2
7	6	1												6		
8	27	3										5	21			1
9	10	5					1		1				5	2		1
10	20	3										2	15			3
11	15	3							1			2	12			
12	22	3										2	18			2
13	25	2											21			4
14	17	2											14			3
15	27	6					1	1			2	3	19			1
16	10	4							1				6	2		1
17	22	2										2	20			
18	16	4								1		1	12			2
19	10	4					2		1				3	4		
20	10	2											8			2
21	3	2	2												1	
22	26	3										8	16			2
23	14	3						2				1	11			
24	30	3										4	23			3
25	12	2										2	10			
26	30	4						1				10	18			1
27	12	4				2							6	2		2
28	10	4							1				6	2		1
29	10	3			1		1						8			
30	21	4							1			2	17			1
31	13	5						1			1	2	8			1
32	10	4					1		1				5	3		
33	10	3											6	1		3
34	24	4			2			2				8	12			
35	12	3						1				1	10			
36	9	4					1	1					5	2		
37	13	4						1	1			1	10			
38	27	3										5	21			1
39	4	2												3	1	
40	27	3									2	11	14			

C.id:= cluster id

#N:= number of nodes (per cluster)

#NR:= number of different nodes roles

(D):= (Dyad)

(S):= Specialist

B:= Broker

C:= Coordinator

HS:= Hot Spot

I:= Integrator

IS:= Information Spreader

ISO:= Isolated

M:= Mediator

MM:= Middleman

PS:= Peripheral Specialist

S:= Specialist

ST:= Star

TW:= Team Worker

Table C-32: Case study. Content-based clustering on linkevents. Node role categorization. Cluster overview. Clusters 0-40

C.id	#N	#NR	(S)	(D)	B	C	HS	I	IS	ISO	M	MM	PS	S	ST	TW
41	30	3										5	24			1
42	8	3					1	2					5			
43	6	2							2					4		
44	25	5					1	1			1	2	20			
45	12	4				4						1	4	3		
46	19	3							1				17			1
47	17	3										3	11			3
48	15	3							1				13			1
49	39	3										7	31			1
50	24	2											21			3
51	12	2										2	10			
52	18	2										5	13			
53	15	3						1				1	13			
54	13	4						1				3	7			2
55	18	3							1			2	15			
56	18	2										4	14			
57	24	3										5	18			1
58	22	3						1				8	13			
59	23	4					1					5	16			1
60	13	3						1					10			2
61	22	2										5	17			
62	20	4						2				2	14			2
63	42	2										13	29			
64	4	2												3	1	
65	17	3										3	13			1
66	16	3							1				14			1
67	15	4					1	1				1	12			
68	5	2							1					4		
69	37	3					1					5	31			
70	23	3						1				6	16			
71	13	4							1			1	10			1
72	5	2							1					4		
73	39	3										8	30			1
74	28	5					1	1			2	3	21			
75	12	3					1				2		9			
76	21	2										6	15			
77	2	1		2												
78	20	4				1						2	13			4
79	15	4			3			3			1		8			
80	23	5						1			1	6	14			1
81	20	4										4	10	4		2
82	9	4							1				2	4		2
83	31	3							1			6	24			
84	24	3										4	17			3
85	16	4							1			1	13			1
86	11	3											6	1		4

C.4.3.5 Knowledge Domain Categorization

Table C-33: Case study. Content-based clustering on linkevents. Knowledge domain categorization. Clusters 0-49

C.id	#N	#LE	Max(#LE)	Ø(#LE)	SD(#LE)	Max(%LE)	Ø(%LE)	SD(%LE)	Knowledge Domain Categorization		
									Homogeneity	Frequency	Activity
0	26	126	33	4.85	8.75	26.19%	3.85%	6.94%	het.	freq.	medium
1	11	104	29	9.45	9.06	27.88%	9.09%	8.71%	het.	infreq.	medium
2	8	48	19	6.00	7.52	39.58%	12.50%	15.66%	het.	infreq.	low
3	9	39	10	4.33	3.43	25.64%	11.11%	8.80%	het.	infreq.	low
4	11	144	50	13.09	16.16	34.72%	9.09%	11.22%	het.	infreq.	high
5	11	104	38	9.45	12.52	36.54%	9.09%	12.04%	het.	infreq.	medium
6	18	132	37	7.33	9.60	28.03%	5.56%	7.27%	het.	freq.	medium
7	6	22	9	3.67	3.77	40.91%	16.67%	17.14%	het.	infreq.	low
8	27	131	20	4.85	6.08	15.27%	3.70%	4.64%	het.	freq.	medium
9	10	110	39	11.00	14.39	35.45%	10.00%	13.09%	het.	infreq.	medium
10	20	149	46	7.45	12.70	30.87%	5.00%	8.52%	het.	freq.	high
11	15	34	7	2.27	1.81	20.59%	6.67%	5.31%	het.	infreq.	low
12	22	141	20	6.41	6.87	14.18%	4.55%	4.87%	hom.	freq.	high
13	25	145	36	5.80	9.36	24.83%	4.00%	6.45%	het.	freq.	high
14	17	60	13	3.53	3.73	21.67%	5.88%	6.21%	het.	infreq.	low
15	27	79	10	2.93	1.92	12.66%	3.70%	2.43%	hom.	freq.	low
16	10	50	20	5.00	5.98	40.00%	10.00%	11.97%	het.	infreq.	low
17	22	51	10	2.32	2.26	19.61%	4.55%	4.44%	het.	freq.	low
18	16	28	3	1.75	0.83	10.71%	6.25%	2.96%	hom.	infreq.	low
19	10	301	132	30.10	38.15	43.85%	10.00%	12.67%	het.	infreq.	high
20	10	22	6	2.20	1.72	27.27%	10.00%	7.82%	het.	infreq.	low
21	3	6	3	2.00	0.82	50.00%	33.33%	13.61%	het.	infreq.	low
22	26	66	6	2.54	1.55	9.09%	3.85%	2.35%	hom.	freq.	low
23	14	35	5	2.50	1.30	14.29%	7.14%	3.70%	hom.	infreq.	low
24	30	156	25	5.27	6.02	15.82%	3.33%	3.81%	het.	freq.	high
25	12	21	3	1.75	0.83	14.29%	8.33%	3.95%	hom.	infreq.	low
26	30	146	30	5.00	6.88	20.00%	3.33%	4.59%	het.	freq.	high
27	12	58	11	4.83	3.39	18.97%	8.33%	5.84%	het.	infreq.	low
28	10	57	24	5.70	7.73	42.11%	10.00%	13.57%	het.	infreq.	low
29	10	23	4	2.30	1.00	17.39%	10.00%	4.37%	hom.	infreq.	low
30	21	139	34	6.62	8.70	24.46%	4.76%	6.26%	het.	freq.	medium
31	13	51	11	3.92	3.22	21.57%	7.69%	6.32%	het.	infreq.	low
32	10	236	74	23.60	26.60	31.36%	10.00%	11.27%	het.	infreq.	high
33	10	30	9	3.00	2.53	30.00%	10.00%	8.43%	het.	infreq.	low
34	24	100	12	4.17	2.84	12.00%	4.17%	2.84%	hom.	freq.	medium
35	12	61	21	5.08	6.38	34.43%	8.33%	10.46%	het.	infreq.	low
36	9	19	5	2.11	1.37	26.32%	11.11%	7.21%	het.	infreq.	low
37	13	60	17	4.62	4.31	28.33%	7.69%	7.18%	het.	infreq.	low
38	27	107	12	3.96	3.47	11.21%	3.70%	3.24%	hom.	freq.	medium
39	4	84	35	21.00	13.19	41.67%	25.00%	15.70%	het.	infreq.	low
40	27	85	11	3.15	2.45	12.94%	3.70%	2.88%	hom.	freq.	low
41	30	79	10	2.63	2.32	12.66%	3.33%	2.93%	hom.	freq.	low
42	8	16	5	2.00	1.41	31.25%	12.50%	8.84%	het.	infreq.	low
43	6	19	8	3.17	2.79	42.11%	16.67%	14.70%	het.	infreq.	low
44	25	87	19	3.48	4.51	21.84%	4.00%	5.18%	het.	freq.	low
45	12	208	42	17.33	16.15	20.19%	8.33%	7.76%	het.	infreq.	high
46	19	396	35	20.79	16.67	8.86%	5.26%	4.22%	hom.	freq.	high
47	17	99	34	5.82	9.58	34.34%	5.88%	9.68%	het.	infreq.	low
48	15	60	17	4.00	4.46	28.33%	6.67%	7.43%	het.	infreq.	low
49	39	117	24	3.00	3.90	20.51%	2.56%	3.33%	het.	freq.	medium

C.id:= cluster id
 N:= node
 LE: intra-cluster linkevent
 SD:= standard deviation
 het.:= heterogeneous
 hom.:= homogeneous
 freq.:= frequent
 infreq.:= infrequent

Table C-34: Case study. Content-based clustering on linkevents. Knowledge domain categorization. Clusters 50-86

C.id	#N	#LE	Max(#LE)	Ø(#LE)	SD(#LE)	Max(%LE)	Ø(%LE)	SD(%LE)	Knowledge Domain Categorization		
									Homogeneity	Frequency	Activity
50	24	74	17	3.08	4.32	22.97%	4.17%	5.84%	het.	freq.	low
51	12	30	7	2.50	1.66	23.33%	8.33%	5.53%	het.	infreq.	low
52	18	47	7	2.61	1.74	14.89%	5.56%	3.69%	hom.	freq.	low
53	15	36	7	2.40	1.89	19.44%	6.67%	5.25%	het.	infreq.	low
54	13	63	15	4.85	4.55	23.81%	7.69%	7.23%	het.	infreq.	low
55	18	39	4	2.17	1.07	10.26%	5.56%	2.74%	hom.	freq.	low
56	18	70	10	3.89	2.83	14.29%	5.56%	4.04%	hom.	freq.	low
57	24	54	11	2.25	2.52	20.37%	4.17%	4.67%	het.	freq.	low
58	22	92	17	4.18	4.26	18.48%	4.55%	4.63%	het.	freq.	low
59	23	93	15	4.04	3.75	16.13%	4.35%	4.03%	het.	freq.	low
60	13	186	69	14.31	23.10	37.10%	7.69%	12.42%	het.	infreq.	high
61	22	66	17	3.00	4.31	25.76%	4.55%	6.52%	het.	freq.	low
62	20	157	47	7.85	12.43	29.94%	5.00%	7.92%	het.	freq.	high
63	42	111	16	2.64	3.03	14.41%	2.38%	2.73%	hom.	freq.	medium
64	4	50	21	12.50	8.05	42.00%	25.00%	16.09%	het.	infreq.	low
65	17	46	8	2.71	2.32	17.39%	5.88%	5.04%	het.	infreq.	low
66	16	310	107	19.38	34.29	34.52%	6.25%	11.06%	het.	infreq.	high
67	15	216	71	14.53	24.59	32.57%	6.67%	11.28%	het.	infreq.	high
68	5	13	3	2.60	0.49	23.08%	20.00%	3.77%	hom.	infreq.	low
69	37	288	29	7.78	6.63	10.07%	2.70%	2.30%	hom.	freq.	high
70	23	118	39	5.13	9.03	33.05%	4.35%	7.65%	het.	freq.	medium
71	13	27	3	2.08	0.83	11.11%	7.69%	3.07%	hom.	infreq.	low
72	5	20	9	4.00	3.69	45.00%	20.00%	18.44%	het.	infreq.	low
73	39	136	18	3.49	2.99	13.24%	2.56%	2.20%	hom.	freq.	medium
74	28	273	66	9.75	16.16	24.18%	3.57%	5.92%	het.	freq.	high
75	12	22	6	1.83	1.40	27.27%	8.33%	6.38%	het.	infreq.	low
76	21	57	9	2.71	2.37	15.79%	4.76%	4.16%	het.	freq.	low
77	2	18	9	9.00	0.00	50.00%	50.00%	0.00%	hom.	infreq.	low
78	20	148	29	7.40	7.46	19.59%	5.00%	5.04%	het.	freq.	high
79	15	93	19	6.20	5.67	20.43%	6.67%	6.10%	het.	infreq.	low
80	23	73	11	3.17	3.03	15.07%	4.35%	4.15%	hom.	freq.	low
81	20	118	25	5.90	8.32	21.19%	5.00%	7.05%	het.	freq.	medium
82	9	72	25	8.00	8.91	34.72%	11.11%	12.37%	het.	infreq.	low
83	31	54	4	1.74	1.05	7.41%	3.23%	1.94%	hom.	freq.	low
84	24	93	11	4.08	2.63	11.22%	4.17%	2.68%	hom.	freq.	low
85	16	260	117	16.25	29.27	45.00%	6.25%	11.26%	het.	infreq.	high
86	11	52	18	4.73	6.09	34.62%	9.09%	11.71%	het.	infreq.	low

Table C-35: Case study. Content-based clustering on linkevents. Overview of knowledge domain categorization

Homogeneity	Frequency	Activity Type	#Clusters	Clusters
homogeneous	frequent	low	10	15, 22, 40, 41, 52, 55, 56, 80, 83, 84
homogeneous	frequent	medium	4	34, 38, 63, 73
homogeneous	frequent	high	3	12, 46, 69
homogeneous	infrequent	low	7	18, 23, 25, 29, 68, 71, 77
homogeneous	infrequent	medium	0	-
homogeneous	infrequent	high	0	-
heterogeneous	frequent	low	8	17, 44, 50, 57, 58, 59, 61, 76
heterogeneous	frequent	medium	7	0, 6, 8, 30, 49, 70, 81
heterogeneous	frequent	high	7	10, 13, 24, 26, 62, 74, 78
heterogeneous	infrequent	low	30	2, 3, 7, 11, 14, 16, 20, 21, 27, 28, 31, 33, 35, 36, 37, 39, 42, 43, 47, 48, 51, 53, 54, 64, 65, 72, 75, 79, 82, 86
heterogeneous	infrequent	medium	3	1, 5, 9
heterogeneous	infrequent	high	8	4, 19, 32, 45, 60, 66, 67, 85

C.4.3.6 Knowledge Profile Categorization

Table C-36: Case study. Content-based clustering on linkevents. Knowledge profile categorization. Nodes 1125-1192

N.id	#C	#LE	Max(%LE)	Ø(%LE)	SD(%LE)	Knowledge Profile Categorization		
						Homogeneity	Diversification	Activity
1125	7	15	26.67%	14.29%	6.60%	het.	spec.	low
1126	32	77	11.69%	3.13%	2.51%	hom.	div.	medium
1127	5	20	45.00%	20.00%	18.44%	het.	spec.	low
1128	6	17	58.82%	16.67%	19.34%	het.	spec.	low
1130	12	81	48.15%	8.33%	12.68%	het.	spec.	medium
1132	21	91	20.88%	4.76%	5.72%	het.	div.	high
1134	42	346	15.03%	2.38%	3.11%	het.	div.	high
1136	4	8	37.50%	25.00%	8.84%	het.	spec.	low
1138	4	5	40.00%	25.00%	8.66%	het.	spec.	low
1139	18	74	37.84%	5.55%	8.41%	het.	div.	medium
1140	4	8	37.50%	25.00%	8.84%	het.	spec.	low
1141	7	14	28.57%	14.29%	6.61%	het.	spec.	low
1142	28	124	20.16%	3.57%	4.09%	het.	div.	high
1143	34	265	17.74%	2.94%	4.18%	het.	div.	high
1144	17	89	21.35%	5.88%	5.74%	het.	div.	high
1145	5	9	33.33%	20.00%	10.89%	het.	spec.	low
1146	13	38	23.68%	7.69%	5.70%	het.	spec.	low
1147	1	2	100.00%	100.00%	0.00%	-	-	low
1150	10	13	15.38%	10.00%	3.52%	hom.	spec.	low
1151	7	15	20.00%	14.28%	4.26%	hom.	spec.	low
1152	17	39	23.08%	5.88%	5.26%	het.	div.	low
1153	5	7	28.57%	20.00%	7.00%	het.	spec.	low
1154	14	63	55.56%	7.14%	13.53%	het.	spec.	medium
1157	24	78	14.10%	4.16%	3.42%	hom.	div.	medium
1158	20	83	15.66%	5.00%	4.85%	het.	div.	medium
1159	1	3	100.00%	100.00%	0.00%	-	-	low
1160	2	3	66.67%	50.00%	16.67%	het.	spec.	low
1162	11	20	15.00%	9.09%	4.17%	hom.	spec.	low
1163	18	43	39.53%	5.56%	8.42%	het.	div.	low
1164	1	1	100.00%	100.00%	0.00%	-	-	low
1165	15	29	13.79%	6.67%	2.94%	hom.	spec.	low
1167	1	2	100.00%	100.00%	0.00%	-	-	low
1169	15	62	24.19%	6.67%	6.28%	het.	spec.	medium
1170	2	2	50.00%	50.00%	0.00%	hom.	spec.	low
1172	4	6	33.33%	25.00%	8.33%	het.	spec.	low
1173	52	578	7.96%	1.92%	2.24%	hom.	div.	high
1174	46	234	12.39%	2.17%	2.35%	hom.	div.	high
1175	43	240	15.00%	2.33%	3.32%	het.	div.	high
1177	8	21	19.05%	12.50%	4.73%	hom.	spec.	low
1178	3	3	33.33%	33.33%	0.00%	hom.	spec.	low
1179	22	70	50.00%	4.55%	9.99%	het.	div.	medium
1181	4	5	40.00%	25.00%	8.66%	het.	spec.	low
1182	19	100	16.00%	5.26%	5.29%	het.	div.	high
1183	4	5	40.00%	25.00%	8.66%	het.	spec.	low
1184	15	28	28.57%	6.66%	6.24%	het.	spec.	low
1185	2	9	77.78%	50.00%	27.78%	het.	spec.	low
1186	23	59	16.95%	4.35%	3.77%	het.	div.	medium
1187	6	10	30.00%	16.67%	9.43%	het.	spec.	low
1188	7	11	27.27%	14.28%	6.62%	het.	spec.	low
1191	17	70	50.00%	5.88%	11.18%	het.	div.	medium
1192	50	679	19.44%	2.00%	3.34%	het.	div.	High

N.id:= node id

#C:= number of clusters

#LE:= number of intra-cluster linkevents

SD:= standard deviation

het.:= heterogeneous

hom.:= homogeneous

div.:= diversified

spec.:= specialized

Table C-37: Case study. Content-based clustering on linkevents. Knowledge profile categorization. Nodes 1193-1275

N.id	#C	#LE	Max(%LE)	Ø(%LE)	SD(%LE)	Knowledge Profile Categorization		
						Homogeneity	Diversification	Activity
1193	17	67	52.24%	5.88%	11.68%	het.	div.	medium
1194	43	643	15.86%	2.33%	3.29%	het.	div.	high
1195	24	110	21.82%	4.17%	4.37%	het.	div.	high
1196	1	3	100.00%	100.00%	0.00%	-	-	low
1198	45	791	13.53%	2.22%	2.89%	hom.	div.	high
1199	6	49	71.43%	16.67%	24.56%	het.	spec.	low
1200	27	131	14.50%	3.70%	3.99%	hom.	div.	high
1201	22	171	68.42%	4.54%	14.11%	het.	div.	high
1202	3	3	33.33%	33.33%	0.00%	hom.	spec.	low
1203	11	35	60.00%	9.09%	16.24%	het.	spec.	low
1206	16	31	16.13%	6.25%	4.34%	hom.	div.	low
1210	6	18	55.56%	16.67%	17.86%	het.	spec.	low
1211	16	76	18.42%	6.25%	5.89%	het.	div.	medium
1212	3	14	71.43%	33.33%	27.56%	het.	spec.	low
1214	11	13	15.38%	9.09%	2.97%	hom.	spec.	low
1215	9	16	25.00%	11.11%	7.67%	het.	spec.	low
1216	7	23	43.48%	14.29%	13.89%	het.	spec.	low
1218	9	27	37.04%	11.11%	9.72%	het.	spec.	low
1221	16	33	15.15%	6.25%	3.30%	hom.	div.	low
1222	11	31	32.26%	9.09%	8.12%	het.	spec.	low
1225	9	35	31.43%	11.11%	8.77%	het.	spec.	low
1226	2	2	50.00%	50.00%	0.00%	hom.	spec.	low
1227	7	14	28.57%	14.29%	7.64%	het.	spec.	low
1229	4	11	63.64%	25.00%	22.62%	het.	spec.	low
1232	28	107	30.43%	3.57%	6.37%	het.	div.	high
1233	1	2	100.00%	100.00%	0.00%	-	-	low
1235	11	16	18.75%	9.09%	4.10%	hom.	spec.	low
1236	6	10	30.00%	16.67%	7.45%	het.	spec.	low
1239	20	48	10.42%	5.00%	3.12%	hom.	div.	low
1240	6	19	52.63%	16.67%	16.20%	het.	spec.	low
1243	1	2	100.00%	100.00%	0.00%	-	-	low
1247	17	49	12.24%	5.88%	3.35%	hom.	div.	low
1248	16	85	41.18%	6.25%	9.30%	het.	div.	medium
1249	2	4	75.00%	50.00%	25.00%	het.	spec.	low
1250	15	42	23.81%	6.67%	5.91%	het.	spec.	low
1252	6	45	77.78%	16.67%	27.42%	het.	spec.	low
1253	8	10	20.00%	12.50%	4.33%	hom.	spec.	low
1254	8	49	71.43%	12.50%	22.34%	het.	spec.	low
1255	12	45	24.44%	8.33%	7.97%	het.	spec.	low
1256	4	6	33.33%	25.00%	8.33%	het.	spec.	low
1257	12	36	27.78%	8.34%	9.00%	het.	spec.	low
1258	12	51	29.41%	8.33%	8.44%	het.	spec.	medium
1260	6	19	52.63%	16.67%	17.03%	het.	spec.	low
1261	36	167	12.57%	2.78%	3.00%	hom.	div.	high
1262	11	29	34.48%	9.09%	8.61%	het.	spec.	low
1263	35	322	16.46%	2.86%	3.62%	het.	div.	high
1264	15	81	24.69%	6.67%	6.56%	het.	spec.	medium
1265	35	157	13.38%	2.86%	3.21%	hom.	div.	high
1266	33	204	25.98%	3.03%	4.65%	het.	div.	high
1269	6	15	40.00%	16.67%	14.27%	het.	spec.	low
1270	2	6	50.00%	50.00%	0.00%	hom.	spec.	low
1271	32	255	29.02%	3.12%	5.49%	het.	div.	high
1272	7	20	50.00%	14.29%	14.98%	het.	spec.	low
1273	18	99	24.24%	5.56%	6.16%	het.	div.	high
1274	9	56	62.50%	11.11%	18.38%	het.	spec.	medium
1275	6	51	68.63%	16.67%	23.42%	het.	spec.	medium

Table C-38: Case study. Content-based clustering on linkevents. Overview of knowledge profile categorization

Homogeneity	Diversification	Activity	#Nodes	Nodes
-	-	low	7	1147, 1159, 1164, 1167, 1196, 1233, 1243
homogeneous	diversified	low	4	1206, 1221, 1239, 1247
homogeneous	diversified	medium	2	1126, 1157
homogeneous	diversified	high	6	1173, 1174, 1198, 1200, 1261, 1265
homogeneous	specialized	low	13	1150, 1151, 1162, 1165, 1170, 1177, 1178, 1202, 1214, 1226, 1235, 1253, 1270
homogeneous	specialized	medium	0	-
homogeneous	specialized	high	0	-
heterogeneous	diversified	low	2	1152, 1163
heterogeneous	diversified	medium	8	1139, 1158, 1179, 1186, 1191, 1193, 1211, 1248
heterogeneous	diversified	high	16	1132, 1134, 1142, 1143, 1144, 1175, 1182, 1192, 1194, 1195, 1201, 1232, 1263, 1266, 1271, 1273
heterogeneous	specialized	low	42	1125, 1127, 1128, 1136, 1138, 1140, 1141, 1145, 1146, 1153, 1160, 1172, 1181, 1183, 1184, 1185, 1187, 1188, 1199, 1203, 1210, 1212, 1215, 1216, 1218, 1222, 1225, 1227, 1229, 1236, 1240, 1249, 1250, 1252, 1254, 1255, 1256, 1257, 1260, 1262, 1269, 1272
heterogeneous	specialized	medium	7	1130, 1154, 1169, 1258, 1264, 1274, 1275
heterogeneous	specialized	high	0	-

C.4.4 Comparison of Clustering Results

See 5.1.5.3 for a detailed description of the metrics for actor-centric and cluster-centric group membership stability.

C.4.4.1 Structural Clustering versus Content-based Clustering on Nodes

The following abbreviations are used in this section:

- C1.id: Cluster from structural clustering
- C2.id: Cluster from content-based clustering on nodes
- GMS1_1: Actor-centric group membership stability 1 of structural clustering
- GMS1_2: Actor-centric group membership stability 1 of content-based clustering on nodes
- GMS2: Actor-centric group membership stability 2 first and second clustering
- $\emptyset(\text{GMS1})$: Average GMS1 for all clusters in clustering (GMS1_1 for structural clustering, GMS1_2 for content-based clustering on nodes)
- $\emptyset(\text{GMS2})$: Average GMS2 for all clusters in clustering

Table C-39: Case study. Cluster comparison. Structural versus content-based clustering on nodes. Cluster-centric group membership stability values

Structural Clustering					Content-based Clustering on Nodes				
C1.id	C2.id	Overlap	$\emptyset(\text{GMS1})$	$\emptyset(\text{GMS2})$	C2.id	C1.id	Overlap	$\emptyset(\text{GMS1})$	$\emptyset(\text{GMS2})$
0	8	2	1	1	0	3	14	0.51	0.568
1	5	5	1	1	0	2	1	0.51	0.568
2	0	1	0.25	0.247	0	6	4	0.51	0.568
2	6	4	0.25	0.247	0	7	1	0.51	0.568
2	2	3	0.25	0.247	1	5	1	0.875	0.691
3	0	14	0.767	0.674	1	7	15	0.875	0.691
3	2	2	0.767	0.674	2	3	2	0.205	0.3
4	2	10	0.477	0.408	2	5	12	0.205	0.3
4	7	8	0.477	0.408	2	2	3	0.205	0.3
5	1	1	0.629	0.294	2	6	14	0.205	0.3
5	6	1	0.629	0.294	2	4	10	0.205	0.3
5	2	12	0.629	0.294	2	7	6	0.205	0.3
5	9	1	0.629	0.294	5	1	5	1	1
6	0	4	0.567	0.333	6	5	1	0.6	0.4
6	2	14	0.567	0.333	6	2	4	0.6	0.4
7	0	1	0.435	0.497	7	7	1	0.778	0.498
7	1	15	0.435	0.497	7	4	8	0.778	0.498
7	2	6	0.435	0.497	8	0	2	1	1
7	7	1	0.435	0.497	9	5	1	0	0
7	9	1	0.435	0.497	9	7	1	0	0

C1.id:= ID of structural cluster

C2.id:= ID of content-based clusters

Table C-40: Case study. Cluster comparison. Structural versus content-based clustering on nodes. Actor-centric group membership stability values

N.id	C1.id	C2.id	GMS1_1	GMS1_2	GMS2	N.id	C1.id	C2.id	GMS1_1	GMS1_2	GMS2
1125	3	0	0.867	0.684	0.765	1198	3	2	0.067	0.022	0.033
1126	7	1	0.609	0.933	0.737	1199	5	2	0.786	0.239	0.367
1127	5	2	0.786	0.239	0.367	1200	6	0	0.167	0.158	0.162
1128	4	7	0.412	0.875	0.56	1201	6	2	0.722	0.283	0.406
1130	6	2	0.722	0.283	0.406	1202	5	1	0	0	0
1132	6	0	0.167	0.158	0.162	1203	5	2	0.786	0.239	0.367
1134	6	2	0.722	0.283	0.406	1206	2	2	0.25	0.043	0.074
1136	6	0	0.167	0.158	0.162	1210	4	7	0.412	0.875	0.56
1138	2	6	0.375	0.75	0.5	1211	4	2	0.529	0.196	0.286
1139	6	2	0.722	0.283	0.406	1212	4	7	0.412	0.875	0.56
1140	3	0	0.867	0.684	0.765	1214	2	6	0.375	0.75	0.5
1141	3	0	0.867	0.684	0.765	1215	7	7	0	0	0
1142	3	0	0.867	0.684	0.765	1216	7	1	0.609	0.933	0.737
1143	4	2	0.529	0.196	0.286	1218	4	7	0.412	0.875	0.56
1144	3	0	0.867	0.684	0.765	1221	2	6	0.375	0.75	0.5
1145	3	2	0.067	0.022	0.033	1222	4	2	0.529	0.196	0.286
1146	7	1	0.609	0.933	0.737	1225	7	2	0.217	0.109	0.145
1147	0	8	1	1	1	1226	7	9	0	0	0
1150	7	1	0.609	0.933	0.737	1227	7	2	0.217	0.109	0.145
1151	7	1	0.609	0.933	0.737	1229	7	2	0.217	0.109	0.145
1152	6	2	0.722	0.283	0.406	1232	5	2	0.786	0.239	0.367
1153	3	0	0.867	0.684	0.765	1233	1	5	1	1	1
1154	3	0	0.867	0.684	0.765	1235	2	0	0	0	0
1157	3	0	0.867	0.684	0.765	1236	7	0	0	0	0
1158	6	2	0.722	0.283	0.406	1238	1	5	1	1	1
1159	1	5	1	1	1	1239	2	2	0.25	0.043	0.074
1160	1	5	1	1	1	1240	4	7	0.412	0.875	0.56
1162	7	1	0.609	0.933	0.737	1243	0	8	1	1	1
1163	6	2	0.722	0.283	0.406	1247	5	2	0.786	0.239	0.367
1164	6	0	0.167	0.158	0.162	1248	5	2	0.786	0.239	0.367
1165	7	1	0.609	0.933	0.737	1249	1	5	1	1	1
1169	7	2	0.217	0.109	0.145	1250	7	1	0.609	0.933	0.737
1170	5	6	0	0	0	1252	5	2	0.786	0.239	0.367
1172	2	2	0.25	0.043	0.074	1253	7	1	0.609	0.933	0.737
1173	6	2	0.722	0.283	0.406	1254	5	2	0.786	0.239	0.367
1174	3	0	0.867	0.684	0.765	1255	4	2	0.529	0.196	0.286
1175	6	2	0.722	0.283	0.406	1256	4	7	0.412	0.875	0.56
1177	6	2	0.722	0.283	0.406	1257	4	2	0.529	0.196	0.286
1178	5	9	0	0	0	1258	4	2	0.529	0.196	0.286
1179	5	2	0.786	0.239	0.367	1260	4	7	0.412	0.875	0.56
1181	7	1	0.609	0.933	0.737	1261	3	0	0.867	0.684	0.765
1182	6	2	0.722	0.283	0.406	1262	4	7	0.412	0.875	0.56
1183	7	2	0.217	0.109	0.145	1263	4	2	0.529	0.196	0.286
1184	7	1	0.609	0.933	0.737	1264	4	2	0.529	0.196	0.286
1186	7	1	0.609	0.933	0.737	1265	3	0	0.867	0.684	0.765
1187	7	2	0.217	0.109	0.145	1266	6	2	0.722	0.283	0.406
1188	2	6	0.375	0.75	0.5	1269	7	1	0.609	0.933	0.737
1191	3	0	0.867	0.684	0.765	1270	7	1	0.609	0.933	0.737
1192	6	2	0.722	0.283	0.406	1271	6	2	0.722	0.283	0.406
1193	3	0	0.867	0.684	0.765	1272	4	2	0.529	0.196	0.286
1194	5	2	0.786	0.239	0.367	1273	4	2	0.529	0.196	0.286
1195	3	0	0.867	0.684	0.765	1274	5	2	0.786	0.239	0.367
1196	7	1	0.609	0.933	0.737	1275	5	2	0.786	0.239	0.367

C.4.4.2 Structural Clustering versus Content-based Clustering on Linkevents

The following abbreviations are used in this section:

- C1.id: Cluster from structural clustering
- C2.id: Cluster from content-based clustering on linkevents
- C1.Member: Cluster members in cluster C1
- C2.Member: Cluster members in cluster C2
- GMS1_1: Actor-centric group membership stability 1 of structural clustering
- GMS1_2: Actor-centric group membership stability 1 of content-based clustering on linkevents
- GMS2: Actor-centric group membership stability 2 first and second clustering
- $\bar{\text{GMS1}}$: Average GMS1 for all clusters in clustering (GMS1_1 for structural clustering, GMS1_2 for content-based clustering on linkevents)
- $\bar{\text{GMS2}}$: Average GMS2 for all clusters in clustering
- S.id: Cluster from structural clustering
- LE.id: Cluster from content-based clustering on nodes

Table C-41: Case study. Cluster comparison. Structural clustering versus linkevents. Cluster-centric group membership stability values. Part I

LE.Cid	S.Cid	ØGMS1	ØGMS2	LE.Cid	S.Cid	ØGMS1	ØGMS2	LE.Cid	S.Cid	ØGMS1	ØGMS2
17	0	100.00%	9.09%	35	3	26.67%	30.77%	30	4	41.18%	37.84%
44	1	25.00%	7.14%	36	3	40.00%	52.17%	34	4	5.88%	5.00%
49	1	75.00%	14.29%	37	3	6.67%	7.41%	38	4	11.76%	9.30%
0	2	12.50%	6.06%	38	3	60.00%	43.90%	39	4	5.88%	10.00%
7	2	12.50%	15.38%	40	3	26.67%	19.51%	40	4	11.76%	9.30%
17	2	12.50%	6.90%	41	3	53.33%	36.36%	41	4	5.88%	4.35%
18	2	50.00%	34.78%	43	3	20.00%	30.00%	44	4	23.53%	19.51%
20	2	12.50%	11.76%	46	3	13.33%	12.12%	48	4	41.18%	45.16%
24	2	37.50%	16.22%	47	3	20.00%	19.35%	49	4	52.94%	32.73%
29	2	12.50%	11.76%	49	3	33.33%	18.87%	50	4	35.29%	30.00%
33	2	12.50%	11.76%	51	3	13.33%	15.38%	51	4	5.88%	7.14%
34	2	37.50%	19.35%	52	3	20.00%	18.75%	52	4	5.88%	5.88%
35	2	12.50%	10.53%	55	3	6.67%	6.25%	55	4	23.53%	23.53%
41	2	25.00%	10.81%	57	3	6.67%	5.26%	56	4	17.65%	17.65%
46	2	50.00%	30.77%	58	3	33.33%	27.78%	57	4	29.41%	25.00%
47	2	12.50%	8.33%	59	3	13.33%	10.81%	58	4	17.65%	15.79%
49	2	50.00%	17.39%	60	3	40.00%	44.44%	59	4	100.00%	87.18%
53	2	12.50%	9.09%	61	3	53.33%	44.44%	60	4	11.76%	13.79%
55	2	12.50%	8.00%	62	3	46.67%	41.18%	61	4	5.88%	5.26%
61	2	12.50%	6.90%	63	3	80.00%	42.86%	62	4	5.88%	5.56%
63	2	50.00%	16.33%	65	3	40.00%	38.71%	63	4	52.94%	31.03%
69	2	25.00%	9.09%	66	3	20.00%	20.00%	64	4	5.88%	10.00%
70	2	50.00%	26.67%	67	3	20.00%	20.69%	66	4	11.76%	12.50%
71	2	12.50%	10.00%	68	3	26.67%	42.11%	67	4	5.88%	6.45%
75	2	12.50%	10.53%	69	3	40.00%	23.53%	69	4	100.00%	64.15%
76	2	12.50%	7.14%	70	3	60.00%	48.65%	70	4	11.76%	10.26%
80	2	62.50%	33.33%	71	3	6.67%	7.41%	73	4	41.18%	25.45%
83	2	37.50%	15.79%	73	3	40.00%	22.64%	74	4	58.82%	45.45%
84	2	12.50%	6.45%	74	3	40.00%	28.57%	78	4	17.65%	16.67%
0	3	40.00%	30.00%	76	3	13.33%	11.43%	80	4	5.88%	5.13%
1	3	60.00%	72.00%	80	3	53.33%	43.24%	82	4	11.76%	16.00%
3	3	26.67%	34.78%	81	3	40.00%	35.29%	83	4	58.82%	42.55%
6	3	6.67%	6.25%	82	3	26.67%	34.78%	85	4	5.88%	6.25%
8	3	20.00%	14.63%	83	3	20.00%	13.33%	6	5	64.29%	58.06%
11	3	46.67%	48.28%	84	3	73.33%	57.89%	8	5	14.29%	10.00%
12	3	33.33%	27.78%	85	3	13.33%	13.33%	9	5	7.14%	8.70%
20	3	13.33%	16.67%	86	3	6.67%	8.00%	10	5	28.57%	24.24%
22	3	53.33%	40.00%	0	4	35.29%	28.57%	11	5	14.29%	14.29%
23	3	13.33%	14.29%	4	4	11.76%	14.81%	13	5	28.57%	21.05%
24	3	53.33%	36.36%	8	4	35.29%	27.91%	14	5	21.43%	20.00%
26	3	66.67%	45.45%	10	4	11.76%	11.11%	15	5	14.29%	10.00%
27	3	6.67%	7.69%	12	4	11.76%	10.53%	16	5	7.14%	8.70%
30	3	40.00%	34.29%	14	4	5.88%	6.06%	22	5	28.57%	20.51%
32	3	6.67%	8.33%	17	4	23.53%	21.05%	24	5	7.14%	4.65%
33	3	26.67%	33.33%	24	4	23.53%	17.39%	25	5	28.57%	32.00%
34	3	60.00%	47.37%	26	4	11.76%	8.70%	26	5	14.29%	9.30%

Table C-42: Case study. Cluster comparison. Structural clustering versus linkevents. Cluster-centric group membership stability values. Part II

LE.Cid	S.Cid	ØGMS1	ØGMS2	LE.Cid	S.Cid	ØGMS1	ØGMS2	LE.Cid	S.Cid	ØGMS1	ØGMS2
30	5	7.14%	5.88%	25	6	16.67%	20.69%	2	7	8.70%	13.33%
38	5	35.71%	25.00%	26	6	38.89%	29.79%	3	7	13.04%	19.35%
40	5	35.71%	25.00%	27	6	27.78%	34.48%	8	7	4.35%	4.08%
41	5	7.14%	4.65%	28	6	33.33%	44.44%	10	7	17.39%	19.05%
42	5	21.43%	28.57%	29	6	38.89%	51.85%	11	7	4.35%	5.41%
44	5	21.43%	15.79%	30	6	5.56%	5.26%	12	7	4.35%	4.55%
46	5	50.00%	43.75%	31	6	61.11%	73.33%	13	7	39.13%	38.30%
50	5	7.14%	5.41%	32	6	38.89%	51.85%	14	7	4.35%	5.13%
51	5	14.29%	16.00%	33	6	11.11%	14.81%	15	7	86.96%	81.63%
53	5	7.14%	7.14%	35	6	11.11%	13.79%	16	7	4.35%	6.25%
54	5	50.00%	53.85%	37	6	50.00%	60.00%	17	7	26.09%	27.27%
55	5	7.14%	6.45%	38	6	11.11%	9.09%	18	7	8.70%	10.53%
56	5	14.29%	12.90%	41	6	55.56%	42.55%	21	7	8.70%	16.00%
57	5	21.43%	16.22%	44	6	33.33%	28.57%	22	7	13.04%	12.50%
61	5	14.29%	11.43%	45	6	61.11%	75.86%	24	7	17.39%	15.38%
62	5	50.00%	42.42%	46	6	5.56%	5.56%	25	7	4.35%	5.88%
63	5	42.86%	21.82%	47	6	55.56%	58.82%	26	7	13.04%	11.54%
66	5	28.57%	27.59%	48	6	33.33%	37.50%	27	7	13.04%	17.65%
67	5	57.14%	57.14%	49	6	38.89%	25.00%	28	7	8.70%	12.50%
69	5	21.43%	12.00%	50	6	38.89%	34.15%	30	7	4.35%	4.65%
71	5	7.14%	7.69%	51	6	16.67%	20.69%	34	7	26.09%	26.09%
73	5	14.29%	7.69%	52	6	16.67%	17.14%	36	7	4.35%	6.45%
76	5	21.43%	17.65%	53	6	5.56%	6.25%	38	7	17.39%	16.33%
78	5	35.71%	30.30%	54	6	22.22%	26.67%	40	7	43.48%	40.82%
83	5	7.14%	4.55%	55	6	22.22%	22.86%	41	7	8.70%	7.69%
0	6	11.11%	9.30%	56	6	50.00%	51.43%	42	7	8.70%	13.33%
2	6	11.11%	16.00%	57	6	33.33%	29.27%	44	7	21.74%	21.28%
4	6	33.33%	42.86%	58	6	50.00%	46.15%	49	7	17.39%	13.11%
5	6	50.00%	64.29%	61	6	27.78%	25.64%	50	7	17.39%	17.39%
6	6	27.78%	28.57%	63	6	5.56%	3.39%	52	7	21.74%	25.00%
7	6	11.11%	17.39%	65	6	38.89%	41.18%	53	7	34.78%	43.24%
8	6	55.56%	45.45%	69	6	16.67%	11.11%	55	7	4.35%	5.00%
9	6	27.78%	37.04%	70	6	16.67%	15.00%	57	7	13.04%	13.04%
10	6	27.78%	27.03%	72	6	5.56%	9.09%	60	7	4.35%	5.71%
12	6	50.00%	46.15%	73	6	44.44%	28.57%	62	7	4.35%	4.76%
13	6	44.44%	38.10%	74	6	22.22%	17.78%	63	7	17.39%	12.50%
14	6	38.89%	41.18%	76	6	33.33%	31.58%	66	7	8.70%	10.53%
15	6	5.56%	4.55%	78	6	27.78%	27.03%	71	7	21.74%	28.57%
16	6	27.78%	37.04%	79	6	66.67%	75.00%	73	7	47.83%	36.07%
17	6	22.22%	20.51%	80	6	22.22%	20.00%	74	7	13.04%	12.00%
18	6	27.78%	30.30%	81	6	27.78%	27.03%	75	7	39.13%	52.94%
19	6	50.00%	66.67%	83	6	27.78%	20.83%	76	7	13.04%	13.95%
20	6	22.22%	29.63%	84	6	38.89%	34.15%	78	7	8.70%	9.52%
22	6	33.33%	27.91%	85	6	50.00%	54.55%	79	7	4.35%	5.41%
23	6	50.00%	58.06%	86	6	44.44%	57.14%	81	7	21.74%	23.81%
24	6	22.22%	17.02%	0	7	21.74%	20.83%	83	7	13.04%	11.32%

Table C-43: Case study. Cluster comparison. Structural clustering versus content-based clustering on linkevents. Actor-centric group membership stability values. Node 1174

N.id	C1.id	C2.id	Overlap	C1.Member	C2.Member	GMS1_1	GMS1_2	GMS2
1174	3	0	6	15	25	40.00%	24.00%	30.00%
1174	3	1	9	15	10	60.00%	90.00%	72.00%
1174	3	3	4	15	8	26.67%	50.00%	34.78%
1174	3	11	7	15	14	46.67%	50.00%	48.28%
1174	3	12	5	15	21	33.33%	23.81%	27.78%
1174	3	22	8	15	25	53.33%	32.00%	40.00%
1174	3	23	2	15	13	13.33%	15.38%	14.29%
1174	3	24	8	15	29	53.33%	27.59%	36.36%
1174	3	26	10	15	29	66.67%	34.48%	45.45%
1174	3	27	1	15	11	6.67%	9.09%	7.69%
1174	3	30	6	15	20	40.00%	30.00%	34.29%
1174	3	34	9	15	23	60.00%	39.13%	47.37%
1174	3	35	4	15	11	26.67%	36.36%	30.77%
1174	3	36	6	15	8	40.00%	75.00%	52.17%
1174	3	37	1	15	12	6.67%	8.33%	7.41%
1174	3	38	9	15	26	60.00%	34.62%	43.90%
1174	3	40	4	15	26	26.67%	15.38%	19.51%
1174	3	41	8	15	29	53.33%	27.59%	36.36%
1174	3	51	2	15	11	13.33%	18.18%	15.38%
1174	3	52	3	15	17	20.00%	17.65%	18.75%
1174	3	55	1	15	17	6.67%	5.88%	6.25%
1174	3	57	1	15	23	6.67%	4.35%	5.26%
1174	3	60	6	15	12	40.00%	50.00%	44.44%
1174	3	61	8	15	21	53.33%	38.10%	44.44%
1174	3	62	7	15	19	46.67%	36.84%	41.18%
1174	3	63	12	15	41	80.00%	29.27%	42.86%
1174	3	65	6	15	16	40.00%	37.50%	38.71%
1174	3	68	4	15	4	26.67%	100.00%	42.11%
1174	3	69	6	15	36	40.00%	16.67%	23.53%
1174	3	70	9	15	22	60.00%	40.91%	48.65%
1174	3	71	1	15	12	6.67%	8.33%	7.41%
1174	3	73	6	15	38	40.00%	15.79%	22.64%
1174	3	74	6	15	27	40.00%	22.22%	28.57%
1174	3	76	2	15	20	13.33%	10.00%	11.43%
1174	3	80	8	15	22	53.33%	36.36%	43.24%
1174	3	81	6	15	19	40.00%	31.58%	35.29%
1174	3	82	4	15	8	26.67%	50.00%	34.78%
1174	3	83	3	15	30	20.00%	10.00%	13.33%
1174	3	84	11	15	23	73.33%	47.83%	57.89%
1174	3	85	2	15	15	13.33%	13.33%	13.33%

Table C-44: Case study. Cluster comparison. Structural clustering versus content-based clustering on linkevents. Actor-centric group membership stability values. Node 1198

N.id	C1.id	C2.id	Overlap	C1.Member	C2.Member	GMS1_1	GMS1_2	GMS2
1198	3	0	6	15	25	40.00%	24.00%	30.00%
1198	3	6	1	15	17	6.67%	5.88%	6.25%
1198	3	11	7	15	14	46.67%	50.00%	48.28%
1198	3	12	5	15	21	33.33%	23.81%	27.78%
1198	3	24	8	15	29	53.33%	27.59%	36.36%
1198	3	26	10	15	29	66.67%	34.48%	45.45%
1198	3	30	6	15	20	40.00%	30.00%	34.29%
1198	3	34	9	15	23	60.00%	39.13%	47.37%
1198	3	35	4	15	11	26.67%	36.36%	30.77%
1198	3	36	6	15	8	40.00%	75.00%	52.17%
1198	3	38	9	15	26	60.00%	34.62%	43.90%
1198	3	40	4	15	26	26.67%	15.38%	19.51%
1198	3	41	8	15	29	53.33%	27.59%	36.36%
1198	3	43	3	15	5	20.00%	60.00%	30.00%
1198	3	49	5	15	38	33.33%	13.16%	18.87%
1198	3	55	1	15	17	6.67%	5.88%	6.25%
1198	3	58	5	15	21	33.33%	23.81%	27.78%
1198	3	59	2	15	22	13.33%	9.09%	10.81%
1198	3	60	6	15	12	40.00%	50.00%	44.44%
1198	3	61	8	15	21	53.33%	38.10%	44.44%
1198	3	62	7	15	19	46.67%	36.84%	41.18%
1198	3	66	3	15	15	20.00%	20.00%	20.00%
1198	3	67	3	15	14	20.00%	21.43%	20.69%
1198	3	68	4	15	4	26.67%	100.00%	42.11%
1198	3	69	6	15	36	40.00%	16.67%	23.53%
1198	3	70	9	15	22	60.00%	40.91%	48.65%
1198	3	71	1	15	12	6.67%	8.33%	7.41%
1198	3	74	6	15	27	40.00%	22.22%	28.57%
1198	3	76	2	15	20	13.33%	10.00%	11.43%
1198	3	80	8	15	22	53.33%	36.36%	43.24%
1198	3	82	4	15	8	26.67%	50.00%	34.78%
1198	3	83	3	15	30	20.00%	10.00%	13.33%
1198	3	84	11	15	23	73.33%	47.83%	57.89%
1198	3	85	2	15	15	13.33%	13.33%	13.33%

Table C-45: Case study. Cluster comparison. Structural clustering versus content-based clustering on linkevents. Actor-centric group membership stability values. Node 1264

N.id	C1.id	C2.id	Overlap	C1.Member	C2.Member	GMS1_1	GMS1_2	GMS2
1264	4	0	6	17	25	35.29%	24.00%	28.57%
1264	4	14	1	17	16	5.88%	6.25%	6.06%
1264	4	24	4	17	29	23.53%	13.79%	17.39%
1264	4	30	7	17	20	41.18%	35.00%	37.84%
1264	4	55	4	17	17	23.53%	23.53%	23.53%
1264	4	58	3	17	21	17.65%	14.29%	15.79%
1264	4	59	17	17	22	100.00%	77.27%	87.18%
1264	4	60	2	17	12	11.76%	16.67%	13.79%
1264	4	64	1	17	3	5.88%	33.33%	10.00%
1264	4	69	17	17	36	100.00%	47.22%	64.15%
1264	4	70	2	17	22	11.76%	9.09%	10.26%
1264	4	74	10	17	27	58.82%	37.04%	45.45%
1264	4	78	3	17	19	17.65%	15.79%	16.67%
1264	4	82	2	17	8	11.76%	25.00%	16.00%
1264	4	83	10	17	30	58.82%	33.33%	42.55%

C.4.4.3 Content-based Clustering on Nodes versus Content-based Clustering on Linkevents

The following abbreviations are used in this section:

- C1.id: Cluster from content-based clustering on nodes
- C2.id: Cluster from content-based clustering on linkevents
- C1.Member: Cluster members in cluster C1
- C2.Member: Cluster members in cluster C2
- GMS1_1: Actor-centric group membership stability 1 of content-based clustering on nodes
- GMS1_2: Actor-centric group membership stability 1 of content-based clustering on linkevents
- GMS2: Actor-centric group membership stability 2 first and second clustering
- $\bar{\text{GMS1}}$: Average GMS1 for all clusters in clustering (GMS1_1 for content-based clustering on nodes, GMS1_2 for content-based clustering on linkevents)
- $\bar{\text{GMS2}}$: Average GMS2 for all clusters in clustering
- N.id: Cluster from content-based clustering on linkevents
- LE.id: Cluster from content-based clustering on nodes

Table C-46: Case study. Cluster comparison. Content-based clustering on nodes versus linkevents. Cluster-centric group membership stability values. Part I

LE.Cid	N.Cid	ØGMS1	ØGMS2	LE.Cid	N.Cid	ØGMS1	ØGMS2
0	0	24.00%	27.91%	65	0	50.00%	47.06%
1	0	100.00%	71.43%	66	0	13.33%	12.12%
2	0	14.29%	8.00%	67	0	14.29%	12.50%
3	0	50.00%	30.77%	68	0	75.00%	27.27%
5	0	10.00%	7.14%	69	0	19.44%	25.93%
8	0	15.38%	18.18%	70	0	36.36%	40.00%
11	0	42.86%	37.50%	73	0	18.42%	25.00%
12	0	23.81%	25.64%	74	0	18.52%	22.22%
13	0	4.17%	4.76%	76	0	10.00%	10.53%
15	0	3.85%	4.55%	80	0	36.36%	40.00%
18	0	6.67%	6.06%	81	0	42.11%	43.24%
20	0	33.33%	22.22%	82	0	37.50%	23.08%
22	0	28.00%	32.56%	83	0	13.33%	16.67%
23	0	38.46%	32.26%	84	0	47.83%	53.66%
24	0	31.03%	38.30%	85	0	13.33%	12.12%
26	0	27.59%	34.04%	86	0	10.00%	7.14%
27	0	18.18%	13.79%	0	1	68.00%	39.08%
30	0	25.00%	26.32%	2	1	71.43%	14.49%
32	0	11.11%	7.41%	3	1	37.50%	8.57%
33	0	55.56%	37.04%	4	1	90.00%	25.00%
34	0	43.48%	48.78%	5	1	80.00%	22.22%
35	0	27.27%	20.69%	6	1	94.12%	40.51%
36	0	62.50%	38.46%	7	1	60.00%	8.96%
37	0	16.67%	13.33%	8	1	73.08%	43.18%
38	0	30.77%	36.36%	9	1	100.00%	25.35%
40	0	11.54%	13.64%	10	1	84.21%	39.51%
41	0	27.59%	34.04%	11	1	50.00%	18.42%
43	0	40.00%	17.39%	12	1	66.67%	33.73%
44	0	4.17%	4.76%	13	1	83.33%	46.51%
45	0	9.09%	6.90%	14	1	100.00%	41.03%
46	0	11.11%	11.11%	15	1	84.62%	50.00%
47	0	18.75%	17.65%	16	1	100.00%	25.35%
49	0	13.16%	17.86%	17	1	66.67%	33.73%
51	0	18.18%	13.79%	18	1	80.00%	31.17%
52	0	17.65%	17.14%	19	1	100.00%	25.35%
56	0	5.88%	5.71%	20	1	55.56%	14.08%
57	0	17.39%	19.51%	21	1	50.00%	3.13%
58	0	23.81%	25.64%	22	1	64.00%	36.78%
59	0	4.55%	5.00%	23	1	53.85%	18.67%
60	0	41.67%	33.33%	24	1	58.62%	37.36%
61	0	33.33%	35.90%	25	1	81.82%	24.66%
62	0	31.58%	32.43%	26	1	65.52%	41.76%
63	0	31.71%	44.07%	27	1	72.73%	21.92%

Table C-47: Case study. Cluster comparison. Content-based clustering on nodes versus linkevents. Cluster-centric group membership stability values. Part II

LE.Cid	N.Cid	ØGMS1	ØGMS2	LE.Cid	N.Cid	ØGMS1	ØGMS2
28	1	100.00%	25.35%	72	1	100.00%	12.12%
29	1	77.78%	19.72%	73	1	76.32%	58.00%
30	1	60.00%	29.27%	74	1	66.67%	40.45%
31	1	91.67%	29.73%	75	1	72.73%	21.92%
32	1	77.78%	19.72%	76	1	80.00%	39.02%
33	1	33.33%	8.45%	77	1	100.00%	3.17%
34	1	47.83%	25.88%	78	1	94.74%	44.44%
35	1	63.64%	19.18%	79	1	92.86%	34.21%
36	1	25.00%	5.71%	80	1	50.00%	26.19%
37	1	75.00%	24.32%	81	1	47.37%	22.22%
38	1	57.69%	34.09%	82	1	50.00%	11.43%
39	1	100.00%	9.23%	83	1	70.00%	45.65%
40	1	76.92%	45.45%	84	1	43.48%	23.53%
41	1	65.52%	41.76%	85	1	66.67%	25.97%
42	1	100.00%	20.29%	86	1	80.00%	22.22%
43	1	40.00%	5.97%	44	4	4.17%	7.14%
44	1	79.17%	44.19%	49	4	7.89%	14.29%
45	1	81.82%	24.66%	7	5	20.00%	20.00%
46	1	72.22%	32.50%	17	5	4.76%	7.69%
47	1	68.75%	28.21%	29	5	11.11%	14.29%
48	1	92.86%	34.21%	40	5	3.85%	6.45%
49	1	60.53%	46.00%	46	5	5.56%	8.70%
50	1	86.96%	47.06%	49	5	2.63%	4.65%
51	1	72.73%	21.92%	53	5	7.14%	10.53%
52	1	58.82%	25.32%	63	5	4.88%	8.70%
53	1	85.71%	31.58%	70	5	9.09%	14.81%
54	1	100.00%	32.43%	75	5	18.18%	25.00%
55	1	88.24%	37.97%	80	5	4.55%	7.41%
56	1	76.47%	32.91%	83	5	3.33%	5.71%
57	1	43.48%	23.53%	8	6	3.85%	5.88%
58	1	66.67%	33.73%	10	6	5.26%	7.41%
59	1	54.55%	28.57%	17	6	4.76%	6.90%
60	1	50.00%	16.22%	30	6	5.00%	7.14%
61	1	57.14%	28.92%	38	6	3.85%	5.88%
62	1	57.89%	27.16%	49	6	5.26%	8.70%
63	1	53.66%	42.72%	57	6	26.09%	38.71%
64	1	100.00%	9.23%	59	6	31.82%	46.67%
65	1	43.75%	17.95%	63	6	2.44%	4.08%
66	1	66.67%	25.97%	69	6	22.22%	36.36%
67	1	78.57%	28.95%	74	6	7.41%	11.43%
69	1	50.00%	36.73%	83	6	3.33%	5.26%
70	1	45.45%	23.81%	17	7	4.76%	9.09%
71	1	83.33%	27.03%	13	8	4.17%	8.00%

Table C-48: Case study. Cluster comparison. Content-based clustering on nodes versus linkevents. Actor-centric group membership stability values. Node 1174

N.id	C1.id	C2.id	Overlap	C1.Member	C2.Member	GMS1_1	GMS1_2	GMS2
1174	0	63	13	18	41	72.22%	31.71%	44.07%
1174	0	65	8	18	16	44.44%	50.00%	47.06%
1174	0	68	3	18	4	16.67%	75.00%	27.27%
1174	0	69	7	18	36	38.89%	19.44%	25.93%
1174	0	70	8	18	22	44.44%	36.36%	40.00%
1174	0	73	7	18	38	38.89%	18.42%	25.00%
1174	0	74	5	18	27	27.78%	18.52%	22.22%
1174	0	76	2	18	20	11.11%	10.00%	10.53%
1174	0	80	8	18	22	44.44%	36.36%	40.00%
1174	0	81	8	18	19	44.44%	42.11%	43.24%
1174	0	82	3	18	8	16.67%	37.50%	23.08%
1174	0	83	4	18	30	22.22%	13.33%	16.67%
1174	0	84	11	18	23	61.11%	47.83%	53.66%
1174	0	85	2	18	15	11.11%	13.33%	12.12%

Table C-49: Case study. Cluster comparison. Content-based clustering on nodes versus linkevents. Actor-centric group membership stability values. Node 1264

N.id	C1.id	C2.id	Overlap	C1.Member	C2.Member	GMS1_1	GMS1_2	GMS2
1264	2	0	17	62	25	27.42%	68.00%	39.08%
1264	2	14	16	62	16	25.81%	100.00%	41.03%
1264	2	24	17	62	29	27.42%	58.62%	37.36%
1264	2	30	12	62	20	19.35%	60.00%	29.27%
1264	2	55	15	62	17	24.19%	88.24%	37.97%
1264	2	58	14	62	21	22.58%	66.67%	33.73%
1264	2	59	12	62	22	19.35%	54.55%	28.57%
1264	2	60	6	62	12	9.68%	50.00%	16.22%
1264	2	64	3	62	3	4.84%	100.00%	9.23%
1264	2	69	18	62	36	29.03%	50.00%	36.73%
1264	2	70	10	62	22	16.13%	45.45%	23.81%
1264	2	74	18	62	27	29.03%	66.67%	40.45%
1264	2	78	18	62	19	29.03%	94.74%	44.44%
1264	2	82	4	62	8	6.45%	50.00%	11.43%
1264	2	83	21	62	30	33.87%	70.00%	45.65%

Table C-50: Case study. Cluster comparison. Content-based clustering on nodes versus linkevents. Actor-centric group membership stability values. Node 1198

N.id	C1.id	C2.id	Overlap	C1.Member	C2.Member	GMS1_1	GMS1_2	GMS2
1198	2	0	17	62	25	27.42%	68.00%	39.08%
1198	2	2	5	62	7	8.06%	71.43%	14.49%
1198	2	4	9	62	10	14.52%	90.00%	25.00%
1198	2	6	16	62	17	25.81%	94.12%	40.51%
1198	2	9	9	62	9	14.52%	100.00%	25.35%
1198	2	11	7	62	14	11.29%	50.00%	18.42%
1198	2	12	14	62	21	22.58%	66.67%	33.73%
1198	2	14	16	62	16	25.81%	100.00%	41.03%
1198	2	24	17	62	29	27.42%	58.62%	37.36%
1198	2	25	9	62	11	14.52%	81.82%	24.66%
1198	2	26	19	62	29	30.65%	65.52%	41.76%
1198	2	30	12	62	20	19.35%	60.00%	29.27%
1198	2	34	11	62	23	17.74%	47.83%	25.88%
1198	2	35	7	62	11	11.29%	63.64%	19.18%
1198	2	36	2	62	8	3.23%	25.00%	5.71%
1198	2	38	15	62	26	24.19%	57.69%	34.09%
1198	2	39	3	62	3	4.84%	100.00%	9.23%
1198	2	40	20	62	26	32.26%	76.92%	45.45%
1198	2	41	19	62	29	30.65%	65.52%	41.76%
1198	2	43	2	62	5	3.23%	40.00%	5.97%
1198	2	49	23	62	38	37.10%	60.53%	46.00%
1198	2	55	15	62	17	24.19%	88.24%	37.97%
1198	2	56	13	62	17	20.97%	76.47%	32.91%
1198	2	58	14	62	21	22.58%	66.67%	33.73%
1198	2	59	12	62	22	19.35%	54.55%	28.57%
1198	2	60	6	62	12	9.68%	50.00%	16.22%
1198	2	61	12	62	21	19.35%	57.14%	28.92%
1198	2	62	11	62	19	17.74%	57.89%	27.16%
1198	2	64	3	62	3	4.84%	100.00%	9.23%
1198	2	66	10	62	15	16.13%	66.67%	25.97%
1198	2	67	11	62	14	17.74%	78.57%	28.95%
1198	2	69	18	62	36	29.03%	50.00%	36.73%
1198	2	70	10	62	22	16.13%	45.45%	23.81%
1198	2	71	10	62	12	16.13%	83.33%	27.03%
1198	2	72	4	62	4	6.45%	100.00%	12.12%
1198	2	74	18	62	27	29.03%	66.67%	40.45%
1198	2	76	16	62	20	25.81%	80.00%	39.02%
1198	2	77	1	62	1	1.61%	100.00%	3.17%
1198	2	78	18	62	19	29.03%	94.74%	44.44%
1198	2	80	11	62	22	17.74%	50.00%	26.19%
1198	2	82	4	62	8	6.45%	50.00%	11.43%
1198	2	83	21	62	30	33.87%	70.00%	45.65%
1198	2	84	10	62	23	16.13%	43.48%	23.53%
1198	2	85	10	62	15	16.13%	66.67%	25.97%

Literature

- Aalbers, R., W. Dolfsma, et al. (2004). "On and Off The Beaten Path: How Individuals Broker Knowledge Through Formal and Informal Networks." ERIM Report Series Reference No. ERS-2004-066-LIS/ORG.
- Abecker, A., K. Hinkelmann, et al. (2002). "Integrationspotenziale für Geschäftsprozesse und Wissensmanagement." *Geschäftsprozessorientiertes Wissensmanagement*. A. Abecker, K. Hinkelmann, H. Maus and H.-J. Müller. Berlin Heidelberg, Springer.
- Abney, S. (1991). "Parsing by Chunks." *Principle-Based Parsing: Computation and Psycholinguistics*. R. Berwick, S. Abney and C. Tenny. Boston, Kluwer Academic Publishers: 257-278.
- Adamic, L. A., R. M. Lukose, et al. (2001). "Search in Power-law Networks." *Physical Review E* 64: 046135.
- Aier, S. and M. Schoenherr (2007). "Integrating an Enterprise Architecture Using Domain Clustering." *Journal Of Enterprise Architecture* 3(4): 25-32.
- Al Hasan, M., V. Chaojia, et al. (2009). "Robust Partitional Clustering by Outlier and Density Insensitive Seeding." *Pattern Recognition Letters* 30(11): 994-1002.
- Alagar, V. S. (1976). "The Distribution of the Distance Between Random Points." *Journal of Applied Probability* 13: 558-566.
- Albert, R. and A.-L. Barabási (2002). "Statistical Mechanics of Complex Networks." *Reviews of Modern Physics* 74: 47-97.
- Aldenderfer, M. S. and R. K. Blashfield (1984). *Cluster Analysis*. Newbury Park, Sage Publications.
- Allen, L. E. (1962). "Some Uses of Symbolic Logic in Law Practice." *M.U.L.L.* 119: 120.
- Allen, L. E. and M. E. Caldwell (1963). "Modern Logic and Judicial Decision Making: A Sketch of One View." *Jurimetrics*. H. W. Baade. New York, USA, Basic Books Inc.
- Allen, T. (1977). *Managing the Flow of Technology*. Cambridge, MA, MIT Press.
- Anderberg, M. R. (1973). *Cluster Analysis for Applications*. New York, Academic Press.
- Anklam, P. (2002). "Knowledge Management: The Collaboration Thread." *Bulleting of the American Society for Information Science and Technology* 28(6 (August/September)).
- Anklam, P. (2005a). "Masterclass: Social-network Analysis (Part I)." *Inside Knowledge* 8(9): 30-33.
- Anklam, P. (2005b). "Masterclass: Social-network Analysis (Part II)." *Inside Knowledge* 8(10): 30-33.
- Anklam, P. (2005c). "Masterclass: Social-network Analysis (Part IV)." *Inside Knowledge* 9(2): 32-35.
- Anklam, P. and A. Wolfberg (2006). "Creating Networks at the Defense Intelligence Agency." *Knowledge Management Review* 9(1): 10-15.
- Arabie, P. and L. J. Hubert (1994). "Cluster Analysis in Marketing Research." *Advanced methods in marketing research*. R. P. Bagozzi. Oxford, Blackwell: 160-189.

- Backer, E. (1978). *Cluster Analysis for Optimal Decomposition of Induced Fuzzy Sets*. Delft, Delft university Press.
- Backer, E. and A. K. Jain (1981). "A Clustering Performance Measure Based on Fuzzy Set Decomposition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI 3: 66-75.
- Bailey, T. A. and R. C. Dubes (1982). "Cluster Validity Profiles." *Pattern Recognition* 15: 61-83.
- Baker, F. B. (1974). "Stability of Two Hierarchical Grouping Techniques - Case 1. Sensitivity to Data Errors." *Journal of the American Statistical Association* 69: 440-445.
- Baker, F. B. and L. J. Hubert (1976). "A Graph Theoretic Approach to Goodness-of-fit in Complete Link Hierarchical Clustering." *Journal of the American Statistical Association* 71: 870-878.
- Ball, G. H. and D. J. Hall (1964). "Some Fundamental Concepts and Synthesis Procedures for Pattern Recognition Preprocessors." *International Conference of Microwaves, Circuit Theory, and Information Theory*. Tokyo.
- Ball, G. H. and D. J. Hall (1967). "A Clustering Technique for Summarizing Multivariate Data." *Behavioural Science* 12: 153-155.
- Barabási, A.-L. (2003). *Linked: How Everything is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*, Plume.
- Barabási, A.-L. and R. Albert (1999). "Emergence of Scaling in Random Networks." *Science* 286: 509-512.
- Barabási, A.-L. and E. Bonabeau (2003). "Scale-free Networks." *Scientific American* 288: 60-69.
- Barnes, J. A. (1954). "Class and Committees in a Norwegian Island Parish." *Human Relations* 7: 39-58.
- Bartell, B. T. (1994). "Optimizing Ranking Functions: A Connectionist Approach to Adaptive Information Retrieval." University of California. San Diego, La Jolla, CA. PhD.
- Bartell, B. T., G. W. Cottrell, et al. (1998). "Optimizing Similarity Using Multi-Query Relevance Feedback." *Journal of the American Society for Information Science* 49(8): 742-761.
- Barzilay, R. and M. Elhadad (1997). "Using Lexical Chains for Text Summarization." *Workshop on Intelligent Scalable Text Summarization*: 10-17.
- Batagelj, V., A. Ferligoj, et al. (1998). "Fitting Pre-specified Blockmodels." *Data Science, Classification and Related Methods*. C. Hayashi, N. Ohsumi, K. Yajima et al. Tokyo, Springer Verlag: 199-206.
- Batagelj, V. and A. Mrvar (1998). "Pajek - Program for Large Network Analysis." *Connections* 21(2): 47-57.
- Battiston, S. and M. Catanzaro (2004). "Statistical Properties of Corporate Board and Director Networks." *European Physics Journal B* 38(345-352).
- Baxter, M. J. (1994). *Exploratory Multivariate Analysis in Archaeology*. Edinburgh, Edinburgh University Press.
- Bayne, C. K., J. J. Beauchamp, et al. (1980). "Monte Carlo Comparisons of Selected Clustering Procedures." *Pattern Recognition* 12: 51-62.

- Beale, E. M. L. (1969a). *Cluster Analysis*. London, Scientific Control System.
- Beale, E. M. L. (1969b). "Euclidean Cluster Analysis." *Bulletin of the International Statistical Institute: Proceeding of the 37th Session (London)*, Book 2, London, Voorburg, Netherlands: ISI.
- Beesley, K. (1998). "Language Identifier: A Computer Program for Automatic Natural-Language Identification of On-line Text." *Languages at Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association*: 47-54.
- Beesley, K. and L. Karttunen (2003). *Finite State Morphology*. Stanford, California, CSLI Publications.
- Bekkerman, R., A. McCallum, et al. (2004). "Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora." UMass CIIR Technical Report IR-418. Amherst, University of Massachusetts.
- Belbin, L. (1987). "The Use of Non-hierarchical Allocation Methods for Clustering Large Sets of Data." *Australian Computer Journal* 19: 32-41.
- Belbin, R. M. (1993). *Team Roles at Work*. Oxford, Butterworth-Heinemann.
- Bellman, R. E., R. Kalaba, et al. (1966). "Abstraction and Pattern Classification." *Journal of Mathematical Analysis and Applications* 13: 1-7.
- Bender-deMoll, S. and D. A. McFarland (2006). "The Art and Science of Dynamic Network Visualization " *Journal of Social Structure* 7(2).
- Berger, P. and T. Luckman (1966). *The Social Construction of Reality*. New York, Anchor Books.
- Berry, M. W., S. T. Dumais, et al. (1995). "Using Linear Algebra for Intelligent Information Retrieval." *SIAM Review* 37(4): 573-95.
- Berry, M. W. and P. Young (1995). "Using Latent Semantic Indexing for Multilanguage Information Retrieval." *Computers and the Humanities* 29(6): 413-429.
- Bertin, J. (1967/1983). *Semiology of Graphics: Diagrams, Networks, Maps* (Translation of "Sémiologie Graphique. Les Diagrammes, les Réseaux, les Cartes (1967)" by William J. Berg). Madison, Wisconsin, University of Wisconsin Press.
- Bezdek, J. C. (1974). "Cluster Validity with Fuzzy Sets." *Journal of Cybernetics* 3: 58-72.
- Bikson, T. K. and J. D. Eveland (1990). "The Interplay of Work Group Structures and Computer Support." *Intellectual teamwork: Social and technological foundations of cooperative work*. J. Galagher, R. Kraut and C. Egidio. Norwood, NJ: Erlbaum: 243-290.
- Bird, S., E. Klein, et al. (2007). "Introduction to Natural Language Processing (Draft), University of Pennsylvania.
- Blanco, R. and A. Barreiro (2007). "Boosting Static Pruning of Inverted Files." *Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM Press.
- Blashfield, R. K. (1976). "Mixture Model Tests of Cluster Analysis. Accuracy of Four Agglomerative Hierarchical Clustering Methods." *Psychological Bulletin* 83: 377-385.
- Blashfield, R. K. and M. S. Aldenderfer (1978). "Computer Programs for Performing Iterative Partitioning Cluster Analysis." *Applied Psychological Measurement* 2: 533-541.

- Blei, D. M., A. Y. Ng, et al. (2003). "Latent Dirichlet Allocation." *Journal of Machine Learning Research* 3: 993–1022.
- Bobrik, A. and M. Trier (2009). "Content-based Community Detection in Social Corpora." *Wirtschaftsinformatik* 1: 295-304.
- Bonacich, P. (1978). "Using Boolean Algebra to Analyze Overlapping Memberships." *Sociological Methodology*: 101-115.
- Bonacich, P. (1991). "Simultaneous Group and Individual Centralities." *Social Networks* 13: 155-168.
- Bonacich, P. and P. Lloyd (2001). "Eigenvector-like Measures of Centrality for Asymmetric Relations." *Social Networks* 23(3): 191-201.
- Booth, T. L. (1969). "Probabilistic Representation of Formal Languages." 10th Annual Symposium on Switching and Automata Theory
- Borgatti, S. P. (2002). "Basic Social Network Concepts." Presentation given at the 2002 Academy of Management Meeting. Denver, CO, USA.
- Borgatti, S. P. (2009). "2-Mode Concepts in Social Network Analysis." *Encyclopedia of Complexity and System Science* (Forthcoming).
- Borgatti, S. P. and R. Cross (2003). "A Relational View of Information Seeking and Learning in Social Networks." *Management Science* 49(4): 432–445.
- Borgatti, S. P. and M. G. Everett (1997). "Network Analysis of 2-Mode Data." *Social networks* 19: 243-269.
- Borgatti, S. P. and M. G. Everett (1999). "Models of Core/Periphery Structures." *Social Networks* 21(4): 375-395.
- Borgatti, S. P. and M. G. Everett (2006). "A Graph-theoretic Perspective on Centrality." *Social Networks* 28(4): 466-484.
- Borgatti, S. P., M. G. Everett, et al. (1992). "UCINET IV, Network Analysis Software, Columbia: Analytic Technologies.
- Borgatti, S. P., C. Jones, et al. (1998). "Network Measures of Social Capital." *Connections* 21(2): 27-36.
- Borgatti, S. P. and J.-L. Molina (2005). "Toward Ethical Guidelines for Network Research in Organizations." *Social Networks* 27(2): 107-117.
- Bouma, G. and G. van Nourd (1993). "Head-driven Parsing for Lexicalist Grammars: Experimental Results." Proceedings of the sixth Conference on European Chapter of the Association for Computational Linguistics Morristown, NJ, USA, Association for Computational Linguistics.
- Bourdieu, P. and L. J. D. Wacquant (1992). *An Invitation to Reflexive Sociology*. Chicago, IL, University of Chicago Press.
- Bourne, C. P. and D. F. Ford (1961). "A Study of Methods for Systematically Abbreviating English Words and Names." *Journal of the ACM (JACM)* 8(4).
- Brandeau, M. L. and S. S. Chiu (1988). "Parametric Facility Location in a Tree Network with an L_p Norm Cost Function." *Transportation Science* 22(59-69).
- Breiger, R. L. (1974). "The Duality of Persons and Groups." *Social Forces* 53(2): 181-190.
- Breiger, R. L., Ed. (1990). *Social Mobility and Social Structure*. Cambridge, England, Cambridge University Press.

- Brill, E. (1995). "Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging." Proceedings of the Third Workshop on Very Large Corpora, Somerset, New Jersey : Association for Computational Linguistics.
- Brill, E. and R. C. Moore (2000). "An Improved Error Model for Noisy Channel Spelling Correction." Annual Meeting of the Association for Computational Linguistics.
- Brown, P. F., S. A. Della Pietra, et al. (1991). "Word-sense Disambiguation Using Statistical Methods." Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics 29: 264-270.
- Brunnert, J., O. Alonso, et al. (2007). "Enterprise People and Skill Discovery Using Tolerant Retrieval and Visualization." Proceedings of the 29th European Conference on IR Research Rome, Italy Springer-Verlag.
- Büttcher, S. and C. L. A. Clarke (2006). "A Document-centric Approach to Static Index Pruning in Text Retrieval Systems." Conference on Information and Knowledge Management, ACM Press.
- Burt, R. S. (1982). *Toward a Structural Theory of Action*. New York, Academic Press.
- Burt, R. S. (1992). *Structural Holes*. New York, Cambridge University Press.
- Burt, R. S. (2001). "Structural Holes versus Network Closure as Social Capital." *Social Capital: Theory and Research*. N. Lin, K. Cook and R. S. Burt, Aldine de Gruyter: 31-56.
- Burt, R. S. (2002). "The Social Capital of Structural Holes." *The New Economic Sociology: Developments in an Emerging Field*. M. F. Guillén, R. Collins, P. England and M. Meyer. New York, Russell Sage Foundation Publications: 148-190.
- Burt, R. S. (2004). "Structural Holes and Good Ideas." *American Journal of Sociology* 100(2): 339-399.
- Calinski, R. B. and J. Harabasz (1974). "A Dendrite Method or Cluster Analysis." *Communications in Statistics - Theory and Methods* 3(1): 1-27.
- Card, S. K., J. D. Mackinlay, et al. (1999). *Information Visualization: Using Vision to Think*. San Francisco, Morgan Kaufmann.
- Carley, K. M. (2002). "Smart Agents and Organizations of the Future." *The Handbook of New Media*. Edited by Leah Lievrouw and Sonia Livingstone. L. Lievrouw and S. Livingstone. Thousand Oaks, CA, Sage: 206-220.
- Carley, K. M. (2003). "Dynamic Network Analysis." *Dynamic Social Network Modelling and Analysis: Workshop Summary and Papers*. R. L. Breiger, K. Carley and P. Pattison.
- Carmichael, J. W., J. A. George, et al. (1968). "Finding Natural Clusters." *Systematic Zoology* 17: 144-150.
- Carmichael, J. W. and P. H. A. Sneath (1969). "Taxometric Maps." *Systematic Zoology* 18(4): 402-415.
- Carstensen, K.-U., C. Ebert, et al., Eds. (1986). *Computerlinguistik und Sprachtechnologie: Eine Einführung*, Spektrum Akademischer Verlag.
- Castellanos, M. (2003). "HotMiner: Discovering Hot Topics from Dirty Text." *Survey of text mining: Clustering, Classification, and Retrieval*. M. W. Berry, Springer.
- Cavnar, W. B. and M. J. Trenkle (1994). "N-gram-based Text Categorization." Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval.

- Chambers, J. M. and B. Kleiner (1982). "Graphical Techniques for Multivariate Data and for Clustering." *Handbook for Statistics*. New York, North-Holland. 2: 206-244.
- Chantree, F., A. Willis, et al. (2006). *Detecting Dangerous Coordination Ambiguities Using Word Distribution*, Springer.
- Chapanoud, A., M. S. Krishnamoorthy, et al. (2005). "Graph Theoretic and Spectral Analysis of Enron Email Data " *Computational & Mathematical Organization Theory* 11(3): 265-281.
- Charniak, E. (1997). "Statistical Parsing with a Context-free Grammar and Word Statistics." *Proceedings of the 14th National Conference on Artificial Intelligence*, Cambridge, MA, AAAI Press.
- Charniak, E. (2000). "A Maximum-Entropy-Inspired Parser." *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, Seattle.
- Charniak, E., C. Hendrickson, et al. (1993). "Equations for Part-of-Speech Tagging." *Proceedings of the 11th National Conference on Artificial Intelligence*, Menlo Park, CA.
- Chen, H., R. Gnanadesikan, et al. (1974). "Statistical Methods for Grouping Cohesion." *Sankhya Ser. B*. 36: 1-28.
- Chomsky, N. (1956). "Three Models for the Description of Language." *Information Theory, IEEE Transactions* 2(3): 113-124.
- Choudhury, M., R. Saraf, et al. (2008). "Investigation and Modeling of the Structure of Texting Language." *International Journal on Document Analysis and Recognition*.
- Church, K. W. (1988). "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text." *ANLP* 2: 136-143.
- Chye, K. H., T. W. Chin, et al. (2004). "Credit Scoring Using Data Mining Techniques." *Singapore Management Review*.
- Clauset, A., M. E. J. Newman, et al. (2004). "Finding Community Structure in Very Large Networks." *Physical Review E* 70: 1-6.
- Cleverdon, C. W., J. Mills, et al. (1966). *Factors Determining the Performance of Indexing Systems*. Cranfield, England.
- Cliff, A. D., P. Haggett, et al. (1995). "The Application of Multidimensional Scaling Methods to Epidemiological Data." *Statistical Methods in Medical Research* 4(102-123).
- Clifford, H. T. and W. Stephenson (1975). *An Introduction to Numerical Classification*. New York, Academic Press.
- Cocke, J. and J. T. Schwartz (1970). "Programming Languages and their Compilers." *Preliminary Notes*. New York, Courant Institute of Mathematical Sciences of New York University.
- Cohen, S. and S. L. Syme (1985). *Social Support and Health*. Orlando, FL, Academic Press.
- Cohen, W. W., R. E. Schapire, et al. (1998). "Learning to Order Things." *Neural Information Processing Systems Conference*, MIT Press.
- Coleman, J. S. (1973). "Loss of Power." *American Sociological Review* 38(1): 1-17.
- Coleman, J. S. (1990). *Foundations of Social Theory*. Cambridge, MA, Harvard University Press.

- Collins, M. (1997). "Three Generative, Lexicalised Models for Statistical Parsing." Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid.
- Commetrix. (2008). "Commetrix - Dynamic Network Visualization Software." 2010-10-17, from <http://www.commetrix.de>.
- Conover, W. J., B. T. R., et al. (1979). "On a Method of Detecting Clusters of Possible Uranium Deposits." *Technometrics* 21: 277-282.
- Contractor, N. S., S. Wasserman, et al. (2006). "Testing Multitheoretical, Multilevel Hypotheses About Organizational Networks: An Analytic Framework and Empirical Example." *Academy of Management Review* 31(3): 681-703.
- Cooper, W. S. (1971). "A Definition of Relevance for Information Retrieval." *Information Storage and Retrieval* 7(1): 19-37.
- Cooper, W. S., A. Chen, et al. (1994). "Full Text Retrieval Based on Probabilistic Equations with Coefficients Fitted by Logistic Regression." *Text Retrieval Conference*.
- Corley, C. D., D. J. Cook, et al. (2010). "Text and Structural Data Mining of Influenza Mentions in Web and Social Media." *International Journal of Environmental Research and Public Health* 7.
- Cormack, R. M. (1971). "A Review of Classification." *Journal of the Royal Statistic Society, Ser. A. (Statistics in Society)* 134: 321-367.
- Coussement, K. and D. Van den Poel (2008). "Improving Customer Complaint Management by Automatic Email Classification Using Linguistic Style Features as Predictors." *Decision Support Systems* 44(4): 870-882.
- CRISP-DM. (1999). "Cross Industry Standard Process for Data Mining." Retrieved 2010-08-23, 2010, from <http://www.crisp-dm.org/Process/index.htm>.
- Croft, W. B. and D. J. Harper (1979). "Using Probabilistic Models of Document Retrieval without Relevance Information." *Journal of Documentation* 35(4): 285-295.
- Cronbach, L. and G. Gleser (1953). "Assessing Similarity Between Profiles." *Psychological Bulletin* 50: 456-473.
- Cross, G. R. (1980). "Some Approaches to Measuring Clustering Tendency." Technical Report TR-80-03. East Lansing, Department of Computer Science, Michigan State University.
- Cross, R., S. P. Borgatti, et al. (2002). "Making Invisible Work Visible: Using Social Network Analysis to Support Strategic Collaboration." *California Management Review* 44(2): 25-46.
- Cross, R., T. Laseter, et al. (2006). "Using Social Network Analysis to Improve Communities of Practice." *California Management Review* 49(1): 32-60.
- Cross, R. and A. Parker (2004). *The Hidden Power of Social Networks: Understanding How Work Really Gets Done in Organizations*. Boston, MA, Harvard Business School Press.
- Cross, R., A. Parker, et al. (2000). "A Bird's-eye View: Using Social Network Analysis to Improve Knowledge Creation and Sharing." *Knowledge Directions* 2(1): 48-61.

- Cucerzan, S. and E. Brill (2004). "Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users." *Empirical Methods in Natural Language Processing*.
- Cunningham, K. M. and L. C. Ogilvie (1972). "Evaluation of Hierarchical Grouping Techniques: A Preliminary Study." *Computer Journal* 15: 209-213.
- Dagan, I. and A. Itai (1994). "Word Sense Disambiguation Using a Second Language for Machine Aided Translation." *Computational Linguistics* 20(4): 563-596.
- Damerau, F. J. (1964). "A Technique for Computer Detection and Correction of Spelling Errors." *Communications of the ACM* 7(3): 171-176.
- Damme, C. V., M. Hepp, et al. (2007). "Folksontology: An Integrated Approach for Turning Folksonomies into Ontologies." *European Semantic Web Conference*.
- Danon, L., J. Duch, et al. (2005). "Comparing Community Structure Identification." *Journal of Statistical Mechanics*.
- Davenport, T. (2008). "Enterprise 2.0: The New, New Knowledge Management?" *Harvard Business Online*, Feb. 19.
- Davenport, T., S. Jarvenpaa, et al. (1996). "Improving Knowledge Work Processes." *Sloan Management Review* 37(4): 53-65.
- Davies, D. L. and D. W. Bouldin (1979). "A Cluster Separation Measure." *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI* 1: 224-227.
- Davis, A., B. B. Gardner, et al. (1941). *Deep South: A Social Anthropological Study of Caste and Class*. Chicago, University of Chicago Press.
- De Boeck, P. and S. Rosenberg (1998). "Hierarchical Classes: Model and Data Analysis." *Psychometrika* 53: 361-181.
- De Sarbo, W. S., J. D. Carroll, et al. (1984). "Synthesized Clustering: A Method for Amalgamating Alternative Clustering Bases with Differential Weighting of Variables." *Psychometrika* 49: 57-78.
- Deerwester, S., S. T. Dumais, et al. (1990). "Indexing by Latent Semantic Analysis." *Journal of the American Society for Information Science* 41(6): 391-407.
- Demers, A. J. (1977). "Generalized Left Corner Parsing." *Proceeding of the 4th annual ACM Symposium on Principles of Programming Languages*: 170-181.
- Denning, P. J. (1997). "A New Social Contract for Research." *Communications of the ACM* 40(2): 132-134.
- DeRose, S. J. (1988). "Grammatical Category Disambiguation by Statistical Optimization." *Computational Linguistics* 14(1): 31-39.
- Dias, G., S. Guilloré, et al. (1999). "Language Independent Automatic Acquisition of Rigid Multiword Units from Unrestricted Text Corpora." *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*, Cargèse, France.
- Dice, L. R. (1945). "Measures of the Amount of Ecological Association Between Species." *Journal of Ecology* 26(3): 297-302.
- Diday, E. (1986). "Orders and Overlapping Clusters by Pyramids." *Multidimensional Data Analysis*. J. De Leeuw, W. Heiser, J. Meulman and F. Critchley. Leiden.
- Diday, E. and G. Govaert (1977). "Classification Automatique Avec Distances Adaptatives." *RAIRO Informatique Théoretique et Applications* 11: 329-329.

- Diehr, G. (1985). "Evaluation of a Branch and Bound Algorithm for Clustering." *SIAM Journal on Scientific and Statistical Computing* 6(2): 268-284.
- Diesner, J. and K. M. Carley (2010). "A Methodology for Integrating Network Theory and Topic Modeling and its Application to Innovation Diffusion." 2010 IEEE Second International Conference on Social Computing (SocialCom).
- Diesner, J., T. L. Frantz, et al. (2005). "Communication Networks from the Enron Email Corpus "It's Always About the People. Enron is no Different"." *Computational & Mathematical Organization Theory* 11: 201-228.
- Diggle, P. J. (1983). *Statistical Analysis of Spatial Point Patterns*. New York, Academic Press.
- Dixon, J. K. (1979). "Pattern Recognition with Partly Missing Data." *IEEE Transactions on Systems, Man and Cybernetics* SMC 9: 617-621.
- Dolcinar, S. (2003). "Using Cluster Analysis for Market Segmentation - Typical Misconceptions, Established Methodological Weaknesses and Some Recommendations for Improvement." *Australasian Journal of Market Research* 11(2): 5-12.
- Doreian, P. (1979). "On the Delineation of Small Group Structures." *Classifying Social Data*. H. C. Hudson. San Francisco, Jossey-Bass.
- Doreian, P., V. Batagelj, et al. (1994). "Partitioning Networks Based on Generalized Concepts of Equivalence." *Journal of Mathematical Sociology* 19: 1-27.
- Doreian, P. and K. Fujimoto (2003). "Structures of Supreme Court Voting." *Connections* 25(3).
- Doreian, P. and F. N. Stockman (1996). "The Dynamics and Evolution of Social Networks." *Evolution of Social Networks*. P. Doreian and F. N. Stokman. New York, Gordon & Breach: 1-17.
- Dorogovtsev, S. N. and J. F. F. Mendes (2003). *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford, Oxford University Press.
- Drucker, P. F. (1988). "The Coming of the New Organization." *Harvard Business Review* 66(1): 45-53.
- Drucker, P. F. (1991). "The New Productivity Challenge." *Harvard Business Review* 69(6): 45-53.
- Dubes, R. C. (1987). "How Many Clusters are Best? - An Experiment." *Pattern Recognition* 20(6): 645-663.
- Dubes, R. C. and A. K. Jain (1979). "Validity Studies in Clustering Methodologies." *Pattern Recognition* 11: 235-254.
- Dubes, R. C. and A. K. Jain (1980). "Clustering Methodologies in Exploratory Data Analysis." *Advances in Computers*. M. C. Yovits. New York, Academic Press. 19: 113-215.
- Dubes, R. C. and G. Zeng (1987). "A Test for Spatial Homogeneity in Cluster Analysis." *Journal of Classification* 4: 33-56.
- Duda, R. O. and P. E. Hart (1973). *Pattern Classification and Scene Analysis*. New York, Wiley.

- Duflou, H. and W. Maenhaut (1990). "Application of Principal Component and Cluster Analysis to the Study of the Distribution of Minor and Trace Elements in the Normal Human Brains." *Chemometrics and Intelligent Laboratory Systems* 9: 273-286.
- Dumais, S. T. (1993). "Latent Semantic Indexing (LSI) and TREC-2." Text Retrieval Conference.
- Dumais, S. T. (1995). "Latent Semantic Indexing (LSI): TREC-3 Report." Text Retrieval Conference.
- Dunbar, R. I. M. (1993). "Coevolution of Neocortical Size, Group Size and Language in Humans." *Behavioral and Brain Sciences* 16(4): 681-735.
- Dunn, J. C. (1974). "A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well-separated Clusters." *Journal of Cybernetics* 3: 32-57.
- Dunn, J. C. (1976). "Indices of Partition Fuzziness and the Detection of Clusters in Large Data Sets." *Fuzzy Automata and Decision Processes*. M. Gupta. New York, Elsevier.
- Dunning, T. (1994). "Statistical Identification of Language." Technical Report 94-273, Computing Research Laboratory, New Mexico State University.
- Earley, J. (1970). "An Efficient Context-Free Parsing Algorithm." *Communications of the ACM* 13: 94-102.
- Ebel, H., L.-I. Mielsch, et al. (2002). "Scale-free Topology of E-mail Networks " *Physical Review E* 66: 035103.
- Eckes, T. and P. Orlik (1993). "An Error Variance Approach to Two-Mode Hierarchical Clustering." *Journal of Classification* 10: 51-74.
- Edelbrock, C. (1979). "Mixture Model Test of Hierarchical Clustering Algorithms - Problem of Classifying Everybody." *Multivariate Behavioral Research* 14(3): 367-384.
- Edelman, D. B. (1992). "An Application of Cluster Analysis in Credit Control." *IMA Journal of Management Mathematics* 4(1): 81-87.
- Edwards, A. W. F. and L. L. Cavalli-Sforza (1965). "A Method for Cluster Analysis." *Biometrics* 21: 362-375.
- Egghe, L. and L. Leydesdorff (2009). "The Relation Between Pearson's Correlation Coefficient r and Salton's Cosine Measure." *Journal of the American Society for Information Science & Technology* 60(5): 1027 - 1036.
- Ehrlich, K., C.-Y. Lin, et al. (2007). "Searching for Experts in the Enterprise: Combining Text and Social Network Analysis." *Proceedings of the 2007 international ACM conference on Supporting group work, Sanibel Island, Florida, USA.*
- El Sayed, A., H. Hacid, et al. (2008). "Involving Validity Indices in Document Clustering." *Conférence en Recherche d'Information et Applications (CORIA).*
- Ellson, J., E. R. Gansner, et al. (2004). "Graphviz and Dynagraph – Static and Dynamic Graph Drawing Tools." *Graph drawing software, Springer-Verlag. Berlin/Heidelberg:* 127-148.
- Elsner, U. (1997). "Graph Partitioning - A Survey." Technical Report 97-27, Technische Universität Chemnitz.
- Erdős, P. and A. Rényi (1959). "On Random Graphs." *Publicationes Mathematicae* 6: 290-297.

- Erdős, P. and A. Rényi (1960). "On the Evolution of Random Graphs." *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5: 17-61.
- Erdős, P. and A. Rényi (1961). "On the Strength of Connectedness of a Random Graph." *Acta Mathematica Hungarica* 12: 261-267.
- Erickson, T. and W. A. Kellogg (2000). "Social Translucence: An Approach to Designing Systems that Mesh with Social Processes." *Transactions on Computer-Human Interaction*, ACM Press, New York. 7: 59-83.
- Everett, M. G. and S. P. Borgatti (1999). "Peripheries of Cohesive Subsets." *Social Networks* 21(4): 397-407.
- Everitt, B. S. (1974). *Cluster Analysis*. New York, John Wiley & Sons, Inc.
- Everitt, B. S. (1979). "Unresolved Problems in Cluster Analysis." *Biometrics* 35(1): 169-181.
- Everitt, B. S. (1980). *Cluster Analysis*. New York, Halsted.
- Everitt, B. S. and G. Dunn (2001). *Applied Multivariate Data Analysis*. London, Arnold.
- Everitt, B. S., S. Landau, et al. (2001). *Cluster Analysis*. New York, Oxford University Press.
- Falkowski, T. and J. Bartelheimer (2008). "Applying Social Network Analysis Methods to Explore Community Dynamics." *Applications of Social Network Analysis 2005*. U. Serdült and V. Täube. Berlin, Wissenschaftlicher Verlag: 189-212.
- Falkowski, T., J. Bartelheimer, et al. (2006). "Mining and Visualizing the Evolution of Subgroups in Social Networks." *Proceedings of Web Intelligence, IEEE Computer Society*: 52-58.
- Falkowski, T., A. Barth, et al. (2007). "DENGRAPH: A Density-based Community Detection Algorithm." *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*: 112-115.
- Falkowski, T., A. Barth, et al. (2008). "Studying Community Dynamics with an Incremental Graph Mining Algorithm." *Proceedings of the 14th Americas Conference on Information Systems (AMCIS)*.
- Falkowski, T. and M. Spiliopoulou (2006). "Observing Dynamics in Community Structures " *Proceedings of Adaptation in Artificial and Biological Systems (AISB'06)* 3: 102-105.
- Falkowski, T. and M. Spiliopoulou (2007). "Users in Volatile Communities: Studying Active Participation and Community Evolution." *Proceedings of User Modeling 2007 - LNAI 4511*, Springer.
- Fallows, D. (2002). "Email at Work: Few Feel Overwhelmed and Most are Pleased with the Way Email Helps Them Do Their Jobs." Retrieved 2012-04-03, 2012, from <http://pewinternet.org/Reports/2002/Email-at-work/Summary-of-Findings.aspx>.
- Fayyad, U. M., G. Piatetsky-Shapiro, et al. (1996). "From Data Mining to Knowledge Discovery: An Overview." *Advances in Knowledge Discovery and Data Mining*. U. M. Fayyad, G. Piatetsky-Shapiro and P. Symth, AAAI Press/MIT Press: 1-34.
- Feldman, J. (1995). "Perceptual Models of Small Dot Clusters." *Partitioning Data Sets. I*. Cox, P. Hansen and B. Julesz. Providence, RI, American Mathematical Society. 19: 331-364.
- Feldman, R. and I. Dagan (1995). "Knowledge Discovery in Textual Databases (KDT)." *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*.

- Feldman, R., M. Fresko, et al. (2008). "Using Text Mining to Analyze User Forums." Proceedings of the International Conference on Service Systems and Service Management (ICSSSM 2008).
- Feldman, R., J. Goldenberg, et al. (2010). "Mine Your Own Business: Market Structure Surveillance Through Text Mining." Marketing Science Institute, Special Report: 10-202.
- Fellbaum, C. D., Ed. (1998). "WordNet – An Electronic Lexical Database. Cambridge, MA, MIT Press.
- Fillenbaum, S. and A. Rapoport (1971). Structures in the Subjective Lexicon. New York, Academic Press.
- Fjällström, P.-O. (1998). "Algorithms for Graph Partitioning: A Survey." Linköping Electronic Articles in Computer and Information Science 3(10).
- Flake, G. W., S. Lawrence, et al. (2000). "Efficient Identification of Web Communities." 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston.
- Flake, G. W., S. Lawrence, et al. (2002). "Self-organization and Identification of Web Communities." IEEE Computer 35(3): 66–71.
- Forgy, E. (1965). "Cluster analysis of Multivariate Data: Efficiency versus Interpretability of Classifications." Biometrics 21: 768 (Abstract).
- Forney, G. D. (1973). "The Viterbi Algorithm." Proceedings of the IEEE 61(3): 268-278.
- Fowlkes, E. B., R. Gnanadesikan, et al. (1988). "Variable Selection in Clustering." Journal of Classification 5(2): 205-228.
- Fowlkes, E. B. and C. L. Mallows (1983). "A Method of Comparing Two Hierarchical Clusterings." Journal of the American Statistical Association 78: 553-569.
- Fox, L. (2003). Enron. The Raise and Fall. Hoboken, N.J., Wiley & Sons.
- Francis, W. N. and H. Kučera (1982). Frequently Analysis of English Usage: Lexicon and Grammar. Boston, MA, Houghton Mifflin.
- Frank, H. and M. Trier (2007). "Projektmanagement." Systemanalyse im Unternehmen - Prozessorientierte Methoden der Wirtschaftsinformatik. H. Krallmann, M. Trier and M. Schönherr, Oldenbourg Wissenschaftsverlag, 5: 187-226.
- Freeman, L. C. (1977). "A Set of Measures of Centrality Based on Betweenness." Sociometry 40: 35–41.
- Freeman, L. C. (2000). "Visualizing Social Networks." Journal of Social Structure 1(1).
- Freeman, L. C. (2003). "Finding Social Groups: A Meta-Analysis of the Southern Women Data." Dynamic Social Network Modelling and Analysis: Workshop Summary and Papers. R. L. Breiger, K. M. Carley and P. Pattison. Washington, The National Academies Press.
- Freeman, L. C. (2004). The Development of Social Network Analysis: A Study in the Sociology of Science. Vancouver, Empirical Press.
- Freeman, L. C. (2006). The Development of Social Network Analysis. Vancouver, Empirical Press.
- Freeman, L. C., S. P. Borgatti, et al. (1991). "Centrality in Valued Graphs: A Measure of Betweenness Based on Network Flow." Social Networks 13(2): 141-154.

- Freeman, L. C. and D. R. White (1993). "Using Galois Lattice to Represent Network Data." *Sociological Methodology*. P. V. Marsden. Cambridge, MA, Blackwell: 127-146.
- Freeman, S. C. and L. C. Freeman (1979). "The Networks Network: A Study of the Impact of a New Communications Medium on Sociometric Structure." *Social Science Research Reports* No. 46. Irvine, CA, University of California.
- Friedkin, N. E. (1980). "A Test of Structural Features of Granovetter's Strength of Weak Ties Theory." *Social Networks* 2: 411-422.
- Friedkin, N. E. (1991). "Theoretical Foundations for Centrality Measures." *American Journal of Sociology* 96(6): 1478-1504.
- Friedman, H. P. and J. Rubin (1967). "On Some Invariant Criteria for Grouping Data." *Journal of the American Statistical Association* 62: 1159-1178.
- Friedman, J. H. and L. C. Rafsky (1979). "Multivariate Generalizations of the Wald-Wolfowitz and Smirnov 2-Sample Test." *Annals of Statistics* 7(4): 697-717.
- Fruchterman, T. M. J. and E. M. Reingold (1991). "Graph Drawing by Force-Directed Placement." *Software - Practice & Experience* 21(11): 1129-1164.
- Fuhr, N. (1989). "Optimum Polynomial Retrieval Functions Based on the Probability Ranking Principle." *ACM Transactions on Information Systems* 7(3): 183-204.
- Fuhr, N. and U. Pfeifer (1994). "Probabilistic Information Retrieval as a Combination of Abstraction, Inductive Learning, and Probabilistic Assumptions." *ACM Transactions on Information Systems* 12(1): 92-115.
- Fusaro, P. C. and R. M. Miller (2002). *What Went Wrong at Enron*. Hoboken, N.J., Wiley & Sons.
- Gabrilovich, E. and S. Markovitch (2006). "Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge." *American Association for Artificial Intelligence*: 1301-1306.
- Gabrilovich, E. and S. Markovitch (2007). "Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis." *Proceedings of the 20th International Joint Conferences on Artificial Intelligence*.
- Gale, N., W. C. Harperin, et al. (1984). "Unclassed Matrix Shading and Optimal Ordering in Hierarchical Cluster Analysis." *Journal of Classifications* 1: 75-92.
- Gale, W. A., K. W. Church, et al. (1992). "A Method for Disambiguating Word Senses in a Large Corpus." *Computers and the Humanities* 26: 415-439.
- Garfield, E., S. I. H., et al. (1964). *The Use of Citation Data in Writing the History of Science*. Philadelphia, The Institute for Scientific Information.
- Garrett, R. G. (1989). "The Chi-square Plot: A Tool for Multivariate Outlier Recognition." *Journal of Geochemical Exploration* 32: 319-341.
- Garside, R. (1995). "Grammatical Tagging of the Spoken Part of the British National Corpus: A Progress Report." *Spoken English on Computer: Transcription, Mark-up, and Application*. G. Leech, G. Myers and J. Thomas. Harlow, Essex, Longman.
- Garside, R., G. Sampson, et al., Eds. (1987). *The Computational Analysis of English: A Corpus-based Approach*. London, Longman.

- Gartner (2010). "Gartner Identifies the Top 10 Strategic Technologies for 2011." Analysts Examine Latest Industry Trends During Gartner Symposium/ITxpo, October 17-21, in Orlando. Gartner Newsroom.
- Gartner (2011). "Gartner's Hype Cycle 2011: Social Analytics And Activity Streams Reach "The Peak". G. Cuccureddu. Gartner Newsroom.
- Garton, L., C. Haythornthwaite, et al. (1997). "Studying Online Social Networks." *Journal of Computer Mediated Communication* 3(1).
- Gerrand, P. (2007). "Estimating Linguistic Diversity on the Internet: A Taxonomy to Avoid Pitfalls and Paradoxes." *Journal of Computer-Mediated Communication* 12(4).
- Gibson, R. and G. Sébastien (2007). "The Style Consistency of Hedge Funds." *European Financial Management* 13(27): 1-18.
- Gingras, Y. (2007). "Mapping the Changing Centrality of Physicists (1900-1944)." *Proceedings of the 11th Conference of the International Society for Scientometrics and Informetrics (ISSI), Madrid, Spain.*
- Girgensohn, A., A. Lee, et al. (2003). "Social Browsers for Visualizing Web Communities." *Proceedings of the ACM WWW 2003, Budapest, Hungary.*
- Girvan, M. and M. E. J. Newman (2002). "Community Structure in Social and Biological Networks." *Proceedings of the National Academy of Sciences USA* 99: 7821-7826.
- Gitman, I. and M. D. Levine (1970). "An Algorithm for Detecting Unimodal Fuzzy Sets and its Application as a Clustering Technique." *IEEE Transactions on Computers* C-19(7): 583-593.
- Gladwell, M. (1999). "Six Degrees of Lois Weisberg." *The New Yorker*: 52-63.
- Gloor, P. A., R. Laubacher, et al. (2004). "Temporal Visualization and Analysis of Social Networks." *North American Association for Computational Social and Organizational Science (NAACSOS) Conference. Pittsburgh PA.*
- Gloor, P. A. and Y. Zhao (2006). "Analyzing Actors and Their Discussion Topics by Semantic Social Network Analysis." *Proceedings of the 2006 Conference on Information Visualization, IEEE CS Press.*
- Gnanadesikan, R., R. J. Kettinger, et al. (1977). "Interpreting and Assessing the Results of Cluster Analysis." *Bulletin of the International Statistical Institute* 47(451-463).
- Gnanadesikan, R., R. J. Kettinger, et al. (1982). "Projection Plots for Displaying Clusters." *Statistics and Probability: Essays in Honor of C. R. Rao. G. Kallianpur and J. K. Ghosh. New York, North-Holland: 269-280.*
- Gnanadesikan, R., R. J. Kettinger, et al. (1995). "Weighting and Selection of Variables for Cluster Analysis." *Journal of Classifications* 12: 113-136.
- Godbole, N., M. Srinivasaiah, et al. (2007). "Large Scale Sentiment Analysis for News and Blogs." *ICWSM'2007 Boulder, Colorado, USA.*
- Goffman, W. (1964). "On Relevance as a Measure." *Information Storage and Retrieval* 2(3): 201-203.
- Goffman, W. and V. A. Newill (1964). "Methodology for Test and Evaluation of Information Retrieval Systems, Comparative Systems Laboratory, Report TSL: TR-2. Cleveland, Ohio, Western Reserve University.

- Goodman, L. A. and W. H. Kruskal (1954). "Measures of Association for Cross-Classifications." *Journal of the American Statistical Association* 49(268): 732-764.
- Gordon, A. D. (1980). *Classification*. London, Chapman & Hall/CRC.
- Gordon, A. D. (1999). *Classification*. Boca Raton, Chapman & Hall/CRC.
- Gould, R. V. and R. M. Fernandez (1989). "Structures of Mediation: A Formal Approach to Brokerage in Transaction Networks." *Sociological Methodology* 19: 89-126.
- Gower, J. C. (1967). "A Comparison of Some Methods of Cluster Analysis." *Biometrics* 23: 623-628.
- Gower, J. C. (1971). "Discussion of a Paper by R. M. Cormack." *Journal of the Royal Statistical Society, Ser. A. (Statistics in Society)* 134: 360-365.
- Gower, J. C. (1974). "Maximal Predictive Classification." *Biometrics* 30(4): 643-654.
- Gower, J. C. (1990). "Cluster Axioms." *Classification Society of North America Newsletter* July, 2-3.
- Gower, J. C. and P. Legendre (1986). "Metric and Euclidean Properties of Dissimilarity Coefficients." *Journal of Classifications* 5: 5-48.
- Granovetter, M. S. (1973). "The Strength of Weak Ties." *American Journal of Sociology* 78(6): 1360-1380.
- Granovetter, M. S. (1983). "The Strength of Weak Ties: A Network Theory Revisited." *Sociological Theory* 1: 201-233.
- Green, P. E., R. E. Frank, et al. (1967). "Cluster Analysis in Test Market Selection." *Management Science* 13: 387-400.
- Greig-Smith, P. (1964). *Quantitative Plant Ecology*. London, Butterworth & Company.
- Grimes, S. (2010). "Unstructured Data and the 80 Percent Rule." Retrieved 2010-08-25, 2010, from <http://www.clarabridge.com/default.aspx?tabid=137&ModuleID=635&ArticleID=551>
- Grineva, M., M. Grinev, et al. (2009). "Extracting Key Terms From Noisy and Multi-theme Documents." *Proceedings of the 18th international conference on World Wide Web, Madrid, Spain*.
- Grippa, F., A. Zilli, et al. (2006). "E-mail May Not Reflect the Social Network." *Proceedings of the NAACOS North American Association for Computational Social and Organizational Science Conference, Notre Dame, IN, USA*.
- Gronau, N. and J. Fröming (2006). "Eine Semiformale Beschreibungssprache zur Modellierung von Wissenskonversionen." *Wirtschaftsinformatik* 48: 349-360.
- Guimerá, R., L. Danon, et al. (2003). "Self-similar Community Structure in a Network of Human Interactions." *Physical Review E* 68: 065103.
- Guiot, J. M. (1976). "A Modification of Milgram's Small World Method." *European Journal of Social Psychology* 6: 503-507.
- Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences*, Cambridge University Press.
- Hahn, U. and K. Schnattinger (1998). "Towards Text Knowledge Engineering." *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) and of the 10th Conference on Innovative Applications*.

- Hakimi, S. L. (1965). "Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems." *Operation Research* 13: 462-475.
- Hall, A. V. (1969). "Group Forming and Discrimination with Homogeneity Functions." *Numerical Taxonomy*. A. J. Cole. New York, Academic Press: 53-67.
- Hamasaki, M., Y. Matsuo, et al. (2008). "Ontology Extraction by Collaborative Tagging with Social Networking." WWW2008 Beijing, China.
- Hamer, R. M. and J. W. Cunningham (1981). "Cluster Analyzing Profile Data Confounded with Interrater Differences. A Comparison of Profile Association Measures." *Applied Psychological Measurement* 5: 63-72.
- Hammer, M. (1983). "'Core' and 'Extended' Social Networks in Relation to Health and Illness." *Social Science & Medicine* 17(7): 405-411.
- Hammersley, J. M. (1950). "The Distribution of Distance in a Hypersphere." *Annals of Mathematical Statistics* 21(3): 447-452.
- Hampel, F. R., E. M. Ronchetti, et al. (1986). *Robust Statistics: The Approach on Influence Functions*. New York, Wiley.
- Hands, S. and B. S. Everitt (1987). "A Monte Carlo Study of the Recovery of Cluster Structure in Binary Data by Hierarchical Clustering Techniques." *Multivariate Behavioral Research* 22: 235-243.
- Hanneman, R. A. and M. Riddle (2005). "Introduction to Social Network Methods.
- Hansen, M. T., N. Nohria, et al. (1999). "What's Your Strategy for Managing Knowledge?" *Harvard Business Review* Mar-Apr 1999: 106-116.
- Hansen, P. and M. DeLattre (1978). "Complete-link Cluster-analysis by Graph Coloring." *Journal of the American Statistical Association* 73: 397-403.
- Hansen, P. and B. Jaumard (1997). "Cluster Analysis and Mathematical Programming." *Mathematical Programming* 79(191-215).
- Harman, D. (1991). "How Effective is Suffixing?" *Journal of the American Society for Information Science* 42: 7-15.
- Hartigan, J. A. (1967). "Representation of Similarity Matrices by Trees." *Journal of the American Statistical Association* 62: 1140-1158.
- Hartigan, J. A. (1975). *Clustering Algorithms*. New York, Wiley-Interscience.
- Hatzivassiloglou, V., L. Gravano, et al. (2000). "An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering." *Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM Press.
- Hayes, P. J. and S. P. Weinstein (1990). "Construe-TIS: A System for Content-based Indexing of a Database of News Stories." *Conference on Innovative Applications of Artificial Intelligence*: 49-66.
- Heaps, H. S. (1978). *Information Retrieval: Computational and Theoretical Aspects*, Academic Press.
- Hearst, M. A. (1999). "Untangling Text Data Mining." *ACL'99 Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics*.

- Heider, F. (1946). "Attitudes and Cognitive Organization." *The Journal of Psychology* 21: 107-112.
- Heisig, P. (2002). "GPO-WM - Methoden und Werkzeuge zum geschäftsprozessorientierten Wissensmanagement." *Geschäftsprozessorientiertes Wissensmanagement*. A. Abecker, K. Hinkelmann, H. Maus and H.-J. Müller. Berlin Heidelberg, Springer: 47-64.
- Heisig, P. (2005). "Integration von Wissensmanagement in Geschäftsprozesse. Berlin, Technische Universität Berlin., Dissertation.
- Henning, J. (2010). "How Text Mining Changes Survey Analysis." Retrieved 2010-08-26, 2010, from <http://blog.vovici.com/blog/bid/25424/How-Text-Mining-Changes-Survey-Analysis>.
- Hevner, A. R., S. T. March, et al. (2004). "Design Science in Information Systems Research." *MIS Quarterly* 28(1): 75-106.
- Heymann, P. and H. Garcia-Molina (2006). "Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems." InfoLab Technical Report.
- Hinchcliffe, D. (2007) "The State of Enterprise 2.0." DOI:
- Hines, W. C. S. and R. J. O. Hines (1979). "The Eberhardt Statistic and the Detection of Nonrandomness of Spatial Point Distributions." *Biometrika* 66(1): 73-79.
- Hoede, C. (1978). "A New Status Score for Actors in a Social Network." Memorandum 243, University of Twente, NL.
- Hoffman, R. and A. K. Jain (1983). "A Test of Randomness Based on the Minimal Spanning Tree." *Pattern Recognition* 1(3): 175-180.
- Hofmann, T. (1999). "Probabilistic Latent Semantic Indexing." Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, ACM Press.
- Hollink, V., J. Kamps, et al. (2004). "Monolingual Document Retrieval for European Languages." *Information Retrieval* 7(1): 33-52.
- Hopkins, B. (1954). "A New Method of Determining the Type of Distribution of Plant Individuals." *Annals of Botany* 18(2): 213-227.
- Horton, R. E. (1945). "Erosional Development of Streams and Their Drainage Basins, Hydrophysical Approach to Quantitative Morphology." *Bull. Geological Society of America* 56: 275-370.
- Hotho, A., A. Nürnberger, et al. (2005). "A Brief Survey of Text Mining." *Zeitschrift für Computerlinguistik und Sprachtechnologie* 20(1): 19-62.
- Hsieh, N.-C. (2005). "Hybrid Mining Approach in the Design of Credit Scoring Models " *Expert Systems with Applications* 28(4): 655-665.
- Hubbel, C. H. (1965). "An Input-Output Approach to Clique Identification." *Sociometry* 28(4): 377-399.
- Hubert, L. J. (1974). "Approximate Evaluation Techniques for the Single-link and Complete-link Hierarchical Clustering Procedures." *Journal of the American Statistical Association* 69: 698-704.
- Hubert, L. J. and P. Arabie (1985). "Comparing Partitions." *Journal of Classification* 2: 193-218.

- Hubert, L. J. and M. J. Subkoviak (1979). "Confirmatory Inference and Geometric Models." *Psychological Bulletin* 14(121-130).
- Hughes, B., T. Baldwin, et al. (2006). "Reconsidering Language Identification for Written Language Resources." *International Conference on Language Resources and Evaluation*: 485-488.
- Huisman, M. and M. A. J. A. J. van Duijn (2005). "Software for Social Network Analysis." *Models and Methods in Social Network Analysis*. P. J. Carrington, J. Scott and S. Wasserman, Cambridge University Press.
- Huisman, M. and M. A. J. A. J. van Duijn (2010). "A Reader's Guide to SNA Software." *Handbook of Social Network Analysis*. P. J. Carrington and J. Scott. London, SAGE.
- Hull, D. A. (1996). "Stemming Algorithms – A Case Study for Detailed Evaluation." *Journal of the American Society for Information Science* 47(1): 70-84.
- Hummon, N. P. and P. Doreian (1989). "Connectivity in a Citation Network: The Development of DNA Theory." *Social Networks* 11: 39-63.
- Huntsman, Y. L. (2010). "The Case Analysis of the Scandal of Enron." *International Journal of Business and Management* 5(10).
- Hyvärinen, L. (1962). "Classification of Qualitative Data." *BIT Numerical Mathematics* 2(2): 83-89.
- IKM Research group (2011). "Knowledge Networks and Semantic Technologies. Lecture Slides.
- Ingaramo, D., D. Pinto, et al. (2008). "Evaluation of Internal Validity Measures in Short-text Corpora." *Proceedings of the 9th international conference on Computational linguistics and intelligent text processing*, Haifa, Israel.
- Ino, H., M. Kudo, et al. (2005). "Partitioning of Web Graphs by Community Topology." *14th international conference on World Wide Web (WWW)*, Chiba.
- Ittner, D. J., D. D. Lewis, et al. (1995). "Text Categorization of Low Quality Images." *Annual Symposium on Document Analysis and Information Retrieval*.
- Jaccard, P. (1901). "Distribution de la Flore Alpine Dans le Bassin des Drouces et Dans Quelques Regions Voisines." *Bulletin de la Société Vaudoise des Sciences Naturelles* 37(140): 241-272.
- Jaccard, P. (1908). "Nouvelles Recherches sur la Distribution Florale." *Bulletin de la Société Vaudoise de Sciences Naturelles* 44: 223-370.
- Jackson, J. E. (1991). *A User's Guide to Principal Components*. New York, Wiley.
- Jain, A. K. and R. C. Dubes (1988). *Algorithms for Clustering Data*. Endelwood Cliffs, New Jersey, Prentice Hall.
- Jancey, R. C. (1966). "Multidimensional Group Analysis." *Australian Journal of Botany* 14(1): 127-130.
- Jansen, D. (2003). *Einführung in die Netzwerkanalyse*. Opladen, Leske + Budrich.
- Jardine, C. J., N. Jardine, et al. (1967). "The Structure and Construction of Taxonomic Hierarchies." *Mathematical Biosciences* 1: 173-179.
- Jardine, N. and R. Sibson (1968). "The Construction of Hierarchic and Non-Hierarchic Classifications." *Computer Journal* 11: 117-184.
- Jardine, N. and R. Sibson (1971). *Mathematical Taxonomy*. London and New York, Wiley.

- Jardine, N. and C. J. van Rijsbergen (1971). "The Use of Hierarchic Clustering in Information Retrieval." *Information Storage and Retrieval* 7: 217-240.
- Jensen, R. E. (1969). "A Dynamic Programming Algorithm for Cluster Analysis." *Operation Research* 17: 1034-1057.
- Jing, H. (2000). "Sentence Reduction for Automatic Text Summarization." Conference on applied natural language processing.
- Johnson-Cramer, M. E., S. Parise, et al. (2007). "Managing Change through Networks and Values." *California Management Review* 29(3): 85-109.
- Johnson, D., V. Malhotra, et al. (2006). "More Effective Web Search Using Bigrams and Trigrams." *Webology* 3(4).
- Johnson, S. C. (1967). "Hierarchical Clustering Schemes." *Psychometrika* 32: 241-254.
- Jones, W. P. and G. W. Furnas (1987). "Pictures of Relevance: A Geometric Analysis of Similarity Measures." *Journal of the American Society for Information Science* 36(6): 420-422.
- Kalakota, R. and M. Robinson (2001). *E-Business 2.0: Roadmap for Success*. Boston, MA, Addison-Wesley Pearson Education.
- Kammenhuber, N., J. Luxenburger, et al. (2006). "Web Search Clickstreams." *ACM SIGCOMM on Internet Measurement*: 245-250.
- Kannan, R., S. Vempala, et al. (2004). "On Clusterings: Good, Bad and Spectral." *Journal of the ACM* 51(3): 497-515.
- Kasami, T. (1965). "An Efficient Recognition and Syntax-Analysis Algorithm for Context-free Languages." Scientific report AFCRL-65-758. Bedford, Air Force Cambridge Research Lab.
- Katz, L. (1953). "A New Status Index Derived from Sociometric Data Analysis." *Psychometrika* 18(1): 39-43.
- Kaufman, L. and P. J. Rousseeuw (1990). *Finding Groups in Data - An Introduction to Cluster Analysis*. Hoboken, New Jersey, Wiley-Interscience.
- Kay, M. (1973). "The MIND System." *Natural Language Processing*. R. Rustin. New York, Algorithmics Press: 155-188.
- Kay, M. (1980). "Algorithm Schemata and Data Structures in Syntactic Processing." Technical Report CSL-80-12. Palo Alto, CA, Xerox PARC, Palo Alto.
- Kelly, F. P. and B. D. Ripley (1976). "A Note on Strauss's Model for Clustering." *Biometrika* 63: 357-360.
- Kendall, M. G. (1938). "A New Measure of Rank Correlation." *Biometrika* 30: 81-89.
- Kendall, M. G. (1988). "Discrimination and Classification." *Multivariate Analysis*. P. R. Krishnaiah. New York, Academic Press: 165-185.
- Kent, A., M. M. Berry, et al. (1955). "Machine Literature Searching VIII. Operational Criteria for Designing Information Retrieval Systems." *American Documentation* 6(2): 93-101.
- Kernighan, M. D. and K. W. Church (1990). "A Spelling Correction Program Based on a Noisy Channel Model." Annual Meeting of the Association for Computational Linguistics.

- Kessler, G. (2010). "Virtual Business: An Enron Email Corpus Study " *Journal of Pragmatics* 42: 262-270.
- Kim, B. J., C. N. Yoon, et al. (2002). "Path Finding Strategies in Scale Free Networks." *Physical Review E* 65: 027103.
- Kim, D.-W., K. H. Lee, et al. (2004). "On Cluster Validity Index for Estimation of the Optimal Number of Fuzzy Clusters." *Pattern Recognition* 37: 2009 – 2025.
- King, B. (1967). "Step-wise Clustering Procedures." *Journal of the American Statistical Association* 69: 86-101.
- Kirriemuir, J. W. and P. Willett (1995). "Use of Cluster Analysis Methods for Analysing the Outputs of Multiple Data-base Searches." *Electronic Library and Visual Information Research: Proceedings of the Second ELVIRA Conference, London, Aslib.*
- Kirschner, A. (2008). *Overview of Common Social Network Analysis Software Platforms.* San Francisco, Monitor Group.
- Kittler, J. (1978). "Classification of Incomplete Pattern Vectors Using Modified Discriminant Functions." *IEEE Transactions on Computers* C 27: 367-374.
- Klein, R. W. and R. C. Dubes (1989). "Experiments in Projection and Clustering by Simulated Annealing." *Pattern Recognition* 22: 213-220.
- Klein, S. and R. F. Simmons (1963). "A Computational Approach to Grammatical Coding of English Words." *Journal of Association for Computing Machinery* 10: 334-347.
- Kleinberg, J. M. (2000a). "Navigation in a Small World." *Nature* 406: 845.
- Kleinberg, J. M. (2000b). "The Small World Phenomenon: An Algorithmic Perspective." *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, Portland, Oregon, United States*
- Kleinfeld, J. S. (2002). "Could It be a Big World After All?" *Society* 39: 61-66.
- Klimt, B. and Y. Yang (2004). "The Enron Corpus: A New Dataset for Email Classification Research." *Machine Learning: ECML 2004*: 217-226.
- Klösgen, W. and J. M. Zytkow (1996). "Knowledge Discovery in Databases Terminology." *Advances in Knowledge Discovery and Data Mining.* U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, AAAI Press: 573-592.
- Knoke, D. and J. H. Kuklinski (1983). *Network Analysis.* Beverly Hills, CA, Sage.
- Kodratoff, Y. (1999). "Knowledge Discovery in Texts. A Definition and Applications." *Lecture Notes in Computer Science* 1609: 16-29.
- Kohonen, T. (1982). "Self-organized Formation of Topologically Correct Feature Maps." *Biological Cybernetics* 43: 59-69.
- Konheim, A. G. (1981). *Cryptography: A Primer,* John Wiley & Sons.
- Koontz, W. L. G., P. M. Narendra, et al. (1975). "A Branch and Bound Clustering Algorithm." *IEEE Transactions on Computer* 24: 908-915.
- Korte, C. and S. Milgram (1970). "Acquaintance Links between White and Negro Populations: Application of the Small World Method." *IBM Journal of Research and Development* 15(2): 101-108.
- Kossinets, G. and D. J. Watts (2006). "Empirical Analysis of an Evolving Social Network." *Science* Jan 6: 88-90.

- Krackhardt, D. (1992). "The Strength of Strong Ties: The Importance of Philos in Organizations." *Organizations and networks: Theory and practice*. N. Nohiram and R. Eccles. Cambridge, MA: Cambridge University Press: 216-239.
- Krackhardt, D. and R. N. Stern (1988). "Informal Networks and Organizational Crises: An Experimental Simulation." *Social Psychology Quarterly* 51(2): 123-140.
- Krallmann, H., M. Trier, et al., Eds. (2007). "Systemanalyse im Unternehmen - Prozessorientierte Methoden der Wirtschaftsinformatik, Oldenbourg Wissenschaftsverlag.
- Krause, E. F. (1975). *Taxicab Geometry: An Adventure in Non-Euclidean Geometry*. Menlo Park, C.A., Addison Wesley.
- Krebs, V. E. (1996). "Managing Core Competencies of the Corporation, The Advisory Board Company.
- Krebs, V. E. (2002). "Post-Merger Integration." Retrieved 2010-09-24, 2010, from <http://www.orgnet.com/merger.html>.
- Krebs, V. E. and J. Holley. (2007). "Building Smart Communities through Network Weaving." Retrieved 2007-06-27, 2007, from <http://www.orgnet.com/BuildingNetworks.pdf>.
- Krovetz, R. J. (1995). "Word Sense Disambiguation for Large Text Databases." University of Massachusetts Amherst. PhD.
- Kruskal, J. B. (1977). "The Relationship of Multidimensional Scaling and Clustering." *Classification and Clustering*. J. Van Ryzin. New York, Academic Press: 7-44.
- Krzanowski, W. J. (1988). "Principles of Multivariate Analysis: A User's Perspective. Oxford, Oxford University Press.
- Kupiec, J., J. Pedersen, et al. (1995). "A Trainable Document Summarizer." *Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM Press.
- Kurgan, L. A. and P. Musilek (2006). "A survey of Knowledge Discovery and Data Mining process models." *The Knowledge Engineering Review* 21(1): 1-24.
- Lakhani, K. R. and A. P. McAfee (2007). "Case Study on Deleting "Enterprise 2.0" Article." Courseware #9-607-712, Harvard Business School.
- Lancaster, F. W. (1979). "Criteria by which Information Retrieval Systems May be Evaluated." *Information Retrieval Systems - Characteristics, Testing and Evaluation*. L. J. D. Wacquant. New York, John Wiley and Sons.
- Lance, G. N. and W. T. Williams (1966). "Computer Programs for Hierarchical Polythetic Classification." *Computer Journal* 9: 60-64.
- Lance, G. N. and W. T. Williams (1967). "A General Theory of Classificatory Sorting Strategies: II. Clustering Systems." *Computer Journal* 10: 271-277.
- Lance, G. N. and W. T. Williams (1968). "Note on a New Information-statistic Classificatory Program." *Computer Journal* 11: 195.
- Lance, G. N. and W. T. Williams (1979). "INVER: A Program for the Computation of Distance-measures Between Attributes of Mixed Types." *Australian Computer Journal* 11: 27-28.

- Larson, R. C. and G. Sardi (1983). "Facility Locations with the Manhattan Metric in the Presence of Barriers to Travel." *Operation Research* 31: 652-699.
- Laseter, T. and R. Cross (2006). "The Craft of Connection." *Strategy+Business* 44: 26-32.
- Laumann, E. O., P. V. Marsden, et al. (1983). "The Boundary Specification Problem in Network Analysis." *Applied Network Analysis. A Methodological Introduction*. R. S. Burt and M. J. Minor. Beverly Hills, CA, Sage: 18-34.
- Lee, A. S. (2000). "Systems Thinking, Design Science, and Paradigms: Heeding Three Lessons from the Past to Resolve Three Dilemmas in the Present to Direct a Trajectory for Future Research in the Information Systems Field." Keynote Address at 11th International Conference on Information Management, Taiwan, May 2000.
- Lefkovich, L. P. (1980). "Conditional Clustering." *Biometrics* 36: 43-58.
- Lefkovich, L. P. (1987). "Cluster Generation and Grouping Using Mathematical Programming." *Mathematical Biosciences* 41: 91-110.
- Leighton, T. and S. Rao (1999). "Multicommodity Max-flow Min-cut Theorems and their Use in Designing Approximation Algorithms." *Journal of the ACM* 46(6): 787-832.
- Leopold, E. and J. Kindermann (2002). "Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?" *Machine Learning* 46: 423-444.
- Lesk, M. (1986). "Automatic Sense Disambiguation: How to Tell a Pine Cone from an Ice Cream Cone." *Proceedings of the 1986 SIGDOC Conference*, New York, Association for Computing Machinery.
- Levine, S. S. and R. Kurzban (2006). "Explaining Clustering in Social Networks: Towards an Evolutionary Theory of Cascading Benefits." *Managerial and Decision Economics* 27(2-3): 173-187.
- Lewis, D. D., Y. Yang, et al. (2004). "RCV1: A New Benchmark Collection for Text Categorization Research." *Journal of Machine Learning Research* 5: 361-397.
- Leydesdorff, L. (2005). "Similarity Measures, Author Cocitation Analysis, and Information Theory." *Journal of the American Society for Information Science & Technology* 56(7): 769-772.
- Li, H., D. Shen, et al. (2006). "Adding Semantics to Email Clustering, Proceedings of the Sixth International Conference on Data Mining." *Proceedings of the Sixth International Conference on Data Mining*.
- Liebertau, A. M. (1977). "Tests of Randomness in Two Dimensions." *Communications in Statistics - Theory and Methods* 6(14): 1367-1383.
- Lin, N., P. Dayton, et al. (1978). "Analyzing the Instrumental Use of Relations in the Context of Social Structure." *Sociological Methods & Research* 7(2): 149-166.
- Ling, R. F. (1972). "On the Theory and Construction of k-Clusters." *Computer Journal* 15: 326-332.
- Ling, R. F. (1973). "Probability Theory of Cluster Analysis." *Journal of the American Statistical Association* 68: 159-164.
- Lingoes, J. C. (1967). "The Multivariate Analysis of Qualitative Data." *Conference on Cluster Analysis of Multivariate Data*, Springfield, Virginia, U.S. Department of Commerce.
- Lita, L. V., A. Ittycheriah, et al. (2003). "tRuEcasIng." *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

- Littman, M. L., S. T. Dumais, et al. (1998). "Automatic Cross-Language Information Retrieval Using Latent Semantic Indexing." *Cross Language Information Retrieval*. G. Grefenstette.
- Liu, C. L. (1968). *Introduction to Combinatorial Mathematics*. New York, McGraw Hill.
- Liu, H., A. R. Aronson, et al. (2002). "A Study of Abbreviations in MEDLINE Abstracts." *Proceedings of AMIA Symposium*.
- Loscalzo, S. and L. Yu (2008). "Social Network Analysis: Tasks and Tools." *Social Computing, Behavioral Modeling, and Prediction*. H. Liu, J. J. Salerno and M. J. Young. New York, Springer: 151-159.
- Losiewicz, P., D. W. Oard, et al. (2000). "Textual Data Mining to Support Science and Technology Management." *Journal of Intelligent Information Systems* 15(2): 99-119.
- Lou, J.-K., K.-T. Chen, et al. (2009). "A Collusion-Resistant Automation Scheme for Social Moderation Systems." *6th annual IEEE Consumer Communications and Networking Conference*.
- Lovins, J. B. (1968). "Development of a Stemming Algorithm." *Translation and Computational Linguistics* 11(1): 22-3.
- Luhn, H. P. (1957). "A Statistical Approach to Mechanized Encoding and Searching of Literary Information." *IBM Journal of Research and Development* 1(4): 309-317.
- Luhn, H. P. (1958). "The Automatic Creation of Literature Abstracts." *IBM Journal of Research and Development* 2(2): 159-165,317.
- MacNaughton-Smith, P., W. T. Williams, et al. (1964). "Dissimilarity Analysis: A New Technique of Hierarchical Sub-division." *Nature* 202: 1034-1035.
- Mahalanobis, P. C. (1936). "On the Generalised Distance in Statistics." *Proceedings of the National Institute of Science of India* 12(1): 49-55.
- Manning, C. D., P. Raghavan, et al. (2008). *Introduction to Information Retrieval*. Cambridge, New York, Melbourne, Cambridge University press.
- Manning, C. D. and H. Schütze (1999). *Foundations of Statistical Natural Language Processing*, MIT Press.
- March, S. T. and G. F. Smith (1995). "Design and Natural Science Research on Information Technology." *Decision Support Systems* 15(4): 251-266.
- Marcus, M. P., B. Santorini, et al. (1993). "Building a Large Annotated Corpus of English: The Penn Treebank." *Computational Linguistics* 19(2): 313-330.
- Markus, M. L., A. Majchrzak, et al. (2002). "A Design Theory for Systems that Support Emergent Knowledge Processes." *MIS Quarterly* 26(3): 179-212.
- Maron, M. E. and J. L. Kuhns (1960). "On Relevance, Probabilistic Indexing, and Information Retrieval." *Journal of the Association for Computing Machinery* 7(3): 216-244.
- Maronna, R. and P. M. Jacovkis (1974). "Multivariate Clustering Procedures with Variable Metrics." *Biometrics* 30: 499-505.
- Marriott, F. H. C. (1971). "Practical Problems in a Method of Cluster Analysis." *Biometrics* 27: 501-514.
- Marriott, F. H. C. (1982). "Optimization Methods of Cluster Analysis." *Biometrika* 69: 417-421.

- Marshall, I. (1987). "Tag Selection Using Probabilistic Methods." *The Computational Analysis of English: A Corpus-based Approach*. R. Garside, G. Sampson and G. Leech. London, Longman: 42-65.
- Matthews, A. (1979). "Standardization of Measures Prior to Clustering." *Biometrics* 35: 892.
- McAfee, A. P. (2006). "Enterprise 2.0: The Dawn of Emergent Collaboration." *Sloan Management Review* 47(3): 21-28.
- McCallum, A., A. Corrada-Emmanuel, et al. (2005). "Topic and Role Discovery in Social Networks." *Proceedings of the International Joint Conference on Artificial Intelligence*.
- McPherson, J. M. (1982). "Hypernetwork Sampling: Duality and Differentiation Among Voluntary Organizations." *Social Networks* 3: 225-249.
- McQueen, J. B. (1967). "Some Methods of Classification and Analysis of Multivariate Observations." *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*: 281-297.
- McQuitty, L. L. (1966). "Similarity Analysis by Reciprocal Pairs for Discrete and Continuous Data." *Educational and Psychological Measurement* 27: 21-46.
- McRae, D. J. (1971). "Micka: A Fortran IV Iterative k-means Cluster Analysis Program." *Behavioural Science* 16: 423-424.
- Mead, M. (1934). *Mind, Self and Society*. Chicago, IL, University of Chicago Press.
- Mead, R. (1974). "A Test of Spatial Pattern at Several Scales Using Data from a Grid of Contiguous Quadrats." *Biometrics* 30(2): 295-308.
- Mending, J. (2007). "Detection and Prediction of Errors in EPC Business Process Models." *Institute of Information Systems and New Media*. Austria, Vienna University of Economics and Business Administration (WU Wien). Ph. D.
- Merkl, D. (2000). "Text Data Mining." *Handbook of Natural Language Processing*. R. Dale, H. Moisl and H. Somers. New York, Dekker: 889-903.
- Merz, C. J., P. M. Murphy, et al. (1997). "UCI Repository of Machine Learning Databases, Dept. of Information and Computer Science, University of California at Irvine.
- Meunier, D. and H. Paugam-Moisy (2006). "Cluster Detection Algorithm in Neural Networks." *ESANN'06, Advances in Computational Intelligence and Learning*: 19-24.
- Miceli, M. A. and G. Susinno (2003). "Using Trees to Grow Money." *Risk*(11-12).
- Miceli, M. A. and G. Susinno (2004). "Ultrametricity in Fund of Funds Diversification." *Physica A* 344(1): 95-99.
- Michelakis, E., I. Androutopoulos, et al. (2004). "Filtron: A Learning-based Anti-spam Filter." *Proc. 1st Conference on Email and Anti-Spam (CEAS 2004)*, Mountain View, CA, USA.
- Milgram, S. (1967). "The Small World Problem." *Psychology Today* 1(1): 60-67.
- Milligan, G. W. (1979). "Ultrametric Hierarchical Clustering Algorithms." *Psychometrika* 44: 343-346.
- Milligan, G. W. (1980). "An Examination of the Effects of Six Types of Error Perturbation of Fifteen Clustering Algorithms." *Psychometrika* 45: 325-342.
- Milligan, G. W. (1981). "A Review of Monte Carlo Test of Cluster Analysis." *Multivariate Behavioral Research* 16: 379-407.

- Milligan, G. W. and M. C. Cooper (1985). "An Examination of Procedures for Determining the Number of Clusters in a Data Set." *Psychometrika* 50: 159-179.
- Milligan, G. W. and M. C. Cooper (1988). "A Study of Standardization of Variables in Cluster Analysis." *Journal of Classifications* 5: 181-204.
- Milne, D. (2007). "Computing Semantic Relatedness Using Wikipedia Link Structure." *Proceedings of the New Zealand Computer Science Research Student Conference (NZCSRSC)*.
- Milne, D. and I. H. Witten (2008). "An Effective, Low-cost Measure of Semantic Relatedness Obtained from Wikipedia Links." *Proceedings of AAAI 2008*.
- Milo, R., S. Itzkovitz, et al. (2004). "Superfamilies of Evolved and Designed Networks." *Science* 303.
- Milo, R., S. Shen-Orr, et al. (2002). "Network Motifs: Simple Building Blocks of Complex Networks." *Science* 298.
- Mirkin, B. (1996). *Mathematical Classification and Clustering*. Dordrecht, Kluwer.
- Moffat, A. and J. Zobel (1998). "Exploring the Similarity Space." *SIGIR Forum* 32(1).
- Mojena, R. (1977). "Hierarchical Grouping Methods and Stopping Rules - An Evaluation." *Computer Journal* 20: 359-363.
- Mojena, R. and D. Wishart (1980). "Stopping Rules for Ward's Clustering Method." *Proceedings of COMPSTAT 1980, Würzburg, Physika-Verlag*.
- Molka-Danielsen, J., M. Trier, et al. (2007). "IRIS (1978-2006): Historical Reflection through Visual Analysis." *Proceedings of IRIS30 (Information systems research seminar in Scandinavia), Tampere, Finland*.
- Monge, P. R. and N. S. Contractor (2003). *Theories of Communication Networks*. New York, NY, Oxford University Pres.
- Moody, J., D. A. McFarland, et al. (2005). "Dynamic Network Visualization." *American Journal of Sociology* 110(4): 1206-1241.
- Moreno, J. L. (1934). *Who Shall Survive?* Washington, DC, Nervous and Mental Disease Publishing Company.
- Nadel, S. F. (1957). *The Theory of Social Structure*. London, Cohen and West.
- Naus, J. I. (1966). "A Power Comparison of Two Tests on Non-random Clustering." *Technometrics* 8: 493-517.
- Naus, J. I. (1982). "Approximations for Distributions of Scan Statistics." *Journal of the American Statistical Association* 77: 177-183.
- Nazir, F., H. Takeda, et al. (2009). "Comparison of Community Identification Techniques for Two-Mode Affiliation Networks Using Wikipedia Data." *IADIS International Conference e-society*. Barcelona, Spain.
- Needham, R. M. (1967). "Automatic Classification in Linguistics." *The Statistician* 17: 45-54.
- Newman, M. E. J. (2001a). "Scientific Collaboration Networks: I. Network Construction and Fundamental Results." *Physical Review E* 64: 016131.
- Newman, M. E. J. (2001b). "Scientific Collaboration Networks: II. Shortest Paths, Weighted Networks, and Centrality." *Physical Review E* 64: 016132.

- Newman, M. E. J. (2001c). "The Structure of Scientific Collaboration." *Proceedings of the National Academy of Sciences* 98: 404-409.
- Newman, M. E. J. (2003a). "Mixing Patterns in Networks." *Physical Review E* 67: 026126.
- Newman, M. E. J. (2003b). "The Structure and Function of Complex Networks." *SIAM Review* 45: 167–256.
- Newman, M. E. J. (2004a). "Analysis of Weighted Networks." *Physical Review E* 70.
- Newman, M. E. J. (2004b). "Coauthorship Networks and Patterns of Scientific Collaboration." *Proceedings of the National Academy of Sciences* 5200--5205.
- Newman, M. E. J. (2004c). "Detecting Community Structure in Networks." *The European Physical Journal B - Condensed Matter and Complex Systems* 38(2): 321-330.
- Newman, M. E. J. (2004d). "Fast Algorithm for Detecting Community Structure in Networks." *Physical Review E* 69: 066133.
- Newman, M. E. J. (2004e). "Who Is the Best Connected Scientist? A Study of Scientific Coauthorship Networks." *Complex Networks In Complex Networks* 337-370.
- Newman, M. E. J. (2006). "Modularity and Community Structure in Networks." *Proceedings of the National Academy of Sciences* 103(23): 8577-8582.
- Newman, M. E. J. (2008). "The Mathematics of Networks." *The New Palgrave Encyclopedia of Economics*. L. E. Blume and S. N. Durlauf. Basingstoke, Palgrave Macmillan.
- Newman, M. E. J., S. Forrest, et al. (2002). "Email Networks and the Spread of Computer Viruses." *Physical Review E* 66: 035101.
- Newman, M. E. J. and M. Girvan (2004). "Finding and Evaluating Community Structure in Networks." *Physical Review E* 69: 026113.
- Noma, E. and M. Shtappenina (2010). "Cluster Analysis as a Funds of Hedge Funds Portfolio Tool " *The Finance Professionals' Post*.
- Nonaka, I. and H. Takeuchi (1995). *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. New York and Oxford, Oxford University Press.
- Ntoulas, A. and J. Cho (2007). "Pruning Policies for Two-tiered Inverted Index with Correctness Guarantee." *Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM Press.
- Offermann, P. P. J. (2009). "Eine Methode zur Konzeption Betrieblicher Software mit einer Serviceorientierten Architektur." *Fakultät Elektrotechnik und Informatik, Fachgebiet Systemanalyse und EDV*. Berlin, Technische Universität Berlin. Dr-Ing.
- Okumura, A. and K. Muraki (1994). "Symmetric Pattern Matching Analysis for English Coordinate Structures." *Proceedings of the 4th Conference on Applied Natural Language Processing*, Stuttgart, Germany.
- Orlikowski, W. J. (2000). "Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations." *Organization Science* 11(4): 404-428.
- Orlikowski, W. J. and S. R. Barley (2001). "Technology and Institutions: What Can Research on Information Technology and Research on Organizations Learn From Each Other?" *MIS Quarterly* 25(2): 145-165.
- Osmer, P. S. (1982). "Quasars as Probes of the Distant and Early Universe " *Scientific American* February: 126-138.

- Paaß, G. and H. deVries (2005). "Evaluating the Performance of Text Mining Systems in Real-World Press Archives." Proc. 29th Annual Conference of the German Classification Society (GfKI 2005), Springer.
- Paice, C. D. (1990). "Another Stemmer." SIGIR Forum 24(3): 56-61.
- Panayirci, E. and R. C. Dubes (1983). "A Test for Multidimensional Clustering Tendency." Pattern Recognition 6: 433-444.
- Pang, B. and L. Lee (2004). "A Sentimental Education: Sentiment Analysis Using Subjectivity." Proceedings of the ACL.
- Papineni, K. (2001). "Why Inverse Document Frequency?" North American Chapter of the Association for Computational Linguistics: 1-8.
- Park, J. and M. E. J. Newman (2005). "Network-based Ranking System for U.S. College Football." Journal of Statistical Physics P10014.
- Parker-Rhodes, A. F. and D. M. Jackson (1969). "Automatic Classification in the Ecology of the Higher Fungi." Numerical Taxonomy. A. J. Cole. New York, Academic Press.
- Pearson, K. (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space." Philosophical Magazine 2: 559-572.
- Pedersen, T. and A. Kulkarni (2006). "Automatic Cluster Stopping with Criterion Functions and the Gap Statistic." Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume: demonstrations, New York, New York.
- Perer, A. and B. Shneiderman (2006). "Balancing Systematic and Flexible Exploration of Social Networks." IEEE Transactions on Visualization and Computer Graphics 12(5).
- Pielou, E. C. (1969). An Introduction to Mathematical Ecology. New York, John Wiley & Sons.
- Polanyi, M. (1966). The Tacit Dimension. London, Routledge & Kegan Paul.
- Pollard, D. (1981). "Strong Consistency of K-Means Clustering." Annals of Statistics 9: 135-140.
- Pool, I. (1980). "Comment on Mark Granovetter's 'The Strength of Weak Ties: A Network Theory Revisited.'" Read at the 1980 Meeting of the International Communications Association. Acapulco, Mexico.
- Porter, M. F. (1980). "An Algorithm for Suffix Stripping." Program 14(3): 130-137.
- Price, D. J. d. S. (1965). "Networks of Scientific Papers." Science 149: 510-515.
- Price, D. J. d. S. (1976). "A General Theory of Bibliometric and Other Cumulative Advantage Processes." Journal of the American Society for Information Science 27(5-6): 292-306.
- Probst, G., S. Raub, et al. (1998). Wissen Managen. Frankfurt a. M., Gabler.
- Pullum, G. K. and G. Gazdar (1982). "Natural Languages and Context-free Languages." Linguistics and Philosophy 4(4): 471-504.
- Punj, G. and D. W. Stewart (1983). "Cluster Analysis in Marketing Research: Review and Suggestions for Application." Journal of Marketing 20: 134-148.
- Putman, R. D. (1993). Making Democracy Work: Civic Traditions in Modern Italy. Princeton, NJ, Princeton University Press.

- Qiu, Y. and H. P. Frei (1993). "Concept Based Query Expansion." Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, ACM Press.
- Radicchi, F., C. Castellano, et al. (2004). "Defining and Identifying Communities in Networks." Proceedings of the National Academy of Sciences.
- Rand, W. M. (1971). "Objective Criteria for the Evaluation of Clustering Methods." Journal of the American Statistical Association 66(846-850).
- Rapoport, A. and S. Fillenbaum (1972). "An Experimental Study of Semantic Structures." Multidimensional Scaling. A. K. Romney, R. N. Shepard and S. B. Nerlove. New York, Seminar Press. 2 - Applications: 93-131.
- Ratnaparkhi, A. (1997). "A Linear Observed Time Statistical Parser Based on Maximum Entropy Models." Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP-97, Providence, RI.
- Ravasz, E. and A.-L. Barabási (2003). "Hierarchical Organization in Complex Networks." Physical Review E 67.
- Ray, S. and R. H. Turi (1999). "Determination of Number of Clusters in K-Means Clustering and Application in Colour Image Segmentation." Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques (ICAPRDT'99), Calcutta, India, Narosa Publishing House, New Delhi, India.
- Recker, J. (2008). "Understanding Process Modelling Grammar Continuance: A Study of the Consequences of Representational Capabilities." Faculty of Information Technology. Brisbane, Queensland University of Technology. Ph. D.
- Reichling, T., K. Schubert, et al. (2005). "Matching Human Actors based on their Texts: Design and Evaluation of an Instance of the ExpertFinding Framework." Proceedings of the Group05 Conference, Sanibel Island, Florida, USA.
- Reichling, T. and V. Wulf (2009). "Expert Recommender Systems in Practice: Evaluating Semi-automatic Profile Generation." Boston, Massachusetts, USA. CHI 2009.
- Resnik, P. (1999). "Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language." Journal of Artificial Intelligence Research 11: 95-1130.
- Ripley, B. D. (1977). "Modeling Spatial Pattern." Journal of the Royal Statistic Society, Ser. B 39: 172-212.
- Ripley, B. D. (1981). Spatial Statistics. New York, John Wiley & Sons.
- Ripley, B. D. and B. W. Silverman (1978). "Quick Tests for Spatial Interaction." Biometrika 65: 641-642.
- Roberts, J. M. J. (2000). "Correspondence Analysis of Two-Mode Networks." Social Networks 22: 65-72.
- Robertson, S. E. (1977). "The Probability Ranking Principle in IR." Journal of Documentation 33: 294-304.
- Robertson, S. E. and K. Spärck Jones (1976). "Relevance Weighting of Search Terms." Journal of the American Society for Information Science 27: 129-146.
- Robins, G. and M. Alexander (2004). "Small Worlds among Interlocking Directors: Network Structure and Distance in Bipartite Graphs." Computational & Mathematical Organization Theory 10(1): 69-94.

- Rogers, A. (1974). *Statistical Analysis of Spatial Dispersion: The Quadrat Method*. London, Pion.
- Rogers, D. J. and T. T. Tanimoto (1960). "A Computer Program for Classifying Plants." *Science* 132: 1115-1118.
- Roget, P. M. (1946). *Roget's International Thesaurus*, Thomas Y. Crowell.
- Rohlf, F. J. (1970). "Adaptive Hierarchical Clustering Schemes." *Systematic Zoology* 19: 58-82.
- Rohlf, F. J. (1975). "A New Approach to the Computation of the Jardine-Sibson B_k Clusters." *Computer Journal* 18: 164-168.
- Rohlf, F. J. and D. R. Fisher (1968). "Test for Hierarchical Structures in Random Data Sets." *Systematic Zoology* 17(407-412).
- Rosen-Zvi, M., T. Griffiths, et al. (2004). "The Author-topic Model for Authors and Documents." *Conference on Uncertainty in Artificial Intelligence, AUAI Press*.
- Rosenkrantz, S. J. and P. M. I. Lewis (1970). "Deterministic Left Corner Parser." *IEEE Conference Record of the 11th Annual Symposium on Switching and Automata*: 139-152.
- Roubens, M. (1978). "Pattern Classification Problems and Fuzzy Sets." *Fuzzy Sets and Systems* 1: 239-253.
- Rousseeuw, P. J. (1987). "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics* 20: 53-65.
- Rousseeuw, P. J., M.-P. Derde, et al. (1989). "Principal Components of a Fuzzy Clustering." *Trends in Analytical Chemistry* 8: 249-250.
- Rousseeuw, P. J. and B. C. Van Zomeren (1990). "Unmasking Multivariate Outliers and Leverage Points." *Journal of the American Statistical Association* 85(411): 633-651.
- Rubin, J. (1967). "Optimal Classification into Groups: An Approach for Solving the Taxonomy Problem." *Journal of Theoretical Biology* 15: 103-144.
- Ruspini, E. H. (1969). "A New Approach to Clustering." *Information and Control* 15: 22-32.
- Salton, G. (1971a). "Cluster Search Strategies and the Optimization of Retrieval Effectiveness." *The SMART Retrieval System – Experiments in Automatic Document Processing*. G. Salton, Prentice-Hall.
- Salton, G., Ed. (1971b). *The SMART Retrieval System – Experiments in Automatic Document Processing*, Prentice-Hall.
- Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, AddisonWesley.
- Salton, G. (1991). "The Smart Project in Automatic Document Retrieval." *Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM Press.
- Salton, G. and C. Buckley (1987). "Term Weighting Approaches in Automatic Text Retrieval." *Technical Report*. Ithaca, NY, Cornell University.
- Salton, G. and C. Buckley (1988). "Term-weighting Approaches in Automatic Text Retrieval." *Information Processing and Management* 24(5): 513-523.

- Salton, G. and M. J. McGill (1984). *Introduction to Modern Information Retrieval*. New York, McGraw-Hill.
- Salton, G., A. Wong, et al. (1975). "A Vector Space Model for Automatic Indexing." *Communications of the ACM* 18(11): 613-620.
- Santorini, B. (1990). "Part-of-speech tagging guidelines for the Penn Treebank Project " Technical Report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.
- Saracevic, T. (1975). "Relevance: A Review of and a Framework for the Thinking on the Notion in Information Science." *Journal of the American Society for Information Science* 26(6): 321-343.
- Saunders, R. and G. M. Funk (1977). "Poisson Limits for a Clustering Model of Strauss." *Journal of Applied Probability* 14: 776-784.
- Savova, G., T. Therneau, et al. (2006). "Cluster Stopping Rules for Word Sense Discrimination." *Proceedings of the workshop on Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*. Association for Computational Linguistics: 9-16.
- Schaeffer, S. E. (2007). "Graph Clustering - Survey." *Computer Science Review* 1(1): 27-64.
- Schilling, M. A. and C. C. Phelps (2007). "Interfirm Collaboration Networks: The Impact of Large-Scale Network Structure on Firm Innovation." *Management Science* 53(7): 1113-1126.
- Schmid, H. (1994). "Probabilistic Part-of-Speech Tagging Using Decision Trees." *International Conference on New Methods in Language Processing*.
- Schonlau, M. (2002). "The Clustergram: A Graph for Visualizing Hierarchical and Non-hierarchical Cluster Analyses." *The Stata Journal* 3: 316-327.
- Schütze, H. (1998). "Automatic Word Sense Discrimination." *Computational Linguistics* 24(1): 97-124.
- Schütze, H. and C. Silverstein (1997). "Projections for Efficient Document Clustering." *Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM Press.
- Scott, A. J. and M. J. Symons (1971). "Clustering Methods Based on Likelihood Ratio Criteria." *Biometrics* 27: 387-398.
- Scott, D. W. (1992). *Multivariate Density Estimation*. New York, Wiley.
- Scott, J. (1991). *Social Network Analysis: A Handbook*. London, Sage.
- Selim, S. Z. and K. Asultan (1991). "A Simulated Annealing Algorithm for the Clustering Problem." *Pattern Recognition* 24(10): 1003-1008.
- Selim, S. Z. and M. A. Ismail (1984). "K-means Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality." *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI* 6: 81-87.
- Seo, H.-C., H. Chung, et al. (2004). "Unsupervised Word Sense Disambiguation Using WordNet Relatives." *Computer Speech & Language* 18(3): 253-273.
- Shaffer, E., R. C. Dubes, et al. (1979). "Single-Link Characteristics of a Mode-Seeking Algorithm." *Pattern Recognition* 11: 65-73.

- Shepard, R. N. and P. Arabie (1979). "Additive Clustering: Representation of Similarities as Combinations of Discrete Overlapping Properties." *Psychological Review* 86: 87-123.
- Shetty, J. and J. Adibi (2004). "The Enron Email Dataset Database Schema and Brief Statistical Report." Distribution.
- Shi, J. and J. Malik (2000). "Normalized Cuts and Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI* 22(8): 888-905.
- Shimbel, A. (1953). "Structural Parameters of Communication Networks." *Bulletin of Mathematical Biophysics* 15(4): 501-507.
- Sifry, D. (2007). "The State of the Live Web, April 2007." Retrieved 2008-09-18, 2008, from <http://technorati.com/weblog/2007/04/328.html>.
- Silver, M. S., M. L. Markus, et al. (1995). "The Information Technology Interaction Model: A Foundation for the MBA Core Course." *MIS Quarterly* 19(3): 361-390.
- Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. London, Chapman & Hall CRC.
- Silverman, B. W. and T. Brown (1978). "Short Distances, Flat Triangles and Poisson Limits." *Journal of Applied Probability* 15: 815-825.
- Silverstein, C., M. Rauch Henzinger, et al. (1999). "Analysis of a Very Large Web Search Engine Query Log." *SIGIR Forum* 33(1): 6-12.
- Simmel, G. (1908/1950). *The Sociology of Georg Simmel*. New York, The Free Press.
- Simon, H. A. (1996). *The Sciences of the Artificial*. Cambridge, MA, MIT Press.
- Simonoff, J. S. (1996). *Smoothing Methods in Statistics*. New York, Springer-Verlag.
- Simpson, E. (2008). "Clustering Tags in Enterprise and Web Folksonomies." *HP Labs Technical Reports*: 222-223.
- Sinclair, A. J. and M. R. Jerrum (1989). "Approximate Counting, Uniform Generation and Rapidly Mixing Markov Chains." *Inf Comput* 82: 93-133.
- Singhal, A., G. Salton, et al. (1996). "Length Normalization in Degraded Text Collections." *Annual Symposium on Document Analysis and Information Retrieval*.
- Singleton, R. C. and W. Kautz (1965). "Minimum Squared Error Clustering Algorithm." *Stanford Research Institute*.
- Skvoretz, J. and K. Faust (1999). "Logit Models for Affiliation Networks." *Sociological Methodology* 29: 253-280.
- Smadja, F. (1993). "Retrieving Collocations from Text: Xtract." *Computational Linguistics* 19(1): 143-177.
- Smith, M. A. and A. T. Fiore (2001). "Visualization Components for Persistent Conversations." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.
- Smith, S. P. and A. K. Jain (1984). "Testing for Uniformity in Multidimensional Data." *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI* 6: 73-81.
- Sneath, P. H. A. and R. R. Sokal (1973). *Numerical Taxonomy*. San Francisco, W. H. Freeman and Company.

- Snijders, T. A. B. (2001). "The Statistical Evaluation of Social Network Dynamics." *Sociological Methodology Dynamics*. M. Sobel and M. Becker. Boston and London, Basil Blackwell: 361-395.
- Snijders, T. A. B. (2004). "Models for Longitudinal Network Data." *Models and Methods in Social Network Analysis*. P. J. Carrington, J. Scott and S. Wasserman. Cambridge, Cambridge University Press.
- Sokal, R. R. and C. D. Michener (1958). "A Statistical Method for Evaluating Systematic Relationships." *University of Kansas Scientific Bulletin* 38: 1409-1438.
- Sokal, R. R. and P. H. A. Sneath (1963). *Principles of Numerical Taxonomy*. London, Freeman.
- Sorensen, T. (1948). "A method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and its Application to Analyses of the Vegetation in Danish Commons." *Biologiske Skrifter* 5: 1-34.
- Spärck Jones, K. (1972). "A Statistical Interpretation of Term Specificity and its Application in Retrieval." *Journal of Documentation* 28(1): 11-21.
- Spärck Jones, K. and D. M. Jackson (1967). "Current Approaches to Classification and Clump finding at the Cambridge Language Research Unit." *Computer Journal* 1: 29-37.
- Späth, H. (1985). *Cluster Disscetion and Analysis*. Chichester, Ellis Horwood.
- Spiliopoulou, M., I. Ntoutsis, et al. (2006). "MONIC - Modeling and Monitoring Cluster Transitions." *Proc. of KDD'06*.
- Spiliopoulou, M., A. Schulz, et al. (2003). "Kursrelevanzprognose von Ad-hoc-Meldungen: Text Mining wider die Informationsüberlastung im Mobile Banking." *Wirtschaftsinformatik 2003*. W. Uhr, W. Esswein and E. Schoop. Heidelberg, Physica: 181-200.
- Sproat, R. W. (1992). *Morphology and Computation*. Cambridge, MIT Press.
- Stamps, J. and J. Lipnack (2000). "A Systems Science of Networked Organisations." *Proceedings of the World Congress on Systems Sciences ISSS 2000*.
- Stenmark, D. (2005). "How Intranets Differ from the Web: Organisational Culture's Effect on Technology." *Proceedings of ECIS2005, Regensburg, Germany, 26-28 May 2005*.
- Stephenson, K. A. and M. Zelen (1989). "Rethinking Centrality: Methods and Examples." *Social Networks* 11(1): 1-37.
- Stevens, S. S. (1946). "On the Theory of Scales of Measurement." *Science* 103: 677-680.
- Strahler, A. N. (1952). "Dynamic Basis for Geomorphology." *Bull. Geological Society of America* 63: 923-938.
- Strauss, D. J. (1975). "Model for Clustering." *Biometrika* 62: 467-475.
- Strauss, J. S. (1973). "Classification by Cluster Analysis." *Report of the International Pilot Study of Schizophrenia*. Geneva, World Health Organisation. 1: 336-356.
- Strauss, J. S., J. J. Bartko, et al. (1973). "The Use of Clustering Techniques for the Classification of Psychiatric Patients." *British Journal of Psychiatry* 122: 351-540.
- Strube, M. and S. P. Ponzetto (2006). "WikiRelate! Computing Semantic Relatedness Using Wikipedia." *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*.

- Subramaniam, L. V., S. Roy, et al. (2009). "A Survey of Types of Text Noise and Techniques to Handle Noisy Text." Proceedings of AND, Barcelona, Spain.
- Sun, L.-X., Y.-L. Xie, et al. (1994). "Cluster Analysis by Simulated Annealing." *Computer and Chemistry* 18: 103-108.
- Symons, M. J. (1981). "Clustering Criteria and Multivariate Normal Mixtures." *Biometrics* 39: 35-43.
- Täube, V. G. (2004). "Measuring the Social Capital of Brokerage Roles." *Connections* 26(1): 29-52.
- Theodoridis, S. and K. Koutroubas (1999). *Pattern Recognition*. New York, Academic Press.
- Thielscher, J. (1999). "Verborgene Schätze: Wissen." *it Management* 5: 22-25.
- Thomas, J. C., W. A. Kellogg, et al. (2001). "The Knowledge Management Puzzle: Human and Social Factors in Knowledge Management." *IBM Systems Journal* 4.
- Thorndike, R. L. (1953). "Who Belongs in the Family." *Psychometrika* 18: 267–276.
- Tibshirani, R., G. Walther, et al. (2001). "Estimating the number of clusters in a data set via the gap statistic." *Journal of the Royal Statistical Society Series B* 63: 411–423.
- Tichy, N. M., M. L. Tushman, et al. (1979). "Social Network Analysis for Organizations." *Academy of Management Review* 4(4): 507-519.
- Tombros, A. and M. Sanderson (1998). "Advantages of Query Biased Summaries in Information Retrieval." *Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, ACM Press.
- Tomlinson, S. (2003). "Lexical and Algorithmic Stemming Compared for 9 European Languages with Hummingbird Search Server at CLEF 2003." *Cross-Language Evaluation Forum*.
- Toutanova, K. and R. C. Moore (2002). "Pronunciation Modeling for Improved Spelling Correction." *Annual Meeting of the Association for Computational Linguistics*.
- Trauwaert, E. (1988). "On the Meaning of Dunn's Partition Coefficient for Fuzzy Clusters." *Fuzzy Sets and Systems* 25(2): 217-242.
- Travers, J. and S. Milgram (1969). "An Experimental Study of the Small World Problem." *Sociometry* 32(4): 425.
- Trier, M. (2005a). "IT-supported Visualization and Evaluation of Virtual Knowledge Communities." *Faculty of Computing Sciences and Electrical Engineering*. Berlin, Technical University of Berlin. PhD: 270.
- Trier, M. (2005b). "IT-supported Visualization of Knowledge Community Structures." *Proceedings of 38th IEEE Hawaii International Conference of Systems Sciences HICCS38 Big Island, Hawaii, United States*.
- Trier, M. (2008). "Towards Dynamic Visualization for Understanding Evolution of Digital Communication Networks." *Information Systems Research* 19(3).
- Trier, M. and A. Bobrik (2007a). "Analyzing the Dynamics of Community Formation using Brokering Activities." *Proceedings of the Third Communities and Technologies Conference, Michigan, Springer Series*.
- Trier, M. and A. Bobrik (2007b). "IT-gestützte Visualisierung und Analyse von virtuellen Kontaktnetzwerken - Anwendungsfelder, Methodik und Vorteile." *Analyse sozialer*

- Netzwerke und Social Software - Grundlagen und Anwendungsbeispiele. C. Mueller and N. Gronau. Berlin, GITO-Verlag.
- Trier, M. and A. Bobrik (2007c). "Systemanalyse und Wissensmanagement." Systemanalyse im Unternehmen - Prozessorientierte Methoden der Wirtschaftsinformatik H. Krallmann, M. Schönherr and M. Trier, Oldenbourg Wissenschaftsverlag. 5: 363-312.
- Trier, M. and A. Bobrik (2008a). "Dynamic Analysis of Electronic Communication Networks - Can We Trust Centrality Measures? (In German: Dynamische Analyse von Netzwerken elektronischer Kommunikation – kann der Zentralität getraut werden?)." Netzwerkanalyse und Netzwerktheorie: Ein neues Paradigma in den Sozialwissenschaften. C. Stegbauer, VS Verlag für Sozialwissenschaften.
- Trier, M. and A. Bobrik (2008b). "Social Search: Exploring and Searching Social Architectures in Digital Networks." IEEE Internet Computing 13(2): 51-59.
- Trier, M., A. Bobrik, et al. (2007). Towards Understanding the Dynamics of Communication Networks. Karlsruhe, Universitätsverlag.
- Trier, M. and C. Mueller (2004). "Towards a Systematic Approach for Capturing Knowledge-Intensive Business Processes." Practical Aspects of Knowledge Management (PAKM). D. Karagiannis and U. Reimer. Berlin, Heidelberg, Springer-Verlag: 239-250.
- Trier, M., M. Mueller, et al. (2009). "Finding Contacts in Dynamic Interaction Networks. A Social Search Approach for CRM (In German: Der richtige Ansprechpartner im dynamischen Interaktionsnetzwerk. Ein lernender softwaregestützter Social Search Ansatz am Beispiel von CRM)." HMD Journal 267.
- Tsichritzis, D. (1997). "The Dynamics of Innovation." Beyond Calculation: The Next Fifty Years of Computing. P. J. Denning and R. M. Metcalfe. New York, Copernicus, Springer Verlag: 259-265.
- Turdakov, D. and P. Velikhov (2008). "Semantic Relatedness Metric for Wikipedia Concepts Based on Link Analysis and its Application to Word Sense Disambiguation." Colloquium on Databases and Information Systems (SYRCoDIS).
- Turi, R. H. and S. Ray (1998). "K-means Clustering for Colour Image Segmentation with Automatic Detection of K." Proceedings of the International Association of Science and Technology for Development (IASTED), Signal and Image Processing (SIP '98), Las Vegas, Nevada, USA, IASTED/ACTA Press, CA, USA.
- Turpin, A. H., Y. Tsegay, et al. (2007). "Fast Generation of Result Snippets in Web Search." Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, ACM Press.
- Tversky, A. (1977). "Features of Similarity." Psychological Review 84: 327-352.
- Tyler, J. R., D. M. Wilkinson, et al. (2003). "Email as Spectroscopy: Automated Discovery of Community Structure within Organizations." Proceedings of Communities and Technologies, Kluwer.
- U.S. Government (Sarbanes-Oxley Act of 2002). "Public Law 107-204 - Sarbanes-Oxley Act of 2002. An Act to Protect Investors by Improving the Accuracy and Reliability of Corporate Disclosures Made Pursuant to the Securities Laws, and for other Purposes. t. Congress, U.S. Government Printing Office. Public Law 107 - 204.
- Urquhart, R. (1982). "Graph Theoretical Clustering Based on Limited Neighborhood Sets." Pattern Recognition 15: 173-187.

- Vadas, D. and J. R. Curran (2007). "Large-Scale Supervised Models for Noun Phrase Bracketing." Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING-2007), Melbourne, Australia.
- Vaishnavi, V. and B. Kuechler. (2004). "Design Research in Information Systems." Retrieved 2006-05-12, 2006, from <http://www.isworld.org/Researchdesign/drisISworld.htm>.
- Valitutti, A., C. Strapparava, et al. (2004). "Developing Affective Lexical Resources." *Psychology Journal* 2(1): 61-83.
- van der Aalst, W. M. P., H. A. Reijers, et al. (2005). "Discovering Social Networks from Event Logs." *Computer Supported Cooperative Work* 14(6): 549-593.
- van Ness, J. W. (1983). "A Method for Comparing Two Hierarchical Clusterings: Comment on Paper by Fowlkes and Mallows." *Journal of the American Statistical Association* 78: 576-579.
- van Rijsbergen, C. J. (1979). *Information Retrieval*, Butterworths.
- van Rijsbergen, C. J. (1986). "A Non-classical Logic for Information Retrieval." *The Computer Journal* 29(6): 481-485.
- VanBuren, V., D. Villarreal, et al. (2009). "Enron Dataset Research: E-mail Relevance Classification." Technical Reports-Computer Science. San Marcos, Texas State University, Department of Computer Science.
- Vendramin, L., R. J. G. B. Campello, et al. (2009). "On the Comparison of Relative Clustering Validity Criteria." *SIAM-SDM*: 733-744.
- Vendramin, L., R. J. G. B. Campello, et al. (2010). "Relative Clustering Validity Criteria: A Comparative Overview." *Statistical Analysis and Data Mining* 3(4): 209 - 235.
- Wainer, H. (1983). "On Multivariate Display." *Recent Advances in Statistics*. M. H. Rizzi, J. S. Rustagi and D. Siegmund. New York, Academic: 469-508.
- Walls, J. G., G. R. Widmeyer, et al. (1992). "Building an Information System Design Theory for Vigilant EIS." *Information Systems Research* 3(1): 36-59.
- Wand, M. P. and M. C. Jones (1995). *Kernel Smoothing*. London, Chapman & Hall CRC.
- Wang, J., B. Yu, et al. (2002). "Classification Visualization with Shaded Similarity Matrix." Technical Report, GSLIS University of Illinois at Urbana-Champaign.
- Ward, J. H. (1963). "Hierarchical Groupings to Optimize an Objective Function." *Journal of the American Statistical Association* 58: 235-244.
- Wasserman, S. and K. Faust (1994). *Social Network Analysis: Methods and Applications*. Cambridge, Cambridge University Press.
- Watts, D. J. (2003). *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton, Princeton University Press.
- Watts, D. J. (2004). *Six Degrees: The Science of a Connected Age*, W. W. Norton & Company.
- Watts, D. J., P. S. Dodds, et al. (2002). "Identity and Search in Social Networks." *Science* 296(5571): 1302-1305.
- Watts, D. J. and S. H. Strogatz (1998). "Collective Dynamics of 'Small-World' Networks " *Nature* 393(6684): 409-10.

- Wei, X. and W. B. Croft (2006). "LDA-based Document Models for Ad-hoc Retrieval." Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, ACM Press.
- Weinstein, C., W. Campbell, et al. (2009). "Modeling and Detection Techniques for Counter-Terror Social Network Analysis and Intent Recognition." Aerospace conference, 2009 IEEE
- Weiss, S. M., N. Indurkha, et al. (2004). Text Mining: Predictive Methods for Analyzing Unstructured Information. Berlin, Springer.
- Wellman, B. (1997). "An Electric Group is Virtually a Social Network." Culture of the Internet. S. Kiesler. Mahwah, NJ, Lawrence Erlbaum: 179-205.
- Wellman, B., J. Salaff, et al. (1996). "Computer Networks as Social Networks: Collaborative Work, Telework, and Virtual Community." Annual Review of Sociology 22: 213-238.
- Wenger, E., R. McDermott, et al. (2002). Cultivating Communities of Practice. Boston, Harvard Business School Press.
- White, H. C. (1992). Identity and Control. A Structural Theory of Social Action. Princeton/New Jersey, Princeton University Press.
- White, H. C., S. A. Boorman, et al. (1976). "Social Structure from Multiple Networks: I. Blockmodels of Roles and Positions." American Journal of Sociology 81: 730-779.
- White, K. F. and W. G. Lutters (2007). "Structuring Cross-Organizational Knowledge Sharing." Proceedings of the 2007 international ACM conference on Supporting group work, Sanibel Island, Florida, USA.
- Williams, W. T. (1971). "Principles of Clustering." Annual Review of Ecology and Systematics 2: 302-326.
- Williams, W. T. and J. M. Lambert (1959). "Multivariate Methods in Plant Ecology. I. Association-Analysis in Plant Communities." J. Ecology 47: 83-101.
- Williams, W. T., G. N. Lance, et al. (1971). "Controversy Concerning the Criteria for Taxometric Strategies." Computer Journal 14: 162-165.
- Wishart, D. (1969). "Mode Analysis: A Generalization of Nearest Neighbor which Reduces Chaining Effects." Numerical Taxonomy. A. J. Cole. New York, Academic Press.
- Wishart, D. (1982). Supplement, CLUSTAN User Manual. Program Library Unit, Edinburgh University.
- Wishart, D. (1987). "Clustan User Manual, Computing Laboratory, University of St. Andrews.
- Wittgenstein, L. (1953). Philosophical Investigations. New York, Macmillan Company.
- Wong, A. K. C. and D. C. C. Wang (1979). "DECA: A Discrete-valued Data Clustering Algorithm." IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI 1: 342-349.
- Wong, M. A. (1982). "A Hybrid Clustering Method for Identifying High-Density Clusters." Journal of the American Statistical Association 77: 841-847.
- Wong, M. A. and T. Lane (1983). "A kth Nearest Neighbour Clustering Procedure." Journal of the Royal Statistic Society, Ser. B 45: 362-368.
- Woodbury, M. A. and K. G. Manton (1982). "A New Procedure for the Analysis of Medical Classification." Methods of Information in Medicine 21: 210-220.

-
- Wu, F. and B. A. Huberman (2003). "Finding Communities in Linear Time: A Physics Approach." *European Physical Journal B - Condensed Matter and Complex Systems* 38(2): 331-338.
- Xu, K., C. Tang, et al. (2010). "A Comparative Study of Six Software Packages for Complex Network Research." *International Conference on Communication Software and Networks*. Singapore.
- Yarowsky, D. (1992). "Word-sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora." *COLING 14*: 454-460.
- Yarowsky, D. (1995). "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods." *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, Cambridge, Massachusetts Association for Computational Linguistics.
- Yeung, C.-m. A., N. Gibbins, et al. (2009). "Contextualising Tags in Collaborative Tagging Systems." *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia* 251-260
- Younger, D. H. (1967). "Recognition and Parsing of Context-free Languages in Time n^3 ." *Information and Control* 10(2): 189-208.
- Zadeh, L. A. (1965). "Fuzzy Sets." *Information and Control* 8: 338-353.
- Zadeh, L. A. (1984). "Making Computers Think Like People." *IEEE Spectrum* 21: 26-32.
- Zagoruiko, N. G. and V. N. Yolkina (1982). "Inference and Data Tables with Missing Values." *Handbook of Statistics*. P. R. Krishnaiah and L. N. Kanal. Amsterdam, North-Holland Publishing Company. 2: 493-500.
- Zhang, J. and M. S. Ackerman (2005). "Searching for Expertise in Social Networks: A Simulation of Potential Strategies." *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*.
- Zmud, R. (1997). "Editor's Comments." *MIS Quarterly* 21(2): xxi-xxii.
- Zobel, J. and P. Dart (1995). "Finding Approximate Matches in Large Lexicons." *Software Practice and Experience* 25(3): 331-345.
- Zobel, J. and P. Dart (1996). "Phonetic String Matching: Lessons from Information Retrieval." *ACM Press*.