Department of Computer Science
University of Potsdam
Am Neuen Palais 10
D-14469 Potsdam

Master's thesis

# Comparing link-based and content-based approaches of topic detection in Wikis using specific clustering algorithms

Janette Lehmann
Matr.-No.: 716498
Janette.Lehmann@gmx.de

| | |
|---|---|
| **Study path:** | Computer Science |
| **Supervisor:** | Dr. Claudia Müller-Birn<br>Institute for Software Research, Carnegie Mellon University |
| **Examiner:** | Prof. Torsten Schaub<br>Knowledge Processing and Information Systems, University of Potsdam |
| **Completed:** | September 23, 2009 |

**Affidavit**

I hereby declare that the following master thesis has been written only by the undersigned and without any assistance from third parties.

Furthermore, I confirm that no sources have been used in the preparation of this thesis other than those indicated in the thesis itself.

(September 23, 2009, Janette Lehmann)

Copyright (c) 2009 Janette Lehmann.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 published by the Free Software Foundation (http://www.gnu.org).

# Contents


**I Introduction** 1

**1 Background** 3
- 1.1 Related Work . . . 4
- 1.2 Structure of the research . . . 7

**2 Research Question** 9

**II Fundamentals** 11

**3 Generic Data Model** 13
- 3.1 Generic Data Model . . . 13
- 3.2 Analysis based on the Data Model . . . 15
  - 3.2.1 Specification of Networks . . . 15
  - 3.2.2 Network Analysis . . . 16
  - 3.2.3 Information Space Analysis . . . 17
- 3.3 Extraction and Transformation of Wiki Data . . . 18
  - 3.3.1 Introduction to Wikis . . . 18
  - 3.3.2 Extraction and Transformation . . . 20

**4 Text Mining - Fundamentals, Methods and Applications** 23
- 4.1 Introduction . . . 23
- 4.2 Fundamentals . . . 27
  - 4.2.1 Linguistic . . . 27
  - 4.2.2 Language Statistic . . . 30
  - 4.2.3 Linear Algebra . . . 33
- 4.3 Text Analysis Methods . . . 35
  - 4.3.1 Preprocessing Methods . . . 36
  - 4.3.2 Representation Methods . . . 42
  - 4.3.3 Dimension Reduction . . . 46
  - 4.3.4 Clustering . . . 63
  - 4.3.5 Further Analysis Methods . . . 75
- 4.4 Research Fields and Applications . . . 77
  - 4.4.1 Information Extraction . . . 77
  - 4.4.2 Information Retrieval . . . 79
  - 4.4.3 Natural Language Processing . . . 80
  - 4.4.4 Text Mining . . . 80

4.4.5 Data Mining . . . 82
4.4.6 Artificial Intelligence . . . 83

**III Analyses** **85**

**5 Data and Methodology** **87**
5.1 Data Sets . . . 88
5.1.1 Data Sets I - Single Topics . . . 89
5.1.2 Data Sets II - Overlapping Topics . . . 89
5.1.3 Data Sets III - Topics of a Main Topic . . . 89
5.1.4 Data Sets IV - Small Topics . . . 90
5.2 Extraction and Transformation of Wiki Data . . . 90
5.3 Implementation of a Text Mining Metric . . . 92

**6 Results and Interpretations** **97**
6.1 Clustering Phase 1 - Clustering Algorithm . . . 98
6.1.1 External Quality Analysis . . . 98
6.1.2 Internal Quality Analysis . . . 100
6.1.3 Performance Analysis . . . 102
6.1.4 Summary . . . 103
6.2 Clustering Phase 2 - Feature Selection . . . 105
6.2.1 External Quality Analysis . . . 105
6.2.2 Summary . . . 108
6.3 Clustering Phase 3 - Feature Transformation . . . 109
6.3.1 Singular Value Analysis . . . 109
6.3.2 External Quality Analysis . . . 111
6.3.3 Summary . . . 114
6.4 Clustering Phase 4 - Content-based vs. Link-based Clustering . . . 115
6.4.1 Improvement Analysis . . . 116
6.4.2 External Quality Analysis . . . 117
6.4.3 Internal Quality Analysis . . . 119
6.4.4 Performance Analysis . . . 121
6.4.5 Wiki-link Network Analysis . . . 122
6.4.6 Summary . . . 130

**7 Conclusion and Outlook** **133**

**IV Appendixes** **139**

**A List of Abbreviations** **141**

**B List of Figures** **151**

**C List of Tables** **153**

**D Bibliography** **155**

**E Topics of Data Sets** **173**
E.1 Data Set I . . . . . 173
E.2 Data Set II . . . . . 175
E.3 Data Set II . . . . . 177
E.4 Data Set IV . . . . . 179

**F Analyses Results** **181**
F.1 Clustering Case 1 - Clustering Algorithm . . . . . 181
F.2 Clustering Case 2 - Feature Selection . . . . . 182
F.3 Clustering Case 3 - Feature Transformation . . . . . 183
F.4 Clustering Case 4 - Content-based vs. Link-based Clustering . . . . . 186

**G German Summary** **191**

# Part I.

# Introduction

# 1. Background

The emergence of social software in the internet shaped human communication, interaction and co-operation substantially. The intensified utilization of social software in private and business areas motivates for a better understanding of the existing structures. Thereby the content and the topics of so-called social information spaces are one aspect of analysis.
A lot of research deals with the analysis of social information spaces such as Wikis, Folksonomies or Blogs. These investigations consider, for example, the development of a social information space or the actor activity. Another very important aspect is the content. Basically, the content is the text of the content elements created by actors. The content includes further information and connections between content elements (e.g. articles in a Wiki, Blog entries) or actors (e.g. the authors of articles).
One field of research which analyzes the content is text mining. Text mining is an extension of data mining for discovering new, previously unknown information from unstructured data. This allows, for example, the identification of topics, the consideration of content-based dependencies between content elements and actors and the improvement of the understanding of collaborative structures and content development.
One very important aspect of content analysis is the detection of topics. The topics describe different content of a social information space. This allows, contrary to the analysis on content element area, the analysis on topics area. Many social information spaces are very large and therefore the analysis can be complex and confusing. To reduce the complexity of analysis, the topics can be used for abstracting data. Consequently, the dependencies between actors based on topics or the dependencies between topics can be analyzed. Furthermore it is possible to analyze the author's activity in topics or the development of topics. Another aspect is the identification of main topics. By considering the development of a social information space, the detection of new topics and their development is possible. If the topics of a social information space are well-known, the content elements and actors can be assigned to them (automatic classification).

The question is now, how topics can be identified. One possibility offers the clustering of content elements based on their terms. A content element includes text and a text consists of terms. The terms represent the content. If content elements with similar terms are grouped, topics will be formed. Clustering is a very popular method in the area of data mining. It is used for the analysis of data, e.g. for the identification of user groups with a typical behavior. Clustering becomes also more and more important in the area of text mining. In the area of text mining clustering is applied on documents for discovering groups of similar documents. Terms are clustered as well, in order to find semantically similar terms. Furthermore, document clustering is used in information retrieval for the generation of better search results.
Because of the high diversity of terms, clustering based on terms is very complex. The base of clustering is a term document matrix, where each row represents a content element and each column a term. A value in the matrix indicates how often a term occurs in a content element. Mostly, the matrix has many columns, because the terms vary from document to document and are numerous. Many methods have been already developed, in order to reduce

the number of terms and dimensions in the matrix, e.g. stop word removal, feature selection and feature transformation. The objective of these methods is the identification of the important terms of the texts.
Another approach is given by Siroker et al. [198]. In this work the articles from CiteSeer[1] are clustered. The term document matrix was not built by terms but by titles of outgoing links of an article. With this concept, clustering is possible on a much smaller matrix, because an article has less links than terms. This concept can be applied on other social information spaces as well. The social information space only needs link-relations between their content elements, whereas by important terms a link to the corresponding content element is set. Consequently, the matrix based on links should only include the important terms of a content element.
The first question is now, which approach gives better results in the context of topic detection. The content-based clustering uses all information in form of terms and identifies the important terms with further methods. For this reason it is substantially more complex. The link-based clustering uses from the beginning only the important terms of a content element. But the importance depends on the linkage. The links of a content element are defined by actors, which leads to a subjective view of the content. The comparison of the approaches is the main object of this work.
The second question is, which existing methods are suitable for an effective and efficient clustering. There are lots of algorithms for clustering and enhancing the clustering results. The results of the clustering methods depend on the underlying data structure. Therefore the existing clustering methods have to be explored first. In this step, methods are identified that can be used. After exploring the fundamentals, clustering based on both approaches and with different clustering algorithms is executed and the results are compared. The methods are applied on Wiki-based social information space. But it is possible to use these methods for other social information spaces as well. Wikis are an appropriate social information space for the analysis, because within a Wiki different topics exist. Furthermore, there are very large Wikis with freely accessible data. The linkage between the articles can be used for the link-based clustering approach.

## 1.1. Related Work

Meanwhile a vast number of studies from different disciplines deal with the linkage of Wikis or other social information spaces. Some of these works use the linkage for executing text mining or natural language processing. In this section, works which are similar to this work are described.

In the following section, the works which are closely related to this thesis are described. In these investigations, clustering and the Wiki linkage form an important basis.
Schonhofen et al. [192] uses the titles and categories of articles for a characterization of documents. An algorithm is presented, which identifies the topics of a document by finding labels or categories. These labels and categories describe the content of a document sufficiently.
Another semantic analysis is given by Gabrilovich et al. [67], called Explicit Semantic Analysis (ESA). The approach computes the semantic relatedness of documents. The methodol-

[1] Scientific Literature Digital Library, http://citeseer.ist.psu.edu/

ogy is similar to LSA (latent semantic analysis, described in Section 4.3.3, p. 48), but it uses knowledge, which is collected and organized by humans. LSA only uses statistical cooccurrence information. By a given word, the corresponding concepts are calculated. Furthermore it is possible to compare two words or texts based on their concepts. A concept is a Wiki article, or more precisely the article name. The concept is represented as a vector of words that occur in the corresponding article. In the preprocessing step a weighted inverted index is generated. This index includes a list of words and for each word a weighted vector of concepts. The strength of association between words and concepts are represented by the TfIdf weight (described in Section 4.3.2, p. 42). Now it is possible to calculate the important concepts of a word or text. When giving a text, words are firstly extracted and represented as a vector. In the next step the value of each word is multiplied with the concept vector of the word. The result is a concept vector for each word in the text. These concept vectors are added together now. Finally a vector including the weighted concepts of a text is given. Text and word relationships are computed and compared with human judgments to evaluate the approach. Furthermore the results are compared with other approaches, which compute text and word relationships. Compared to the other approaches, the ESA method enhances the computing of text and word relationships. Jambunathan et al. [113] improves this approach by identifying the good concepts in the set of concepts. The approach is, that good concepts are related to each other, while the bad ones are unrelated. To identify the good concepts, clustering is applied.

Banerjee et al. [11] represents a clustering approach which uses additional features from Wikipedia. The method for extracting features from Wikipedia is similar to the Explicit Semantic Analysis from Gabrilovich et al. [67]. Short articles from Google News[2] were clustered. Thereby the bag of words model (vector space model, described in Section 4.3.2, p. 44) is compared with the Wiki method. The Wiki method extracts additional concepts for each Google News article. First, a Lucene [3] index of all Wikipedia articles is created. The titles and descriptions of Google News articles are used as query strings for the index. The titles of the retrieved Wikipedia articles are the concepts of the Google News articles. The concept vector is augmented with the bag of words vector. Six different clustering algorithms from the SenseCluster[4] package are used. The measures show that the use of additional features from Wikipedia enhances clustering. Banerjee et al. [10] uses this approach in a further article as well. Here Wikipedia features are used for enhancing the text classification.

Adafre et al. [2] developed an approach for discovering missing links in Wikipedia. The hypothesis is, that a similar link structure indicates to similar pages. The approach consists of two steps. First, the topically related pages will be identified. Then, the missing links will be determined. The first step comprises two phases. In the first phase, a page is represented by its citations, i.e. by the titles of pages that link to the page (full title representation). Lucene is used for indexing each page with its full title representation. Given a page, a cluster of similar pages is computed. Two pages will be similar, if they have similar citations. Therefore its full title representation is used as a query against the index. The result is a ranked list of pages. The method is called *LTRank* (Ranking based on Links and Titles). In the second phase, a page is represented by the first ten pages of the ranked list (short title representation). Lucene is used again for indexing each page with its short title representation. Given a page, a cluster of similar pages is calculated as in the first phases. But in the second phase, the new

[2] http://news.google.com/
[3] http://lucene.apache.org/
[4] http://senseclusters.sourceforge.net/

index is used. From the new list, the first entries are selected. Now the second step can be executed. For each selected page all outgoing links, that not exist in the analyzed page, are identified. For the missing links an anchor text in the analyzed page is searched. If there is an anchor text for a missing link, the link will be added. The approach was evaluated and showed acceptable results.
Siroker et al. [198] uses a similar approach for clustering articles from CiteSeer[5]. Here the titles of outgoing links for representing an article are used. Standard k-means (described in Section 4.3.4, p. 68) is applied for generating the clusters. Additionally the work describes a graph-based visualization approach for representing the clusters. Finally a possibility for the definition of cluster names is given.

In the area of text analysis, many studies and analyses are possible. In the research field of natural language processing a lot of works based on Wikis exist. Because of the structure of Wikis, they can be applied as a knowledge extraction corpus. In most cases the linkage of articles and categories is used.
Syed et al. [208] uses Wikipedia as an ontology (described in Section 4.2.1, p. 29). A document is associated with a set of topics. These topics are derived from the Wikipedia ontology. The ontology based on Wiki articles as well as their article and category links. Wiki categories represent generalized concepts and article links can be used for getting more specific concepts. Similar approaches are given from Cui et al. [44] and Suchanek et al. [206].
Wikis can be used for the automatic detection of word sense disambiguations (described in Section 4.3.1, p. 39) as well, as it is described in the study from Mihalcea et al. [163]. In the study, the article links of Wikipedia are used for the identification of the meanings from ambiguous words. Depend on the meaning of a word, it is linked to different Wiki articles.
Another field of application is taxonomy generation (described in Section 4.2.1, p. 29). Kassner et al. [126] uses the category linking from the German Wikipedia to generate a taxonomy.
A construction of a thesaurus (described in Section 4.2.1, p. 29) based on the Wikipedia is given from Nakayama et al. [172]. The article link structure is very suitable for extracting a thesauruses. If there is a link between two articles, they should have a semantic relationship. In this work the link mining methods *pfibf* (Path Frequency - Inversed Backward link Frequency) and *FB weighting* (forward / backward link weighting) are used for measuring the relationship between two articles. Similar approaches are given by Kinzler et al. [129] and Ito et al. [110]. Here the link co-occurrences analysis is used.
The work of Wu et al. [230] represents a method that autonomously *semantifying* the Wikipedia. The approach identify the structures (e.g. link structure, infoboxes, etc.) and improve them automatical. With this approach, the Wikipedia is augmented to a semantic web.

The described studies are similar to this thesis, because they use the linkage of Wikis or other social information spaces for their research. Most works substantiate that the use of the linkage enhances the research results. That shows that the developed approach is suitable for topic detection.
The works which are closely related to this thesis describe words or documents based on their links. The description is used for clustering ([11], [198]), classification ([10]) topic detection ([192]), the discovering of missing links ([2]) and for the identification of concepts for a document or word ([67], [113]). In this thesis, two approaches are compared: Topic

[5] Scientific Literature Digital Library, http://citeseer.ist.psu.edu/

detection based on content and based on links. Thereby, topic detection uses clustering. The other researches use divers approaches for the link-based method. Schonhofen et al. [192] uses labels, categories and titles. Gabrilovich et al. [67], Jambunathan et al. [113] and Banerjee et al. [11] [10] applied the explicit semantic analysis. Compared to the approach in this study, the explicit semantic analysis and the computations in [192] are more complex. The question is now, whether the approach in this work, which is more simple, enhances the clustering results as well. The approaches from Siroker et al. [198] and Adafre et al. [2] are similar to this thesis, because they only use the links of an article. But Siroker et al. don't analyze, whether the link-based approach gives better results. Adafre et al. uses the linkage for discovering missing links and not for the detection of topics.
The other researches use the linkage of Wikis in the area of natural language processing. They show that the linkage is suitable for ontology construction ([208], [44], [206]), the detection of word sense disambiguations ([163]), taxonomy generation ([126]) and thesaurus construction ([172], [129], [110]). Thereby different link types are differentiated. Category-links are used for ontology construction and taxonomy generation. Article-links are used for ontology construction, the detection of word sense disambiguations and thesaurus construction. This indicates that category-links reflect the hierarchical structure of words, whereas article-links reflect the similarity of words. The researches show that the linkage includes important information about articles and words. This information are used in this research.

## 1.2. Structure of the research

The steps within this work are visualized in Figure 1.1. Two fundamental things are needed for the work: Methods for topic detection and data for a practical comparison. The analysis methods of this work are implemented in the SONIVIS tool[6]. The tool based on a generic data model with a model for the analysis of the data. One part of this work integrates the topic detection methods into the tool.
An explanation of the generic data model is given in Chapter 3 (p. 13). Firstly, the data of each social information space are extracted and saved in this model. Based on this model, analyses independent from the social information space are possible. The integration of analysis methods based on the data model is introduced as well. Additionally, the example social information space Wiki and their transformation in the data model is described.
The existing methods for data preprocessing, representation, dimension reduction and clustering are explored in Chapter 4 (p. 23). Additionally, an overview about text mining and adjacent research fields is given. The methods, which are suitable for this study, are described in detail. Works, which concern themselves with the quality of the methods, are introduced.
After describing the data structure, introducing text mining methods and comparing different clustering methods, the practical part of this work follows. Based on the fundamental part, different data sets and methods are selected and used for the analysis. Since the quality of clustering methods depends on the underlying data structure, several methods are selected. Furthermore methods for preprocessing and representing the documents with terms and links are chosen. A description about the data sets, the preprocessing of the data, the selected methods and the process of the analyses can be found in Chapter 5 (p. 87).
The subsequent chapter (Chapter 6, p. 97) includes the results of the analysis and their eval-

[6] SONIVIS is an open source software for analyzing and visualizing virtual information spaces. http://www.sonivis.org/

uation. Different clustering cases are executed and compared. Thereby different clustering algorithms are used and the influences of dimension reduction methods (feature selection, feature transformation) are assessed. The comparison of content-based and link-based clustering is a central part within each clustering case. The last chapter (Chapter 7, p. 133) concludes the results and gives motivations for the future.

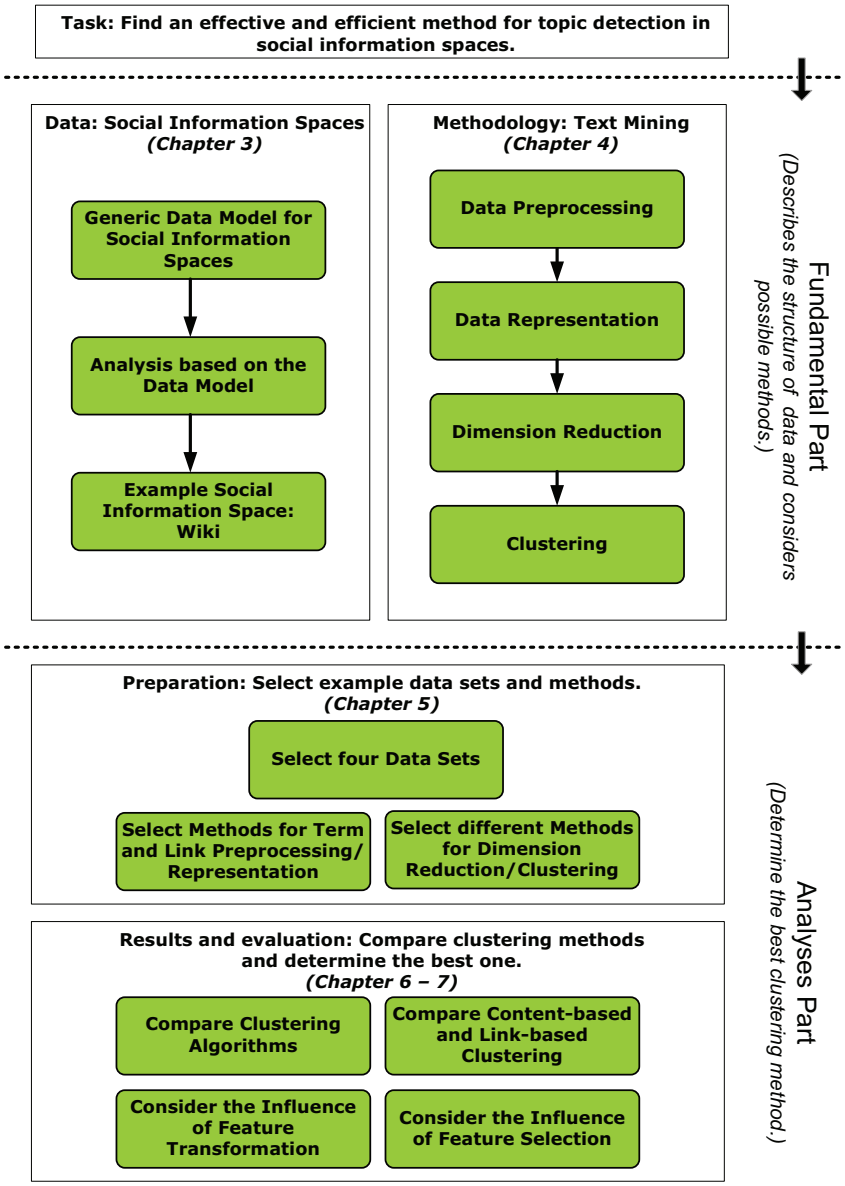


**Figure 1.1.:** Overview of this work

**Summary:** This chapter gives an introduction to the content of this research. The goal of this work and the steps for the achievement of the goal are described. First the background for this research is given and related works are introduced. The last section describes the steps, which are executed in this work. Based on this description a research question and hypothesis can be formulated and answered.

# 2. Research Question

The objective of this work is the identification of a method which allows the clustering of content elements in topics. Wikis are taken as an example of social information spaces. Wiki articles should be grouped based on their content in such a way that a cluster contains articles with related topics. Two possible approaches are link-based and content-based clustering. In order to find out which approach is better suitable, different data sets are clustered with both approaches and the results are compared. The data sets include Wiki articles of different topics.
The objective of this study can be summarized as the following research question:

**Which of the two approaches, the link-based or the content-based clustering is better suitable to detect topics in social information spaces?**

The hypothesis is that for detecting topics in social information spaces the link-based clustering approach is more efficient in terms of computing time, while the content-based clustering approach produces more detailed but also more diffuse results, however at much higher requirements in terms of time and efforts.
In the first part of the work, the possibilities, which text mining offer for the analysis of content elements and for the analyses in this work, are considered. The consideration of the possibilities forms the base of the second part. For this reason an overview about text mining is given and the methods, which are important for this work, are described in detail. The possibilities of text mining are presented. In order to clarify the goals of text mining, the methods and applications are described. Additionally the fundamentals, which are used by the methods, are presented. The fact that there are other research fields which can be used for text analysis, the differences between text mining and the other research fields are highlighted.
Based on these investigations, it is possible to test the hypothesis and thus to answer the research question. The chosen methods for dimension reduction and clustering are used for the analyses. Since there are many methods and the appropriateness of a method depends on the underlying data set, several methods are chosen.
The methods and the evaluation of the results are implemented in the SONIVIS tool. Thus the results can be used for further researches, e.g. the definition of metrics or the visualization of the results in networks. Since the tool uses a generic data model, the methods can be applied on other social information spaces as well. But the data have to be extracted and saved in the data model before.
The content-based clustering needs an initial preprocessing of the articles. The preprocessing step extracts the terms of the articles and removes the stop words (Section 4.3.1, p. 36). Using the link-based clustering, the links have to be identified. Given an article, the titles of the linked articles are the terms. With the data of both approaches the term document matrix is created. In the next step the terms of the term document matrix are weighted with the *Term Significance Weight* (TfIdf) (Section 4.3.2, p. 42). The use of dimension reduction methods

should improve the clustering. The influence of these methods is additionally analyzed. In this work the feature selection method *Term Weight Thresholding* (Section 4.3.3, p. 60) and the feature transformation method *Latent Semantic Analysis* (Section 4.3.3, p. 48) are applied. Now the distance matrix can be calculated using the *Cosine Coefficient* (Section 4.3.4, p. 65) measure. Based on this distance matrix, different clustering methods are applied. In this work two center-based clustering algorithms (*Partitioning Around Medoids* (PAM), *bisecting PAM*) (Section 4.3.4, p. 68) and two hierarchical clustering algorithms (*Agnes*, *Diana*) (Section 4.3.4, p. 67) are used.
To evaluate the approaches, the performance of the clustering and the validity of the clustering results are considered. The performance is measured by the computing time. The validity of the clustering results is measured by internal and external quality measures. In this work the Dunn Index (Section 4.3.4, p. 72) and the SD Validity Index (Section 4.3.4, p. 72) are used as internal quality measures. As an external quality measure the F-measure (Section 4.3.4, p. 71) is used. Therefore a user-specific clustering is accomplished.

**Summary:** In this chapter the research question is defined and a hypothesis is formulated. Furthermore, the steps for answering the question are described. This includes an overview about the fundamental and the applied part.

# Part II.

# Fundamentals

# 3. Generic Data Model

This chapter describes the generic data model, which is used for the practical part of this work. The data of each social information space are saved in this model. This allows the use of analysis methods, independently from the social information space. The application of analysis methods based on the data model is described as well, because in this work, topic detection methods are implemented. Since Wikis are used for the evaluation of the topic detection methods, the storage of Wiki data into the data model is described. The following descriptions based on the fundamentals of the SONIVIS tool. The generic data model and the model for the analysis of social information spaces are implemented in this tool.

## 3.1. Generic Data Model

The idea of the data model is to represent all components of a social information space as generic as possible. This means any kind of social information space should be representable in the data model - independent of its appearance and dedication. A description about the data model is given in an ongoing work from Erber et al. [58] and from Müller et al. [169]. Additionally it is described in the SONIVIS Wiki[1].
The data model is an universal actor content model. Figure 3.1 includes an UML diagram of the model. A social information space is represented by *InfoSpace*. Here the specific information about an information space are saved. An information space is able to contain or aggregate other information spaces. The model includes two different types of entities and all possible relations between these which sum up to three. The relations and entities have properties, which reflect the specific characteristics. The entities represent what can be found in any social information space:

1. Actors: Acting things (authors, readers, etc.).
2. Content elements: Things that are acted upon (articles, images, etc.).

The relations between entities of the same or different kind can be grouped as:

1. Knowledge relations: actors ↔ content (actor editing content, etc.).
2. Contextual relations: content ↔ content (hyperlinks, common topics, etc.).
3. Interactions: actor ↔ actor (collaborations, acquaintances, communication, etc.).

It is possible to build network graphs based on entities and relations. The networks, nodes and edges have properties as well.

1. Network graphs: A graph description including a collection of nodes and edges.
2. Nodes: Actors, content elements.

[1] http://www.sonivis.org/wiki/index.php/Data_Model

3. Edges: Knowledge relations, contextual relations, interactions.

Based on this description a social information space can defined as:

**Definition 1 (Social Information Space)** *A social information space* $S = (A, P, S_d)$ *contains a set of actors* $A$*, a set of content elements* $P$ *and optionally a set* $S_d$ *of additional contained social information spaces.*

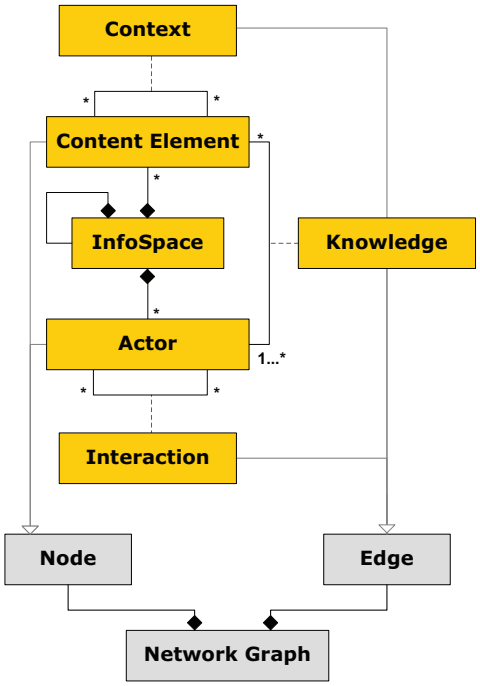


**Figure 3.1.:** UML class diagram of an universal actor content-element model
*Defining an universal actor-content-model for exploring temporal social and information networks* [169]

With these entities and relations the data of social information spaces can be extracted and saved in the data model. This process is implemented in an extractor. There is information, which cannot be extracted directly from the original data. Transformer gain this information in further steps. After extracting and transforming the data, a independent analysis is possible. Each entity and relation can be associated a timestamp to model the dynamic nature of the a social information space.

**Definition 2 (Extractor)** *An extractor transforms the data of a social information space into the generic data model.*

**Definition 3 (Transformer)** *A transformer gains further information based on the data in the generic data model. The new information is saved in the data model as well.*

## 3.2. Analysis based on the Data Model

Based on the data model, different analysis methods can be implemented. Figure 3.2 visualizes the analysis possibilities based on the data model. There are methods which are either independent or dependent of the social information space. The basic idea is the definition of metrics, which deliver results for interpretations. Dependent metrics are defined for the selected social information space (InfoSpace analysis). Independent metrics are defined for each social information space (InfoSpace analysis) or they depend on a network, which have to be defined before (network definition and network analysis). The metrics are described now and some examples are given.

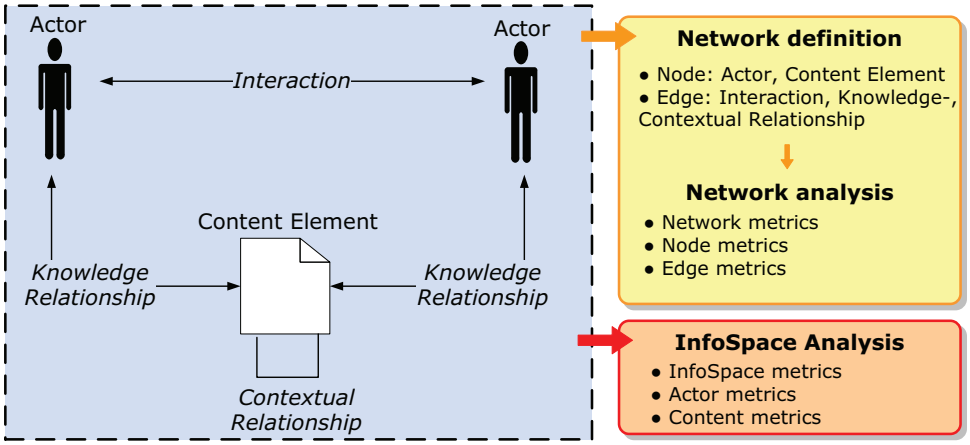


**Figure 3.2.:** Model of the social information space analysis

### 3.2.1. Specification of Networks

The definition of networks allows the analysis, independent of the social information space, but requires the definition of nodes, edges and optionally weights. Based on a social information space, nodes are *Actors* or *Content Elements*. Edges are defined by: *Contextual Relationship*, *Interactions* and *Knowledge Relationship*. A network is defined by:

**Definition 4 (Dynamic Network)** *A dynamic network* $G = (V, E, W, t_{start}, t_{end})$ *consists a set of vertices (nodes)* $V$ *and a set of edges* $E$*. A weighted graph requires additionally a set of edge weights* $W$*. A dynamic network includes a start* $t_{start}$ *and end time* $t_{end}$ *as well.*

In most cases a network represents a subset of a social information space, limited for example by the definition of a specific time period or content element type. In Wikis, these are a start and end date, a category or a namespace. But also the representation of the whole social information space by a network is possible. The definition of dynamic networks allows the analysis of the social information space development. Actually, two networks are implemented in the SONIVIS tool.
A social perspective is given by the *Collaboration Network*. In a *Collaboration Network* the nodes are *Actors* and the edges represent a collaboration between two *Actors*. A collaboration will be given, if two *Actors* edit mutually one *Content Element*.

Another network is the *WikiLink Network*, which is part of the information perspective of the social information space. In this network, nodes are *Content Elements* and the edges are generated by *Contextual Relationships* between the *Content Elements*. A formal definition of the *Collaboration Network* and *WikiLink Network* can be found in [169]:

**Definition 5 (Collaboration Network)** *A temporal Collaboration Network is an undirected graph* $G_C(V_A, E_C, W_{REV}, t_{start}, t_{end})$ *including a set of authors* $V_A$ *as nodes and a set* $E_C = (A_i, A_j); A_i, A_j \in V_A$ *of collaborations with* $i,j \in N, i \neq j$ *as edges.*
*Basically, an author* $A_i$ *creates a revision* $r_{t_i} \in REV$ *at a certain time* $t_i$*. This revision* $r_{t_i}$ *belongs to a page P, which consists of* $P = \{r_{t_0}, ..., r_{t_i}, ..., r_{t_n}\} \subseteq REV$ *revisions. A collaboration* $C = (A_i, A_j)$ *exist, if* $A_i$ *is author of* $r_{t_i}$*,* $A_j$ *is author of* $r_{t_j}$ *and* $r_{t_i}, r_{t_j} \in P$*. Based on the defined interval* $[t_{start}, t_{end}]$ *the revision is incorporated in the network.*
*The temporal Collaboration Network is a weighted graph, each edge has a specific weight based on the number of mutual revisions* $W_{REV}$ *of two connected authors. Only these revisions are considered which rely on the same content elements and which where established in the defined period of time* $[t_{start}, t_{end}]$*.*

**Definition 6 (WikiLink Network)** *A temporal WikiLink Network is a directed graph* $G_{WL}(V_P, E_L, t_{start}, t_{end})$ *consisting of a set* $V_P$ *of wiki pages and a set* $E_L = (P_1, P_2)$ *of Wikilinks from a page* $P_1 \in V_P$ *to a page* $P_2 \in V_P$*.*
*The first revision* $r_{t_0}$ *of a page* $P_i$ *should be before the upper boundary of the analyzed time interval:* $t_0 \leq t_{end}$*. Furthermore, links* $E_L$ *should exist in a revision* $r_{t_i}$ *with* $t_i \leq t_{end}$*. Only a subset of the previously introduced page types is used for this network:* $V_P = V_{AR} \cup V_{ME}$ *and* $E_L = E_{ARL} \cup E_{MEL}$*, where* $V_{AR}$ *are articles,* $V_{ME}$ *media,* $E_{ARL}$ *article-links and* $E_{MEL}$ *media-links. The existence of broken links are considered in a single measure.*

### 3.2.2. Network Analysis

Based on the described networks, independent metrics are defined now. Thus metrics for nodes, edges and the whole network can be defined. Metric examples are listed in Table 3.1.

***Node metrics***

| | |
|---|---|
| **Degree Centrality** | $C_D(v) = \frac{grad(v_i)}{\|V\|-1}$ |

***Edge metrics***

| | |
|---|---|
| **Betweenness Centrality** | $C_B(e) = \sum_{i,j \in V, i \neq j} \frac{\sigma_{ij}(e)}{\sigma_{ij}}$ ($\sigma$ is number of shortest paths between two vertexs i and j that go through e) |

***Network metrics***

| | |
|---|---|
| **Density** | $D(G) = \frac{2\|E\|}{\|V\|(\|V\|-1)}$ |

**Table 3.1.:** Metrics for the analysis of networks

A positional analysis is given by using node and edge metrics. These metrics analyze the nodes and edges concerning their properties and positions. Centrality metrics calculate for example the importance of a node or edge in the network.
On the other hand, network metrics are used to analyze the macro level. There are employed as a structural analysis by characterizing the global properties of a network. For example, the density of links in a *WikiLink Network* can be measured with the *Density* metric. In a *Collaboration Network* the *Density* describes the collaboration level between all *Actors*.

### 3.2.3. Information Space Analysis

Beside the definition of network metrics the definition of metrics, which analyzes the social information space directly is possible. Between general metrics and information space specific metrics are differentiated. At this point metrics for actors, the content and the whole information space are defined. General metrics are defined independent from the underlying social information space (see Table 3.2). Information space specific metrics depend on the social information space (see Table 3.3).
For example, the number of actors and content elements provide information about the information space size. The activity of an actor or content element in a social information space is computed by the *Edit Count* and the *Revision Count*.

| ***InfoSpace metrics*** | |
|---|---|
| **Total Author Count** | The number of authors. |

| ***Actor metrics*** | |
|---|---|
| **Edit Count** | The number of edits an author made. |
| **Minor Edit Count** | The number of minor edits an author made. |

| ***Content metrics*** | |
|---|---|
| **Revision Count** | The number of changes on a content element. |
| **Content Element Size** | The size of a content element in kByte. |

**Table 3.2.:** Metrics for the analysis of social information spaces

| ***InfoSpace metrics*** | |
|---|---|
| **Total User Count** | The number of users. |
| **Total Wiki Link Count** | The number of existing internal Wiki-links. |

| ***Actor metrics*** | |
|---|---|
| **Last Edit Date** | Date of last edit of an actor. |
| **Added File Count** | The number of added files of a certain actor. |

| ***Content metrics*** | |
|---|---|
| **Included Template Count** | The number of templates of a page. |
| **Included Wiki Link Count** | The number of included Wiki links on a page. |

**Table 3.3.:** Metrics for the analysis of Wikis

## 3.3. Extraction and Transformation of Wiki Data

The data for the analyses are Wikis in this research. Therefore the extraction and transformation of Wiki data have to be described. First an introduction to Wikis is given, then the extraction and transformation is introduced.

### 3.3.1. Introduction to Wikis

In a Wiki personal knowledge is made available and exchangeable in the form of articles, whereby a collectively created knowledge base evolves. There are differences between Wikis, concerning their goals, the used guidelines and the intended use.
A definition for Wikis is given by Müller et al. [168]. A **Wiki** is an ”open” website, in which each person can consume and edit the pages of the website. The versions of a page are saved in a history. The information, which is saved in pages, can be merged or split. Furthermore it is possible to define dependencies between the information context by linking specific pages. Thereby it is possible to define links to non-existing pages. The structure of a Wiki is not pre-defined. It will be built by the users in a collaborative self-organized process. Müller et al. writes further: ”Usually the final product of knowledge creation is visible but not the actual process of knowledge creation. With the page history the process from the first draft to the final version is traceable. A critical issue of knowledge work is solved in this way.”.

| | Pages | Users | Edits |
|---|---|---|---|
| **[en] Wikipedia** | 15,831,582 | 8,868,132 | 282,645,816 |
| **[ge] Wikipedia** | 2,463,032 | 694,762 | 58,282,909 |
| **[en] Wiktionary** | 1,245,398 | 146,349 | 6,205,862 |
| **[ge] Wiktionary** | 110,173 | 15,284 | 946,558 |
| **[en] Wikibook** | 119,917 | 172,702 | 1,425,304 |
| **[ge] Wikibook** | 42,824 | 13,656 | 435,514 |
| **[en] Wikinews** | 87,762 | 81,219 | 762,157 |
| **[ge] Wikinews** | 38,489 | 4,886 | 447,606 |
| **[en] Wikiversity** | 70,434 | 83,571 | 402,239 |
| **[ge] Wikiversity** | 15,858 | 4,314 | 144,463 |

**Table 3.4.:** Size and activity of MediaWiki projects [2009/02/04]

There are a lot of Wiki implementations, like TWiki[2], Trac[3], MediaWiki[4] and JSPWiki[5], available. One of the most popular Wiki software is MediaWiki. **MediaWiki** is an open source software (GPL[6]), that is developed by the Wikimedia Foundation[7] and volunteers. A

[2] http://twiki.org/
[3] http://trac.edgewall.org/
[4] http://www.mediawiki.org/
[5] http://www.jspwiki.org/
[6] GPL: General Public License, http://www.gnu.org/licenses/gpl-3.0.html
[7] Wikimedia: The Wikimedia Foundation is a non-profit foundation, whose aim is the promotion of free knowledge.

list of important Wiki projects, based on the MediaWiki software, is given in Table 3.4. One project which based on the MediaWiki software are Wikipedia. **Wikipedia** is an internet encyclopedia, which is developed by the cooperation of multitude of users. The following analyses are based on the English Wikipedia, wherefore some parts of the Wikipedia are extracted.

A Wiki contains a set of pages, which are linked with each other. To each page belongs a list of revisions, which expresses the page content at a certain time. Each revision is written by one author. The last revision of a page presents the current page content. It is distinguished between different page types (namespaces). The main pages are called articles including the texts of specific topics. Additionally, there are further page types such as discussion, user and category pages. In a Wiki the following entities exists:

***Page:*** A Wiki page is a text resource in a Wiki project, which can be edited by everyone. It has an unique name within the Wiki project. To each page another page is assigned, the so-called discussion page. The discussion page includes discussions about the content of a page.

***Redirect:*** The redirection mechanism allows that a page can be accessible under several names. This is an effective technique for synonym handling, e.g. the page about pail can have a redirect to bucket: `#REDIRECT [[bucket]]`.

***Revision:*** Another very important concept in Wikis is the storage of the complete history of a page using revisions. Each editing of a page creates a new revision.

***Wiki-Links:*** Wiki-links are internal links between pages. A Wiki-link corresponds to a hyperlink in the world wide web, whereas Wiki-links are constricted by the corresponding Wiki project. There are the following link types:

- Internal links between pages
  `[[PAGE]]`,
  `[[PAGE|PAGEDESCRIPTION]]`, `[[PAGE#CHAPTER]]`
- Links to image description pages
  `[[Image:IMAGENAME|IMAGEDESCRIPTION]]`
- Links to files
  `[[Media:FILENAME|FILEDESCRIPTION]]`

***Namespace:*** With namespaces a differentiation based on the function of pages is established. It allows the summarization of pages to groups. All namespaces are addressed over a prefix by the page name, e.g. `[[User:Alice]]` links to the user page of *Alice*.
There is a list of pre-defined namespaces (see Table 3.5), but further namespaces can be introduced. Some important namespaces are:

- ***Main***: The main namespace includes the articles of a Wiki. Not all pages in a Wiki are articles. An article is a page, which defines a concept. For example a Wikipedia article includes special knowledge to certain topics or terms. Counterexamples are user pages, category pages, redirects, etc.
- ***Template***: Templates allow the integration of the content of a page into another page. If a content is used many times, it is possible to define a template for the content. The

template will be included in the corresponding pages. Templates can be used for text formations, the definition of a navigation, text which is used several times, etc.

- ***Category***: Another special kind of pages are categories, which realizes the summarization of pages to groups, e.g. the grouping of pages according to different topics. The category is described on the category page. A page can belong to more than one category. A category can be assigned to another category as well. Links are used to assign a page to a category, e.g. `[[Category:University]]`.

| Index | Name | Purpose |
|---|---|---|
| 0 | Main | *Real* content; articles |
| 1 | Talk | |
| 2 | User | User pages |
| 3 | User talk | |
| 4 | Project | Information about the Wiki |
| 5 | Project talk | |
| 6 | File | Media description pages |
| 7 | File talk | |
| 8 | MediaWiki | Site interface customisation |
| 9 | MediaWiki talk | |
| 10 | Template | Template pages |
| 11 | Template talk | |
| 12 | Help | Help pages |
| 13 | Help talk | |
| 14 | Category | Category description pages |
| 15 | Category talk | |

**Table 3.5.:** WikiMedia namespaces [2009/06/17]
http://www.mediawiki.org/wiki/Manual:Namespace

***Users:*** A user is a person who has an account in the corresponding Wiki. A user can have a role[8] [9], e.g. announced users, administrators, stewards, bots, etc.

***Author:*** Authors are persons, which are editing articles of a Wiki. Authenticated authors (authors who can be identified by their user account) and anonymous authors (authors who can be identified by their IP address) can be differentiated.

### 3.3.2. Extraction and Transformation

The extraction and transformation of the Wiki data to the generic data model is described now. Therefore the items in a Wiki (Section 3.3.1, p. 18) have to be saved as entities and relations in the generic data model (Section 3.1, p. 13). The extractions and transformations are listed in Table 3.6.
In the data model, two types of entities exists: Actors and content elements. Actors are the users and authors of a Wiki. Content elements are pages (articles, categories, templates, etc.)

[8] http://en.wikipedia.org/wiki/Wikipedia:User_access_levels
[9] http://de.wikipedia.org/wiki/Hilfe:Benutzer

and revisions. The different types of pages and authors are differentiated by properties of the actors and content elements.
Additionally the following relations are saved in the data model. A knowledge relation is given, if an author had created a revision of a page. Wiki-links are contextual relations. Interactions are not explicitly given in a Wiki. A transform step is needed. If two authors mutually edited the same page, an interactions is given.

| **Data Model** | **Wiki Data** | **Data Acquisition** |
|---|---|---|
| Content elements | Pages | Extraction |
| | Revisions | Extraction |
| Actors | Authors | Extraction |
| | Users | Extraction |
| Knowledge relations | ”Author creates a revision” | Extraction |
| Contextual relations | Wiki-links | Extraction |
| | Category-links | Extraction |
| Interactions | ”Two authors mutually edited the same page” | Transformation |

**Table 3.6.:** Extraction and Transformation of wiki data into the data model

**Summary:** In this chapter the generic data model for the analyses is described. Since Wikis are the example social information space in this work, the extraction and transformation of Wiki data to the data model is introduced. Furthermore it is described how analysis can be executed based on the data model. The content of this chapter makes it possible to analyze Wiki data in the data model and to integrate analysis methods for topic detection in the SONIVIS tool. In the next step, the analysis methods for topic detection have to be selected.

# 4. Text Mining - Fundamentals, Methods and Applications

This chapter introduces the different possibilities of text analysis. Based on this chapter a selection of methods for data preprocessing, data representation, dimension reduction and clustering is possible. The fundamentals and methods which are used in this work are described. First a definition of text mining and a dissociation to the adjacent research fields is given. Another part of this chapter describes the important fundamentals of the research fields. Afterwards an overview of text analysis methods is given. The last section includes a definition of all research areas, describes which methods each research field has developed and gives some application examples.

## 4.1. Introduction

For the analysis of documents different research fields are existing. Beside text mining, also information retrieval, information extraction, natural language processing and artificial intelligence are research fields for the analysis of text documents.

| **Research fields** | **General** | **Scholar** |
|---|---|---|
| Data Mining [DM] | 14,000,000 | 548,000 |
| Artificial Intelligence [AI] | 9,770,000 | 1,020,000 |
| Information Retrieval [IR] | 2,800,000 | 579,000 |
| Natural Language Processing [NLP] | 1,160,000 | 202,000 |
| Text Mining [TM] | 682,000 | 25,100 |
| Information Extraction [IE] | 513,000 | 49,500 |

**Table 4.1.:** Publicity of research fields (estimated)
http://www.google.com/ [2009/04/15]

The definition of text mining and adjacent research fields are not clear. Some definitions indicates that adjacent research fields are a part of text mining or the disciplines overlaps. Often different names are used for a research field. Consequently a differentiation of text mining in relation to other disciplines is not easily possible. One reason for this is the age of each research field. Recent research fields like the text mining have usually still no exact definition. The tasks of older research fields are supplemented by further requirements with the time, in order to obtain better results. Thus many methods are used by several research fields as well. A view of the google hits in Table 4.1 gives a quantitative overview of the size and the age of each research field. A large or an old research field has more hits, like Data Mining or Artificial Intelligence. Small and new research fields have fewer links, like Text Mining or Information Extraction. The age of a research field can influences the size. This means that an older research field is mostly larger than new research fields.
Kosala et al [135] defines information extraction and text categorization as a subarea of

text mining, whereas Hearst et al [96] excludes this research fields from text mining. Another definition is given by Dorre et al [56], who regard information extraction and text categorization as equal. Thereby he constitutes an intersection between text mining and text categorization.

| | **Novel knowledge** | **Already-known facts** |
|---|---|---|
| **Struc. data** | DM | Database Query |
| **Unstruc. data** | TM | IE, IR |

**Table 4.2.:** Differences between research fields, Yoo et al [236]

A differentiation between the research fields is given by Yoo et al [236]. He distinguishes text mining, data mining, information retrieval and information extraction by the structure of data and the type of knowledge. In his opinion text mining is different from data mining, because text mining handles unstructured data. Information extraction fetches already-existing documents, whereby text mining looks for novel patterns in text. Information extraction extracts facts or events of interest, whereby text mining discovers novel knowledge. Table 4.2 illustrates this differentiation.

| | |
|---|---|
| **Information Retrieval Perspective**<br>Jacobs [111], Salton [187],<br>Allan [6], Goser [74] | **Data Mining Perspective**<br>Rajman [180], Kodratoff [131], Losiewicz [149], Hearst [96] [97], Dorre [56], Feldman [62], Han [88] |
| **Methodically Perspective**<br>Joachims [116] | **Knowledge-based Perspective**<br>Hearst [96] [97], Kao [122], Nahm [171], Haralambos [92], Hotho [104], Heyer [98], Feldman [61] [62] |

**Table 4.3.:** Perspectives of text mining definitions

Mehler et al [161] tries to distinguish the definitions by the division into four perspectives. The found text mining definitions are classified in his perspectives in Figure 4.3. Some definitions are part of more than one perspective.

1. **Information retrieval perspective:** Text mining was introduced, in order to adjust deficits of the information retrieval. In this perspective the information retrieval is extended by text analysis methods like text summarization and the extraction of relevant information. It is called content-based information retrieval.

2. **Data mining perspective:** In reference to data mining, text mining is a natural extension of data mining. The goal of text mining is the extraction of useful information from texts data sources through the identification of interesting patterns.

3. **Methodical perspective:** This perspective defines text mining as a set of methods for the analysis of huge amount of texts.

4. **Knowledge-oriented perspective:** This is the knowledge-based containment. It defines text mining as the discovery of heretofore unknown and never before encountered information, by automatically extracting information from texts and linking them together.

In this work the research fields are differentiated by their goals. Figure 4.1 visualized my approach. Natural language processing includes the analysis of natural language, information extraction extracts information from a text, information retrieval searchs for relevant information (e.g. documents,...) and text mining discovers new information. Data mining discovers new information as well, but data mining uses structured data. Artificial intelligence is the research field for developing intelligent machines. Data Mining and artificial intelligence uses no unstructured data, but they are closely related to the other research fields. The methods of each research field are mostly used in other research fields. Sometimes research fields are connected for developing an application. For example search engines uses additionally text mining to enhance search results.
First, there is a fundamental area. The using of fundamentals, like linguistic and mathematics, makes it possible to define methods. For example, the fundamentals of linguistic are used by tokenization and stemming. The statistic supports the identification of co-occurrences with (conditional) probabilities. Zipfs law is based on statistic as well. The vector space model for representing text documents was defined by linear algebra. The graph theory can be used by network analysis. The logic fundamentals include the propositional and predicate logic, which can be used by boolean retrieval, word-sense disambiguation and by methods from artificial intelligence. Another fundamental area is visualization, which includes the basics for the representation of data. For example, with the visualization of data the visual abilities of the viewer should be considered. The fundamentals can be used by creating charts, networks, etc.
The developed methods are not used only by one research field, but they have their origin in a certain one. That is highlighted by the color of the rectangles. Text mining uses for example natural language methods (stemming), data mining methods (clustering) and information retrieval methods (vector space retrieval).
The research fields natural language processing, information extraction, information retrieval and text mining concern themselves with text analysis. Data mining and artificial intelligence are adjacent research fields. The research fields are differentiated by their goals.
Based on the research fields different applications are developed. An application can be assigned to a certain research field as well. In addition, the combination of different research fields is possible. Nahm et al. [171] developed a framework, called DiscoTEX (Discovery from Text EXtraction). It is a text mining framework which includes information extraction and data mining methods as well. This framework transforms text into more structured data which can be mined for interesting relationships. Jarosch et al. [114] represents an approach, where methods of artificial intelligence are merged into communication with an information retrieval system. Knowledge based techniques for information retrieval are represented by Tong et al. [214].
In the following sections each area of Figure 4.1 is described. Therefore the diagram is taken up several times and the area, which is described in the according section, is highlighted. The following examples are based on the book *Alice's Adventures in Wonderland*[1]. Each chapter of the book represents a text documents.

[1]Lewis Carroll, *1832, http://www.gutenberg.org/files/11/11-h.zip

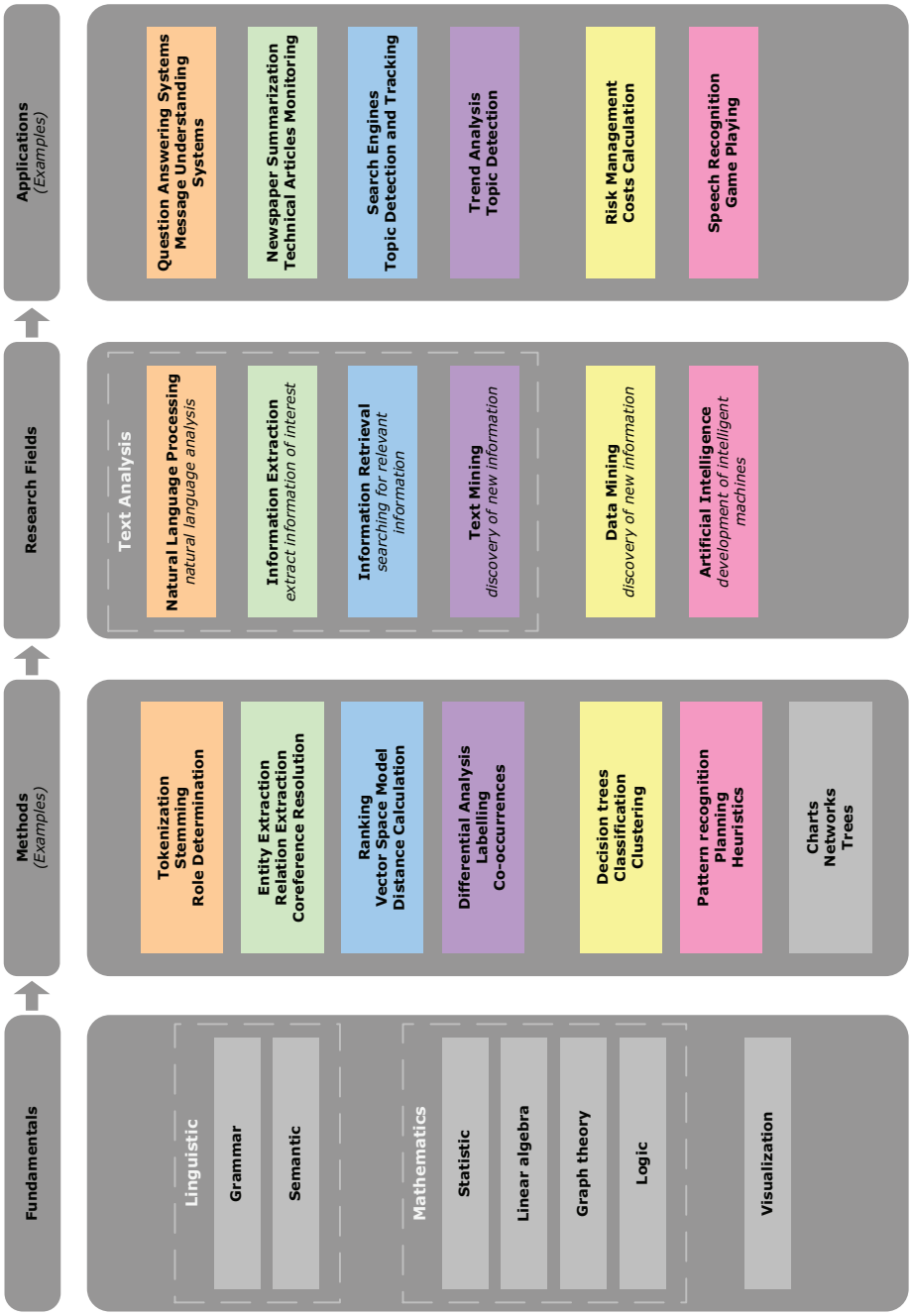


**Figure 4.1.:** Research fields

## 4.2. Fundamentals

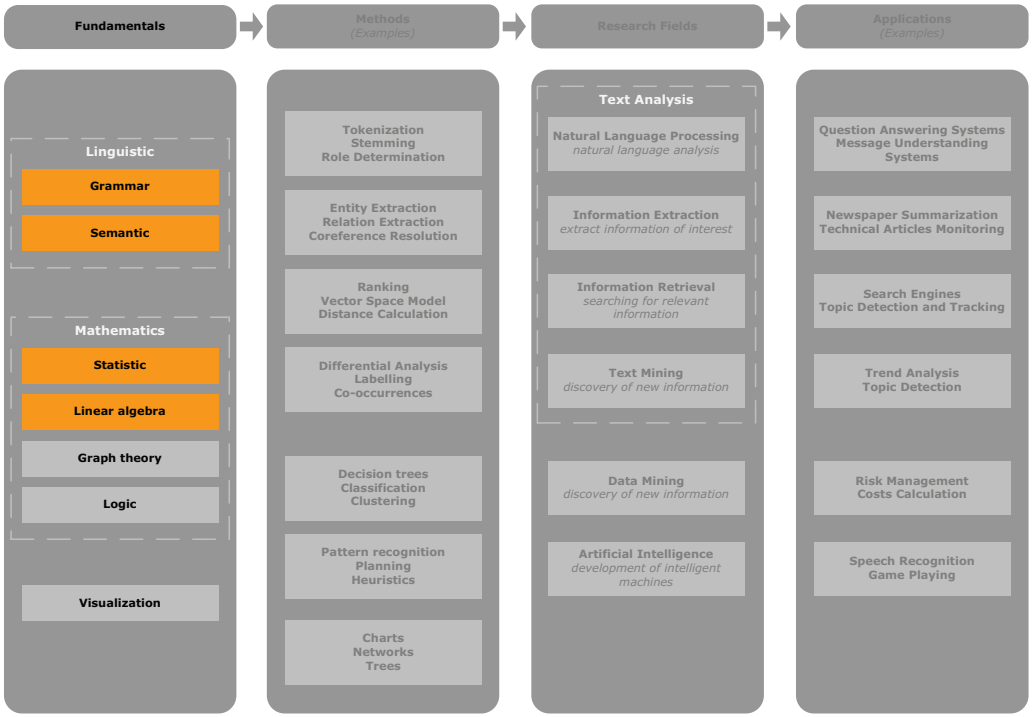


**Figure 4.2.:** Text analysis: fundamentals

This section introduces the important fundamentals of this work. These are highlighted in figure 4.2. Based on this fundamentals, methods of text analysis are developed. The basics of linguistic, language statistic and linear algebra are described.
The theory of linguistic is used for structuring the text, e.g. the extraction of words, phrases, etc. or the identification of nouns, verbs, etc. Furthermore the linguistic can be used for the identification of keywords. The language statistic detects statistic regularities in texts. These language rules can be used, for example, for the identification of keywords or co-occurrences. The vectors and matrices from linear algebra can be used to represent the terms of a document. Based on these vectors and matrices, operations can be executed to gain more information about a document or a set of documents.

### 4.2.1. Linguistic

In this section the linguistic fundamentals used by computer-assisted text analysis are presented. Compared to data mining, text analysis is adopted on unstructured data, which are saved in documents. A document contains text, which was formed by a natural language. It is possible to extract important information from a text, because a natural language is based on rules and therefore, a text structure exists. In this context, grammar and semantics are very important for text analysis.

Figure 4.3 represents the fields of linguistic and highlights the important ones for text analysis. A description of the linguistic fundamentals and its usage for text analysis is given in [226], [98] and [201].

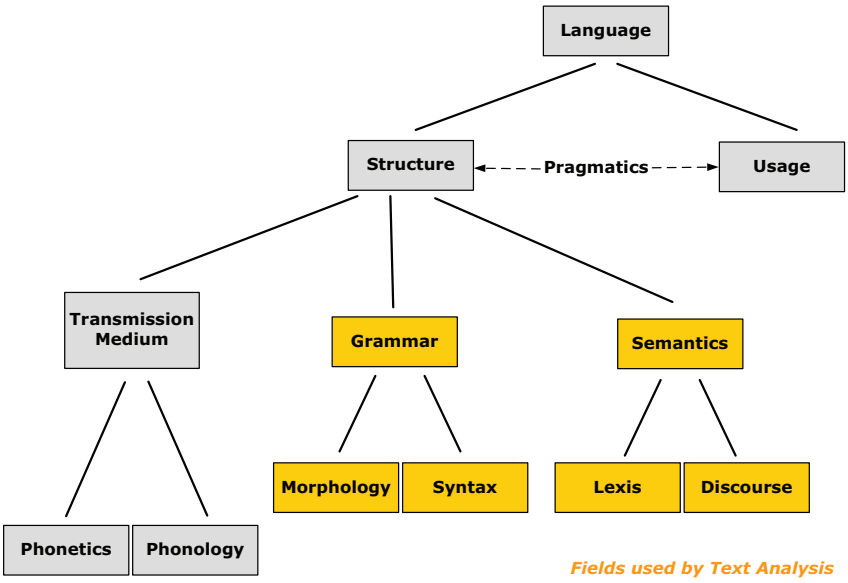


**Figure 4.3.:** Structural model of language,
*The Cambridge Encyclopaedia of Language* [43]

### Grammar

The field of grammar contains the rules, which constitute the system of a language. It defines the structure of words (morphology) and sentences (syntax).
**Morphology** is the study of the composition of words. A **word** is the combination of morphemes and a **morpheme** is the smallest meaningful unit in a language. Thereby free and bound morphemes are distinguished. A morpheme is free if it can exist meaningfully on its own, like *book*, *read* and *nice*. It is the base or stem of a word. A bound morpheme never occurs by itself, but is used alongside a free morpheme. The use of a bound morpheme changes the meaning of the stem. Affixes (prefix, suffix, infix and circumfix) are bound morphemes. Examples are *un*read (prefix), book*s* (suffix) and read*ing* (suffix). A set of words belong to a base word called **lexeme**. For example the words *read*, *reads* and *reading* belong to the lexeme *read*. Lexemes are the fundamental unit of a lexicon.
The rules of **syntax** describe, the way that words in phrases and sentences may be combined. The units of a phrase or sentence correlating to each other. A **phrase** is a group of two or more linked words that functions as a single unit in the syntax of a sentence, like *white house* or *by the way*. A **sentence** is the largest independent unit of grammar, that expresses a complete idea. Conventionally, a sentence includes a subject and a verb. It begins with a capital letter and concludes with an end mark (a period, an exclamation point or a question mark).

For example *Alice led the way, and the whole party swam to the shore.*
For the computer-assisted processing of documents, linguistically relevant units (words, phrases and sentences) have to be extracted first. The extracted words can be reduced to their word stem with the help of morphology. This process makes it possible to reduce the complexity of the analysis and to represent the word meanings. On the number of words with the same word stem can be decided how important these are. The syntactical analysis allows the identification of nouns, verbs, adjectives etc. Using morphology informations and syntactical rules can help to recognize structures in the form of word- and phrases patterns. These are the basis of the semantic analysis.

### Semantic

The field of semantics concerns itself with the meaning of words, phrases, sentences and texts. It is important to understand language in social contexts. The semantic analysis comprises for example the study of how and what the words of a language denote (**lexis**) and the analysis of sentence-comprehensive semantic relations (**discourse**).
The differentiation of **homonyms** (word with different meanings, such as *bank*) and **polysems** (word with similar meanings, such as *wood*), the identification of **synonyms** (different words with an identical or similar meaning, such as *bucket* and *pail*) and **antonyms** (different words with an opposite meaning, such as *short* and *tall*) and the search for **collocations** (expressions of two or more words, which make sense together, such as *Your Majesty*) are central parts of the semantic. Another example of semantic analysis is the search for co-occurrences of a word. **Co-occurrences** are words, which often occur together in a text.
A **glossary** is a list of difficult, technical or specialized terms with their definitions. In the context of semantics a **taxonomy** is a hierarchical structure of words. This type of classification points out similarities between words. A **thesaurus** is a dictionary which contains the semantic relations of words, like synonyms, antonyms and contrasting words. A **topic map** is a collection of topics, their relations and information sources, which are relevant for the topics. It defines a model of knowledge structures. In this model, the knowledge of a topic is collected. Based on topic maps, ontologies can be formulated. The definition of an **ontology** is different. Heyer et al. [98] defines an ontology as a formalization of a common understanding of words for an application or subject. Furthermore it is possible to define relations and derivation rules between the words.
Computer-assisted text analysis is mostly applied on a corpus. A **corpus** is a collection of documents, e.g. articles, newspapers, web pages and eMails. Some analyses uses a second collection for comparative purposes, called **reference corpus**. With a reference corpus the significant terms of a corpus can be extracted. These are the terms, which occur more frequently in the corpus than in the reference corpus. For analyzing a corpus, the terms of each document are extracted first. A **term** is a word or a phrase. The set of all unique terms in the corpus is called **lexicon**.
Further methods allow the extraction of keywords. **Keywords** are significant or descriptive terms, which describe a document very well. These terms are often used for indexing a document.
A corpus of a specific field includes a specialized vocabulary, called **technical terminology**. It includes all technical terms of the field, which are generally usual. If the technical terms are well-known, the analysis will be eased.

### 4.2.2. Language Statistic

The words in a text are not combined randomly, but according to the syntactic rules of a natural language. Thus it is possible to detect statistic regularities in texts. These can be used by further analysis, e.g. the difference analysis or the co-occurrences analysis. A detailed description of statistic fundamentals is given in [98].

#### Zipfs Law

The first analysis of the word frequency distribution is given by George K. Zipf[2]. He detected that a natural language follows the *principle of least effort* [246]. Frequently used words are usually short or they become shorter over the years, such as *automobile-auto-car* or *television-tele-TV*.

Zipf detected that the `rank r` multiplied with the `frequency n` of any word is nearly constant (see Table 4.4, p. 31). The value of the `constant k` however is determined by the size of the text corpus.

$$r \cdot n \approx k \qquad n \approx \frac{k}{r} = k \cdot r^{-1} \tag{4.1}$$

This behaviour arises because the frequency of any word is inversely proportional to its rank. This means that the rank depends on the frequency.

$$n \sim \frac{1}{r} \tag{4.2}$$

Using the logarithm the equation will be transformed into linear form. Now the equation can be draw as a straight line in a diagram.

$$log(n) = log(k) - log(r) \tag{4.3}$$

Let `N` be the `total number of all words` and `n` the `frequency of a certain word`. The `relative frequency` of a word is defined by:

$$\frac{n}{N} \tag{4.4}$$

The `rank position` $r_n$ defines the highest rank of all words, which have the frequency `n`. The `constant c` depends on the selected text.

$$r_n = c \cdot \frac{N}{n} \tag{4.5}$$

The language dependent constant c will be determined with the following equation:

$$c = r \cdot \frac{n}{N} \approx \frac{k}{N} \tag{4.6}$$

[2]George Kingsley Zipf, 1902–1950, Harvard University, Linguist who studied the major statistical regularities in language.

Let `t` be the `size of the lexicon`. It is the set of all unique terms. The size of a lexicon can be estimated with the Equation 4.5:

$$t = max(r_n) = c \cdot \frac{N}{1} = c \cdot N \tag{4.7}$$

The `number of words, which occur n times` is defined by $I_n$ and can be calculated by:

$$I_n = r_n - r_{n+1} = c \cdot \frac{N}{n} - c \cdot \frac{N}{n+1} = c \cdot \frac{N}{n(n+1)} \tag{4.8}$$

Since the important words of a text do not occour very frequently, this equation allows an estimation about the number of important words in a text. For example half of the words occur only one time (n=1).

$$I_1 = c \cdot \frac{N}{2} = \frac{t}{2} \tag{4.9}$$

Zipfs law allows statements about the word frequency distribution, the vocabulary in a text and the increase of a vocabulary with growth of the text. Zipfs law can be used to estimate the minimum size of the reference corpus.
In this law all words of a text are ordered by decreasing frequency and each word has a rank. In most cases the word *the* is the most frequently word and gets the rank one. An example is given in Table 4.4. In the book *Alice's Adventures in Wonderland* the term *Alice* has a high frequency and gets the rank 10. The rank multiplied by the frequency results in 3860. The results of the other words are similar (4150, 4800, 3500, 3000). Consequently the law is reflected by the book.

| **Rank r** | **Word** | **Frequency n** | **r · n** |
|---|---|---|---|
| 10 | alice | 386 | 3860 |
| 50 | again | 83 | 4150 |
| 100 | never | 48 | 4800 |
| 500 | listen | 7 | 3500 |
| 1000 | pieces | 3 | 3000 |

**Table 4.4.:** Rank table *(Alice's Adventures in Wonderland)*

### Mandelbrot-Zipfs Law

The Mandelbrot-Zipfs law enhances Zipfs law in the range of low and high ranks. It was defined by Mandelbrot[3]. The law includes to constants $c_1$ and $c_2$. The constants depend on the distribution of word frequencies. Word frequencies depend on the language and the kind and field of activity of a text corpus. Generally, the exponent $1 + c_2$ is close to 1. The

[3]Benoit B. Mandelbrot, *1924, Mathematician, best known as the father of fractal geometry.

`constant` $c_1$ influences the range of low ranks and the `constant` $c_2$ influences the range of high ranks.

$$(r + c_1)^{1+c_2} \cdot n \approx k \tag{4.10}$$

$$n \sim \frac{1}{(r + c_1)^{1+c_2}} \tag{4.11}$$

Using the logarithm, the equation will be transform as follows:

$$log(n) = log(k) - (1 + c_2) \cdot log(r + c_1) \tag{4.12}$$

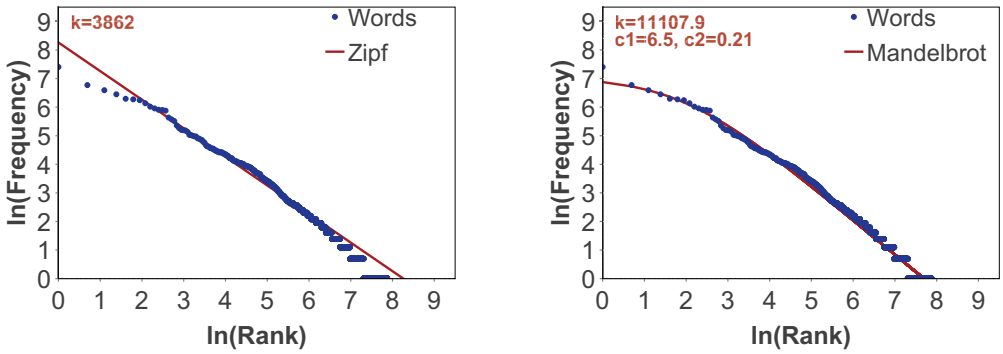


**Figure 4.4.:** Zipf and Mandelbrot *(Alice's Adventures in Wonderland)*

The goal of Zipfs law is to find an approximated value of the constant k, which describe the distribution very well. Zipfs law of the book *Alice's Adventures in Wonderland* is described by the function $n = \frac{3862}{r}$. The Mandelbrot-Zipfs law needs additionally the values for $c_1$ and $c_2$: $n = \frac{11109.7}{(r+6.5)^{1.21}}$. The charts of 4.4 represent the word distribution of *Alice's Adventures in Wonderland*. The laws are helpful by the extraction of the important words of a text. Words with a high frequency (low ranked words) are unimportant and not relevant for further analysis. The analysis of the dependency between rank and frequency can be helpful for stylistic analysis and author detection as well.

### Conditional Probabilities

Conditional probabilities are another statistical instrument used by text analysis. The probability of an event A under the condition that the event B already occurred is called conditional probability $P(A|B)$. For example, the conditional probability of the co-occurrence of two words is used by the detection of significant co-occurrences.

### N-gram Model

The n-gram model uses conditional probabilities. It is a probabilistic language model and for example it concerns itself with the correctness of sentences. The unigram assigns a probability $P(w_i)$ to each word. By a bigram additionally the previous word is considered: $P(w_i|w_{i-1})$. A n-gram consider n previous words: $P(w_i|w_{i-(n-1)}...w_{i-1})$. The probability of a sentence can be calculated by the multiplication of all word probabilities.

#### Hidden Markov Model

The Hidden Markov model (HMM) is an extension of the n-gram model. This statistical model allows to assign properties to a language and probabilities to linguistic events. It is defined as a stochastic finite state machine $(S, A, B, O, \Pi)$: S - hidden states, A - transition probabilities, B - emission (output) probabilities, O - observations (output alphabet) and Π - initial state probabilities. The model is used by tagging, machine learning, typing errors correction and speech recognition.

### 4.2.3. Linear Algebra

Since the linear algebra is a fundamental part of many text analysis methods, this section introduces vectors, matrices and linear algebra rules. The mathematical notations are used in the further sections and chapters of this work as well. The use of these fundamentals are descripted detailed in Section 4.3 (p. 35).

#### Vector

A vector $\vec{v} = v_1, \ldots, v_n$ includes $n$ values. $v_i$ indicates the value of the i-th position. The mean of the values of a vector $\vec{v} = v_1, \ldots, v_n$ are defined by $\overline{v}$:

$$\overline{v} = \frac{1}{n} \sum_{i=1}^{n} v_i \tag{4.13}$$

In the context of text analysis, a document or a term can be represented by a vector. For example a document vector includes the frequency of each term in the document. A term vector includes the frequency of the term in each given document.

#### Matrix

A $r \times c$ matrix $X = (x_{ij})$ includes $r$ rows and $c$ columns. $x_{ij}$ indicates the value of the i-th row in the j-th column. The values of the i-th row can be indicated by $x_{i.}$ and $x_{.j}$ represents all values of the j-th columns. In these cases the values are represented by a vector. For getting all values, $x_{..}$ can be used.
A matrix can represents a set of documents and their terms. In this case each row includes a document vector. Furthermore a matrix can be used for describing the dependencies between terms, documents or authors. Here each row and column is a term, document or author and the values of the matrix describing their dependencies. Based on the matrices, different matrix operations can be executed.

#### Singular Value Decomposition

The singular value decomposition (SVD) converts the matrix $A$ into 3 matrices. The matrix $T$ includes the left singular vectors and span the rows of $A$. The singular vectors (columns of $T$) are the eigenvectors of $A^T A$. The matrix $D$ is the matrix of right singular vectors and span the columns of $A$. The singular vectors (columns of $D$) are the eigenvectors of $AA^T$. $S$ is

the diagonal matrix of singular values. The size of a singular value describes the diffusion of the dimension (dilation factor). The singular values have a decreasing order and characterize, similar to eigenvalues, the properties of a matrix. Singular values are the roots of the eigenvalues of $A^T A$. Because the columns of $T$ and the columns of $D$ including singular vectors and they have an orthonormal base, the matrices $T$ and $D$ are orthogonal ($TT^T = I$, $DD^T = I$).

$$\underset{(n \times m)}{A} = \underset{(n \times r)}{T} \quad \underset{(r \times r)}{S} \quad \underset{(r \times m)}{D^T} \tag{4.14}$$

The SVD can be applied on matrices which are formed by terms, documents and authors. This allows the revealing of hidden relations between terms, documents or authors. For example, the latent semantic analysis uses SVD to reveal the semantic information. The latent semantic analysis is described in Section 4.3.3 (p. 48).

### Matrix Diagonalization

Matrix diagonalization, known as eigendecomposition or spectral decomposition as well, factorizes a matrix $A$ into 3 matrices. The matrix $\Phi$ includes the eigenvectors $\varphi_1, \varphi_2, ..., \varphi_n$ of the matrix $A$. The matrix $\Lambda$ includes the eigenvalues $\lambda_1 > \lambda_2 > ... > \lambda_n$ as diagonal. The eigenvalues have a decreasing order and the eigenvectors have to be unit, i.e. their lengths are one.

$$\underset{(n \times n)}{A} = \underset{(n \times n)}{\Phi} \quad \underset{(n \times n)}{\Lambda} \quad \underset{(n \times n)}{\Phi^T} \tag{4.15}$$

Like singular value decomposition, matrix diagonalization can be applied on matrices to reveal hidden relations. For example, the principal components analysis uses matrix diagonalization for their calculations. The principal components analysis is described in Section 4.3.3 (p. 51).

## 4.3. Text Analysis Methods

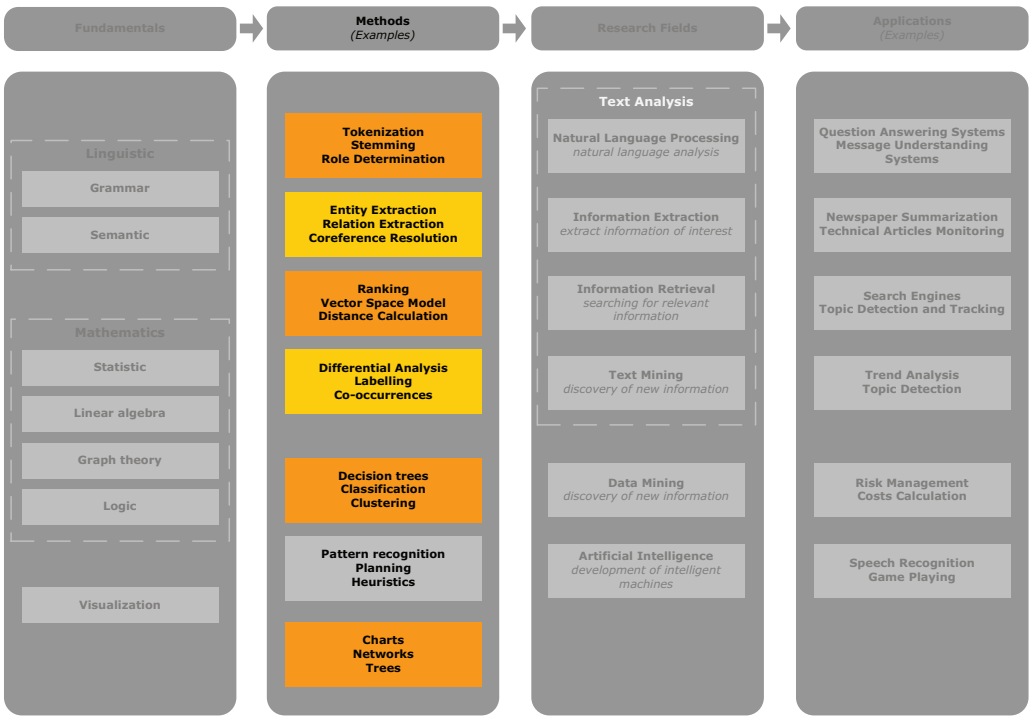


**Figure 4.5.:** Text analysis: methods

In this section the text analysis methods, used in this work, are described in detail (tokenization, stop word removal, stemming, term weighting, table model, vector space model, dimension reduction and clustering). Additionally an overview about further methods, which are used in text mining, is given. Based on the given fundamentals, which are described in Section 4.2 (p. 27), methods for text analyses can be defined. Figure 4.5 highlights the methods, which are described now.

Figure 4.6 visualizes the important text mining methods. The methods can be differentiated into preprocessing, representation and analysis methods. Given a set of documents, called corpus (defined in Section 4.2.1, p. 29), the text has to be preprocessed in most cases. The preprocessing methods extract the terms of a text, remove stop words, extract multiword expressions, etc. In this step the text will be structured. In the next step the structured text is transformed in a processable form, e.g. tables, vector space model or a simple string. Now the structured text can be saved in a database. Other storage possibilities are files. This allows a later re-using without the preprocessing step. Based on the representation different analysis methods can be executed. Sometimes a reference corpus is used [98]. The results of the analysis can be used for a second preprocessing step.

The preprocessing and representation methods are described in the following two sections (preprocessing methods in Section 4.3.1, p. 36 and representation methods in Section 4.3.2, p. 42). The analysis methods dimension reduction (Section 4.3.3, p. 46) and clustering (Section 4.3.4, p. 63) are described in separate sections. Further analysis methods are explained in the last section (Section 4.3.5, p. 75).

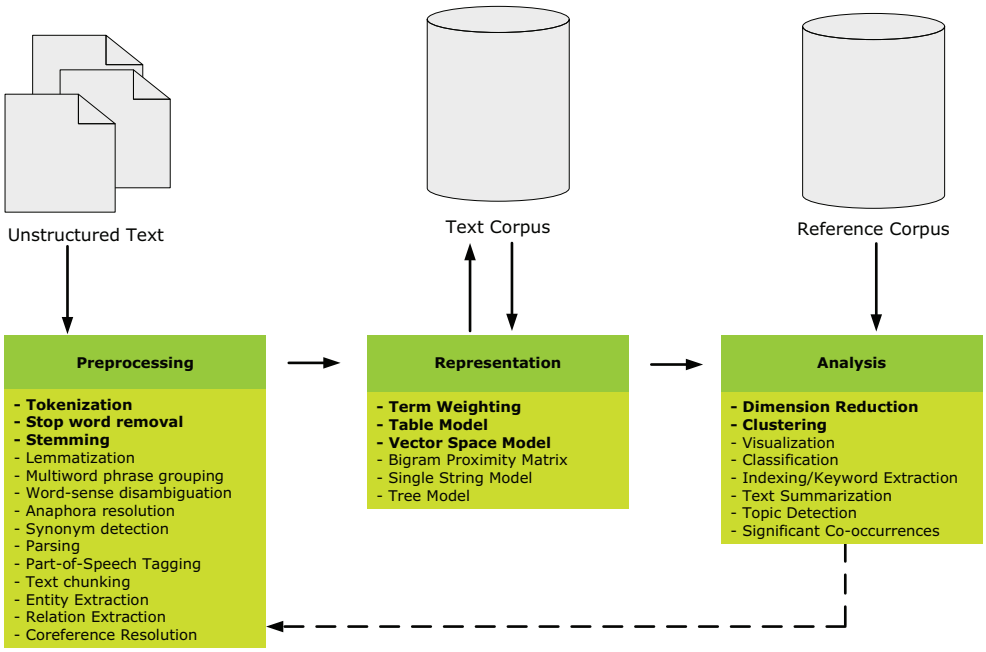


**Figure 4.6.:** Important analysis methods for text mining

### 4.3.1. Preprocessing Methods

In this section methods for text preparation, such as term extraction and text structuring, are presented.

#### Tokenization

Tokenization, called lexical analysis as well, is the process of splitting a text document into meaningful units, like sentences or terms (word or phrases). Thereby all punctuation marks, tabs and non-text characters are replaced by single white spaces. The main problem is the identification of sentence boundaries, which can be recognized by a punctuation mark. But a punctuation mark is applied in many ways, like *Mr.* or *Dr.*. For this reason abbreviation lists are used to differentiate between abbreviations and the end of a sentence.
Let D be a set of text documents called corpus. The result of the tokenization process is a list of terms $T_i = t_1, \ldots, t_n$ and maybe a list of sentences $S_i = s_1, \ldots, s_n$ for each document $d_i \in D$. In most cases the terms are only words and no phrases. Later preprocessing steps summarize words into phrases, like the multiword phrase grouping (described in Section 4.3.1, p. 39). The field of grammar builds the linguistic fundamentals of this method (described in Section 4.2.1, p. 28).

### Stop Word Removal

All terms of a document are extracted by the tokenization step. The generated lexicon is very large and contains unimportant words. The goal of stop word removal is the decrease of the size of the lexicon. In this context the not meaningful content will be removed, like articles (*the*), conjunctions (*and, or*), prepositions (*on, in*), etc. The not meaningful contents are the **stop words** of the text documents. Stop word removal is applied on the level of words. There are two approaches for the stop word removal process.
One possibility is the removal of words by using a stoplist. Commonly-used stoplists are for example:

- Van Rijsbergen, 250 stop words
  *Cornelis Joost van Rijsbergen, 1979* [220]
  http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words
- SMART, 571 stop words
  *Information Retrieval System, Gerard Salton and Chris Buckley, 1968*
  http://www.lextek.com/manuals/onix/stopwords2.html
- Fox, 519 words
  *Christopher Fox, derived from the Brown corpus, 1989* [64]
  ftp://sunsite.dcc.uchile.cl/pub/users/rbaeza/irbook/
- Brown corpus, 1,014,000 stop words
  *Nelson W. Francis and Henry Kucera, 1982*
  http://khnt.aksis.uib.no/icame/manuals/brown/

Beside stoplists it is possible to define the stop words dependent from the corpus. In this case (index) term selection methods can be used. Wilbur et al. [223] evaluated a method for the automatic identification of stop words.
The definition of stop words based on the language statistic fundamentals from Zipfs Law, described in Section 4.2.2 (p. 30).

### Stemming

Stemming methods try to reduce words to its root or stem by removing suffixes and prefixes, like the *s* from *books* or the *ing* from *reading*. Stemming is applied on the level of words, like stop word removal. The stem is a morphem and the basic form of a word. The field of morphology builds the linguistic fundamentals of this method (described in Section 4.2.1, p. 28). The words with similar meaning are summarized. This reduces the number of unique words and therefore the size of the lexicon. For example the word *protecting*, *protected*, *protects*, and *protection* will be summed up to the root word *protect*.
Stemming is not a perfect process, because there are special cases like *skies* → *sky*, *lying* → *lie*, *cosmos* → *cosmos* (none plural) and *news* → *news* (only plural). Lists which include these special cases have to be defined.
There are many stemming algorithms, which can be differentiated as follows.

- Simple stemming: The use of a table including all words and their stem is the simplest way of stemming, called simple stemming. But this approach causes a storage overhead.

- Successor variety stemming: A successor variety stemmer uses the frequencies of letter sequences for stemming.
  Examples: Hafer and Weiss [85]
- N-gram stemming: Language independent stemming by using statistic methods. The words are splitted into digrams (e.g. orders → {or,rd,de,er,rs}, ordering → {or,rd, de,er,ri,in,ng}) and words are merged based on the number of common digrams.
  Examples: Adamson and Boreham [3]
- Affix removal stemming: These stemming algorithms remove the affixes (suffixes and/or prefixes). Sometimes the computed stem will be transformed.
  Examples: Lovins [151], Dawson [48], Porter [177], Paice [174], Krovetz [138]

Frakes et al. [65] includes a section about the experimental evaluations of affix removal stemming: "The Porter algorithm is more compact than Lovins, Salton, and Dawson, and seems, on the basis of experimentation, to give retrieval performance comparable to the larger algorithms. The Paice stemmer is also compact, but experimental data was not available for the Paice algorithm."
The Porter stemmer is one of the most well-known stemmer for English. It includes a set of condition rules: conditions on the stem, conditions on the suffix and conditions on the rules. The rules iteratively transform words into their stems.

### Lemmatization

Compared to stemming the lemmatization transform a word to its basic form, called the lemma. While stemming removes suffixes and prefixes from a word, lemmatization uses rules for the transformation. This is a complex process, because the inflectional and variant forms (the part of speechs) of every word in a sentence has to be known. This requires, for example, knowledge of the grammar of a language. The difference to stemming is that a stemmer considers a single word without knowledge of the context. For this reason a stemmer would not transform the word *better* to *good*, as it is with lemmatization.
For a lemmatizer different methods exist. One possibility is the use of a lexicon including the lemma for each word. If a word exists in the lexicon, the specified lemma will be returned. Another method defines a list of rules which replace affixes for generating the lemma form. For example: *If a word ends by "ing", then strip "ing" and add "e"*. These rules are often called rules of detachment. In practice stemming or a combination of stemming and lemmatization is frequently used. The field of morphology builds the linguistic fundamentals of this method (described in Section 4.2.1, p. 28).
Examples are the MorphAdorner [4] and the WMTrans Lemmatizer [5]. A lot of lemmatization algorithm approaches are given by Mladenic et al. [176]:

- A rule induction system [ATRIS, 1993-2002],
- If-then classification rules [2002],
- Naive Bayes [2002],
- A first-order rule learning system [CLog, 2004] and
- Ripple Down Rule (RDR) learning [2004].

[4] Java command-line program, http://morphadorner.northwestern.edu/
[5] WordManager Transducers, http://www.canoo.com/wmtrans/home/index.html

### Multiword Phrase Grouping

In natural languages, groups of words can occur as a single unit, called multiword expressions (MWEs), such as *add up*, *telephone box* or *Los Angeles*. There are linguistic and statistic approaches (described in Section 4.2.1, p. 27 and Section 4.2.2, p. 30) for the detection and grouping of multiwords to n-word terms, which is a key problem in natural language processing. Multiword phrase grouping is well-known under the names multiword recognition and multiword extraction as well.
Sag et al. [184] introduces different kinds of MWEs and describes some analytic techniques. A good overview about linguistic and statistic techniques are given by [189] as well.

### Word-Sense Disambiguation

In natural language, some terms have different meanings, like *bank*: a *financial institution*, the *border of a river or lake* or *a bench*. These terms are called homonyms (defined in Section 4.2.1, p. 29). Word sense disambiguation (WSD) tries to resolve the ambiguity. Thereby the meaning of a term can only be determined by examining its context. Different approaches are possible [189]: knowledge based, corpus based and hybrid. Knowledge based approaches using information from a lexicon or knowledge base, whereas corpus based approaches using information which are extracted by a training corpus. A hybrid approaches combines knowledge and corpus approaches. The field of semantic builds the linguistic fundamentals of this method (described in Section 4.2.1, p. 29).

### Anaphora resolution

Anaphora resolution is the process of resolving indirect references of noun phrase, like *he* or *herself*. A noun phrase can be a pronoun, a demonstrative, a reflexive etc. The two main approaches are the knowledge based and machine learning approach. Knowledge based approaches are based on linguistic analysis of the sentences. Machine learning approaches are based on an annotated corpus.
Feldman et al. [62] gives a good overview about the fundamentals and methods for anaphora resolution.

### Synonym Detection

Synonym detection is the process of the detection of synonymous terms and the following conflation by using a list of synonyms. Another possibility is the recognition of synonyms by using a text corpus. The field of semantic builds the linguistic fundamentals of this method (described in Section 4.2.1, p. 29). The denotation of synonyms are defined in Section 4.2.1 (p. 29).

### Parsing

Parsing, known as role determination as well, is the syntactic analysis of the structure of a sentence according to a certain grammar theory (described in Section 4.2.1, p. 28). The sen-

tence structure will be analyzed and each term will be annotated according to its position, e.g. subject, object, etc.
In many languages, it is possible to recognize the role of a term by its position in the sentence. Typically the sentence will be represented as a tree, called parsetree. In this tree the root node is the sentence, branch nodes are non-terminals and leaf nodes are the terminals of the grammar. One parsing problem is the syntactic ambiguity. Sometimes it is possible to create several grammatically correct syntax trees for one sentence. One possibility to solve this problem is probabilistic parsing [226]. The task of probabilistic parsing is the identification of ambiguities and the selection of the most probable syntax tree under the possible syntax trees for the sentence.
Parsing can be differentiated into deep and shallow parsing. Deep parsing extracts the complete syntactic structure of a sentence. However, that is expensive, not robust enough and not needed in most cases. Shallow parsing, on the other hand, extracts only a limited amount of syntactic information. Thus, shallow parsing is faster and more robust. Typically a shallow parser includes additionally POS-Tagging and text chunking.

#### Part-of-Speech Tagging

Part-of-Speech tagging (POS-Tagging), known as morphosyntactic tagging as well (see morphology, Section 4.2.1, p. 28), is the grammatical tagging of a text. It determines the part of speech of each word. In classical linguistic eight kinds of words are known: noun, verb, adjective, preposition, pronoun, adverb, conjunction, and interjection. The knowledge about the part of speech of each word can support the syntactic analysis (parsing) and the stemming (described in Section 4.3.1, p. 37).
There are rule-based, stochastic and transformation-based tagger. Rule-based tagger are based on a list of disambiguation rules (the grammatic), e.g. the *ENGTWOL Tagger*. The stochastic approach based on the idea that certain combinations of words are very probable (e.g. article+noun) and others are rather improbable (e.g. article+verb). A training corpus is used to compute the probability of a given word in a given context. An example tagger is the *HMM tagger*, which is based on the Hidden-Markov-Model (see Section 4.2.2, p. 33). Transformation-based tagger are combinations of rule-based and stochastic tagger. For example the *Brill Tagger*.
An overview about POS-Tagging and the examples are given in [226].

#### Text Chunking

Text chunking is also known as phrase recognition. Sentences are divided into non-overlapping phrases in order to group adjacent words in a sentence. Text chunking can be used by keyphrases detection and text summarization. The field of grammar builds the linguistic fundamentals of this method (described in Section 4.2.1, p. 28).

#### Entity Extraction, Relation Extraction, Coreference Resolution

Entity extraction, also known as named entity recognition and entity identification, is the process of locating and classifying elements in text. There are predefined categories such as

the names of persons, organizations, locations, dates, times, etc. The Hidden-Markov-Model (see Section 4.2.2, p. 33) is successfully used by this process. For example in the *Nymple system*, later called *Identifinder system*, from *Bikel et al.* [21].
Relation extraction is the process of locating and classifying relationships between entities. A relationship is defined over two or more entities, for example *is employee of* or *capital of*.
Coreference resolution is the process of locating noun phrases which refer to a entity. The noun phrases describes the entity.
On one hand, there are rule-based information extraction systems with manually codec rules, later with algorithms for automatically learning rules. On the other hand, there are statistic based information extraction systems. There are generative models based on Hidden-Markov-Models and conditional models based on maximum entropy.
*Sarawagi et al.* [190] give an overview about the fundamentals and some applications are presented. *Feldman et al.* [62] describe some entity extraction methods as well.

### 4.3.2. Representation Methods

In this section, methods for representing text in a computer processable form are described.

#### Term Weighting

The first step for describing the terms of a text is the definition of a weight $w(t_i, d_k)$ for each term $t_i$ in a document $d_k$. In a corpus $C$ a collection of documents $d_k \in C$ exists. A document $d_k$ contains a set of terms $t_i \in d_k$. The set of all unique terms in the corpus are saved in a lexicon $L$. The function $f(t_i, d_k)$ defines the frequency of a term $t_i$ in a document $d_k$.
The definition of terms weights based on the fundamentals of language statistic (described in Section 4.2.2, 30), e.g. Zipfs law.

***Term Existing Weight:*** The simplest weighting approach declares in which documents a term exist. If term $t_i$ exists in the document $d_k$ the weight is 1, otherwise 0. This weight allows a binary representation of documents and thus the using of boolean operations.

$$g(t_i, d_k) = \begin{cases} 1 & iff(t_i, d_k) > 0 \\ 0 & else \end{cases} \quad ) \tag{4.16}$$

***Term Frequency Weight:*** Another weighting approach uses the frequency $f(t_i, d_k)$ of a term $t_i$ in a document $d_k$.

$$f(t_i, d_k) \tag{4.17}$$

If a term occurs six times in a small document, the term describes the document very well. That will be not the case, if the document is very large. The term frequency has to be normalized by defining the relative frequency $tf$.

$$tf(t_i, d_k) = \frac{f(t_i, d_k)}{\sum_{t_j \in d_k} f(t_j, d_k)} \tag{4.18}$$

Another possibility is the normalization via the frequency of a term with the highest frequency in the document. [8]:

$$tf(t_i, d_k) = \frac{f(t_i, d_k)}{max_j f(t_j, d_k)} \tag{4.19}$$

***Term Distribution Weight:*** A term $t_i$ is characteristic for a document $d_k$, if it occurs mainly in this document. That is described by the inverse document frequency $idf$.

$$idf(t_i) = log \frac{|d|}{|d : t_i \in d|} \tag{4.20}$$

***Term Significance Weight:*** The combination of relative term frequency and term distribution $TfIdf$ describes the significance of a term $t_i$ in a document $d_k$. Significant terms for a document are those terms which occur mainly in this document and then frequently.

$$TfIdf(t_i, d_k) = tf(t_i, d_k) \cdot idf(t_i) \tag{4.21}$$

In Salton et al. [187] another term weight is described that is used in practice. This term weight is normalized by the terms of a document.

$$Tfidf(t_i, d_k) = \frac{tf(t_i, d_k) \cdot idf(t_i)}{\sqrt{\sum_{t_j \in d_k} tf(t_j, d_k)^2 \cdot idf(t_i)^2}} \tag{4.22}$$

In practice mostly the term existing weight and term significance weight are used. A weight threshold can be defined for reducing the complexity. In this case only the terms with a term weight higher or equal this threshold are used in further analysis. A detailed introduction is given by Heyer et al. [98].

### Table Model

One possibility for representing and saving terms is the use of tables. By using a database, this allows an efficient saving and access of terms. In this case, SQL queries can be used for getting specific terms. There are several possibilities for the construction of the table model. One is described in the following section.

The first table includes the original documents. That is necessary, if the original text has to be accessed later again. Each text document gets its own ID and it is possible to save further information, such as the original document text (text), the document name (name) and the creation date (date). An example of a document table is given in 4.5. If a document has several revisions, the revisions will be saved separately in the table. The revisions are distinguished by its ID.

| d_id | text | name |
|---|---|---|
| 1 | Alice was beginning to get ... | Down the Rabbit-Hole AAW-CH1 |
| ... | ... | ... |
| 11 | The King and Queen of ... | Who Stole the Tarts? AAW-CH11 |
| 12 | 'Here!' cried Alice ... | Alice's Evidence AAW-CH12 |

**Table 4.5.:** Example of a document table

The next table includes the sentences of text documents. Each sentence is saved only one time, even if it arises several times. If texts (e.g. revisions) are very similar, this will be an efficient method. Table 4.6 shows the structure of a sentence table.

| s_id | sentence |
|---|---|
| 1 | Alice was beginning to get very tired of... |
| 2 | Oh dear! |
| 3 | On this the White Rabbit blew... |
| ... | ... |

**Table 4.6.:** Example of a sentence table

The terms are saved in the term table. Additional columns are for example the total frequency or the subject area. An example is given in Table 4.7.
The definition of a co-occurrence table including the significant co-occurrences (terms those arise very often together, see Section 4.2.1, p. 29) is possible as well (Table 4.8). A further possibility is to connect the term table with a dictionary to get more information about the term, like the synonyms.
To combine the document with the sentence table and the sentence with the term table, inverse tables can be used. The Tables 4.9 and 4.10 are example inverse tables. Additionally, the frequency or position of a sentence in a document or a term in a sentence can be saved.
Other structures of the table model are possible. For example the sentence table could be removed and the terms can be directly combined with the document inverse table. If the documents have revisions, the differences between the revisions can be saved. Between two revisions the added and deleted sentences can be saved.
For further informations the German book *Text Mining: Wissensrohstoff Text* [98] describes the construction of a text database very detailed.

| t_id | term | total frequency |
|---|---|---|
| 1 | the | 1643 |
| 2 | alice | 386 |
| 3 | white | 30 |
| 4 | rabbit | 47 |
| ... | ... | ... |

**Table 4.7.:** Example of a term table

| t1_id | t2_id | significance |
|---|---|---|
| 3 | 4 | 20 |
| ... | ... | ... |

**Table 4.8.:** Example of a co-occurrence table

| d_id | s_id | frequency |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 2 | 2 |
| 11 | 3 | 1 |
| ... | ... | ... |

**Table 4.9.:** Example of a document-sentence inverse table

| s_id | t_id | frequency |
|---|---|---|
| 1 | 1 | 3 |
| 1 | 2 | 2 |
| 4 | 4 | 1 |
| ... | ... | ... |

**Table 4.10.:** Example of a sentence-term inverse table

### Vector Space Model

The vector space model (VSM), also known as vector space retrieval or bag of words model, is probably the most well-known model in information retrieval. Documents and search queries will be represented as vectors. The model was developed in the range of the SMART project [28]. Today the vector space model is used in other text analysis areas such as text mining as well. The mathematical fundamentals are described in section 4.2.3 (p. 33).
In the vector space model a document $d_k$ is represented as a n-dimensional vector $TDV(d_k)$.

$$TDV(d_k) = (w(t_1, d_k), \ldots, w(t_n, d_k)) \tag{4.23}$$

Each element of the vector is a term of the document represented by their term weight $w(t_i, d_k)$ (see Section 4.3.2, p. 42). The size $n$ of the vector is defined by the number of all terms in the corpus. The transformation of a document into a vector allows the use linear algebra to analyze the document collection. For example the similarity of two documents can the calculated by computing the similarity of the corresponding vectors.
The documents of a corpus can be saved in a **term document matrix** $A$ with n rows and m columns. In this matrix each row represents a term and each column represents a document. The element $a_{ij}$ of the matrix contains the weight of the i-th term in the j-th document. An example is given in Table 4.11. In this table the chapters of the book are the documents. Furthermore it is possible to derive a **dissimilarity matrix** $D$ from the term document matrix. Each column and row represents a document or term and the values describe the pairwise dissimilarity of the documents or terms.

| | $ch_1$ | $ch_2$ | $ch_3$ | $ch_4$ | ... | $ch_{12}$ |
|---|---|---|---|---|---|---|
| king | 0 | 0 | 0 | 0 | ... | 22 |
| day | 1 | 2 | 1 | 0 | ... | 2 |
| rabbit | 9 | 3 | 0 | 12 | ... | 8 |
| ... | ... | ... | ... | ... | ... | ... |
| dogs | 0 | 3 | 0 | 0 | ... | 0 |

**Table 4.11.:** Example of a term document matrix

Because of the high number of terms, the term document matrix has many dimensions. The matrix is very sparse, because in each text only a small number of terms arise. Thinning the dimensions out by removing stop words is not sufficient. Further methods, such as dimension reduction, have to be applied.

### Bigram Proximity Matrix

A bigram proximity matrix, known as text proximity matrix as well, encodes each document as a matrix [156]. If two terms $t_1$ and $t_2$ arise one behind the other, there will be a term pair $(t_1, t_2)$. A row in the matrix represents the first term in the pair, and a column represents the second term in the pair. The element $x_{ik}$ of the matrix is the number of times the i-th term appears before the j-th term. Thus there will be n matrices with m rows and m columns, if the corpus includes $n$ documents and $m$ terms. In a bigram proximity matrix each sentence ends with a period. All other special characters will be removed. This allows a more detailed encoding of the documents. An example is given in Table 4.12.

| | ! | a | curious | feeling | what |
|---|---|---|---|---|---|
| ! | 0 | 0 | 0 | 0 | 0 |
| a | 0 | 0 | 1 | 0 | 0 |
| curious | 0 | 0 | 0 | 1 | 0 |
| feeling | 1 | 0 | 0 | 0 | 0 |
| what | 0 | 1 | 0 | 0 | 0 |

**Table 4.12.:** Example of a bigram proximity matrix, for the sentence *What a curious feeling!*

### 4.3.3. Dimension Reduction

Given a data set, e.g. a set of documents and their terms, the dimension reduction is a method which can be used for preprocessing the data. The result is an approximated matrix in that the noise is removed from the data and the important features (e.g. terms) are highlighted. In text analysis the dimension reduction is mostly applied on a term document matrix or on the derived dissimilarity matrix (see Section 4.3.2, p. 44). This preprocessing allows a more efficient and effective clustering, classification, topic detection, data selection, etc. Sometimes dimension reduction is executed during the clustering or classification.

| | **Feature Transformation** | **Feature Selection** |
|---|---|---|
| **Supervised** | LDA [63] [17], Supervised PPCA [237] | Quinlan's C4.5 [179], Optimal Brain Damage [144], Random Forests [26], Chi-Square measure [231], Information Gain [179], Odds Ratio [164], Mutual Information [173], Forward Selection [134], Backward Elimination [134], Genetic Search [150], Simulated Annealing [150] |
| **Unsupervised** | PCA [118] [196] [17] [201], LSA [50] [17] [201], MDS [42] [75] [79] [89] [141] [175] [188] [215] [216], ISOMAP [211] [9], LLE [181] [182], LDM [38] [37], EFA [49] [41], ICA [29], local PCA Fukunaga et al. [120], PPCA [213], SOM [133], GTM [22], RM [117], SM [40], CCA [51], CDA [146], KDA [14], local FDA [207], CentroidQR [115], | Category Utility [53], NMF [145], Laplacian Score [95], Q-$\alpha$ [229], Biclustering [27], FSSEM [57], DF Thresholding [234], TS [223], mean TFIDF [210], TfV [210], TW Thresholding [Sec. 4.3.3, p. 60] |

**Table 4.13.:** Dimension reduction techniques

There are a lot of techniques to reduce the number of dimensions. The techniques can be divided into feature transformation and feature selection methods and into supervised and unsupervised learning methods [45]. An overview about the different techniques are given in Table 4.13. The abbreviations are listed in chapter A.
Feature transformation maps the original data set into a lower dimensional space. The new space is generated by combinations or transformations of the original ones. Sometimes this is problematic, e.g. if the original dimensions were built by terms. In this case the new dimensions don't have an obvious reference to the terms. Feature transformation techniques are applied on the term document matrix or on a derived dissimilarity matrix. The implementation can be linear or non-linear. Some techniques have more than one implementation with linear and non-linear approaches. An overview about the classification is given in Table 4.14.
On the other hand, feature selection discover an important subset of the original data set. The techniques can be divided into filter, wrapper and embedded methods 4.15 [45]. Filter based

approaches remove irrelevant features, whereas wrapper based approaches choose the relevant features. By the embedded approach the feature selection is integrated into the learning algorithm. Some feature selection techniques are executed during classification or clustering. Supervised methods have a supervised learning process. Each object (e.g. document) has class labels which are taken into account by the dimension reduction process. Using class labels, objects are grouped in different classes.
By unsupervised methods no additional information about the data exists.

| | **Term document matrix** | **Dissimilarity matrix** |
|---|---|---|
| **Linear** | ICA, EFA, PCA, LSA, LDA, local PCA, PPCA, supervised PPCA, RM, SM, CentroidQR | MDS |
| **Non-linear** | ICA, EFA, KDA, local FDA | ISOMAP, LLE, LDM, MDS, SOM, GTM, CCA, CDA |

**Table 4.14.:** Feature transformation techniques

| | | |
|---|---|---|
| **Filter Methods**<br>Chi-Square measure, Information Gain, Odds Ratio, Mutual Information<br>Laplacian Score, DF Thresholding, TS, mean TFIDF, TfV, TW Thresholding | **Wrapper Methods**<br>Forward Selection, Backward Elimination, Genetic Search, Simulated Annealing<br>Category Utility, FSSEM | **Embedded Methods**<br>Quinlan's C4.5, Optimal Brain Damage, Random Forests<br>NMF, Q-$a$, Biclustering |

**Table 4.15.:** Feature selection techniques

### Latent Semantic Analysis

The goal of Latent Semantic Analysis (LSA), known as Latent Semantic Indexing (LSI) as well, is to reveal the semantic information from a corpus by the transformation into a semantic space. The semantic space includes *latent semantics* between terms and documents. For example terms can be associated with a document even if they did not appear in the text. By this approach the problems of synonyms are resolved and the problems of homonyms are partially detected as well. The terms are summarized to clusters, whereby the dimensions of the matrix are merged to a common concept. LSA afford additionally the removing of noise in the data.
LSA is applied on a $n \times m$ term document matrix $A = (a_{ij})$, whereby the rows are terms (n terms) and the columns are documents (m documents). For creating the semantic space, the matrix will be approximated by using singular value decomposition and denoising. The method was introduced by Deerwester et al. [50] and was used in information retrieval first. The fundamentals of linear algebra (definition of vectors and matrices and operations on them) are introduced in Section 4.2.3 (p. 33). The semantic fundamentals are described in Section 4.2.1 (p. 29).

***Singular Value Decomposition:*** The Singular Value Decomposition (SVD) is described in section 4.2.3 (p. 33). It converts the term document matrix $A$ into 3 matrices. The matrix $T$ includes the left singular vectors and spans the term space of $A$. The matrix $D$ is the matrix of right singular vectors and spans the document space of $A$. $S$ is the diagonal matrix of singular values.

$$\begin{matrix} A & = & T & S & D^T \\ (n \times m) & & (n \times r) & (r \times r) & (r \times m) \end{matrix} \tag{4.24}$$

The SVD can be adapted by the creation of a document-document matrix (4.25) or a term-term matrix (4.26) as well. To find documents for a query $q$ (information retrieval), the query have to be represented as a term vector $X_q$ (4.27). A matrix $D_q$ will be derived. The equations are valid after the approximation of the matrix (Denoising) as well.

$$A^T A = D(SS^T)D^T \tag{4.25}$$

$$AA^T = T(SS^T)T^T \tag{4.26}$$

$$D_q = X_q^T T S^{-1} \tag{4.27}$$

***Denoising:*** The next step is denoising, whereby the noise in the data are removed. The noise of the data are unimportant relations between documents and terms. For example if a term occur in a document, but the term is not relevant for the document. That will be achieved by the dimension reduction of the matrix $A$. The matrix has the rank $r$. The rank defines the maximum number of linear independent row vectors and the maximum number of linear independent column vectors.

SVD allows the approximation of a matrix by matrices of lower rank/dimension $k < r$ (*low-rank approximation*). Thus the lowest singular values are truncated and only the first k singular values are kept.

$$\begin{array}{lcccc} A \approx A' & = & T_k & S_k & D_k^T \\ & & (n \times k) & (k \times k) & (k \times m) \end{array} \tag{4.28}$$

Since zeros were introduced, the representation can be simplified by deleting the zero rows and columns. The value $k$ should be small enough to filter the unimportant details and large enough to keep the important features. The choice of a good value is an open issue in the factor analytic literature. One possibility is to plot the singular values and to look for a flattening or a break. Another possibility is the computation of the differences between the result matrices with different $k$ values.
The approximated matrix $A'$ represents the semantic space of the term document matrix $A$. The cosine between two vectors of the semantic space supplies the semantic similarity of two words or documents. The space of the matrix $A'$ is smoothed concerning the original matrix $A$. In the semantic space terms and documents that are closely associated are placed near to each other. Thus synonyms are detected, because the values of adjacent terms are more similar and terms get a value for a document even if they did not appear in the text. The computational complexity of this transformation is $O((|T| + |D|)^2 \cdot k^3)$.

***Example:*** Now the LSA will be explained by an example. First, the term document matrix of 4.16 will be transformed by the Singular Value Decomposition.

| | $ch_7$ | $ch_8$ | $ch_9$ | $ch_{10}$ | $ch_{11}$ |
|---|---|---|---|---|---|
| king | 0 | 11 | 2 | 0 | 26 |
| majesty | 0 | 3 | 1 | 0 | 4 |
| queen's | 0 | 5 | 1 | 0 | **0** |
| duchess's | 0 | 0 | 1 | 0 | 1 |
| shark | 0 | 0 | 0 | 1 | 0 |
| chair | 1 | 0 | 0 | 0 | 0 |
| room | 3 | 0 | 0 | 0 | 1 |
| turtles | 3 | 0 | **0** | 2 | 0 |
| turtle | 0 | 0 | 27 | 29 | 0 |

**Table 4.16.:** Term document matrix for LSA

$$A = \begin{pmatrix} 0 & 11 & 2 & 0 & 26 \\ 0 & 3 & 1 & 0 & 4 \\ 0 & 5 & 1 & 0 & \mathbf{0} \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 \\ 0 & 0 & \mathbf{0} & 2 & 0 \\ 0 & 0 & 27 & 29 & 0 \end{pmatrix}$$

$$T = \begin{pmatrix} 0.08 & -0.98 & 0.11 & -0.05 & 0.04 \\ 0.02 & -0.17 & -0.24 & 0.03 & -0.16 \\ 0.02 & -0.07 & -0.95 & 0.13 & 0.06 \\ 0.02 & -0.03 & 0.06 & -0.01 & -0.43 \\ 0.02 & 0.00 & 0.02 & 0.00 & 0.40 \\ 0.00 & 0.00 & 0.02 & 0.32 & 0.00 \\ 0.00 & -0.03 & 0.14 & 0.94 & 0.00 \\ 0.04 & 0.01 & 0.04 & -0.01 & 0.79 \\ 1.00 & 0.08 & 0.01 & 0.00 & -0.03 \end{pmatrix}$$

$$S = \begin{pmatrix} 39.73 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 28.75 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 4.83 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 3.14 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 1.70 \end{pmatrix}$$

$$D^T = \begin{pmatrix} 0.00 & 0.03 & 0.68 & 0.73 & 0.05 \\ 0.00 & -0.40 & 0.00 & 0.08 & -0.91 \\ 0.09 & -0.90 & -0.11 & 0.11 & 0.41 \\ 1.00 & 0.08 & 0.01 & -0.01 & -0.04 \\ 0.01 & 0.17 & -0.72 & 0.67 & -0.01 \end{pmatrix}$$

Now the dimension reduction by choosing a $k$ is possible. The matrices for $k = 3$, $k = 2$ and $k = 1$ are calculated.

$$A_3 = \begin{pmatrix} 0.05 & 10.89 & 2.10 & 0.13 & 26.02 \\ -0.10 & 3.02 & 0.66 & 0.06 & 4.01 \\ -0.41 & 4.96 & 1.04 & -0.09 & \mathbf{-0.01} \\ 0.03 & 0.11 & 0.51 & 0.54 & 0.94 \\ 0.01 & -0.07 & 0.53 & 0.59 & 0.08 \\ 0.01 & -0.09 & -0.01 & 0.01 & 0.04 \\ 0.06 & -0.27 & -0.07 & 0.01 & 1.06 \\ 0.02 & -0.24 & \mathbf{1.06} & 1.20 & -0.11 \\ 0.00 & 0.23 & 27.01 & 29.19 & -0.09 \end{pmatrix}$$

$$T_3 = \begin{pmatrix} 0.08 & -0.98 & 0.11 \\ 0.02 & -0.17 & -0.24 \\ 0.02 & -0.07 & -0.95 \\ 0.02 & -0.03 & 0.06 \\ 0.02 & 0.00 & 0.02 \\ 0.00 & 0.00 & 0.02 \\ 0.00 & -0.03 & 0.14 \\ 0.04 & 0.01 & 0.04 \\ 1.00 & 0.08 & 0.01 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} 39.73 & 0.00 & 0.00 \\ 0.00 & 28.75 & 0.00 \\ 0.00 & 0.00 & 4.83 \end{pmatrix}$$

$$D_3^T = \begin{pmatrix} 0.00 & 0.03 & 0.68 & 0.73 & 0.05 \\ 0.00 & -0.40 & 0.00 & 0.08 & -0.91 \\ 0.09 & -0.90 & -0.11 & 0.11 & 0.41 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0.00 & 11.37 & 2.16 & 0.07 & 25.80 \\ 0.00 & 1.98 & 0.54 & 0.19 & 4.49 \\ 0.00 & 0.83 & 0.54 & 0.42 & \mathbf{1.87} \\ 0.00 & 0.37 & 0.54 & 0.51 & 0.82 \\ 0.00 & 0.02 & 0.54 & 0.58 & 0.04 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.34 & 0.00 & -0.07 & 0.78 \\ 0.00 & -0.07 & \mathbf{1.08} & 1.18 & -0.18 \\ 0.00 & 0.27 & 27.02 & 29.19 & -0.11 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 0.00 & 0.10 & 2.16 & 2.32 & 0.16 \\ 0.00 & 0.02 & 0.54 & 0.58 & 0.04 \\ 0.00 & 0.02 & 0.54 & 0.58 & \mathbf{0.04} \\ 0.00 & 0.02 & 0.54 & 0.58 & 0.04 \\ 0.00 & 0.02 & 0.54 & 0.58 & 0.04 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.05 & \mathbf{1.08} & 1.16 & 0.08 \\ 0.00 & 1.19 & 27.02 & 29.00 & 1.99 \end{pmatrix}$$

A comparison of the matrices shows, that the best approximation is given with $k = 2$. The singular values have a break at this point as well. In the matrix $A_2$ *turtles* is regarded as synonymous for *turtle* and *queen's* is associated with *majesty*. The values are highlighted red in each matrix to visualize the change.

Figure 4.7 represents the two dimensional approximated matrix as a chart. The first two columns of $T$ represent the values for the terms and the first two columns of $D$ represent the values for the chapters. Because of the high approximation a lot of points are nearby null. *Chapter 9* and *chapter 10* are near together, since the term *turtle* arises very frequently in both chapters. Because the term *king* occurs in *chapter 11* very frequently, these points are adjacent as well.

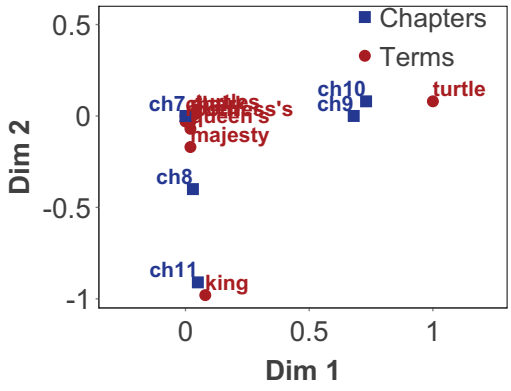


**Figure 4.7.:** Latent semantic analysis 2D-plot

***Similar Methods:*** A similar method to LSA is **Random Mapping** (RM), known as random projection (RP) as well, from Kaski et al. [125] [121]. Compared to LSA the reduced space is a composition of random vectors. The method based on the lemma from Johnson et al. [117]. To reduce the complexity of RM, **Sparse Random Mapping** (SRM) was introduced.

### Principal Components Analysis

The Principal Components Analysis (PCA) is a method which reduces the dimensions of a data set, so that only few information is lost. Thus the noise are removed and pattern can be detected, because the similarities and differences are highlighted by the PCA.
Mathematically it is a transformation of the data into a lower dimensional space, in which the main part of the data variance is. The lower dimensional space is defined by eigenvectors, called *principal components*. The variables are summarized to few linear combinations based on the principal components, which resolve together a large as possible part of the variance.
A detailed description about PCA can be found in the book *Principal Component Analysis* [118]. The tutorial about PCA from Shlens et al. [196] includes the background mathematics (covariance and matrix algebra) and describes the PCA by an example.

PCA will be applied on a $n \times m$ matrix $A = (a_{ij})$, where the columns are objects (samples) and the rows are the variables (features) of the objects. The matrix has $n$ variables (terms) and $m$ objects (documents, authors, etc.). Because the rows represent the dimensions of the matrix, the matrix has n dimensions. In the area of text analysis the matrix is called term document matrix. In the majority of cases the objects are documents and the terms are variables. But it is also possible to take the terms as objects. The fundamentals of linear algebra (definition of vectors and matrices and operations on them) are introduced in Section 4.2.3 (p. 33).

***Centered Matrix:*** The first step of PCA is the calculation of the centered matrix $B = (b_{ij})$ to adjust the data set. The mean of each variable (row) is calculated and substracted from each object. Therefore the mean vector $\overline{A} = (\overline{a}_{1.}, \ldots, \overline{a}_{n.})$ including the mean values for each variable will be calculated. For each column $b_{.i}$ of $B$ the mean vector $\overline{A}$ will be substracted from each column $a_{.i}$ of $A$

$$b_{.i} = a_{.i} - \overline{A} \tag{4.29}$$

***Covariance Matrix:*** Now the covariance matrix $\Sigma = (\sigma_{ij})$ of $B$ can be calculated. The **covariance** is a measure which describes the correlation between two variables $i$ and $j$. It defines the dispersion around the mean concerning each other. The **variance** will be given, if $i = j$.

$$\sigma_{ij} = \frac{1}{m-1} \sum_{k=0}^{m} (a_{ik} - \overline{a_i})(a_{jk} - \overline{a_j}) \tag{4.30}$$

Since the centered matrix $B$ includes the adjusted data set, the covariance matrix $\Sigma$ can be defined as follow:

$$\Sigma = \frac{1}{m-1} BB^T \tag{4.31}$$

$\Sigma$ has n rows and n columns representing the variables. The value $\sigma_{ij}^{\Sigma}$ is the covariance between the i-th and j-th variable. The variables will grown in the same direction, if the covariance value is positive. If the covariance value is negative, the variables will grown in opposite directions.

***Matrix Diagonalization:*** The next step is the diagonalization of the covariance matrix. Therefore the eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$ and eigenvectors $\varphi_1, \varphi_2, ..., \varphi_n$ of $\Sigma$ have to be calculated. An introduction into the matrix diagonalization can be found in section 4.2.3 (p. 33).

$$\underset{(n \times n)}{\Sigma} = \underset{(n \times n)}{\Phi} \quad \underset{(n \times n)}{\Lambda} \quad \underset{(n \times n)}{\Phi^T} \tag{4.32}$$

The rows of $\Phi$ refer to the variables and the columns (eigenvectors) represent the components. The values are the transformation coefficients of a variable, concerning a specific component. The eigenvalues in $\Lambda$ define the variance, which is resolved by the corresponding component (eigenvector). Thus it defines the significance of each component.

***Dimension Reduction:*** Since the eigenvalues have a decreasing order, the components are given in order of significance. The eigenvector with the highest eigenvalue reproduces the data best and is called the first **principal component**. To reduce the dimensional space, components of lesser significance can be ignored. If the eigenvalues are small, only few information will be lost.
Different methods exist to define the number of components $k$ which should be remained. One possibility is to plot the eigenvalues and to look for a flattening or a break. A simple approach is the *Kaiser-Guttman Criteria*. In this case only the components are selected, whose eigenvalues are greater than one. Another possibility is to choose the first $k$ eigenvalues so that about $x$ percent of the variance by the components are reproduced, e.g. 85 percent:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \geq 0.85 \tag{4.33}$$

There are two possibilities to use the lower dimensional space for the original matrix $A$. The objects of the matrix can be described by the components. Otherwise the matrix can be approximated by the components.

***Transforming:*** The original matrix $A$ can be transformed to a new basis $F$ using the first $k$ components.

$$F = \Phi_k^T A \tag{4.34}$$

If all $n$ components are used, the original matrix can be rebuilt:

$$A = \Phi^T F \tag{4.35}$$

***Approximation:*** With $k$, the best rank-k approximation to $\Sigma$ in the least square sense is given [245].

$$\begin{array}{ccccc} \Sigma \approx \Sigma' & = & \Phi_k & \Lambda_k & \Phi_k^T \\ (n \times n) & & (n \times k) & (k \times k) & (k \times n) \end{array} \tag{4.36}$$

For an approximation of the matrix $B = (b_{ij})$ to $B' = (b'_{ij})$ the matrix $\Phi = (\phi_{ij})$ including the principal components are used:

$$b'_{\cdot i} = \sum_{j=1}^{k} w_{ij} \cdot \phi_{\cdot j} \qquad w_{ij} = \phi_{\cdot j}^T \cdot b_{\cdot i} \tag{4.37}$$

An object of the matrix $B'$ is the weighted sum ($w_{ij}$) of the principal components ($\phi_{\cdot j}$). An approximation $A'$ of the original matrix $A$ can be derived as well using the mean vector $\overline{A} = (\overline{a}_{1\cdot}, ..., \overline{a}_{n\cdot})$:

$$a'_{i\cdot} = b'_{i\cdot} + \overline{A} \tag{4.38}$$

***PCA using SVD:*** Instead of calculating the eigenvectors and eigenvalues of the covariance matrix $\Sigma = \frac{1}{m-1} BB^T$, it is possible to use SVD for the identification of the eigenvectors and eigenvalues [5] [199].
Finding the eigenvalues and eigenvectors of $\frac{1}{m-1} BB^T$ is the same as finding the eigenvalues and eigenvectors of $BB^T$, at which the eigenvalues of $BB^T$ are (m-1)-times greater than the eigenvalues of $\frac{1}{m-1} BB^T$.
In SVD the matrix $B$ is decomposed to $B = USV^T$, whereby $BB^T = US^2U^T$. The eigenvalues of $BB^T$ are the squares of the singular values on S and the columns of $U$ are the eigenvectors (section 4.2.3, p. 33).

***Example***: The example data are known from the LSA example in figure 4.16 (p. 49). First the centered matrix $B$ have to be calculated. Afterward the covariance matrix $\Sigma = \frac{1}{4} BB^T$ can be defined.

$$A = \begin{pmatrix} 0 & 11 & 2 & 0 & 26 \\ 0 & 3 & 1 & 0 & 4 \\ 0 & 5 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 27 & 29 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} -7.8 & 3.2 & -5.8 & -7.8 & 18.2 \\ -1.6 & 1.4 & -0.6 & -1.6 & 2.4 \\ -1.2 & 3.8 & -0.2 & -1.2 & -1.2 \\ -0.4 & -0.4 & 0.6 & -0.4 & 0.6 \\ -0.2 & -0.2 & -0.2 & 0.8 & -0.2 \\ 0.8 & -0.2 & -0.2 & -0.2 & -0.2 \\ 2.2 & -0.8 & -0.8 & -0.8 & 0.2 \\ -0.4 & -0.4 & -0.4 & 1.6 & -0.4 \\ -11.2 & -11.2 & 15.8 & 17.8 & -11.2 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 124.20 & 19.15 & 2.55 & 3.10 & -1.95 & -1.95 & -1.30 & -3.9 & -95.70 \\ 19.15 & 3.30 & 1.60 & 0.45 & -0.40 & -0.40 & -0.60 & -0.8 & -15.65 \\ 2.55 & 1.60 & 4.70 & -0.35 & -0.30 & -0.30 & -1.20 & -0.6 & -10.05 \\ 3.10 & 0.45 & -0.35 & 0.30 & -0.10 & -0.10 & -0.15 & -0.2 & 1.15 \\ -1.95 & -0.40 & -0.30 & -0.10 & 0.20 & -0.05 & -0.20 & 0.4 & 4.45 \\ -1.95 & -0.40 & -0.30 & -0.10 & -0.05 & 0.20 & 0.55 & -0.1 & -2.80 \\ -1.30 & -0.60 & -1.20 & -0.15 & -0.20 & 0.55 & 1.70 & -0.4 & -11.20 \\ -3.90 & -0.80 & -0.60 & -0.20 & 0.40 & -0.10 & -0.40 & 0.8 & 8.90 \\ -95.70 & -15.65 & -10.05 & 1.15 & 4.45 & -2.80 & -11.20 & 8.9 & 235.70 \end{pmatrix}$$

Now the calculation of the eigenvalues and eigenvectors of $\Sigma$ is possible.

$$\Phi = \begin{pmatrix} -0.50 & 0.85 & 0.05 & -0.03 & -0.16 & 0.00 & 0.00 & 0.00 & 0.00 \\ -0.08 & 0.13 & -0.24 & 0.17 & 0.79 & 0.48 & 0.01 & -0.04 & -0.16 \\ -0.03 & -0.04 & -0.89 & 0.00 & -0.36 & 0.15 & -0.01 & -0.24 & 0.02 \\ 0.00 & 0.05 & 0.05 & 0.42 & 0.17 & -0.47 & -0.02 & -0.75 & 0.06 \\ 0.02 & 0.01 & 0.02 & -0.39 & 0.08 & 0.01 & 0.89 & -0.22 & 0.07 \\ 0.00 & -0.04 & 0.10 & 0.05 & -0.20 & 0.03 & 0.09 & -0.11 & -0.96 \\ -0.03 & -0.10 & 0.37 & 0.16 & -0.33 & 0.72 & -0.03 & -0.41 & 0.18 \\ 0.03 & 0.02 & 0.05 & -0.78 & 0.16 & 0.00 & -0.45 & -0.40 & -0.07 \\ 0.86 & 0.50 & -0.02 & 0.04 & -0.05 & 0.07 & 0.00 & -0.01 & -0.01 \end{pmatrix}$$

$$\Lambda = \begin{pmatrix} 293.59 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 71.39 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 5.38 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.74 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{pmatrix}$$

The columns of the matrix $\Phi$ are the principal components (eigenvectors) and the matrix $\Lambda$ includes the eigenvalues. In the next step the number of component, which should be represent the objects, have to be chosen. It is defined that at least 80 percent of the variance have to be reproduced. By taking the first component ($k = 1$) about 79.11 percent of the variance are reproduced. 98.35 percent of the variance will be reproduced, if $k = 2$. Consequently $k = 2$ components will be chosen.

$$\frac{\sum_{i=1}^{1} \lambda_i}{\sum_{i=1}^{9} \lambda_i} = 0.79 < 0.8 \qquad \frac{\sum_{i=1}^{2} \lambda_i}{\sum_{i=1}^{9} \lambda_i} = 0.98 > 0.8$$

Now the transformation to the matrix $F = \Phi_k^T A$ is possible and the approximated matrix $A'$ can be defined. The objects of the original matrix $A$ are represented by two components now, whereby 98.35 percent of the variance are reproduced. The new presentation has only two dimensions.

$$F = \begin{pmatrix} -0.10 & -5.88 & 22.16 & 25.08 & -13.28 \\ -0.33 & 9.54 & 15.23 & 14.43 & 22.56 \end{pmatrix}$$

$$A' = \begin{pmatrix} -0.10 & 11.17 & 2.07 & -0.06 & 25.92 \\ 0.47 & 2.17 & 0.65 & 0.32 & 4.39 \\ 1.87 & 1.70 & 0.51 & 0.44 & 1.47 \\ -0.17 & 0.30 & 0.50 & 0.46 & 0.91 \\ 0.01 & -0.01 & 0.50 & 0.54 & -0.04 \\ 0.79 & 0.38 & -0.01 & 0.01 & -0.16 \\ 2.19 & 1.42 & 0.01 & 0.00 & 0.39 \\ 0.02 & -0.02 & 0.99 & 1.08 & -0.07 \\ 0.03 & -0.06 & 26.94 & 29.06 & 0.03 \end{pmatrix}$$

Figure 4.8 includes a plot of the chapters represented by two principal components. The diagram is similar to the LSA diagram in figure 4.7. The chapters 9 and 10 are adjacent as well.

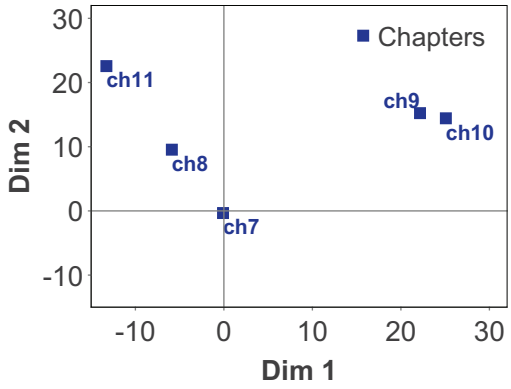


**Figure 4.8.:** Principal components analysis 2D-plot

***Similar Methods:*** A similar method to PCA is **Independent Component Analysis** (ICA). ICA is a technique which identify statistically independent components from a data set to identify global patterns [29]. PCA identify orthogonal components. A PCA extension is the **Local Principal Component Analysis** (local PCA) from Fukunaga et al. [66] and Kambhatla et al. [120]. By this methods, the data are clustered before applying PCA on each cluster. Other PCA extensions are the **Probabilistic Principal Component Analysis** (PPCA) [213] and the **Supervised Probabilistic Principal Component Analysis** (Supervised PPCA) [237].

**Linear Discriminant Analysis** (LDA) is a technique for data classification and matrix approximation, developed from Fisher et al. [63]. There are different non-linear implementations for this technique as well, e.g. the **Kernel Discriminant Analysis** [14] and the **Local Fisher Discriminant Analysis** [207]. The LDA is similar to PCA, but the LDA considers the class labels of the data. The objects are assigned to classes and the transformation maximize the separation between classes and minimize the separation within classes [45].

### Multidimensional Scaling

Multidimensional Scaling (MDS) is a multivariate analysis method. It transforms a dissimilarity matrix (described in Section 4.3.2, p. 45) into a lower-dimensional representation whereby the relationships should be maintained. MDS is used for dimension reduction, visualization etc. Mostly, the result of MDS is one-, two- or three-dimensional, because it allows the plotting of the objects. More dimensions are possible and sometimes required for getting a good fitting of the original data. Cox et al. [42], Handl et al. [89] and Groenen et al. [79] give a good overview about MDS.

Given a $n \times m$ matrix with $m$ objects, the $m \times m$ dissimilarity matrix includes pairwise dissimilarities between the objects. The MDS defines for each object a point in the reduced space, which describes the dissimilarities of the objects very well. It is called the configuration.

The developed MDS techniques can be differentiated between metric or non-metric and inner product (classical) or distance. The fundamental of metric MDS are ratio or interval data.

The metric MDS uses the dissimilarities between the objects for the spatial arrangement of the objects. The distances and the dissimilarities should be as similar as possible. Using the non-metric MDS the proximities of objects are based on the scaled or ranked data. In this case the monotonic relationship between the dissimilarities are represented. The distances and the dissimilarities should have the same order.
The first MDS techniques were proposed from Torgerson et al. (1952 [215], (1958 [216]) and Gower et al. (1966 [75]). The theoretical results were originated in Schoenberg (1935), Young and Householder (1938). It is a classical metric-based MDS. The techniques are known under the name of *Torgerson-Scaling*, *Torgerson-Gower-Scaling* and *Classical Multidimensional Scaling*. Another very famous technique is the *Shepard-Kruskal-Scaling* [141]. It is a non-metric distance-based MDS developed from Shepard and Kruskal, two pioneers of non-metric scaling. Sammon et al. [188] developed another non-metric distance-based MDS technique known as non-linear mapping.

Given a $m \times m$ dissimilarity matrix $\Delta = (\delta_{ij})$ with dissimilarities $\delta_{ij}$, the result of MDS is a configuration of points, saved in a matrix $X = (x_{ij})$. A row $x_{i.}$ includes the coordinates of an object in the reduced space. The distances $d_{ij}$ between the points in $X$ are saved in a distance matrix $D = (d_{ij})$. By metric MDS $d_{ij}$ and $\delta_{ij}$ should agree as well as possible. Using the non-metric MDS, $d_{ij}$ should be ordered like $\delta_{ij}$.
Sometimes a transformation of the original data is needed [79], e.g. for ordinal MDS. There are ratio, interval, ordinal or spline transformations for example. The transformed data are called **disparities** $\hat{a}_{ij}$. Now the configuration should represent the disparities as good as possible. E.g. by ordinal MDS the disparities have to satisfy $\hat{a}_{ij} < \hat{a}_{ab}$, if $\delta_{ij} < \delta_{ab}$.

***Distance Measures:*** There are a lot of possibilities for measuring the distance between two points. In MDS the Euclidean distance, the City-Block-Metric and the Minowski-Metric are often used. A list of distance measures is given in section 4.3.4 (p. 65).

***Quality factor:*** The quality of a configuration can be measured in different ways. One possibility is a **Shepard-diagram**. The Shepard-diagram allows the comparison between $\hat{a}_{ij}$ and $d_{ij}$ to evaluate the quality of the scaling. The dissimilarities will be plotted against the calculated distances.
Another quality factor is the **STRESS** value. It is a value between 0 and 1 which defines the *Badness of Fit*. Different STRESS definitions exist. $\omega_{ij}$ is a non-negative weight, but in most cases the weight is not defined.

- $STRESS = \sqrt{\sum_{i<j}^{m} \omega_{ij}(\delta_{ij} - d_{ij})^2}$ (Raw Stress, Kruskal [139] [140], least-squares MDS)
- $STRESS = \sqrt{\sum_{i<j}^{m} \omega_{ij}(\hat{a}_{ij} - d_{ij})^2}$ (Raw Stress, Kruskal [139] [140])
- $STRESS = \sqrt{\frac{\sum_{i<j}^{m} \omega(\hat{a}_{ij} - d_{ij})^2}{\sum_{i<j}^{m} \omega_{ij}\hat{a}_{ij}^2}}$ (Normalized raw Stress)
- $STRESS1 = \sqrt{\frac{\sum_{i<j}^{m} \omega(\hat{a}_{ij} - d_{ij})^2}{\sum_{i<j}^{m} \omega_{ij}d_{ij}^2}}$ (Kruskal [89])
- $STRESS2 = \sqrt{\frac{\sum_{i<j}^{m} \omega(\hat{a}_{ij} - d_{ij})^2}{\sum_{i<j}^{m} \omega_{ij}(d_{ij} - \overline{d})^2}}$ (Kruskal)
- $STRESS = \frac{1}{\sum_{i<j}^{m} \delta_{ij}} \sum_{i<j}^{m} \frac{\delta_{ij} - d_{ij}}{\delta_{ij}}$ (Sammon [188])

A small STRESS value refers to a good configuration. Experience with the STRESS values are given from Kruskal (Table 4.17). A very small STRESS value is an indication for the fact that the objects collect in the center of the coordinate system.
Beside STRESS, the **RSQ** ($R^2$) measure is another quality factor [175]. A value near one indicates a good configuration ($R^2 > 0.9$ is acceptable).

| | STRESS1 | STRESS2 |
|---|---|---|
| poor | 0.2 | 0.4 |
| fair | 0.1 | 0.2 |
| good | 0.05 | 0.1 |
| excellent | 0.025 | 0.05 |
| perfect | 0 | 0 |

**Table 4.17.:** Kruskal STRESS experiences

***Dimensions:*** The number of dimensions have to be defined dependent on the original data. Using *Classical Multidimensional Scaling* it is possible to define the number of dimensions dependent on the size of the eigenvalues, like the PCA technique (described in Section 4.3.3, p. 52).
In general, a plot of the minimal STRESS value and the number of dimensions is possible to read off a good number of dimensions. Another possibility to define the number of dimensions is the use of the data reduction coefficient which is defined by the number of similarities divided by the number of coordinates.

***Metric MDS:*** The *Classical Multidimensional Scaling* is explained as an example of metric MDS. Given a $m \times m$ dissimilarity matrix $\Delta = (\delta_{ij})$, the following steps have to be executed [89] [42]:

1. Define $m \times m$ matrix $A = (a_{ij})$ as follow:
$$a_{ij} = -0.5\delta_{ij}^2 \tag{4.39}$$
2. Create inner product matrix $B = (b_{ij})$
$$b_{ij} = a_{ij} - \overline{a}_{i.} - \overline{a}_{.j} + \overline{a}_{..} \tag{4.40}$$
3. Spectral decomposition of matrix $B$
$$B = \Phi\Lambda\Phi^T \tag{4.41}$$
4. Define MDS configuration $X_k$
Create a k-dimensional matrix $X_k$, whereby $\Lambda_k$ includes the k greatest eigenvalues and $\Phi_k$ the corresponding eigenvectors. The eigenvalues have to be unit, i.e. their lengths are one. The matrix is the MDS configuration including the points as row vectors.
$$X_k = \Phi_k\Lambda_k^{\frac{1}{2}} \tag{4.42}$$
5. Calculate the quality factor
Calculate the distance matrix $D = (d_{ij})$ of the new data points and evaluate the quality of the scaling with the Shepard-diagram.

***Non-metric MDS:*** Now a short overview about non-metric distance MDS is given. For example the *Shepard-Kruskal-Scaling* algorithm is more complex and included more steps. The general steps are [175]:

1. Configurations
   Define the number of dimensions and a distance measure for $D$.

2. Initializations
   Define an initial configuration $X$. Random or a metric MDS configuration for example.

3. Disparity calculation ($\hat{d}_{ij}$)
   Transform the original data, e.g. that $\delta_{ij} < \delta_{ab} \Rightarrow d_{ij} < d_{ab}$ [157].

4. STRESS value calculation
   The STRESS value describes the difference between the disparities and the distances.

5. Termination
   If the STRESS value is small enough or didn't changed very much, the final MDS configuration will be found. The algorithm is terminated.

6. Configuration adaptation ($X$)
   Adjust the points $X_{i.}$ of the objects based on the differences between the disparities and MDS-distances. This enhances the quality factor.

7. Continue with step 3

***Example:*** Examples for metric MDS and non-metric MDS are given in Figure 4.9 and 4.10. The clustering based on the data in Table 4.16 (p. 49). For metric MDS the `cmdscale` method and for non-metric MDS the method `sammon` from the statistic tool R[6] are used.

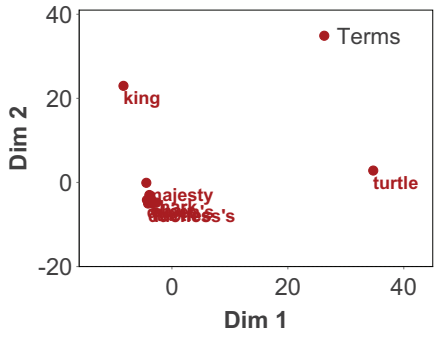


**Figure 4.9.:** Metric MDS 2D-plot (R: cmdscale)

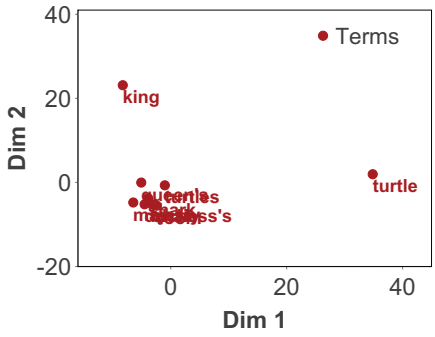


**Figure 4.10.:** Non-metric MDS 2D-plot (sammon, R)

***Similar Methods:*** The non-linear mapping from Sammon et al. [188] is improved by Demartines et al. with the **Curvilinear Component Analysis** (CCA) [51].

[6] http://www.r-project.org/

### Isometric Feature Mapping

Isometric Feature Mapping (ISOMAP) is a non-linear method for dimension reduction which uses isometric mapping, developed from Tenebaum et al. [211]. The algorithm extends metric MDS (described in Section 4.3.3, p. 57) concerning the definition of the dissimilarity matrix (described in Section 4.3.2, p. 45). ISOMAP uses a non-linear distance method instead of a linear distance measure (described in Section 4.3.4, p. 65). The base of the non-linear measure is a weighted graph generated by the neighborhoods of each data point. The nodes of the graph are the data points. The geodesic distance of two points is defined by the shortest path between two nodes. The problem is the determination of the neighborhood size, because there are no approaches to find a good value.

Before the algorithm will be executed, the value of $k$ or $\varepsilon$ and the number of new dimension $d$ have to be defined. Given a $n \times m$ matrix $A = (a_{ij})$ at which each column represents an object (data) and the rows are the data points, the following steps have to be executed [211]:

1. Construct neighborhood graph $G$
   Calculate the distances $d_A(i,j)$ (e.g. the euclidean distance) between all pairs of objects (data points) $a_{.i}$ and $a_{.j}$. In the neighborhood graph the $k$ nearest neighbors or all points within a certain radius $\varepsilon$ are connected directly to a point. The edge weight is $d_A(i,j)$.

2. Calculate the geodesic distance matrix $D_G = (d_{ij})$
   The values $d_{ij}$ of the geodesic distance matrix $D_G$ are the distances $d_G(i,j)$ between all pairs of nodes. A distance between two nodes (data points) is the shortest path between them. Since the graph is weighted, the sum of the edge weights defines the distance value. The algorithms of *Floyd's* or *Djkastra's* can be used for the computation.

3. Apply Classical MDS
   Calculates a $d$-dimensional representation with metric MDS for the distance matrix $D_G$ using eigenvalues and eigenvectors (see section 4.3.3, p. 55).

The article from Balasubramanian et al. [9] analyses the algorithm and detects that ISOMAP is topologically unstable and require an excellent preprocessing of the data. Furthermore the definition of the neighborhood size have to be good. A too small value lead to a too sparse matrix and if the value is too large, *short-circuit* edges will be introduced.

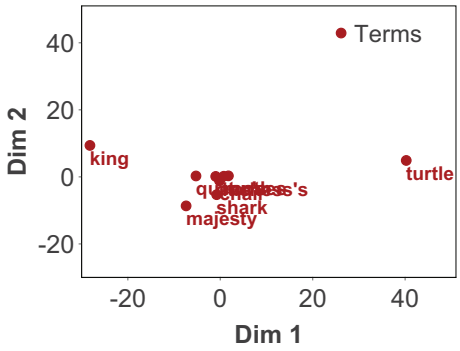


**Figure 4.11.:** Isometric feature mapping 2D-plot (R: isomap)

***Example:*** An example is given in Figure 4.11. The clustering based on the data in Table 4.16 (p. 49). It based on the `isomap` method from the statistic tool R[7].

***Similar Methods:*** A similar method to ISOMAP is given by the **Curvilinear Distance Analysis** (CDA) [146]. ISOMAP is based on metric MDS, whereas CDA is based on non-metric MDS (Sammon's non-linear mapping [188]) and the Curvilinear Component Analysis.
Roweis et al. [181] developed a method, called **Local Linear Embedding** (LLE), for dimension reduction by discovering non-linear structures in a data set. The structures are transformed into a lower dimensional space. It is a local approach, because only the neighbors of a data point are used for the computation of the approximated data point. Just like PCA, LLE do not involve local minima in its optimizations.

### Special Methods for Text Analysis

A lot of analysis approaches for text documents uses the standard dimension reduction methods. But there are dimension reduction methods developed for text analysis as well. One of the simplest dimension reduction approach for text data is the detection and selection of the keywords. Afterwards only the keywords will be used for further analysis. Another feature selection method is **Document Frequency Thresholding** (DF Thresholding) [234]. First the number of documents in which a term occur will be calculated. Then terms whose document frequency is greater than a defined threshold will be selected. The **Term Strength** (TS) [223] [233] [235] is a method for vocabulary reduction. It defines the importance of a term based on how likely it occurs in closely related documents. Similar feature selection methods are **Mean TFIDF** [210] and **Term Frequency Variance** (TfV) [210]. Another possibility is the definition of a threshold which is applied on the actual term weight. Terms whose weight is smaller than the threshold will be deleted. The definition of a threshold value depends on the selected term weighting. The method is called **Term Weight Thresholding** (TW Thresholding).
Jeon et al. [115] represents a method which apply orthogonal transformation on centroids, called **CentroidQR**. The method based on the vector subspace computation.
A combination of dimension reduction, clustering and visualization is given by **Self- Organizing Maps** (SOM) from Kohonen et al. [133]. By this non-linear method the data will be plotted in a one or two dimensional space, whereby similar data are grouped and represented by a color. The method uses other dimension reduction methods. An alternative to SOM is **Generative Topographic Map** (GTM) from Bishop et al. [22]. This method is non-linear as well.

### Evaluation for Text Analysis

Dimension reduction improves clustering and classification - this insight can be found in nearly all regarded evaluations. Table 4.18 summarizes the results of the evaluations.
Yang et al. [234] compared the following feature selection methods using text categorization: Document Frequency Thresholding (DF) [234], Information Gain (IG) [179], Mutual Information (MI) [173], CHI-Square test [76] and Term Strength (TS) [223] [233] [235]. Two data sets and the classifier k-Nearest Neighbor and Linear Least Squares Fit mapping

[7] http://www.r-project.org/

were used. To measure the effectiveness of the methods the standard recall and precision were used. IG and the CHI-Square test are the most effective feature selection methods. The performance of DF is similar. There are a strong relation between the term values of IG, CHI-Square test and DF. The performance of the other techniques are poor.
An evaluation of Independent Component Analysis (ICA), Latent Semantic Indexing (LSI), Document Frequency Thresholding (DF) [234], Random Projection (RP) [125] [121] is given by Tang et al. [209]. DF is the only feature selection approach by the measures. Five benchmark data sets were used. The dimensionally reduced data sets were clustered using k-means. The quality of the clusters was measured by a modified Purity [241] version. Based on the results the following ranking of the techniques can be derivated: ICA>LSI>DF>RP. ICA and LSI have the best overall performance, while RP consistently have the worst performance.
In another work from Tang et al. [210] three feature transformation techniques (Independent Component Analysis (ICA), Latent Semantic Indexing (LSI), Random Projection (RP) [125]) and three feature selection techniques (Document Frequency Thresholding (DF) [234], mean TFIDF [210] and Term Frequency Variance (TfV) [210]) for document clustering are evaluated. Based on three data sets the methods were compared with the k-means clustering algorithm. A modified Purity measure [241] was used for defining the quality of the clusters. Like the further analysis, ICA and LSI have a better performance than RP. Using text clustering, mean TFIDF and TfV are superior than DF. A comparison of feature selection and transformation methods offer, that the best results of mean TFIDF and TfV are similar to ICA and LSI, but they need much more dimensions. A combination of feature selection and feature transformation methods improves the dimension reduction again. Combining ICA with mean TFIDF or TfV compared to ICA and LSI reduces the computational cost while the clustering performance is preserved.

| **Paper** | **Methods** | **Result** |
|---|---|---|
| Yang et al. [234] | DF, IG, MI,<br>CHI-Square test, TS | IG, CHI-Square, DF<br>> MI, TS |
| Tang et al. [209] | ICA, LSI, DF, RP | ICA > LSI > DF > RP |
| Tang et al. [210] | ICA, LSI, RP, DF,<br>mean TFIDF, TfV | ICA+mean TFIDF, ICA+TfV<br>> ICA, LSI, mean TFIDF, TfV<br>> RP, DF |
| Underhill et al. [218] | PCA, MDS, ISOMAP,<br>LLE, LDM | MDS, ISOMAP<br>> PCA, LLE, LDM |
| Huang et al. [106] | PCA, MDS, SOM | MDS<br>> PCA, SOM |

**Table 4.18.:** Evaluation of dimension reduction methods

Underhill et al. [218] compared Principal Component Analysis (PCA), Multidimensional Scaling (MDS), Isometric Feature Mapping (ISOMAP), Local Linear Embedding (LLE) and Lafon's Diffusion Maps (LDM) for text classification using the kNN classifier (k-Nearest

Neighbor Classifier) and Linear Classifier. Three data sets (Science News, Science & Technology, Google News) are used. Dimension reduction improves classification in most of the tests. The results of PCA depends on the data set. By the Science & Technology data set, the results of PCA are poor. But the results of the other data sets are very well. MDS and ISOMAP have the best performance. Also by a small number of dimensions, they achieve very good results. The advantage of MDS compared to the other techniques, which are mostly newer and more complex, was surprising. Because text documents have linear properties, the non-linear techniques have no advantage concerning MDS or ISOMAP. In summary, the measures show that the best dimension reduction technique depends on the underlying data set and non-linear techniques are not suitable for dimension reduction of text data sets.
Huang et al. [106] compared different dimension reduction methods for text visualization. Dimension reduction was used for transforming the data set into a two-dimensional space. The methods Principal Component Analysis (PCA), Multidimensional Scaling (MDS), Self-Organizing Maps (SOM) and similarity-based dimension clustering were used. Like the evaluation before, MDS generates good transformations also by eliminating many dimensions. PCA needs three to five dimensions (components) for a good classification and visualization of the text documents. SOM has a discretization problem. Sometimes the problem can be solved, however the computation time can become unacceptable.

### 4.3.4. Clustering

Clustering is a technique which sort *N* objects (e.g. documents) into *M* groups, called clusters, in which similar objects are in one group. There is a high similarity between objects of the same group (homogeneity within the cluster) and a low similarity between objects of different groups (heterogeneity between the clusters) [7]. By many clustering techniques *M* is unknown at the beginning. The following descriptions are based on the publications [68], [65] and [98].
The resulting clusters can be disjunct or overlapping. Using disjunct clustering each object will be assigned to one cluster. With overlap clustering each object can be assigned to more than one cluster.
The clustering methods can be differentiated into hierarchical, fuzzy, center-based, search-based, graph-based, grid-based, density-based, model-based and subspace. The classification is derived from Gan et al. [68]. Some example methods are given in Figure 4.19. Most of the methods are described in Gan et al. [68]. There are methods which belong to more than one field.

***Hierarchical Clustering:*** This clustering technique creates a hierarchy of clusters, that can be represented in a tree diagram, called dendrogram. Each leaf of the dendrogram represents one object. The nodes are clusters and the root is the cluster including all objects. For the creation of the hierarchy two possibilities exist. Bottom-up clustering is an agglomerative approach. Initial, each object forms a cluster. By each step two clusters are summarized. A divisive (partition) approach is the top-town clustering. Initially, there are one cluster. By each step the clusters is divided.

***Fuzzy Clustering:*** Fuzzy clustering affords overlap clustering. Each data point is associated with every cluster using a membership function. The approach based on the fuzzy sets concept from Zadeh et al. [238].

***Center-based Clustering:*** By this algorithms each cluster is represented by a center. Center-based clustering uses objective functions. The goal is to minimize these functions. The algorithms are appropriate for clustering large and high-dimensional data sets (Gan et al. [68]).

***Search-based Clustering:*** Many clustering algorithms find a local optimal solution. Search-based clustering algorithms explore the solution space and find a globally optimal solution.

***Graph-based Clustering:*** Graph-based clustering apply clustering on a graph to partition them. Thus the data set has to be transformed to a graph before.

***Grid-based Clustering:*** Grid-based clustering algorithms work with the value space of the data objects and not with the data objects like other clustering algorithms. By this approach a grid structure will be created and used for the clustering. Because of the low computational complexity the grid-based clustering algorithms can be used for large data sets.

***Density-based Clustering:*** Density-based clustering uses connectivity and density functions for clustering. The resulting clusters are dense areas separated by low-density areas. At the start of the algorithm the number of clusters are not required. Like grid-based clustering the algorithms has a low computational complexity. Thus they can be used for large data sets.

| | |
|---|---|
| **Hierarchical**<br>SLINK [197], Group Average [112], Centroid [112], Median [112], Ward [222], AGNES [127], BIRCH [240], CURE [80], DIANA [127], WBSC [148] | **Fuzzy**<br>Fuzzy k-means [18], Fuzzy k-modes [107], Fuzzy c-means [19] [23], Gath-Geva [70], Gustafson-Kessel [82] |
| **Search-based**<br>Tabu Search [72] [73], Variable Neighborhood Search [91], j-means [90], GKA [137], SARS [105] | **Center-based**<br>k-means [153], PAM [127], k-modes [109] [108], bisecting k-means [147] [204], spherical k-means [55], Buckshot [46], Fractionation [46] |
| **Graph-based**<br>Chameleon [124], CACTUS [69], ROCK [81], STC [239] | **Grid-based**<br>STING [221], OptiGrid [100], GRIDCLUS [191], GDILC [242], WaveCluster [195] |
| **Density-based**<br>DBSCAN [59], BRIDGE [47], DBCLASD [232], DENCLUE [99] | **Model-based**<br>COOLCAT [12], STUCCO [15], AutoClass [33] [34] |
| **Subspace**<br>CLIQUE [4], ENCLUS [35], MAFIA [170], DOC [178] | |

**Table 4.19.:** Classification of clustering techniques

***Model-based Clustering:*** Model-based clustering uses models for clustering and tries to find the best fit between the data set and the models. Possible models are the Gaussian models or the Hidden Markov models. Using mixture likelihood models, the Expectation-Maximization (EM) algorithm is mostly used for finding the maximum likelihood estimation. The result are the parameters of a finite mixture probability density. The Bayesian Information Criterion (BIC) can be used for determining the final model, whereas the best model is the model with the highest BIC.

***Subspace Clustering:*** Subspace clustering will be used for an effective and efficient clustering of high-dimensional data sets without using of dimension reduction techniques. However the clusters are embedded in subspaces of the data set. The subspaces have their own dimensions. Compared to clustering with dimension reduction no information will be lost.

Clustering allows the summarization of similar documents or terms. In information retrieval, clustering is be applied for improving the efficiency and effectiveness. Text Mining uses clustering for finding similar documents or terms, the identification of different topic areas, etc.

### Distance and Similarity Measures

A lot of clustering techniques use a distance or similarity measure. A similarity matrix includes the similarities between the objects. The definition of a similarity measure depends on the structure of the data set and on the analysis problem. It is possible to group documents with the same length or the same content, semantically similar terms, terms of the same part of speech, etc.
The similarity of two documents, two words or a document and a query is the similarity of two vectors. The calculation of the similarity of an object and a cluster is possible as well. There are a lot of distance and similarity measures. Following, the measures are explained by two vectors $\vec{v} = v_1, ..., v_n$ and $\vec{w} = w_1, ..., w_n$ of the length $n$.
First the distance measures $dist(\vec{v}, \vec{w})$ are described. The smallest value corresponds to the most similar vector pair.

| | |
|---|---|
| Manhattan distance:<br>(City-Block-Metric) | $dist(\vec{v}, \vec{w}) = \sum_{i=1}^{n} \lvert v_i - w_i \rvert$ |
| Euclidean distance: | $dist(\vec{v}, \vec{w}) = \sqrt{\sum_{i=1}^{n} \lvert v_i - w_i \rvert^2}$ |
| Minkowski distance: | $dist^p(\vec{v}, \vec{w}) = (\sum_{i=1}^{n} \lvert v_i - w_i \rvert^p)^{\frac{1}{p}}$ |
| Canberra distance: | $dist^p(\vec{v}, \vec{w}) = \sum_{i=1}^{n} \frac{\lvert v_i - w_i \rvert}{v_i + w_i}$ |

The Minkowski distance is a generalization of the Manhattan and Euclidean distance. If $p = 1$, the Manhattan distance is given. The Euclidean distance will be defined, if $p = 2$.
Similarity measures $sim(\vec{v}, \vec{w})$ describes the similarity between two vectors $\vec{v}$ and $\vec{w}$. The greatest value corresponds to the most similar vector pair.

| | |
|---|---|
| Simple-Matching coeffcient: | $sim(\vec{v}, \vec{w}) = \sum_{i=1}^{n} v_i \cdot w_i$ |
| | Binary: $\lvert X \bigcap Y \rvert$ |
| Cosine coefficient: | $sim(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{n} v_i \cdot w_i}{\sqrt{\sum_{i=1}^{n} v_i^2 \cdot \sum_{i=1}^{n} w_i^2}}$ |
| | Binary: $\frac{\lvert X \bigcap Y \rvert}{\sqrt{\lvert X \rvert \times \lvert Y \rvert}}$ |

| | |
|---|---|
| Dice coefficient: | $sim(\vec{v}, \vec{w}) = \frac{2 \cdot \sum_{i=1}^{n} v_i \cdot w_i}{\sum_{i=1}^{n} v_i + \sum_{i=1}^{n} w_i}$ |
| | Binary: $\frac{2\|X \bigcap Y\|}{\|X\|+\|Y\|}$ |
| Jaccard coefficient: (Tanimoto) | $sim(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{n} v_i \cdot w_i}{\sum_{i=1}^{n} v_i + \sum_{i=1}^{n} w_i - \sum_{i=1}^{n} v_i \cdot w_i}$ |
| | Binary: $\frac{\|X \bigcap Y\|}{\|X \bigcup Y\|} = \frac{\|X \bigcap Y\|}{\|X\|+\|Y\|-\|X \bigcap Y\|}$ |
| Overlap coefficient: | $sim(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{n} min(v_i, w_i)}{min(\sum_{i=1}^{n} v_i, \sum_{i=1}^{n} w_i)}$ |
| | Binary: $\frac{\|X \bigcap Y\|}{min(\|X\|, \|Y\|)}$ |

Haenelt et al. [84] introduces important similarity measures for document clustering and they analyze their behavior. Applying the Euclidean distance on normalized vectors, the resultant ranking is the same like the Cosine coefficient. The Cosine coefficient for normalized vectors is equal to the Simple-Matching coefficient. Using the Simple-Matching coefficient the results are not normalized. If the number of non-zero values is very different, the Cosine coefficient will be have higher values than the Dice coefficient. By a small number of common values, the Jaccard coefficient is smaller than the Dice coefficient. Using the Overlap coefficient the result will be 1.0, if each value of the first vector is greater than the value of the second value or in reverse.
A distance measure can be converted to a similarity measures [98] [24]. The variable $m$ defines the greatest distance value.

$$sim(\vec{v}, \vec{w}) = \frac{1}{dist(\vec{v}, \vec{w}) + 1} \tag{4.43}$$

$$sim(\vec{v}, \vec{w}) = 1 - \frac{dist(\vec{v}, \vec{w})}{m} \tag{4.44}$$

The result of a similarity measure can be converted to a distance measure as well [24]. If $0 \leq sim(\vec{v}, \vec{w}) \leq 1$, the following transformations are existed. Otherwise the 1 can be replaced by the maximum value.

$$dist(\vec{v}, \vec{w}) = 1 - sim(\vec{v}, \vec{w}) \tag{4.45}$$

$$dist(\vec{v}, \vec{w}) = \sqrt{1 - sim(\vec{v}, \vec{w})} \tag{4.46}$$

$$dist(\vec{v}, \vec{w}) = -log(sim(\vec{v}, \vec{w})) \tag{4.47}$$

$$\tag{4.48}$$

Anderberg et al. [7] [65] and Cha et al. [32] represent a variety of distance and similarity measures. A discussion about the most suitable measures for documents is given by Salton et al. [186] [65]. Willett et al. [225] tests some similarity coefficients and show that the measure should be normalized by the length of the document vectors [65]. For document clustering the Dice, Jaccard and Cosine coefficients are often used, because these similarity measures are simple and normalized. For query matching, the Cosine coefficients is one of the most common similarity measure [16]. The Overlap coefficient is not suitable for the comparison of documents.

### Hierarchical Clustering

For Hierarchical Clustering a similarity measure have to be chosen, e.g. the Euclidean distance or the Cosine coefficient (see Section 4.3.4, p. 65). Furthermore a method for the determination of the similarity of two clusters $C_1$ and $C_2$ are needed. $|C|$ defines the number of objects in a cluster. The following methods exists:

- Single-linkage: $sim_C(C_1, C_2) = max_{x \in C_1, y \in C_2}(sim(x, y))$
  The similarity between two clusters is the similarity between the most similar objects.
- Complete-linkage: $sim_C(C_1, C_2) = min_{x \in C_1, y \in C_2}(sim(x, y))$
  The similarity between two clusters is the similarity between the most dissimilar objects.
- Group Average (UPGMA): $sim_C(C_1, C_2) = \frac{1}{|C_1| \cdot |C_2|} \sum_{x \in C_1, y \in C_2}(sim(x, y))$
  The similarity between two clusters is the average similarity between the objects.
- Centroid
  The similarity between two clusters is the similarity between the centroids of the clusters. The centroid vector $c$ of a corpus $D$ with the documents $d$ is calculated by $c = \frac{1}{|D|} \sum_{d \in D} d$.
- Median
  The similarity between two clusters is the similarity between the medians of the clusters.
- Minimum variance (Ward)
  This method based on the sum of squares between each cluster pair, summed over all objects. The goal is the merging of clusters in order to minimize the sum of squares within clusters. The error sum of squares is the total sum of squared distances of all points from the mean. For measuring the similarity the squared Euclidean distance is used.

In the following, the algorithm for bottom-up clustering is explained. The algorithm for top-town clustering is reversed. It starts with all objects in one cluster and at each step a cluster will be split. The algorithm ends, if each object represents one cluster.

1. Initializations
   Creation of initial clusters, each object represents one cluster.

2. Similarity calculations
   Compute the similarity between each cluster based on the chosen similarity measure (see Section 4.3.4, p. 65) and method (single-linkage, complete-linkage, etc.).

3. Cluster merging
   Merging of clusters with the highest similarity.

4. Termination
   If only one cluster exists, the algorithm is terminated. Based on the computed dendrogram the clusters can be extracted.

5. Continue with step 2.

The number of clusters haven't be defined at the beginning of the clustering. Additionally to the clusters, a hierarchical order of the clusters will be found. The algorithms are very complex, because they have high space and time requirements. Good algorithms have a minimal complexity of $O(N^2)$ ($N$ objects). Furthermore not all technologies are stable against noise.

***Example:*** An example of group average hierarchical clustering is given in Figure 4.12. The data of the clustering can be found in Table 4.16 (p. 49). It based on the `agnes` method from the statistic tool R[8]. The distance measure is the Manhattan distance. Initially, each chapter corresponds to one cluster. In the first step, the chapter 9 and 10 are summed up to one cluster. The next step merges the chapters 8 and 11 to one cluster. Now there are three clusters: (ch7), (ch8, ch11) and (ch9, ch10). In the next step, the cluster including the chapter 7 and the cluster including the chapters 8 and 11 are merged. The final step sums up the clusters (ch7, ch8, ch11) and (ch9, ch10) to one cluster.

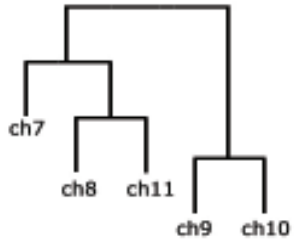


**Figure 4.12.:** Example: hierarchical agglomerative clustering
(R: agnes, UPGMA, Manhattan distance)

### Center-based Clustering using K-means

The k-means clustering algorithm, developed from MacQueen et al. [153], is one of the most used clustering algorithm. It is center-based clustering, because each cluster are represented by a centroid. The centroid is a virtual object with a minimal distance to all objects (e.g. documents) of the cluster. The number of resulting clusters are defined at the start of the algorithm.
The k-means algorithm is efficient, because the computational complexity is linear. It depends on the size of the data set. The performance and quality of the result depends on the initialization phase. The algorithm starts with a randomly definition of intial clusters. Depend on the initial definition, a clustering solution will be searched. Mostly, a local optimum can be found.

Given a data set $D$ with document vectors $d$. The algorithm work as follow [68]:

1. Configurations
   Definition of the number of clusters $k$ and selection of a distance measure $dist$.
2. Initializations
   Randomly definition of the initial clusters $C = C_1, ..., C_k$.
3. Centroid computation
   Calculation of the centroid $c_i$ of each cluster $C_i$: $c_i = \frac{1}{|C_i|} \sum_{d \in C_i} d$.

[8] http://www.r-project.org/

4. Distance computation
   Compute the distance $dist(d_j, c_i)$ between each document vector $d_j$ and each centroid $c_i$ (see Section 4.3.4, p. 65).

5. Cluster adaption
   Move each document $d_j$ to the cluster with the highest centroid similarity: $min(\sum_{i=1}^{k} dist(d_j, c_i))$.

6. Termination
   If no document was moved or the error function $E$ does not change significantly, the algorithm is terminated.

$$E = \sum_{i=1}^{k} \sum_{d \in C_i} dist(d, c_i) \tag{4.49}$$

7. Continue with step 3.

***Example:*** An example of k-means clustering is given in Table 4.20. The clustering based on the data in Table 4.16 (p. 49). The algorithm was executed using the method `pam` from the statistic tool R[9]. The distance measure is the Manhattan distance.

| Cluster | Chapters |
|---|---|
| 1 | ch7, ch8, ch11 |
| 2 | ch9, ch10 |

**Table 4.20.:** Example: k-means
(R: pam, Manhattan distance)

***Similar Methods:*** There are a lot of extensions of k-means, like fuzzy k-means [18], Buckshot [46], bisecting k-means [147] [204] or spherical k-means [55]. Spherical k-means is an algorithm for document clustering based on a similarity measure. A more robust version of k-means is given by Kaufman et al. [127], called PAM.
Bisecting k-means is a combination of divisive hierarchical clustering and k-means. Initially one cluster exists. At each step, a cluster will be split into two subclusters using k-means. Thereby k-means will be executed n times and the best result will be used.

#### Graph-based Clustering using Chinese Whispers

Graph-based Clustering is based on the (Weighted) Graph Partitioning problem. Given a Graph $G = (V, E)$ with vertexes $V$ and weighted edges $E$, the goal is to find a set of edges whose removal partitions the graph into subgraphs [205]. The set of edges have a minimum sum of edge weights.

Chinese Whispers is a randomized graph-clustering algorithm. The algorithm considered the neighbors of each node. Compared to other clustering algorithm no distance metric is used.

[9] http://www.r-project.org/

The similarity between two objects is represented by an edge between the corresponding nodes. The strength of the similarity can be represented by an edge weight.
The clustering algorithm is time-linear $O(k)$ corresponding to the number of edges $k$. If there are edges between all nodes, the algorithm has a quadratic time complexity $O(k) = O(n^2)$ while $n$ is the number nodes.

Given a weighted Graph $G = (V, E)$ with vertexes $(v_i) \in V$ and weighted edges $(v_i, v_j, w_{ij}) \in E$. The method $class(v_i)$ returns the class (cluster number) of the node $v_i$. The algorithm work as follow [20]:

1. Initializations
   Definition of the initial classes, all nodes get different classes: $\forall v_i in V : class(v_i) = i$.

2. Class (Cluster) adaption
   Determine for each node the strongest class in the local neighborhood. This is the class with the greatest sum of edge weights to the current node:
   $\forall v_i in V : class(v_i) =$ *highest ranked class in neighborhood of* $v_i$

3. Termination
   If no node was moved to another class, the algorithm will be terminated.

4. Continue with step 2.

There are different approaches for the definition of the highest rank class in the neighborhood of a node. Beside the definition of the sum of edge weights, also the degree can be used.

***Example:*** An example is given in Figure 4.13. The clustering based on the data in Table 4.16 (p. 49). The graph clustering was executed with the Chinese Whispers tool[10]. The edge weights are calculated with the cosine similarity (see Section 4.3.4, p. 65).

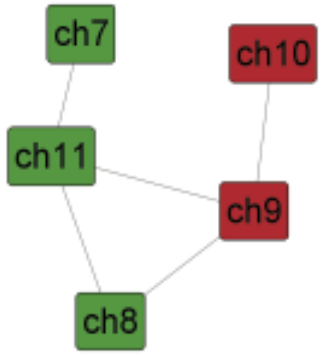


**Figure 4.13.:** Example: Chinese whispers
(Cosine similarity)

[10]http://wortschatz.informatik.uni-leipzig.de/~cbiemann/software/CW.html

### Cluster Quality Measurement

The validity of clusters can be measured in different ways. There are external and internal quality measures.

***External Quality Measures:*** External quality measures use external criteria for the cluster evaluation, e.g. a user-based reference clustering.
Examples are Precision, Recall and the F-measure, which are described in [143]. $|C_i|$ is the number of documents in the cluster $C_i$ and $|X_i|$ is the number of correct documents in the cluster $C_i$. $|Y_i|$ defines the number of all possible correct documents. This includes the correct documents of the cluster $C_i$ and the documents which should be belonged to the cluster $C_i$. The range of the F-measure is between null and onw. By an optimal clustering solution, the F-measure value is one.

| | |
|---|---|
| Precision: | $P = \frac{\|X_i\|}{\|C_i\|}$ |
| Recall: | $R = \frac{\|X_i\|}{\|Y_i\|}$ |
| F-measure: | $F = \frac{2PR}{P+R}$ |

***Internal Quality Measures:*** The internal quality measure based on metrics [136]. The metrics are applied on the data set and the clustering result. Their calculations based solely on the resulted clustering.
Classical cluster quality measures are Dunn Index [203], Davies Bouldin Index [203] and SD Validity Index [87]. Further examples are given in [136], [87] and [203].
Given a set of clusters $C = C_1, ..., C_n$, each cluster $C_i$ has a cluster center $c_i$. $D$ is the set of all documents. The following measures are needed for the definition of the internal quality measures. $D_{max}$ defines the maximum distance between cluster centers and $D_{min}$ defines the minimum distance between cluster centers.

| | |
|---|---|
| $dist(C_i, C_j)$ | Cluster distance measure<br>Example: The distance between two clusters is the distance between the centers of the clusters. |
| $\Delta(C_i)$ | Cluster diameter measure<br>Example: The distance between the most dissimilar documents. |
| $\sigma(C_i)$ | Cluster scatter measure<br>Example: The average distance of all documents in the cluster $C_i$ to the cluster center $c_i$. |
| $Dis = \frac{D_{max}}{D_{min}} \cdot \frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{n} \|c_i - c_j\|}$ | Total separation between clusters |
| $Scatt = \frac{1}{n} \sum_{i=1}^{n} \frac{\|\sigma(C_i)\|}{\|\sigma(D)\|}$ | Average scattering for clusters |

| | |
|---|---|
| $\omega$ | Weighting factor (for example the number of clusters) |

Now the quality measures can be defined. Since the terms of the SD Validity Index have a different range, a weighting factor $\omega$ is needed. A high SD Validity Index indicates a good clustering, whereas a low Dunn Index or a low Davies Bouldin Index point to that the clustering is well.

| | |
|---|---|
| Dunn Index: | $D = \frac{min_{i \neq j}(dist(C_i, C_j))}{max_{i<k<n}(\Delta(C_k))}$ |
| Davies Bouldin Index: | $DB = \frac{1}{\lvert C \rvert} \sum_{i=1, i \neq j}^{n} max(\frac{\Delta(C_i)+\Delta(C_j)}{dist(C_i, C_j)})$ |
| SD Validity Index: | $\omega \cdot Scatt + Dis$ |

### Special Methods for Text Analysis

Since text document clustering becomes ever more important, there are special clustering techniques developed for document clustering.
Cutting et al. [46] developed two partition-based clustering algorithm called Buckshot and Fractionation. **Buckshot** selects a small random sample of documents and cluster them. Afterwards the rest of the documents can be assigned to the clusters. The complexity of the algorithm is $O(kn)$ while $k$ is the number of clusters and $n$ the number of documents. **Fractionation** breaks the documents into buckets. Afterwards clustering on each bucket will be applied. The clusters off all buckets are now treated as individual documents and the whole process is repeated until $k$ clusters will exist. The complexity is $O(mn)$ while $m$ is the number of documents in a buckets and $n$ the number of all documents.
A **Word-based Soft Clustering** (WBSC) approach is given by Lin et al. [148]. Initially, each word form a cluster including all documents that have that word. Using a hierarchical agglomerative clustering and the Jaccard coefficient (see Section 4.3.4, p. 65), clusters with similar sets of documents will be merged. With this approach additionally the words are clustered. The result are cluster of words (the topics) containing a set of documents. In this work, the algorithm are compared with k-means, Fractionation and Buckshot. 2000 web documents and 2000 newsgroup documents are used. Since the complexity of the algorithm is linear to the number of documents, it have the best execution times by the measures.
A method, called **Spherical K-means**, for document clustering based on k-means are given by Dhillon et al. [55]. The resulting clusters of this algorithm are represented by a concept vector. A concept vector is the centroid of the cluster. Compared to standard k-means a similarity measure is used.
Zamir et al. [239] represents a clustering method, called **Suffix Tree Clustering** (STC). The clustering are based on phrases shared between text document. Dhillon et al. [54] developed a method which cluster simultaneously documents and their words.

### Evaluation for Text Analysis

By the multiplicity of clustering algorithms not all are suitable for document or word clustering. Depending on application and text corpus some clustering algorithms are better than

others. An overview about the clustering evaluations is given in table 4.21.
Zhao et al. [243] evaluate some Hierarchical Clustering algorithms for text corpora. Six divisive (partition) and nine agglomerative methods are compared. Additionally a new constrained agglomerative approach are tested. The F-measure and 12 different text corpora are used for comparing the results. The experiments show that divide clustering approaches are better suitable for text clustering than agglomerative approaches. The constrained agglomerative algorithm outperform the agglomerative algorithms and in most cases they have better results than the divide approach.

| **Paper** | **Methods** | **Result** |
|---|---|---|
| Zhao et al. [243] | Divisive hierarchical<br>Agglomerative hierarchical<br>Constrained agglomerative | Constrained<br>> divisive<br>> agglomerative |
| Steinbach et al. [204] | Agglomerative hierarchical<br>K-means<br>Bisecting k-means | Bisecting k-means<br>> k-means<br>> agglomerative |
| Zhong et al. [244] | Model-based (models: mBM,<br>mNM, vMF; strategies: hard,<br>stochastic, soft, DA)<br>Graph-based (CLUTO, BSC) | CLUTO, DA<br>> vMF<br>> BSC |
| Strehl et al. [205] | SOM, k-means<br>hyper GP, weighted GP | Weighted GP<br>> SOM, hyper GP<br>> k-means |
| Steinbach et al. [204] | PDDP, EM | EM > PDDP |

**Table 4.21.:** Evaluation of clustering methods

Another comparison of document clustering techniques is given by Steinbach et al. [204]. The two main approaches of text clustering are compared: agglomerative hierarchical clustering and k-means (standard k-means [153] and bisecting k-means [147]). Basically, the cluster quality of agglomerative hierarchical clustering is higher, but the algorithms has a quadratic time complexity. Using k-means, the cluster quality is not so well, but the time complexity is linear. Therefore both approaches are combined. Internal (overall similarity) and external quality measures (entropy [194], F-measure [143]) are used for comparing the results. The result of the measures show that standard k-means generates better clusters than hierarchical clustering. In this paper, bisecting k-means is the best clustering technique. The produced clusters are better than the clusters from k-means and as good or better than the clusters from the agglomerative hierarchical approach.
Zhong et al. [244] compares some model-based clustering techniques. Model-based clustering is another possibility for a more efficient clustering than the hierarchical agglomerative clustering. The experiments based on three probabilistic models (multivariate Bernoulli model (mBM) [159], multinomial model (mNM) [202], von Mises-Fisher model (vMF)

[155]) in conjunction with four data assignment strategies (hard, stochastic, soft, deterministic annealing (DA)). Additionally the model-based approaches are compared with two graph-based approaches (CLUTO [123], bipartite spectral co-clustering method (BSC) [54]). All twelve algorithms are tested with and without previous feature selection. 20 newsgroups and several data sets from CLUTO [123] are used for the evaluation. In the experiments CLUTO and DA have the best performance but they have the highest computational costs. The vMF model has good performance and is not computationally expensive. In most cases the vMF model is better than the bipartite spectral co-clustering method. Meila et al. [162] published another study of model-based clustering methods. Here the EM algorithm are the best.
Strehl et al. [205] compare similarity measures (Euclidean, Cosine, Pearson correlation, extended Jaccard) in combination with different clustering methods (random, self-organized feature maps (SOM), hyper-graph partitioning (hyper GP), generalized k-means, weighted graph partitioning (weighted GP)). The data base are web documents (industry web-pages, news web-pages). Cosine and extended Jaccard have the best measure, whereas Euclidean performs poorest. The weighted graph partitioning method performs better than the other clustering algorithms. Gaussian-based methods, e.g. k-means, have a very poor performance [205].
A comparison between the Principal Direction Divisive Partitioning (PDDP) [25] and the EM algorithm [52] is given by He et al. [94]. The results of the EM algorithm are the best, because the similarity measure of the Principal Direction Divisive Partitioning does not use the distance. Documents will be similar, if the distance between the documents are small.

### 4.3.5. Further Analysis Methods

Beside clustering and dimension reduction, further text analysis methods exist. Well-known analysis methods are described in this section. Since the methods are not used in this work, only a short introduction is given.

#### Indexing/Keyword Extraction

Is the automatic extraction/detection of keywords in text. There are linguistic and statistic approaches. A keyword extraction algorithm can be corpus-based or document-based. Corpus-based algorithms extract keywords based on a corpus (document collection), like KEA [228] or Differential Analysis [98]. Document-based approaches only need the document for the extraction of keywords. Examples are the Repeated-String-Patterns (RSP) [217], B&C [13] or Cooccurrence [158]. An overview about the techniques is also given in [130].

#### Visualization

The visualization of words or documents and their dependencies is another very important field of activity. There are a lot of possibilities to visualize data sets based on text, e.g. the visualization in a diagram (see section 4.3.3) or a graph. One example is the **Crossmapping** technique [166]. It visualizes relations in a kind of diagram. For example, terms are plotted on the x axis and documents are plotted on the y axis. A point in the diagram visualizes the number of times a term occur in the document. Another example is the **Word Cloud** visualization. It based on the the tag cloud visualization. A tag cloud represents the most popular tags [93]. The importance of a tag is visualized by the font size or the font color. This concept can be applied on text as well. Using texts, the important terms (keywords) can be visualized. After defining the keywords and their importance, a layout algorithm visualizes them. An example is given in Figure 4.14.

**Figure 4.14.:** Word cloud example

### Classification

Classification is a technology for grouping objects into categories. Compared to clustering the categories are defined at the beginning.

### Text Summarization

Witte et al. [226] describes text summarization as a method for shortening text without reducing the information content. Another definition is given by Luhn et al. [152]. He defines text summarization as the creation of a summary of a text that satisfy the user requirements.

### Significant Co-occurrences

Is the detection of semantic relations in text. Significant co-occurrences are terms that arise very often together. Statistical fundamentals are used here [98].

## 4.4. Research Fields and Applications

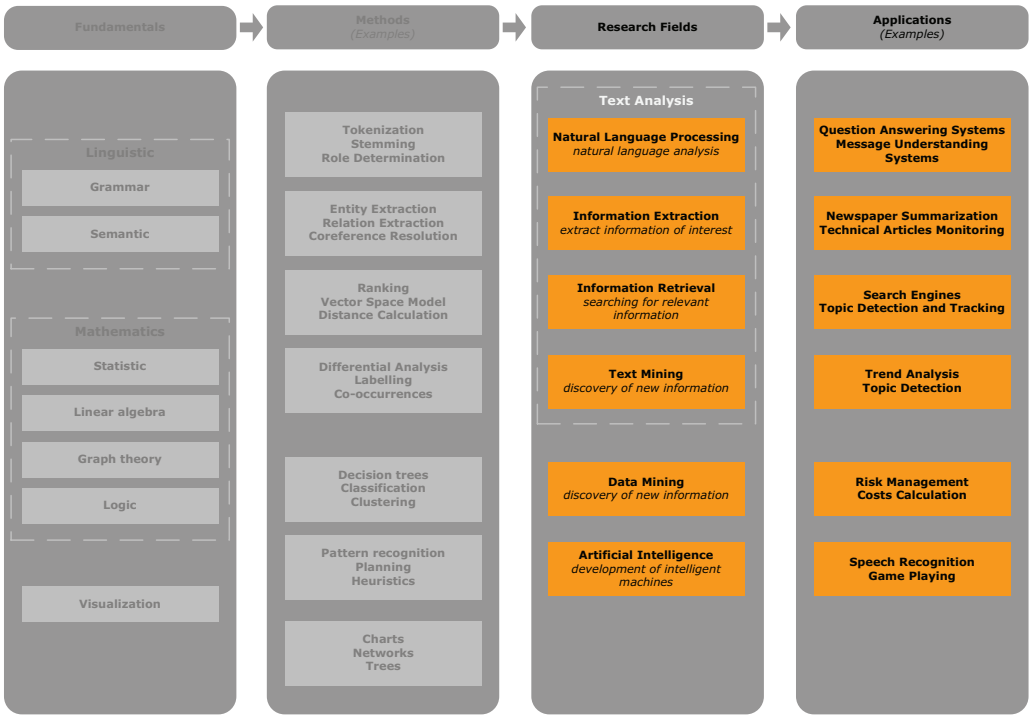


**Figure 4.15.:** Text analysis: research fields and applications

In the following section, the research fields with their definitions and developed methods are presented. Some applications of each discipline are listed additionally. Figure 4.15 visualizes the research fields and applications. This section allows a dissociation from text mining to the adjacent research fields. Like described in Section 4.1 (p. 23) the differentiation of text analysis research fields is difficult. In this work they are differentiated by their goals (see Figure 4.15). This section describes the research fields detailed and dissociates them from each other.

### 4.4.1. Information Extraction

*Abbreviation: IE*
*Synonyms: Entity Extraction*

**Method Examples**
- Entity Extraction
- Relation Extraction
- Coreference Resolution

**Application Examples**
- Analyse newspaper articles for finding and summarizing instances of business joint ventures [167]
- Support underwriters in analysing life insurance applications [71]
- Information Extraction for the Global Legal Information Network [103]

This research field is mainly influenced by the *Message Understanding Conference (MUC)* [36] [78] and the *Automatic Content Extraction (ACE) Evaluation*[11] [1]. Information extraction is interested in the structure of text, whereas one could say that, from an IR point of view, texts are just bags of unordered words [224]. It refers to the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources [190] [165]. For example the summarization of medical patient records by extracting diagnoses, symptoms, physical findings, test results, and therapeutic treatments in order to assist health care providers or support insurance processing [200]. Kaufman et al. [167] describes a method for monitoring technical articles describing microelectronic chip fabrication in order to capture information on chip sales, manufacturing advances, and the development or the use of chip processing technologies.
The publications *Relational learning techniques for natural language information extraction* [30] and *Empirical methods in information extraction* [31] give a very good introduction to information extraction. The article *Information Extraction* [190] describes the extraction task, the techniques used for extraction, the variety of input resources exploited, and the type of output produced. It includes rule-based and statistical methods for entity and relationship extraction.

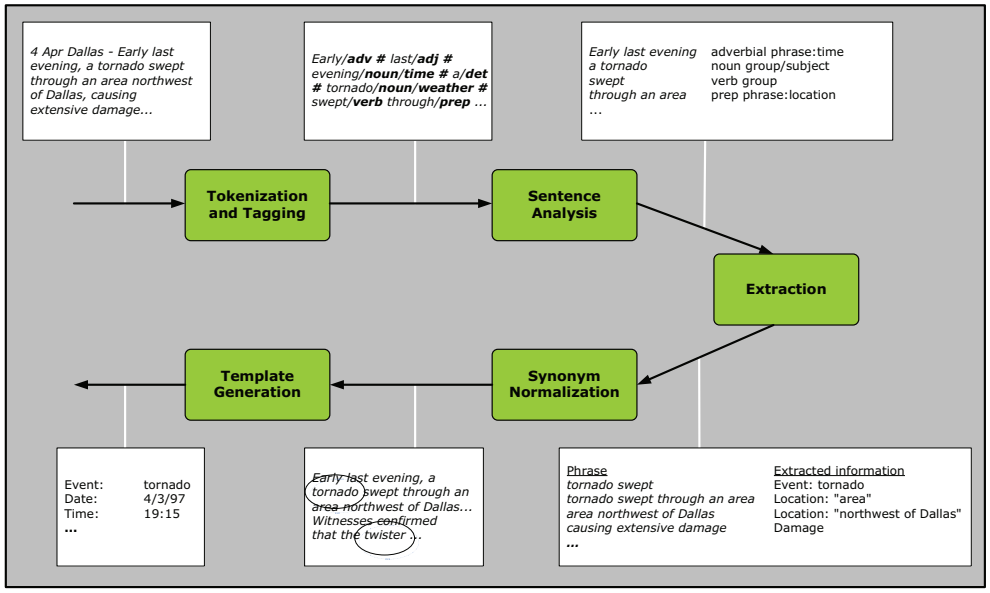


**Figure 4.16.:** Information extraction process,
*Empirical methods in information extraction* [31]

[11] http://www.itl.nist.gov/iad/mig/tests/ace/

### 4.4.2. Information Retrieval

*Abbreviation: IR*
*Synonyms: Text Retrieval, Document Retrieval, Information Access*

**Method Examples**
- String searching/matching
- Stop word removal
- Ranking
- Boolean retrieval
- Probabilistic retrieval
- Predicate-logical models
- Relevance feedback
- Index construction
- Term weighting
- Vector space model
- Distance measures
- Dimension reduction

**Application Examples**
- Web search engines
- Search engines for books, journals etc.
- Search engines for messages-, mail-, news-, blog-archives
- Search engines for documents
- Topic detection and tracking

Early definitions are very general, for example from Salton et al. [185]: "Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information". Automated information retrieval systems were originally developed for managing the huge scientific literature that has been developed since the 1940s [65]. In this paper, information retrieval is defined as follows: "The goal of information access is to help users find documents that satisfy their information needs." [8]. The information retrieval process is presented in Figure 4.17.
The books *Introduction to Information Retrieval* [154] and *Understanding Search Engines: Mathematical Modelling and Text Retrieval* [16] introduces methods of information retrieval and represents some software tools. An overview about the data structures and algorithms for building an information retrieval system is given in *Information Retrieval: Data Structures & Algorithms* [65].

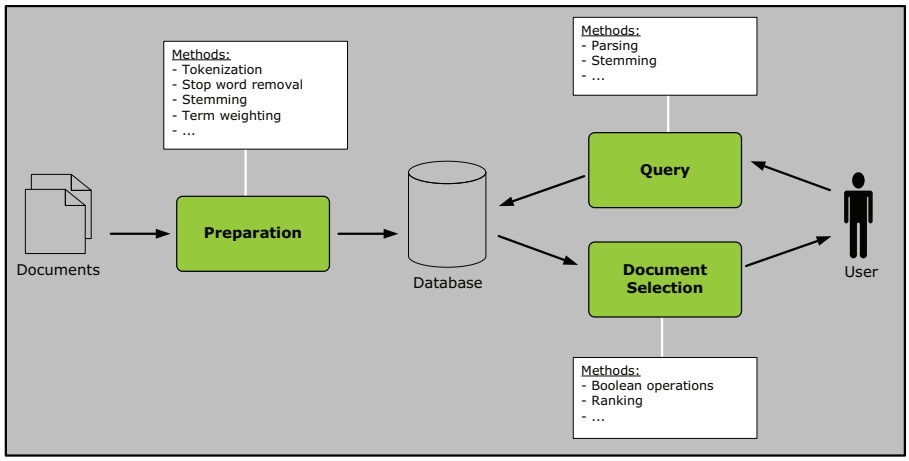


**Figure 4.17.:** Information retrieval process,
*Information Retrieval: Data Structures & Algorithms* [65]

### 4.4.3. Natural Language Processing

*Abbreviation: NLP*
*Synonyms: Computational Linguistic (CL)*

| **Method Examples** | **Application Examples** |
|---|---|
| • Tokenization | • Question answering systems |
| • Stemming | • Message understanding systems |
| • Lemmatization | • Information filtering |
| • Multiword phrase grouping | • Text categorization |
| • Word-sense disambiguation | • Automatic abstracting |
| • Anaphora resolution | • Thesaurus construction |
| • Parsing | • Speech-to-speech translation |
| • Part-of-Speech tagging | • Speech recognition |
| • Text chunking | • Speech synthesis |

The general goal of natural language processing is to provide a better natural language understanding through the use of computers [77] [132]. It is a research field between linguistics and computer science which is concerned with the computational aspects of the human language faculty [219]. Natural language processing has developed various techniques that are typically linguistically inspired, i.e. text is typically syntactically parsed using information from a formal grammar and a lexicon. The resulting information is then interpreted semantical to extract information about what was said [122]. These investigations are interdisciplinary and involve concepts from artificial intelligence, linguistics, logic and psychology. By connecting the closely interrelated fields of computer science and linguistics, NLP plays a key role in cognitive science [128].
A short instruction about NLP is given by Rindflesch et al.. The book *Speech and Language Processing* written from Jurafsky et al. [119] gives a good description about the syntax, semantic and pragmatic techniques.

### 4.4.4. Text Mining

*Abbreviation: TM*
*Synonyms: Text Data Mining, Textual Data Mining, Text Knowledge Engineering, Knowledge Discovery in Texts, Knowledge Discovery in Textual Databases, Web Farming*

| **Method Examples** | **Application Examples** |
|---|---|
| • Text summarization | • Topic identification |
| • Keyword extraction | • Trend detection |
| • Topic Detection | • Technical terms identification |
| • Differential analysis | • Extraction of characteristics from texts |
| • Significant co-occurrences | • Creation of semantic networks for knowledge management |
| | • Detection of similar documents |
| | • Text classification |

Text mining is a comparatively new discipline. For this reason a lot of definitions exists and there are a lot of denotations, like *Text Knowledge Engineering* [86] and *Web Farming* [83]. Different text mining definitions are introduced in the previous section. In this thesis a combination of the data mining and knowledge-oriented perspective described by MEHLER ET AL. [161] is used. HEARST ET AL. [97] defines text mining as a natural language extension of data mining for discovering new, previously unknown information: "The goal is to find out interesting patterns from large databases. The difference between regular data mining and text mining is that in text mining the patterns are extracted from natural language text rather than from structured databases of facts. Text Mining is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. A key element is the linking together of the extracted information together to form new facts or new hypotheses to be explored further by more conventional means of experimentation." Thereby text mining uses a lot of methods from other disciplines like tokenization, stemming, vector space retrieval, clustering. Figure 4.18 visualizes a text mining process.
A detailed introduction into text mining is given by the German books *Text Mining: Wissensgewinnung aus natürlichsprachigen Dokumenten* [226] and *Text Mining: Wissensrohstoff Text* [98]. Here the linguistic and statistic fundamentals and some methods, like clustering or tagging are described. *Text Data Mining: Theory and Methods* [201] and *Survey of Text Mining II: Clustering, Classification, and Retrieval* [17] are very good publications describing the vector space retrieval, the dimensional reduction of matrices and clustering in the context of text mining. [201] includes additionally a lot of examples. Another recommendable book is given by FELDMAN ET AL., the book *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data* [62] introduces into clustering, categorization, information extraction, network analysis and visualization, and a lot of applications are introduced.

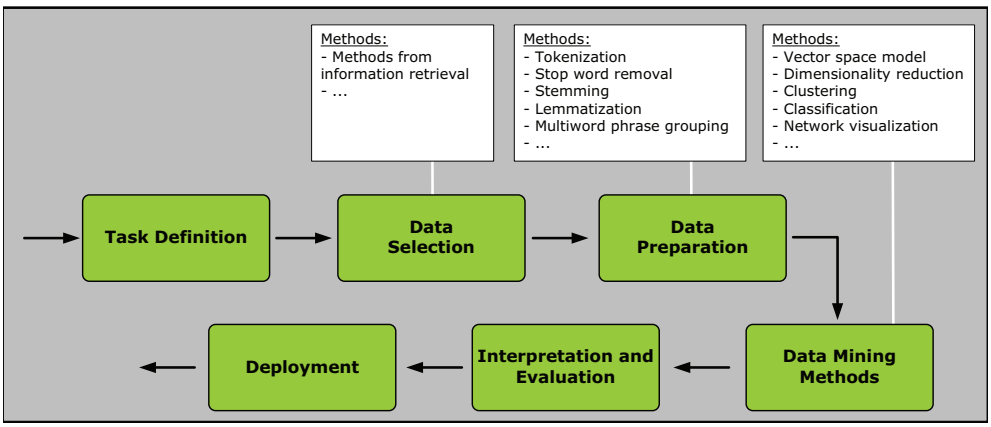


**Figure 4.18.:** Text mining process, *Text Mining* [101]

### 4.4.5. Data Mining

*Abbreviation: DM*
*Synonyms: Knowledge Mining from Data, Knowledge Extraction, Data/Pattern Analysis, Data Archaeology, Data Dredging, Knowledge Discovery from Data, Knowledge Discovery in Databases*

**Method Examples**
- Decision trees
- Neural networks
- Clustering
- Classification
- Categorization
- SNA
- Prediction
- Frequent pattern mining
- Regression

**Application Examples**
- Costs reduction
- Risk management
- Market basket analysis
- Customer Relationship Management
- Detection of target groups for marketing

Data mining is a process that uses a variety of data analysis tools to discover patterns and relationships in data that may be used to make valid predictions [39]. Another definition is given from Fayyad et al. [60]: "Data mining is a step in the KDD process that consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns (or models) over the data." (KDD = Knowledge Discovery in Databases). Practical applications are the *Terrorism Information Awareness (TIA) Program* or the *Computer-Assisted Passenger Prescreening System (CAPPS II)* described in [193]. For example, data mining techniques are used for building mathematical models with financial data [142]. Figure 4.19 visualizes the data mining process.
The books *Data mining : practical machine learning tools and techniques* [227] and *Data Mining: Concepts and Techniques* [88] explain the concepts and techniques used by data mining and represents tool examples.

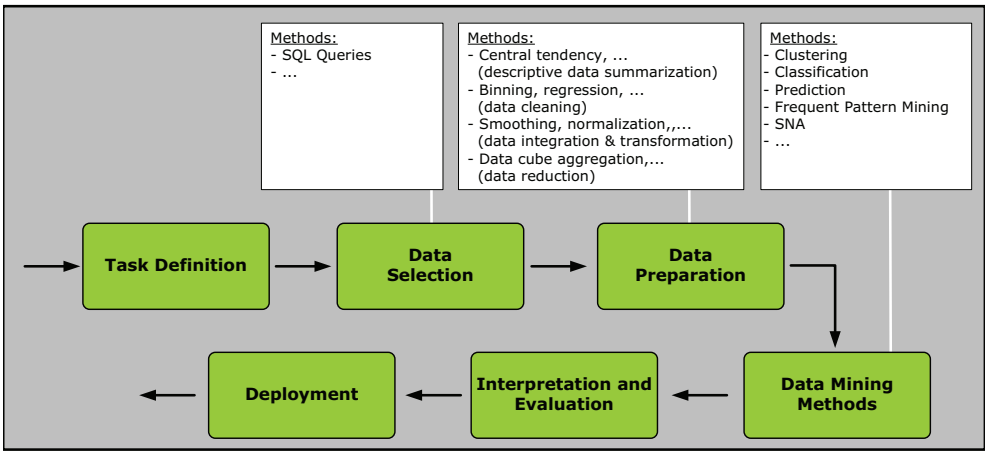


**Figure 4.19.:** Data mining process,
*Der Prozess des Data Mining im Marketing* [102]

### 4.4.6. Artificial Intelligence

*Abbreviation: AI*
*Synonyms: Computational Intelligence, Logic Intelligence*

**Method Examples**
- Logical AI
- Search
- Pattern recognition
- Inference
- Common sense knowledge and reasoning
- Planning
- Learning from experience
- Epistemology
- Ontology
- Heuristics
- Genetic programming

**Application Examples**
- Game playing
- Speech recognition
- Understanding natural language
- Expert systems
- Heuristic classification

Artificial intelligence is the subfield of computer science devoted to developing programs that enable computers to display behavior that can (broadly) be characterized as intelligent [212]. It is the science of creating intelligent machines, especially intelligent computer programs [160].
The book *Artificial Intelligence: A Modern Approach* written from Russell et al. [183] gives an introduction into the theory and practice of artificial intelligence.

**Summary:** This chapter gives an overview about text analysis with a detailed description about text mining. Text mining is a central part in the practical part of this work. Since the definitions of text mining and adjacent research fields are not clear, an introduction into the other research fields is given as well. In the first section an introduction to text analysis with a detailed consideration of text mining is presented. The next section describes the fundamentals of the text analysis methods: linguistic, language statistic and linear algebra. Based on these descriptions, different text analysis methods can be developed. The methods which build the base for this work are described detailed. These are methods for preprocessing and representing text documents, dimension reduction and clustering. In the last section an overview about the research fields with their methods and applications is given.

# Part III.

# Analyses

# 5. Data and Methodology

In this chapter the data sets and methodology for the analyses are represented. The described methods for text mining (Section 4.3, p. 35) have to be integrated into the SONIVIS Tool[1], which provides a generic data model and a model for the implementation of analysis methods (Chapter 3, p. 13). The text mining methods are differentiated into methods for preprocessing, representation and analysis (see Figure 4.6, p. 36). The integration is described in Table 5.1.

| **Text Mining Methods** | **Integration into SONIVIS** |
|---|---|
| Data preprocessing<br>- Tokenization (Sec. 4.3.1, p. 36)<br>- Stop word removal (Sec. 4.3.1, p. 37)<br>- Stemming (Sec. 4.3.1, p. 37) | Term Transformer |
| Data representation<br>- Term weighting (Sec. 4.3.2, p. 42)<br>- Vector space model (Sec. 4.3.2, p. 44) | Content Element Cluster Number Metric |
| Data analysis<br>- Dimension reduction (Sec. 4.3.3, p. 46)<br>- Distance calculation (Sec. 4.3.4, p. 65)<br>- Clustering (Section 4.3.4, p. 63)<br>- Cluster quality measures (Sec. 4.3.4, p. 71) | Content Element Cluster Number Metric |

**Table 5.1.:** Integration of text mining into SONIVIS

The first part of this chapter describes the data sets for the analysis. The data sets include Wiki articles. The extraction and transformation of the Wiki data into the generic data model is introduced in the next section. The transformation includes the preprocessing of the data (see Section 4.3.1, p. 36). The methods for representing and analysing the data are implemented in a metric. Metrics are part of the analysis model in SONIVIS. The representation methods are introduced in Section 4.3.2 (p. 42) and the analysis methods are described in Section 4.3.3 (p. 46), 4.3.4 (p. 63) and 4.3.5 (p. 75). The implementation of the metric is described in the last section of this chapter.
SONIVIS is a network mining software, integrating network visualization, network analysis and further statistics. In this open source software tool a lot of analysis methods are already implemented. SONIVIS is implemented in Java based on the Eclipse Rich Client Platform. The calculation of metrics is carried out by GNU R[2]. In this work, the term-based and link-

[1] http://www.sonivis.org/
[2] http://www.r-project.org/

based clustering approach are integrated into the tool.
For this libraries of GNU R, which are already available, are used. Since the source code of the SONIVIS tool including this implementation is free available[3], it is not added in the appendix of this thesis.

## 5.1. Data Sets

This section describes the data sets which are used for the analyses. The data sets are part of the English Wikipedia[4], which started in 2001. The English Wikipedia was selected because of its size. It is the largest Wiki project (see Table 3.4, p. 18). The English Wikipedia includes many different topics, which can be used in this work. The structure of Wikis is described in Chapter 3 (Section 3.3, p. 18). In this work, the data sets are based on the English Wikipedia from 17 July 2009. An overview about the size and activity of the English Wikipedia is given in Table 5.2.
Since the English Wikipedia is very large, only the needed parts were extracted. A sub area of the Wikipedia was selected by a set of article titles. Based on chosen topics, articles which are related to these topics were randomly selected. After the set of articles was selected, the corresponding data were extracted. Therefore the export page was used [5]. An export of a page includes page information (ID, title) and information about its revisions (ID, timestamp, author, text). For the following analyses only the actual revision is required.
Because the quality of clustering depends on the underlying data structure, four differently structured data sets were extracted. The data sets have different characteristics. The analyses examine, how the clustering diverges with the different characteristics of the data sets. In the context of text mining a data set is called corpus. The articles of each data set topic are listed in Appendix E.

| | |
|---|---|
| **Pages** | 17,417,619 |
| **Articles** | 2,954,545 |
| **Revisions** | 320,177,020 |
| **Average revisions per page** | 18.38 |
| **Registered users** | 10,101,805 |

**Table 5.2.:** Statistics of the English Wikipedia http://en.wikipedia.org/wiki/Special:Statistics [2009/07/17]

An overview about the topics is given in Figure 6.38 (p. 123) and Figure 6.39 (p. 124). The figures visualize the data sets in Wiki-Link networks. The articles are grouped by their topics and the linkage between the topics is visualized.

[3] http://www.sonivis.org:8080/cb/user
[4] http://en.wikipedia.org/
[5] http://en.wikipedia.org/wiki/Special:Export/

### 5.1.1. Data Sets I - Single Topics

The first data set includes articles of two different topics: *Astronomy* and *Oceanography*. The topics are completely different and have the same amount of articles. Based on this data set, the clustering should provide good results and the results of the different clustering methods should be equal. This data set allows a first evaluation of the clustering methods. Table 5.3 gives an overview about this data set.

| | **#Articles** | **∅ Revisions*** | **∅ Users*** | **∅ Links*** | **∅ Terms*** |
|---|---|---|---|---|---|
| Astronomy | 50 | 2055.96 | 491.92 | 285.86 | 810.42 |
| Oceanography | 50 | 1097.80 | 281.14 | 164.00 | 501.86 |

**Table 5.3.:** Properties of data set I *(*per Article)*

### 5.1.2. Data Sets II - Overlapping Topics

The second data set includes articles of two different topics as well: *Artificial Intelligence* and *Computer Programming*. Compared to the first data set, the two topics are more similar. Like the first data set, the topics have the same size concerning the number of articles. The clustering of this data set is more difficult, because of the similarity of the topics. An overview about the topics is given in Table 5.4.

| | **#Articles** | **∅ Revisions*** | **∅ Users*** | **∅ Links*** | **∅ Terms*** |
|---|---|---|---|---|---|
| Artificial Intelligence | 25 | 487.32 | 130.04 | 109.08 | 497.00 |
| Computer Programming | 25 | 348.08 | 100.16 | 77.08 | 277.24 |

**Table 5.4.:** Properties of data set II *(*per Article)*

### 5.1.3. Data Sets III - Topics of a Main Topic

The third data set includes four different topics in the field of medicine: *Genetics*, *Immunology*, *Medical* and *Toxicology*. The topics are similar concerning their main topic. Compared to the other data sets, there are more topics with a different size. The topics are described in Table 5.5.

| | **#Articles** | **∅ Revisions*** | **∅ Users*** | **∅ Links*** | **∅ Terms*** |
|---|---|---|---|---|---|
| Genetics | 25 | 767.76 | 201.84 | 104.20 | 380.24 |
| Immunology | 17 | 690.12 | 178.59 | 159.47 | 513.12 |
| Medical | 10 | 656.20 | 155.70 | 143.00 | 475.20 |
| Toxicology | 14 | 458.00 | 125.14 | 109.21 | 451.29 |

**Table 5.5.:** Properties of data set III *(*per Article)*

### 5.1.4. Data Sets IV - Small Topics

The last data set includes six topics: *Astronomy*, *Genetics*, *Oceanography*, *Computer Programming*, *Tourism* and *Toxicology*. The topics are small and have the same size concerning the number of articles. With this data set, the clustering algorithm will be tested based on small topics. Since the topics are small, only little information about each topic is available. An overview about the topics is given in Table 5.6.

| | **#Articles** | **∅ Revisions*** | **∅ Users*** | **∅ Links*** | **∅ Terms*** |
|---|---|---|---|---|---|
| Astronomy | 5 | 3429.80 | 838.00 | 428.20 | 1107.60 |
| Genetics | 5 | 2329.60 | 571.40 | 284.80 | 770.40 |
| Oceanography | 5 | 1680.20 | 419.20 | 216.00 | 634.40 |
| Computer Programming | 5 | 676.60 | 199.80 | 113.60 | 409.40 |
| Tourism | 5 | 1150.80 | 322.20 | 117.60 | 356.60 |
| Toxicology | 5 | 390.20 | 103.60 | 96.00 | 246.60 |

**Table 5.6.:** Properties of data set IV *(*per Article)*

## 5.2. Extraction and Transformation of Wiki Data

The selected articles have to be extracted and saved in the new data model. Furthermore transformations have to be executed to gain additional information. The extraction and transformation of Wiki data into the generic data model is described in Section 3.3.2 (p. 20). In this thesis, two approaches of clustering are compared: content-based and link-based clustering. The new content elements and relations are listed in Table 5.7.

| **Data Model** | **Wiki Data** |
|---|---|
| Content elements | Terms |
| Contextual relations | ”Term was added/deleted in the revision text” |
| Properties | ”Difference value of a term between two revisions” |

**Table 5.7.:** Term Transformer

The data for link-based clustering are saved in the contextual relations of type *WikiLink* and *RevisionPartOfCategory*. Thereby only the last revision of an article is used in this work. The terms of link-based clustering are the titles of the target content elements of these relations. Content-based clustering needs the terms of each article text. The terms have to be detected in a further transformation step, called term transformer. For the detection of terms, the content elements of type *Page* and *Revision* are used. After detecting the terms, they are saved as content elements of type *Term*. Only the term difference between revisions is saved. Therefore contextual relations of type *KeywordPartOfRevision* between the content elements of type *Term* and *Revision* are created. The difference value of a term is saved as a property of the relation. The property name is *Diff*.

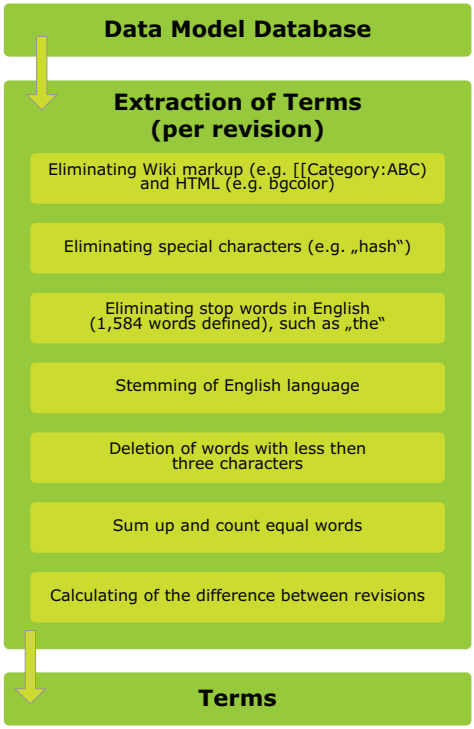


**Figure 5.1.:** Preprocessing the data sets (term extraction per revision)

In the context of text mining the term transformer includes the preprocessing of the data (see Figure 4.6, p. 36 and Section 4.3.1, p. 36). In addition to the detection of the terms, the terms have to be manipulated as well. The term detection per revision is described in Figure 5.1. It is implemented in the statistical tool GNU R. The R packages *tm* and *RStem* are used. The preprocessing methods tokenization, stop word removal and stemming are used. They are described in Section 4.3.1 (p. 36).

The first step is the elimination of unimportant characters and character sequences. A lot of Wiki texts include Wiki markups and HTML-Syntax. But these parts of the text are of no interest and reduce the quality of the analyses outcome. For this reason they are deleted. After that the elimination of special characters is accomplished.

The result is a character sequence including only alphabetic characters. The words are now separated by whitespaces. Based on this character sequence, the tokenization is adopted (see Section 4.3.1, p. 36). The tokenization includes the segmentation of character sequences into words (tokens).

Now the words of the text are detected. But there are many words which are not interesting for later analyses. For example, the word *the* gives no more information of a specific text, whereas the words *stars*, *planets* and *meteors* give a reference that the text might be about *astronomy* or *science fiction*. The unimportant words, called stop words, are deleted in the next step by using the method `removeWords` of the R package *tm*. The method is called stop word removal (see Section 4.3.1, p. 37). A defined list of stop words (1,584 English and 34

general stop words) determines the words which should be removed.
The word sequence includes now the important words of the revision text. But there are also equal words of a different form, like *stars* and *star*. These words should be covered as the same. The reduction of words to their stem by using the R package *RStem* solves this problem (the method is described in Section 4.3.1, p. 37). In this case the word *stars* is reduced to *star*. The stemming is based on the *Porter Stemming Algorithm* which is realized by *Snowball* [6].
The deletion of stemmed words with less than three characters removes words which are not representative for the text. For example the *s* from *he's* contains no relevant information and worsen the following analysis. Additionally the term *s* have a high frequency, because they occur in many other words as well.
The last step is summing up and counting equal words. Now the terms and their frequency in a revision are calculated. In order to reduce the memory load in the data base, only the differences between revisions are stored. The difference between two revisions is the difference between the frequencies of all terms in the revision texts. The preprocessing of the data is completed now. With the detected terms the content can be analyzed.

## 5.3. Implementation of a Text Mining Metric

The clustering of the content elements is implemented in a metric, called *Content Element Cluster Number Metric*. The metric is a content metric (see Section 3.2.3, p. 17), which returns the cluster ID for each content element. Furthermore the quality of the clusters is calculated and represented. Text mining methods for representing and analyzing data are used from the metric. The methods term weighting (Section 4.3.2, p. 42) and vector space model (Section 4.3.2, p. 44) are used for representing the terms and links of content elements. The methods dimension reduction (Section 4.3.3, p. 46), distance calculation (Section 4.3.4, p. 65), clustering (Section 4.3.4, p. 63) and cluster quality measurement (Section 4.3.4, p. 71) are used for the topic detection.
There are many algorithms available for the analysis methods. The algorithms for term weighting, dimension reduction, distance calculation and clustering have to be selected. The term weight is the *Term Significance Weight* by default, because it is very popular in the text analysis area (described in Section 4.3.2, p. 42). The distance measure is the *Cosine Coefficient* by default, because the Cosine achieve good results in the area of text analysis (described in Section 4.3.4, p. 65).
The selection of adequate algorithms for dimension reduction and clustering is more difficult, because a lot of algorithms exist. Table 5.8 describes the choice of adequate algorithms. The selection is supported by two conditions. The algorithms have to be used in further text analysis works. This is an indication that the algorithms are suitable for this thesis. The other condition is, that an implementation in R exists or an implementation of the method in R is reasonable. This condition allows an appropriate implementation in the SONIVIS tool.
In the field of dimension reduction, two algorithms are used. For feature selection a method, called *Term Weight Thresholding* (described in Section 4.3.3, p. 60), is used. Feature selection discover the important terms of the matrix. A threshold is defined and all values which are smaller than the threshold will be deleted. *Latent Semantic Analysis* is used for feature transformation (described in Section 4.3.3, p. 48).

[6] Snowball: A small string processing language designed for creating stemming algorithms, http://snowball.tartarus.org/texts/introduction.html

| | Text analyses evaluation found | R implementation exists/easy | Selected algorithms | Description |
|---|---|---|---|---|
| ***Dimension Reduction*** | | | | |
| Feature Selection (Supervised) | X | X | | |
| Feature Selection (Unsupervised) | X | X | TW Thresholding | Low complexity |
| Feature Transformation (Supervised) | | X | | |
| Feature Transformation (Unsupervised) | X | X | LSA | Implemented for text analysis (Information Retrieval) |
| ***Clustering*** | | | | |
| Density-based | | X | | |
| Hierarchical | X | X | Diana<br>Agnes | Divisive hierarchical clustering<br>Agglomerative hierarchical clustering |
| Fuzzy<br>Search-based | | X | | |
| Center-based | X | X | PAM<br>bisecting PAM | More robust version of k-means<br>Achieved good results in [204] |
| Graph-based<br>Grid-based | X | | | |
| Model-based<br>Subspace | X | X | | |

**Table 5.8.:** Selection of analysis methods

For clustering four clustering algorithms are selected. *PAM* is a center-based clustering algorithm, which is a more robust variant of k-means (described in Section 4.3.4, p. 67). *Agnes* belongs to the agglomerative hierarchical clustering algorithms (method: group average) and *Diana* belongs to the divisive hierarchical clustering algorithms. Hierarchical clustering is described in Section 4.3.4 (p. 68). These three algorithms are implemented in the R package `cluster`. The fourth clustering algorithm *bisecting PAM* is a composition of center-based and divisive hierarchical clustering (described in Section 4.3.4, p. 67). It is implemented in an additional R method. The number of iterations that the PAM algorithm will execute for one clustering step is 100.
Internal and external quality measures are used for evaluating the clustering. The selected measures are listed in Table 5.9. The measures are explained in Section 4.3.4 (p. 71). An external cluster quality measure is given by the *F-measure*. The F-measure compares the result of the clustering with the optimal solution. The optimal clustering solution is given by the topics of the defined data sets. The optimal clusters are the topics of each data set. Since the clustering depends on the underlying data set and the selected clustering approach, the internal cluster quality is measured by the *Dunn Index* and *SD Validity Index* additionally. The R package `clv` includes methods for the calculation of the Dunn Index and SD Validity Index. The performance of the clustering is evaluated by the *Computing Time*.

| | **Measure Type** | **Settings** |
|---|---|---|
| Computing Time | Performance | |
| Dunn Index | Internal quality | Cluster distance: Group Average<br>Cluster diameter: Group Average |
| SD Validity Index | Internal quality | Cluster scatter: Manhattan distance |
| F-measure | External quality | |

**Table 5.9.:** Clustering evaluation measures

The topic detection includes seven steps, which are visualized in Figure 5.2. The dimension reduction steps are optional and not necessary for clustering. First a term document matrix has to be created (see Section 4.3.2, p. 44). This depends on the selected clustering approach. The content-based clustering uses the terms of the content elements. Link-based clustering takes contextual relations of type Wiki-link and category-link as terms. After this step, both approaches follow the same routine.
Now the terms have to be weighted. Therefore the *Term Significance Weight* is used. The weight is described in Section 4.3.2 (p. 42).
The next two steps include the dimensional reduction of the matrix. This reduces the complexity and increases the performance of clustering. Dimension reduction methods are described in Section 4.3.3 (p. 46). Feature selection discovers the important terms of the matrix. Here a method, called *Term Weight Thresholding* (TW Thresholding) is used. Feature transformation is executed with the *Latent Semantic Analysis* (LSA) technique. The dimension reduction techniques are described in Section 4.3.3 (TW Thresholding, p. 60) and Section 4.3.3 (LSA, p. 48).
Now the distance matrix can be created. Based on the *Cosine Coefficient* the similarity be-

tween each document will be calculated first (described in Section 4.3.4, p. 65). Then the similarity values will be transformed into distance measures using the formula $dist(\vec{v}, \vec{w}) = m - sim(\vec{v}, \vec{w})$, where $m$ defines the greatest similarity value. Finally the values are saved in an object of class `dist`. The class can be found in the R package `stats`.
Based on the distance matrix, the clustering can be executed. Four different clustering algorithms are used: *PAM*, *bisecting PAM*, *Agnes* and *Diana*. The algorithms are described in Section 4.3.4 (hierarchical clustering, p. 67) and Section 4.3.4 (center-based clustering, p. 68).
The results are evaluated by comparing the performance and validity of the clustering (see Table 5.9). The internal quality measures are the *Dunn Index* and the *SD Validity Index*. The external quality measure is the *F-measure*. The measures are explained in Section 4.3.4 (p. 71). The computing time is used for evaluating the performance.

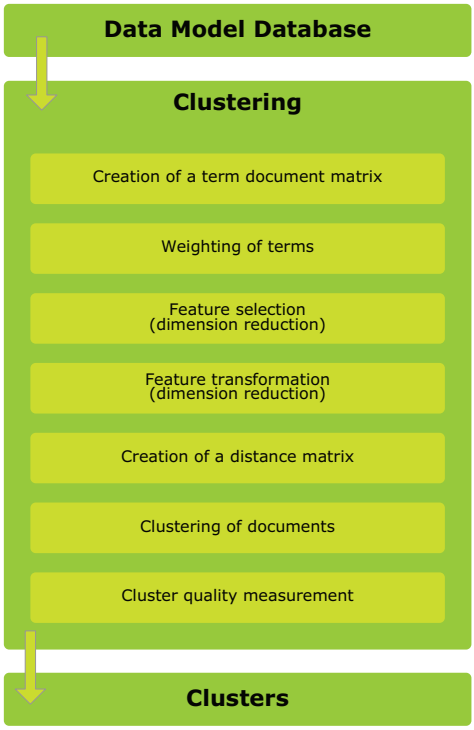


**Figure 5.2.:** Topic detection process for analyses

**Summary:** The content of this chapter describes the implementation of text mining methods for topic detection into the SONIVIS tool. A transformer for the detection of the terms is needed. The analysis methods are implemented in a metric. Furthermore data sets, which are used in the analyses, are described. With the selected data sets, the *Term Transformer* and the *Content Element Cluster Number Metric* the measures for the analyses can be executed as shown in Chapter 6.
The analyses are separated into four steps. The steps are listed in Table 5.10. First each clustering algorithm without dimension reduction is used for the content-based and link-based clustering approach. The next two steps define the optimal values for dimension reduction.

In the last step clustering with and without dimension reduction is compared. Additionally the results are compared with the optimal clustering solution.

| | **Feat. Sel. (TW Threshold)** | **Feat. Transf. (Dimensions)** | **Clustering** | **Evaluation** |
|---|---|---|---|---|
| ***Phase 1*** | | | | |
| CB: | - | - | PAM, bisPAM, Agnes, Diana | F-measure, Dunn Index, SD Validity Index, Computing Time |
| LB: | - | - | PAM, bisPAM, Agnes, Diana | |
| ***Phase 2*** | | | | |
| CB: | 10%-90% | - | PAM, Diana | F-measure |
| LB: | | - | PAM, Agnes | |
| ***Phase 3*** | | | | |
| CB: | 50% | 1-100 | PAM | F-measure |
| | - | | Diana | |
| LB: | 90% | | PAM | |
| | 90% | | Agnes | |
| ***Phase 4*** | | | | |
| CB: | 50% | DS I: 40<br>DS II: 40<br>DS III: 6<br>DS IV: - | PAM | F-measure, Dunn Index, SD Validity Index, Computing Time, Wiki-link network |
| | - | DSI: -<br>DS II: -<br>DS III: 9<br>DS IV: 5 | Diana | |
| LB: | 90% | DS I: 8<br>DS II: 15<br>DS III: 15<br>DS IV: - | PAM | |
| | 90% | - | Agnes | |
| OS: | - | - | - | |

**Table 5.10.:** Properties of clustering phases *(DS=Data Set, CB=Content-based clustering, LB=Link-based clustering, OS=Optimal Solution)*

# 6. Results and Interpretations

In this chapter, the results of the analyses are represented and interpreted. The content-based and link-based clustering are compared and evaluated. Furthermore the clustering algorithms and dimension reduction methods which are suitable for the approaches are detected. Additionally the quality of clustering per data sets are evaluated.
The objective of this study is the evaluation of the clustering approaches in combination with different clustering algorithm and dimension reduction methods. Therefore four different experiments with different methods are executed. The four phases are described in Table 5.10 (p. 96). Each phase uses the results of the previous phases for their analysis. In the first phase the content-based and link-based clustering approach is executed with each clustering algorithm (PAM, bisecting PAM, Agnes, Diana, described in Section 4.3.4, p. 63). The goal is the selection of appropriate clustering algorithms for both approaches. Within this phase, no dimension reduction is used. The next two phases define optimal values for dimension reduction (TW Thresholding, described in Section 4.3.3, p. 60; LSA, described in Section 4.3.3, p. 48). In the last phase, the clustering approaches are compared. Furthermore the results are compared with the optimal clustering solution. In the optimal clustering solution, the clusters are created manually by the given topics (see Section 5.1, p. 88).
In the phases, the results are evaluated with the F-measure, Dunn Index, SD Validity Index and computing time (described in Section 4.3.4, p. 71). The most important quality measure is the F-measure (external quality analysis), because this measure compares the optimal solution with the clustering solution. The computing time (performance analysis) is used to compare the performance of the different clustering configurations and clustering approaches. The Dunn Index and SD Validity Index (internal quality analysis) measure the quality of a clustering result based on density measures. These indices describe the distance between clusters and the distance between documents in a cluster. They can be used to determine the compactness of each data set. For example a high Dunn Index or a small SD Validity Index indicate that the clusters of a data set are far apart and the articles in one cluster are close together. Furthermore they can be used to compare clustering algorithms and the approaches content-based and link-based clustering. It has to be pointed out, that the results of the indices depend on the underlying term document matrix (described in Section 4.3.2, p. 44). The matrix depends on the selected clustering approach. Thus the consideration of these indices should be combined with the F-measure.
In most cases the results (F-measure, Dunn Index, SD Validity Index and computing time) are visualized in two views. The first view is the clustering algorithm view (left diagram). The view allows a comparison of the clustering algorithms. The second view is the data set view (right diagram). The view allows a comparison of the data sets. In each diagram the content-based and link-based clustering results are visualized. The content-based clustering is represented by a dark color and the link-based clustering is represented by a light color. The F-measure is visualized with blue, the Dunn Index with green, the SD Validity Index with red and the computing time with yellow. Other diagrams give a detailed insight in the results. In these diagrams, the data set results of each clustering algorithm are not summed up. The four data set values from each clustering algorithm are visualized. Thereby the data

set I has the color blue, the data set II has the color green, the data set III has the color red and the data set IV has the color yellow.

## 6.1. Clustering Phase 1 - Clustering Algorithm

In the first analysis phase, content-based and link-based clustering are executed without dimension reduction. For each approach, four different clustering algorithms are used (described in Section 4.3.4, p. 63): center-based clustering (PAM), agglomerative hierarchical clustering (Agnes), divisive hierarchical clustering (Diana) and a combination of center-based and divisive hierarchical clustering (bisecting PAM [bisPAM]). The goal is to find two clustering algorithms for each clustering approach, which are suitable for the content-based and link-based clustering approach. Clustering on four data sets with two clustering approaches and four clustering algorithms are executed and the values for the measures (F-measure, Dunn Index, SD Validity Index and Computing Time, described in Section 4.3.4, p. 71) are calculated.
The results are compared by an external quality analysis using the F-measure (see Section 6.1.1, p. 98), an internal quality analysis using the Dunn Index and SD Validity Index (see Section 6.1.2, p. 100) and by an analysis of the performance using the computing time (see Section 6.1.3, p. 102). The results are summed up in Section 6.1.4 (p. 103).
The values of the charts are listed in Appendix F.1. The figures allow a first comparison between the content-based and link-based clustering. In addition, a comparison between the data sets and clustering algorithms is needed for a better understanding of these results.

### 6.1.1. External Quality Analysis

Because the F-measure (described in Section 4.3.4, p. 71) is the most important quality measure, the F-measure for each data set and clustering algorithm is displayed in Figure 6.1 (content-based clustering) and Figure 6.2 (link-based clustering). The x-axis visualizes the clustering algorithms. The clustering algorithms are executed on each data set. The F-measure values for the clustering results of each data set are displayed on the y-axis.

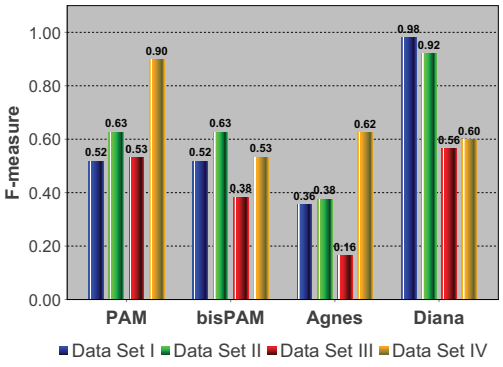


**Figure 6.1.:** F-measure: Content-based clustering

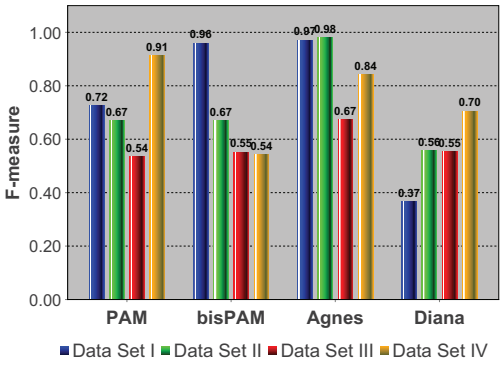


**Figure 6.2.:** F-measure: Link-based clustering

Using content-based clustering (Figure 6.1), the data sets I-III (including single topics, overlapping topics, topics of a main topic) achieve the best results with the clustering algorithm Diana. Data set IV (including small topics) has the best results with PAM. The poorest results of the data sets I-III are achieved with Agnes. Data set IV obtains the poorest results with bisPAM. The results of link-based clustering are different (Figure 6.2). The best results of the data sets I-III (including single topics, overlapping topics, topics of a main topic) are achieved with the clustering algorithm Agnes. Like the content-based approach, PAM clusters the fourth data (including small topics) set best. The poorest results are achieved with the clustering algorithms bisPAM and Diana.
The differences within the results show, that the quality of clustering depends on the selected clustering algorithm and on the structure of the data (the selected data set and clustering approach). The selected clustering algorithm determines the quality of the clustering, because the results of the clustering algorithms are different. Comparing the data sets, the results of data set IV (including small topics) are different to the other data sets. Furthermore the data structure of both clustering approaches have to be different, because the results of content-based and link-based clustering with Agnes and Diana are completely contrary. The content-based clustering uses all terms of a text, whereas the link-based clustering uses only the links. Because all in all link-based clustering achieved better results, the data structure of this approach has to be more suitable for topic detection.

For the next phases two clustering algorithms for each clustering approach have to be selected. Since the first two figures are not suitable for the definition of appropriate algorithms, the average of the results have to be calculated and compared. Furthermore the results of each data set are compared. Thus one can find out whether there are data sets which can be better clustered than others. The average values give an overview which is more useful by the selection of appropriate algorithms. For example the definition of minimum or maximum values are not suitable, because the results depend on the data set and thus the values strongly vary. It is not useful to select an algorithm that clusters one data set very well, but the other data sets especially bad. For this reason the dispersion exerts no influence on the selection of appropriate algorithms as well.

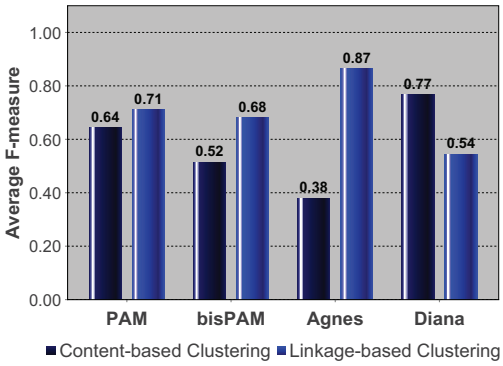


**Figure 6.3.:** F-measure:
Clustering algorithm view

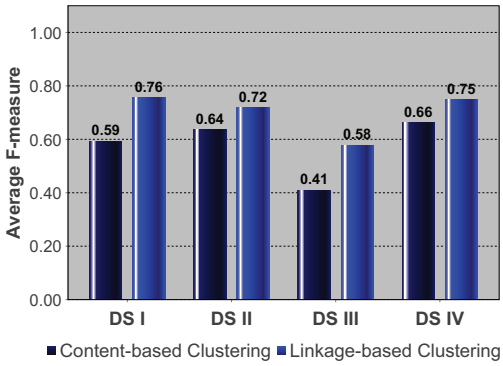


**Figure 6.4.:** F-measure:
Data set view *(DS = Data Set)*

The average results of the F-measure are visualized in 6.3 (clustering algorithm view) and 6.4 (data set view). The left diagram compares the results of the clustering algorithms. On the

x-axis the clustering algorithms can be found. The clustering algorithms calculate the clusters of four data sets. Thus per clustering algorithm four F-measure values exist. The average of the values per clustering algorithm is calculated and visualized on the y-axis. With this diagram it is possible to select the best clustering algorithms based on the F-measure. The diagram on the right side visualizes the average results of the data sets. On the x-axis the data sets can be found. Each data set is clustered with four clustering algorithm. The average of the clustering algorithm values per data set is calculated and visualized on the y-axis. With this diagram the quality of clustering per data set can be compared.
Figure 6.3 and 6.4 clarify that link-based clustering achieves better results than content-based clustering (Figure 6.4). With each data set the individual topics are identified better by link-based clustering. With exception of Diana, each clustering algorithm has better results in combination with link-based clustering (Figure 6.3).
Using link-based clustering, PAM and Agnes seem to be the best clustering algorithms (Figure 6.3). The clustering algorithms PAM and Diana are the best for content-based clustering. Although bisecting PAM is an optimized variant of PAM, the algorithm does not generate good results.
The F-measure of the data sets (Figure 6.4) depends on the similarity of the topics. Data set I (including single topics) has the most different topics and the best F-measure value, closely followed from data set IV (including small topics) with different and similar topics. As expected the third data set (including topics of a main topic) has the poorest clustering results (see Figure 6.4). In this data set the topics are very similar.
Like the measurements before, the figures visualize that the results depend on the data set, the clustering algorithm and the clustering approach. The cluster quality of a data set depends on the similarity of the topics within this data set. The link-based clustering has better results than the content-based clustering. The terms which are used for creating the term document matrix are more suitable to identify the topics. Thus the link-based clustering approach identifies more the important terms of a text, whereas the content-based clustering approach includes more terms which are unimportant and confuse the clustering. Based on the F-measure, center-based clustering (PAM) and agglomerative hierarchical clustering (Agnes) are the best clustering algorithms for link-based clustering and center-based clustering (PAM) and divisive hierarchical clustering (Diana) are the best clustering algorithms for content-based clustering.

### 6.1.2. Internal Quality Analysis

In this section the internal quality of the clustering is analyzed by using the Dunn Index and SD Validity Index (described in Section 4.3.4, p. 71). Like in Section before, the average results are visualized. Figures 6.5 (Dunn Index) and 6.7 (SD Validity Index) represent the average results of the clustering algorithms (clustering algorithm view). In these diagrams the x-axis represents the clustering algorithms and the y-axis sums up the data set results per clustering algorithm. The average of the four data set values per clustering algorithm is calculated. Based on these diagrams the clustering algorithms can be compared concerning the internal quality of the clusters. Figures 6.6 (Dunn Index) and 6.8 (SD Validity Index) represent the average results of the data sets (data set view). In these diagrams the y-axis represents the data sets and the y-axis visualizes the average clustering results of each data sets. Based on these diagrams the internal quality of the clusters per data set can be evaluated. Compared to the F-measure, the Dunn Index and SD Validity Index show different results.

The Dunn Index (Figure 6.5) indicates that the clustering results of content-based and link-based clustering using PAM or Diana are very similar. In the case of bisecting PAM, the content-based clustering forms better clusters than the link-based clustering. Only the results of the clustering algorithm Agnes are similar to the F-measure. The same difference is shown in Figure 6.6. Only with data set II and data set III link-based clustering is better than content-based clustering. The SD Validity Index shows the same results as well (Figure 6.7 and 6.8). Here only bisecting PAM provides better results using link-based clustering. Each data set has a better internal quality using the content-based clustering.

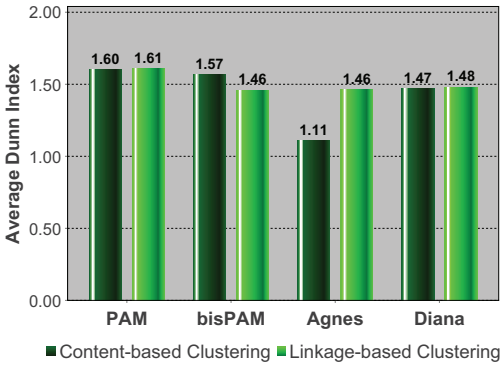


**Figure 6.5.:** Dunn Index: Clustering algorithm view

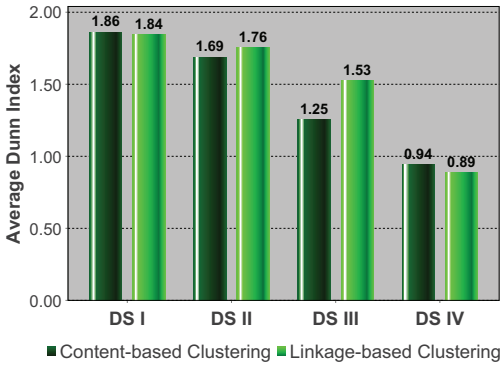


**Figure 6.6.:** Dunn Index: Data set view *(DS = Data Set)*

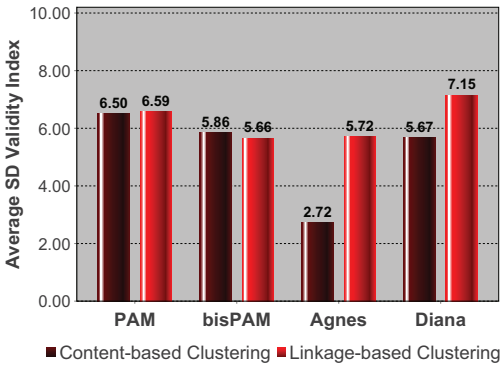


**Figure 6.7.:** SD Validity Index: Clustering algorithm view

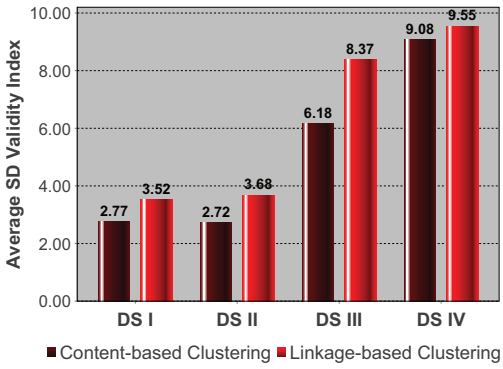


**Figure 6.8.:** SD Validity Index: Data set view *(DS = Data Set)*

Because the results of each clustering algorithm are very similar, the selection of appropriate algorithms is not clear. Based on the Dunn Index (Figure 6.5) the algorithms PAM and bisPAM have the best internal quality with content-based clustering. Using link-based clustering, the algorithm PAM and Diana have the best results. The results of the SD Validity Index (Figure 6.7) are different. The algorithm Agnes and Diana creates the best clusters with the content-based clustering approach. The best results with the link-based clustering approach are achieved with the algorithms bisPAM and Agnes. It is surprising that Agnes has the poorest clustering results based on the Dunn Index, but the SD Validity Index indicates that Agnes is the best clustering algorithm.
The internal quality of the data sets is equal by both approaches. The data sets I+II (includ-

ing single topics and overlapping topics) have the best internal quality (Dunn, Figure 6.6; SD Validity Index, Figure 6.8). Data set IV obtains the poorest (including small topics) internal quality. The internal quality values do not depend on the topic similarity of a data set, as is the case with the F-measure.
The analysis of the internal quality shows that the results of link-based clustering are not better than the results of content-based clustering. The Dunn Index and SD Validity Index are internal quality measures. The results depend on the structure of the data sets and on the clustering results. The results show that the data structure of content-based clustering allows more precise results. The clusters are not close together and the documents within a cluster are closer together. But the F-measure shows that this is no indication for a good clustering. Based on the data structure of content-based clustering wrong clusters are built. One reason for this could be that the content-based clustering has more terms. Accordingly, the term document matrix has more dimensions and the clusters of a data set are far away from each other. The identification of suitable clustering algorithms is not easy, because the results of the Dunn Index and SD Validity Index are different and the results within one diagram are similar. The contrary evaluation of the clustering algorithm Agnes shows that the measures are not suitable to define appropriate clustering algorithms. In the diagrams of the data set view no dependencies between the internal quality values and the structure of the data sets (similarity of the topics, data set size, etc.) can be found.

### 6.1.3. Performance Analysis

This section analyzes the performance of the clustering. The computing time is measured and the average results are calculated. The results are visualized in Figure 6.9 (clustering algorithm view) and Figure 6.10 (data set view). Like the sections before, the x-axes represent the clustering algorithms (Figure 6.9) and data sets (Figure 6.10). The average computing time per clustering algorithm is calculated and visualized on the y-axis in Figure 6.9. The y-axis in Figure 6.10 visualizes the average computing time of each data sets. The figures allow the comparison of the clustering algorithms and data sets based on the computing time.

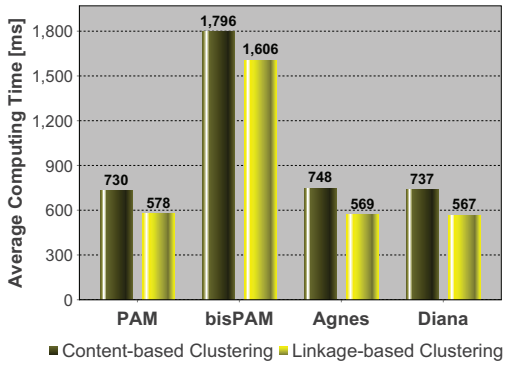


**Figure 6.9.:** Computing Time: Clustering algorithm view

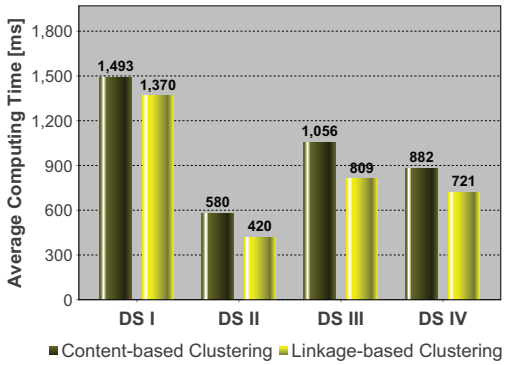


**Figure 6.10.:** Computing Time: Data set view *(DS = Data Set)*

Considering the computing time (Figure 6.9 and 6.10), link-based clustering is better than content-based clustering. Each clustering algorithm and data set has a lower computing time

using the link-based clustering approach.
The clustering algorithm bisPAM has the poorest performance. Between the performance of the other algorithms no crucial differences can be recognized. The computing time of PAM, Agnes and Diana is less than one second.
There are differences between the computing time of the data sets. The computing time depends on the size of the data set. Data set I (including single topics) has 100 articles and the poorest computing time, followed from data set III (including topics of a main topic) with 66 articles. The best computing time has data set IV (including small topics). It is the smallest data set including 30 articles.
Using the content-based clustering approach more terms are used to describe an article. Consequently the term document matrix is very large. This fact increases the computing time. Since the data sets are small, the computing time of the hierarchical clustering algorithm are not higher than PAM. Bisecting PAM has the poorest performance, because at each clustering step, PAM will be executed 100 times. The computing time of the data sets corresponds to their size (number of articles). The more articles a data set has, the higher the computing time is.

### 6.1.4. Summary

The analysis of the first clustering phase gives a first overview about the behavior of the clustering approaches, the clustering algorithms and the data sets. The credibility of the internal quality analysis is questioned. The results contradict those of the external quality analysis. Furthermore the results of the internal quality measures (Dunn Index and SD Validity Index) are not compatible (e.g. Agnes, Figure 6.5 and 6.7). The results depend on the structure of the data sets. Therefore the content-based clustering creates better results concerning the internal structure of the data sets and clusters. But the internal structure causes a wrong clustering concerning the topics of the data sets.
Link-based clustering creates better clusters than content-based clustering concerning the external quality and performance analysis. The clusters reflect more precisely the defined topics. Because the external quality is better, the term document matrix of link-based clustering includes less unimportant terms and consequently the computing time is shorter. Because the external quality of the content-based clustering is poorer, the preprocessing of the data doesn't seem to remove enough unimportant terms.
The cluster quality of the data sets depends on the similarity of the topics (external quality analysis) and on the size of the data set (performance analysis). The size of a data set is defined by the number of articles.
The goal of this clustering phase is the identification of suitable clustering algorithms. Because the correctness of the internal quality analysis is not proven, only the external quality and performance analysis are used. Table 6.1 describes the choice of adequate algorithms. The computing time in seconds is combined with the F-measure value. In such a way that the combined value increases, the higher the F-measure value and the smaller the computing time. The two clustering algorithms with the highest values are chosen. PAM and Diana are the best clustering algorithms for content-based clustering and PAM and Agnes are the best clustering algorithms for link-based clustering.
The question is now, whether the clustering results can be enhanced by using dimension reduction methods and whether it is possible that content-based clustering can have a better cluster quality than link-based clustering by using dimension reduction methods. In order to

be able to answers these questions, the selection of suitable dimension reduction parameters is part of the next two clustering phases. The dimension reduction methods remove further unimportant terms of the content-based clustering and if unimportant terms exist, from the link-based clustering as well.

| | **F-measure** $FM$ | **Computing time [s]** $CT$ | **Clustering quality** $(\frac{1}{1+CT^{0.25}})^{0.5} \cdot FM^{0.5}$ |
|---|---|---|---|
| ***Content-based Clustering*** | | | |
| PAM | 0.643 | 0.730 | 0.578 |
| bisPAM | 0.515 | 1.796 | 0.489 |
| Agnes | 0.380 | 0.748 | 0.444 |
| Diana | 0.766 | 0.737 | 0.631 |
| ***Link-based Clustering*** | | | |
| PAM | 0.711 | 0.578 | 0.616 |
| bisPAM | 0.681 | 1.606 | 0.566 |
| Agnes | 0.866 | 0.569 | 0.681 |
| Diana | 0.545 | 0.567 | 0.540 |

**Table 6.1.:** Selection of clustering algorithms

## 6.2. Clustering Phase 2 - Feature Selection

Although the results of the first measures show that link-based clustering is better than content-based clustering, optimal clustering solutions are not achieved. The F-measures show that only data set I is nearly optimally clustered (Figure 6.2, p. 98). Other data sets mostly have a F-measure around 0.6. Thus the next two clustering phases analyze the influence of dimension reduction. The question is whether dimension reduction enhances the clustering results of both approaches. Possibly, that dimension reduction enhances the content-based clustering approach more than the link-based approach. This possibility will be explored.
In this clustering phase the influence of the dimension reduction method Term Weight Thresholding (described in Section 4.3.3, p. 60) is explored. It is a feature selection method. Based on a defined term weight threshold only the term weights which are equal or higher than this threshold are used in further analyses. Because there is no pre-defined threshold and the definition of an appropriate threshold is not easy, the search of a suitable threshold is part of this section. Furthermore it is examined, whether feature selection improves the results. For each clustering approach, the two best clustering algorithms are combined with feature selection. The content-based approach uses the clustering algorithm PAM and Diana and the link-based approach uses PAM and Agnes (see Table 6.1, p. 104). Consequently, clusters for four data sets with two clustering approaches each with two clustering algorithms are created.
In order to determine an appropriate threshold, a stepwise clustering within the term weight range is executed. For each clustering algorithm and clustering approach, the threshold with the best average value is used for further analyses. Since each data set and clustering approach has its own term weight range, the use of absolute ranges and absolute step values is extensive. Thus the thresholds are defined in percent and clustering with 10%-100% (steps of 10%) of the features (term weights) is executed. Clustering with 100% of the features is equal to clustering without Term Weight Thresholding. The results of 100% are compared with the other results to find out whether and how feature selection enhances the cluster quality. The term weight threshold in percent defines the position of a term weight. The value of this term weight defines the minimum term weight value which should not be deleted. For example if there are 30 term weights in the term document matrix and 50% of the term weights should be used in further analyses. The 30 term weights are ordered and the term weight value at position 15 defines the minimum term weight value.
The F-measure is used to evaluate the results. The influence of feature selection on the Dunn Index, SD Validity Index and computing time is explored in the last clustering phase (see Section 6.4, p. 115). Section 6.2.1 (p. 105) includes the external quality analysis based on the F-measure. In Section 6.2.2 (p. 108) the results are summed up. Appendix F.2 includes the corresponding values of the diagrams. Based on the diagrams, the feature selection threshold can be defined and it can be shown whether and how Term Weight Thresholding enhances the cluster quality.

### 6.2.1. External Quality Analysis

Figure 6.11 and 6.12 display the results of content-based clustering and Figure 6.13 and 6.14 display the results of link-based clustering. The x-axis represents the number of features (term weights) which are used for clustering in percent. The other features are removed. On the y-axis the F-measure value of each data set can be found. Additionally the average value of the four data sets is visualized. Table 6.2 (p. 107) gives an overview of the best

feature selection thresholds. In the table the number of terms, the absolute term weight threshold and the deleted terms after feature selection are listed. The number of terms and the number of deleted terms are not equal to the number of term weights. One term can occur in many documents and therefore one term can have many term weights. A term will be only deleted, if the term weight thresholding deletes all corresponding term weights. The number of deleted term weights is defined by the term weight threshold in percent.

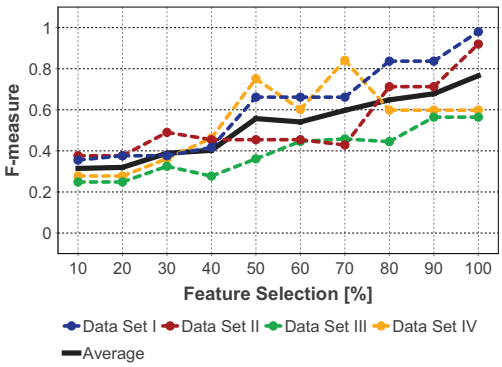


**Figure 6.11.:** F-measure:
Content-based clustering, Diana

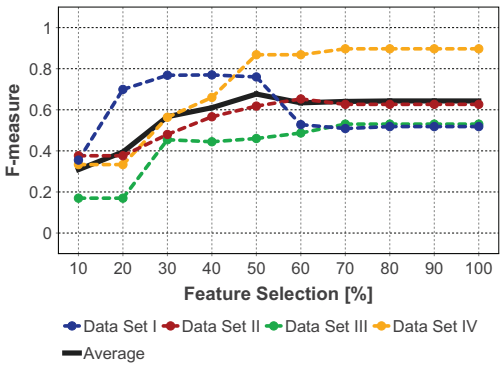


**Figure 6.12.:** F-measure:
Content-based clustering, PAM

On average, feature selection doesn't much enhance the clustering with Diana (Figure 6.11). Only data set IV (including small topics) has better results by using 70% or 50% of the term weight values. Looking at the other data sets each dimension reduction reduces the cluster quality.
Using PAM with content-based clustering (Figure 6.12), the best results are given by 50% of the term weight values. The cluster quality of data set I (including single topics) are higher and the cluster quality of the other data sets are nearly constant. It is surprising, that clustering with 60%-100% of the term weights are nearly equal. In this range of feature selection, only unimportant term weights are removed. The deleted term weights don't influence the clustering results. Table 6.2 (p. 107) gives an overview about the absolute term weight thresholds and the number of deleted terms. The absolute term weight thresholds are different. Its range from 0.052 to 0.122. The absolute value depends on the term document matrix and the term document matrix depends on the data set. A feature selection of 50% deletes 311 terms in the term document matrix of data set I (including single topics). This and the deletion of further term weights enhances the cluster quality. Although the size of data set IV (including small topics) is smaller concerning the number of terms and articles, more terms are deleted. But the deletion of 551 terms in data set IV don't worsen the results.

The best enhancement via feature selection is achieved by using Agnes and link-based clustering (Figure 6.13). By using 90% of the term weight values the F-measure of all data sets are almost one. By using fewer than 90% of the term weight values the F-measure values of the data sets are decreasing. The F-measure values are constant between 10% and 50%, because this threshold defines that the distance between all articles is null. In this case, clustering with Agnes is not possible. The range of the thresholds is 0.087 to 0.333 (Table 6.2). Like the content-based clustering with PAM, the number of terms don't depends on the number of articles. Presumably it depends additionally on the size of the articles and on the similarity

of the topics. Compared to the content-based clustering, the number of terms are smaller and fewer terms are deleted. That proves that the term document matrix of the link-based clustering approach is smaller and includes more the important terms.
The clustering algorithm PAM based on link-based clustering (Figure 6.14) gets the best results with 90% of the term weight values as well. The F-measure of data set II (including overlapping topics) decreases concerning clustering without feature selection (100%), but the F-measure of data set IV (including small topics) increases highly. The F-measure values are constant between 10% and 50%, because this threshold defines that the distance between all articles is null. In this case, PAM groups the articles to a big cluster. Since the term document matrix depends not on the clustering algorithm, the number of terms, the absolute threshold and the number of deleted terms is the same as link-based clustering with Agnes (Table 6.2). Only the F-measure values are different (Figure 6.13 and 6.14), because the clustering algorithms creates different clusters.

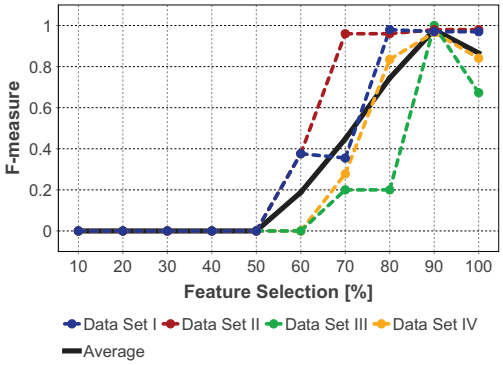


**Figure 6.13.:** F-measure:
Link-based clustering, Agnes

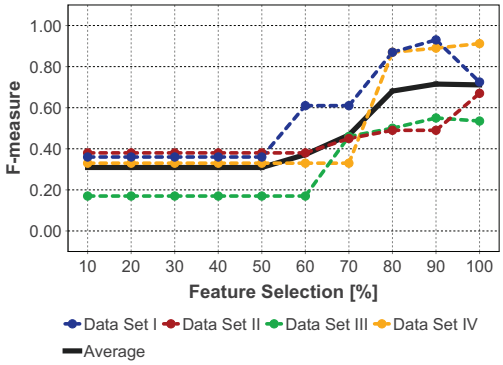


**Figure 6.14.:** F-measure:
Link-based clustering, PAM

| | # Terms | Term Weight Threshold | # Terms after Feature Selection |
|---|---|---|---|
| ***Content-based Clustering with 50% of the term weights*** | | | |
| Data set I | 15,514 | 0.052 | 15,203 |
| Data set II | 6,285 | 0.110 | 6,041 |
| Data set III | 9,311 | 0.122 | 8,970 |
| Data set IV | 7,377 | 0.200 | 6,826 |
| ***Link-based Clustering with 10% of the term weights*** | | | |
| Data set I | 14,667 | 0.087 | 14,570 |
| Data set II | 3,613 | 0.200 | 3,571 |
| Data set III | 6,065 | 0.174 | 6,004 |
| Data set IV | 5,169 | 0.333 | 5,092 |

**Table 6.2.:** Top term weight thresholds

### 6.2.2. Summary

In summary, feature selection enhances the cluster quality. The challenge is the definition of a suitable threshold. The threshold depends on the clustering algorithm and data structure (data set and clustering approach). If too many term weight values are deleted, the cluster quality decreases rapidly. In this analysis with a threshold of 90% the best link-based clustering results are achieved. Using content-based clustering, the thresholds are different. The clustering algorithm Diana achieves the best results without feature selection. With a feature selection threshold of 50% the algorithm PAM has the best results. Thus the definition of a term weight threshold seems to be more stable with link-based clustering than with content-based clustering. The best overall clustering quality is achieved with Agnes and link-based clustering. The F-measure values of all data sets are nearly one. The selection of suitable thresholds is listed in Table 6.3.
The absolute term weight thresholds are diversified, because the term weight range depends on the data set and clustering approach. Thus the definition of relative thresholds in percent is more practical. The number of terms does not depend on the number of articles. Probably the size of the articles and the similarity of the topics influence the number of terms as well. The link-based clustering approach has fewer terms than the content-based clustering approach. Since the cluster quality of the link-based clustering approach is higher, the term document matrix of the link-based clustering approach includes more important terms. Using the content-based clustering approach further terms have to be removed.

| | ***Content-based Clustering*** | | ***Link-based Clustering*** | |
|---|---|---|---|---|
| | **Diana** | **PAM** | **Agnes** | **PAM** |
| Term Weight Threshold | - | 50% | 90% | 90% |

**Table 6.3.:** Selection of term weight thresholds

## 6.3. Clustering Phase 3 - Feature Transformation

Another dimension reduction method, which should improve the clustering, is feature transformation. In this section, the influence of Latent Semantic Analysis (LSA, described in Section 4.3.3, p. 48) is evaluated.
Before executing LSA, feature selection (Term Weight Thresholding) is applied. The best feature selection thresholds of the previous section are used (see Table 6.3, p. 108). The clustering algorithm *Diana* uses no feature selection, because the best results are achieved by 100%. In the context of content-based clustering, the algorithm *PAM* achieved the best results with 50% of the term weights. In the context of link-based clustering, both clustering algorithms (Agnes, PAM) achieved the best clustering quality by 90%.
LSA needs the definition of a final number of dimensions. The maximum number of dimensions is given by the number of articles in a data set. The largest data set is data set I including 100 articles. Thus the maximum number of dimensions is 100. The maximum number of dimensions of the other data sets are different, because they include another number of articles. The maximum number of dimensions shows the clustering without feature transformation (LSA). To define a suitable dimension value, a stepwise clustering within the dimension range is executed. Compared to the section before, the absolute values are used, because in most cases a small number of dimensions should be already sufficient. Thereby the step within the range of 1-10 dimensions was chosen to be 1. In the range from 10 to 100 dimensions the step was set at 5. One possibility to define the number of dimensions is the plotting of the singular values. A flattening or break indicates a good value.
The F-measure is used to evaluate the results. The influence of feature transformation on the Dunn Index, SD Validity Index and computing time is explored in the last clustering phase (see Section 6.4, p. 115). In the following subsection the singular values are analyzed. In some cases the singular values can be helpful to define the number of dimensions. Section 6.3.2 includes an external quality analysis based on the F-measure (p. 111). The results are summed up in the last section (Section 6.3.3, p. 114). Appendix F.3 includes the values of the diagrams.

### 6.3.1. Singular Value Analysis

The singular values of LSA are displayed in the Figures 6.15, 6.16 and 6.17. Figure 6.15 (no feature selection uses by Diana) and Figure 6.16 (50% feature selection uses by PAM) visualize the singular values of the content-based clustering approach and Figure 6.17 (90% feature selection uses Agnes and PAM) visualizes the singular values of the link-based clustering approach. On the x-axis the actual selected number of dimensions are given. The y-axis shows the singular values concerning the number of dimensions.
A flattening or break in the diagrams indicates a good number of dimensions value. Unfortunately there is no flattening or break in the plots (see additionally Table F.14 (p. 184), Table F.13 (p. 184) and Table F.15 (p. 185)). The singular values have a decreasing order. Thus the higher the number of dimensions, the smaller the singular value. The singular values in each diagram are very similar. The values differ only in an insignificant way. Furthermore the singular values depend on the size of the term document matrix. In each diagram, data set I (including single topics) has the highest singular values. The term document matrix of data set I is the largest with 100 articles and around 15,000 terms (15,514 terms with content-based clustering and no feature selection; 15,203 terms with content-based clustering and

50% feature selection; 14,570 terms with link-based clustering and 90% feature selection). Data set III (including topics of a main topic, 66 articles) is the second largest data set and has the second largest singular values. Data set II (including overlapping topics, 50 articles) and data set IV (including small topics, 30 articles) are the smallest data sets.

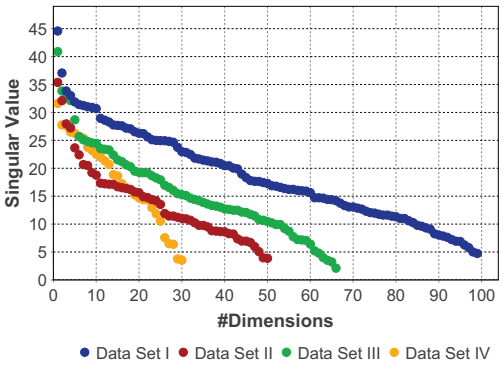


**Figure 6.15.:** Singular Values: Content-based clustering, no feature selection

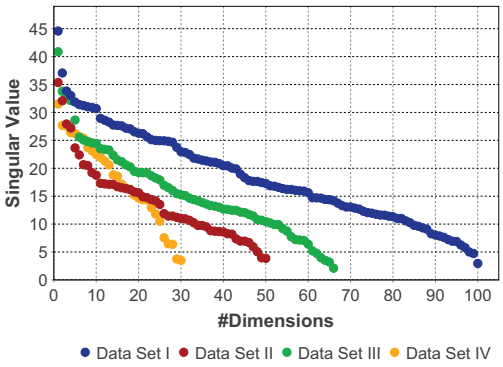


**Figure 6.16.:** Singular Values: Content-based clustering, 50% feature selection

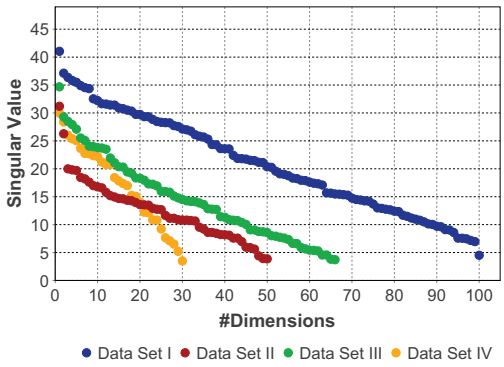


**Figure 6.17.:** Singular Values: Link-based clustering, 90% feature selection

The diagrams show that the singular values of each clustering approach and feature selection value are very similar. Furthermore the singular values depends on the size of the term document matrix and thus on the number of articles and terms. Based on the diagrams a selection of a number of dimensions value is not possible. There is no flattening or break in the plots. Accordingly, the number of dimensions have to be defined based on the F-measure (following section). The highest F-measure defines the number of dimensions for the further analysis.

### 6.3.2. External Quality Analysis

Because the number of dimensions cannot be defined based on the singular values, the F-measure has to be used. The Figures 6.18 (Diana, content-based), 6.19 (PAM, content-based), 6.20 (Agnes, link-based) and 6.21 (PAM, link-based) display the F-measures for different final number of dimensions. On the x-axes the actual number of dimensions for feature transformation (LSA) is visualized. The range is between 1 to 100. The maximum number of dimensions is 100, because data set IV has 100 articles. The other data sets have fewer articles and thus the maximum number of dimensions by this data sets is lower. On the y-axis the F-measure is displayed.
The values of the F-measure vary very strongly. There is no dependency between the F-measure and number of dimensions recognizable. In some cases, the F-measure increases again by a small number of dimensions (e.g. Figure 6.19 (data set I and III), Figure 6.20 (data set I, II, III)). After clustering with the maximum number of dimensions the values decrease very strongly (e.g. Figure 6.20, data set I in the range of 95-100 dimensions, data set II in the range of 45-50 dimensions, data set III in the range of 60-66 dimensions and data set IV in the range of 30-30 dimensions). Contrary to the singular values (Figure 6.15, 6.16 and 6.17), the F-measure values in each diagram are not similar.
Based on the diagrams the best dimensions value is selected. It is the value with the highest F-measure. Because the term document matrix and thus the number of dimensions is different for each data set, clustering approach and clustering algorithm, the final dimensions value is defined for each data set, clustering approach and clustering algorithm.
Based on Figure 6.18 the final dimensions value for the clustering algorithm Diana (content-based clustering) can be defined. Data set I (including single topics) and II (including overlapping topics) don't achieve a better clustering quality while using feature transformation with LSA. The cluster quality of data set III (including topics of a main topic) is enhanced by using only nine dimensions. Five dimensions enhance the cluster quality of data set IV (small topics). Probably the cluster quality of data set I and II cannot be enhanced, because the data sets achieve already a nearly optimal clustering without LSA.
In context of content-based clustering with PAM (Figure 6.19), the cluster quality of data set I, II and III can be enhanced. Data set I (including single topics) and data set II (including overlapping topics) achieve the best results with 40 dimensions. Data set III (including topics of a main topic) has the best results with six dimensions. Like the content-based clustering with Diana, the data sets which have a F-measure over 0.8 (without feature selection) cannot be enhanced while using feature selection. In this case, it is the data set IV (including small topics).
While using link-based clustering with Agnes (Figure 6.20) the cluster quality cannot be enhanced with LSA. The cluster quality without feature selection is already nearly optimal.
Like content-based clustering with PAM, the cluster quality of link-based clustering with PAM (Figure 6.21) can be enhanced by the data sets I, II and III. Data set II (including overlapping topics) and data set III (including topics of a main topic) have the best results with 15 dimensions. Although data set I (including single topics) has a very good clustering quality without dimension reduction, the quality can be enhanced marginally by using only eight dimensions. Since data set IV (including small topics) has good results without feature transformation, the results cannot be enhanced with LSA.

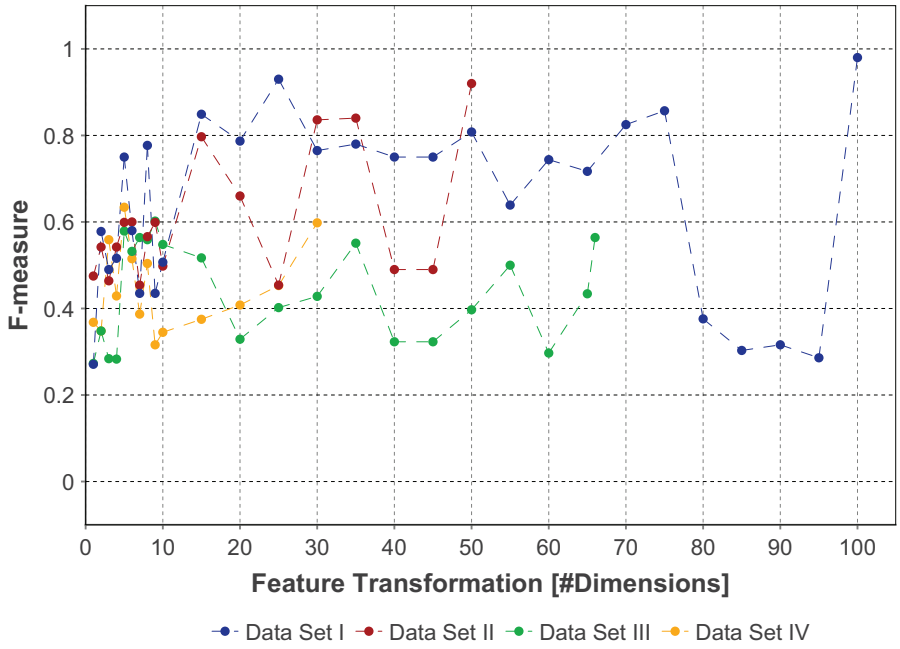


**Figure 6.18.:** F-measure: Content-based clustering, Diana

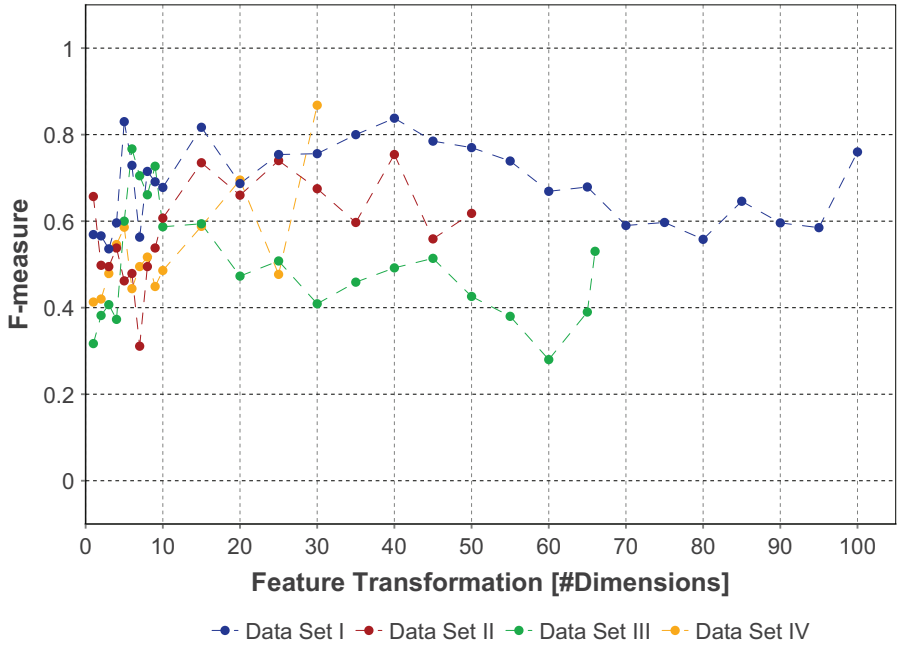


**Figure 6.19.:** F-measure: Content-based clustering, PAM

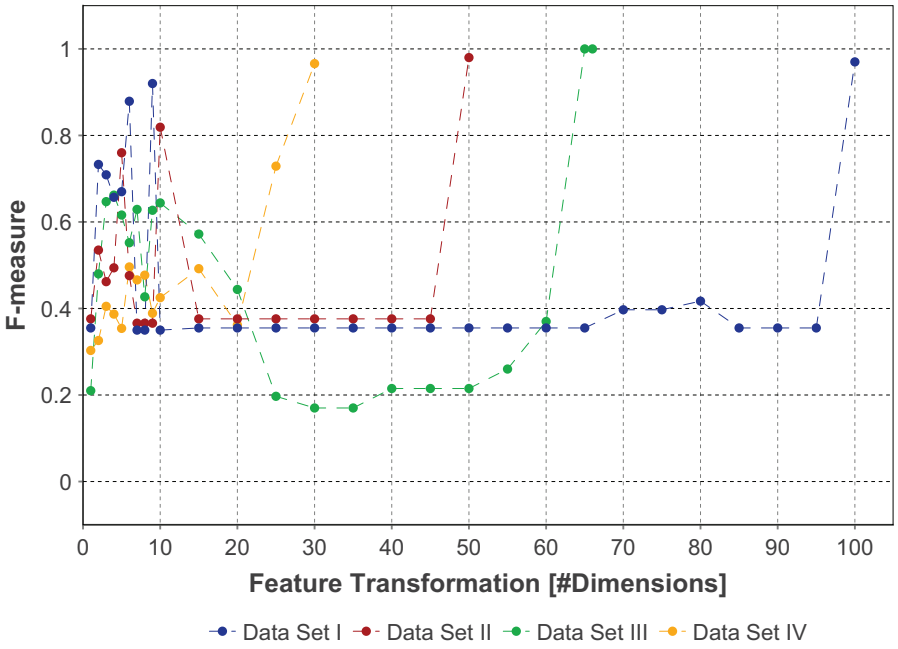


**Figure 6.20.:** F-measure: Link-based clustering, Agnes

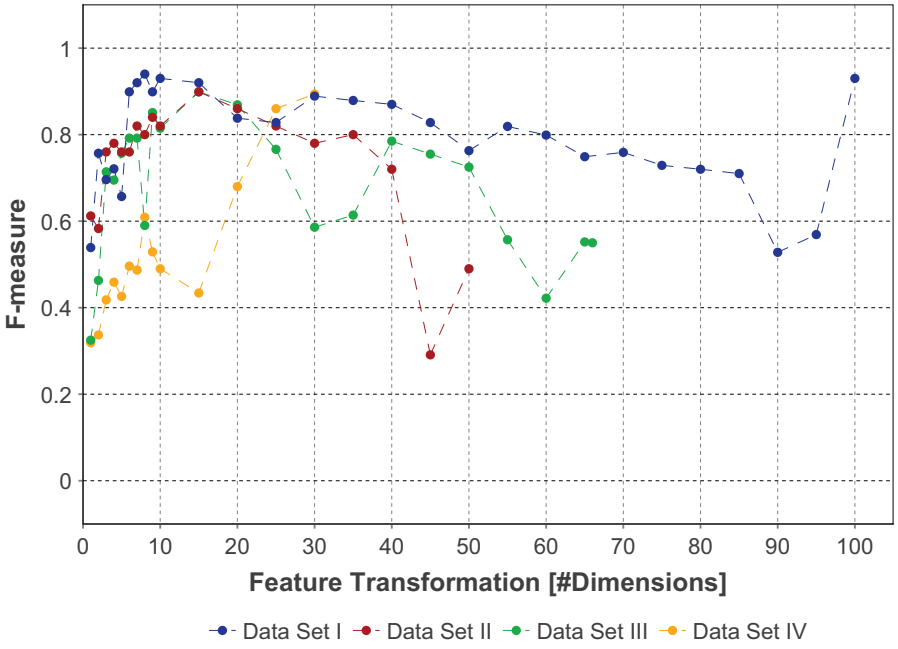


**Figure 6.21.:** F-measure: Link-based clustering, PAM

### 6.3.3. Summary

In some cases dimension reduction enhances the cluster quality. In the context of content-based clustering, LSA enhances the cluster quality in most cases. Using link-based clustering with PAM, the results of the data sets I-III are enhanced. Table 6.4 includes the optimal number of dimensions for each clustering configuration. Since the optimal number of dimensions depends on the size of the data set, each data set gets its own dimension value. A uniform value could not be found. Thus the defined values are not suitable for other data sets and clustering algorithms.
If a good clustering quality is already achieved (F-measure over 0.8), LSA decreases the results. This is the case for data sets I and II with the clustering algorithm Diana and for data set IV with the clustering algorithm PAM. Link-based clustering has the same characteristic. Since the clustering algorithm Agnes has achieved nearly optimal clustering results, LSA doesn't enhance the clustering solution.
The problem is the definition of an optimal number of dimensions. For some clustering configurations the results are enhanced by using LSA, but these enhancements are not visible in the singular value plots. Consequently it is difficult to define an optimal dimension value. Additionally, there is the danger that the final number of dimensions reduces the cluster quality. The approach which is used in this analysis is impractical for the use of feature selection in practice, because this approach is very complex. Another approach to define the final number of dimensions has to be found. For all practical purposes the measures show that in most cases a small value (1-15 dimensions) improves the cluster quality, furthermore a dimension value nearly the maximum number of clusters is not suitable.

| | ***Content-based Clustering*** | | ***Link-based Clustering*** | |
|---|---|---|---|---|
| | **Diana** | **PAM** | **Agnes** | **PAM** |
| Data set I | - | 40 | - | 8 |
| Data set II | - | 40 | - | 15 |
| Data set III | 9 | 6 | - | 15 |
| Data set IV | 5 | - | - | - |

**Table 6.4.:** Selection of optimal number of dimensions

## 6.4. Clustering Phase 4 - Content-based vs. Link-based Clustering

In this section the content-based and the link-based clustering approach are compared based on the findings of the previous sections of this chapter. In the first section (Section 6.1, p. 98), suitable clustering algorithms for the content-based and link-based clustering approach are chosen (clustering algorithms are described in Section 4.3.4, p. 63). The next two sections define optimal values for the dimension reduction techniques Term Weight Thresholding (Section 6.2, p. 105; Term Weight Thresholding is described in Section 4.3.3, p. 60) and Latent Semantic Analysis (Section 6.3, p. 109; Latent Semantic Analysis (LSA) is described in Section 4.3.3, p. 48). The content-based and link-based clustering approach are compared using the optimized clustering configurations. In the first section (Section 6.1, p. 98) a first comparison between content-based and link-based clustering is given. The analysis shows that link-based clustering creates superior clusters when compared to the content-based clustering approach. But in the first section no dimension reduction methods are used. With the use of dimension reduction techniques, the results may change. While the goal of the first section and the overall objective of this paper is the identification of suitable clustering algorithms, the subject of this section is an intensified analysis and comparison of the clustering approaches as such.
Like the sections before, the clustering algorithms Diana and PAM are used for the content-based clustering approach. The link-based clustering approach uses the algorithms Agnes and PAM (see Table 6.1, p. 104). Feature selection with Term Weight Thresholding is executed on PAM (both approaches) and Agnes (see Table 6.3, p. 108). Thereby PAM uses 50% of the features by the content-based clustering approach. By the link-based clustering approach both clustering algorithms use 90% of the features. Feature transformation based on Latent Semantic Analysis is defined for each clustering approach, clustering algorithm and data set (see Table 6.4, p. 114). Based on the content-based clustering, Diana uses feature transformation for data set III (nine dimensions) and data set IV (five dimensions) and PAM uses feature transformation for data set I (40 dimensions), data set II (40 dimensions) and data set III (six dimensions). Based on the link-based clustering, PAM uses feature transformation for data set I (eight dimensions), data set II (15 dimensions) and data set III (15 dimensions). Agnes does not use feature transformation. An overview about the cluster configurations can be found in Table 5.10 (p. 96).
Like in the first section, the F-measure, Dunn Index, SD Validity Index and the computing time (described in Section 4.3.4, p. 71) are used for comparing the results. Additionally the results of the best clustering algorithm of the content-based and link-based clustering approach are visualized in Wiki-link networks (described in Section 3.2.1, p. 16). In the first section (Section 6.4.1, p. 116) the results of Section 6.1 and 6.3 are compared. The enhancements due to the use of dimension reduction methods are evaluated. Furthermore the results of content-based clustering and link-based clustering with dimension reduction are compared by an external quality analysis using the F-measure (see Section 6.4.2, p. 117), an internal quality analysis using the Dunn Index and SD Validity Index (see Section 6.4.3, p. 119) and by an analysis of the performance using the computing time (see Section 6.4.4, p. 121). Section 6.4.5 (p. 122) analyzes the results based on Wiki-link networks. The Wiki-link networks allow another view on the results. They visualize the articles, the links between the articles and their topics and clusters. The results are summed up in Section 6.4.6 (p. 130). The values of the diagrams are listed in Appendix F.4.

### 6.4.1. Improvement Analysis

Firstly, the improvement of the clustering quality while using dimension reduction is evaluated. The differences of the results from Section 6.1 and 6.3 are visualized in Figure 6.22 and 6.23. *C* stands for content-based clustering and *L* for link-based clustering. With these figures the differences between clustering with dimension reduction and clustering without dimension reduction are highlighted. The x-axes represent the clustering algorithms of the content-based and link-based clustering approach. The y-axis of Figure 6.22 represents the differences between the F-measure values of Section 6.1 and 6.3. The F-measure values from Section 6.1 are subtracted from the optimized F-measure values from Section 6.3. Thus the enhancement for clustering with dimension reduction can be evaluated. The y-axis of Figure 6.23 represents the difference between the computing time of Section 6.1 and 6.3. The use of dimension reduction methods causes a time increase, because further steps are executed.

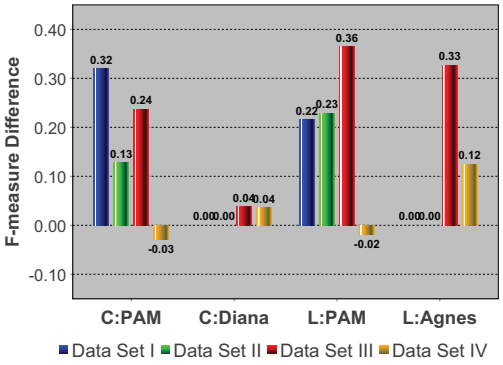


**Figure 6.22.:** F-measure: Clustering improvement through dimension reduction *(C = content-based clustering, L = link-based clustering)*

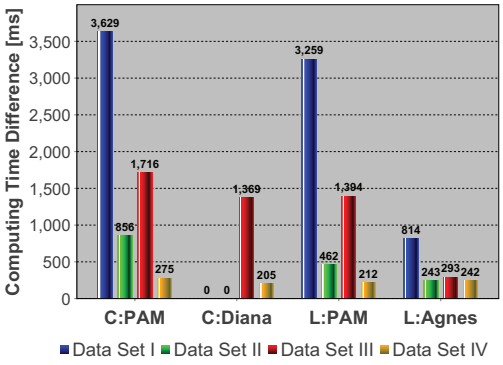


**Figure 6.23.:** Computing Time: Time increase through dimension reduction *(C = content-based clustering, L = link-based clustering)*

Figure 6.22 visualizes that dimension reduction improves the clustering quality. The F-measure value of almost each clustering algorithm and data set increases while using dimension reduction methods. But one recognizes as well, that the definition of suitable dimension reduction parameters depends on the clustering algorithm and data structure (data set and clustering approach). The clustering quality of data set IV (C:PAM and L:PAM) decreases. The best improvements are achieved with PAM. The use of dimension reduction methods by PAM with content-based clustering (C:PAM) and link-based clustering (L:PAM) enhances the results of the data sets. Thereby the best overall enhancement achieves PAM with link-based clustering (L:PAM). The results of data set I (F-measure difference of 0.215), data set II (F-measure difference of 0.229) and data set III (F-measure difference of 0.364) are improved while using dimension reduction. It is interesting to note that the results of data set IV are worsening by both clustering approaches (C:PAM and L:PAM). Thus the structure of data set IV has to be completely different. Using Diana with content-based clustering (C:Diana), dimension reduction enhances the results to a minimal extent. Diana uses only LSA and no Term Weight Thresholding. With Agnes based on link-based clustering (L:Agnes) it is different. Without dimension reduction, only data set I and II yield good results. With dimension reduction, these data sets give good results as well and the other two data sets exhibit an

improvement of the results.
The use of dimension reduction methods increase the performance. This fact is visualized in Figure 6.23. The increase of the computing time depends on the size of the data set as well. For this reason, data set I has the highest computing time difference. The clustering algorithm PAM has the largest increase of the computing time by both approaches (C:PAM and L:PAM). The computing time shows a significant increase, because PAM uses feature selection and feature transformation for the optimization. The computing time of Diana (C:Diana) increases not so much, because Diana doesn't use feature selection. On the other hand, Diana provides the poorest enhancement of the clustering results while using dimension reduction methods. The computing time of Agnes (L:Agnes) increases not so much as well, because Agnes doesn't use feature transformation. Agnes uses only feature selection and achieves with dimension reduction a nearly optimal solution.
In summary, dimension reduction improves the clustering results, but the computing time increases as well. Furthermore the definition of suitable dimension reduction parameters depends on the structure of the data set. In this analysis, data set IV has a different data structure. Thus the clustering results of data set IV decrease (see C:PAM and L:PAM), whereby the cluster quality of the other data sets increases. The clustering algorithm PAM (see C:PAM and L:PAM) has the best improvement of the clustering quality, but on the other side, the highest increase of computing time as well. The computing time of Agnes (see L:Agnes) increases not so much and with the use of dimension reduction methods nearly optimal solutions are achieved.

### 6.4.2. External Quality Analysis

The following diagrams compare content-based and link-based clustering based on the F-measure. The structure of this analysis is similar to the external quality analysis of the first section (Section 6.1, p. 98). The first two figures display the results in detail (Figure 6.24 and Figure 6.25). The other two figures sum up the results and highlight the best clustering algorithms and the data sets which are clustered best (Figure 6.26 and Figure 6.27). The * indicates that the clustering is optimized using dimension reduction.

Figure 6.24 (content-based clustering) and Figure 6.25 (link-based clustering) visualize the F-measure values for each clustering algorithm and data set. The x-axes visualizes the clustering algorithms. The clustering algorithms are executed on each data set. The F-measure values for the clustering results of each data set are displayed on the y-axis.
As is to be inferred from Figure 6.22 (p. 116), the results of PAM with dimension reduction are much better (Figure 6.24 and 6.25). The clustering with Diana (Figure 6.24) is hardly enhanced. Figure 6.25 shows, that Agnes in combination with link-based clustering has nearly optimal results. Data set III includes very similar topics, but it is clustered in an optimal way. All data sets are clustered best from Agnes with link-based clustering. With the exception of the results from Diana the results per data set are very similar. Thus it is not possible to say, which data set contains similar topics.
The use of dimension reduction techniques enhances the quality of clustering. Except for Diana with content-based clustering, appropriate clustering results are achieved. Even data sets with similar topics are clustered very well. The best clustering quality is achieved with Agnes in combination with link-based clustering.

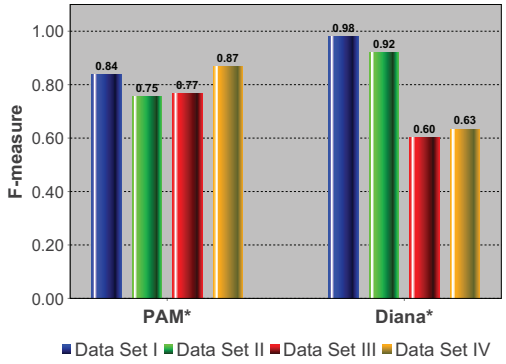


**Figure 6.24.:** F-measure: Content-based clustering

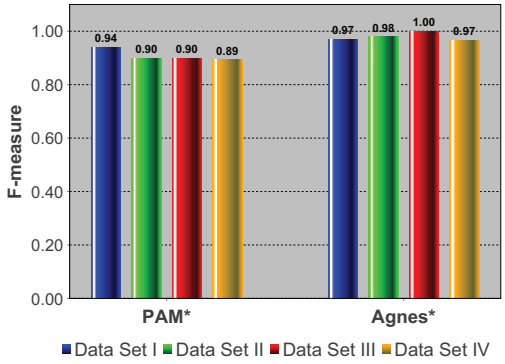


**Figure 6.25.:** F-measure: Link-based clustering

To select the best clustering algorithm and clustering approach, the average of the results have to be calculated and compared. Furthermore the results of each data set are evaluated.
The average results of the F-measure are visualized in 6.26 (clustering algorithm view) and 6.27 (data set view). The x-axis of the clustering algorithm view represents the clustering algorithm. The average of the values per clustering algorithm (F-measures of the corresponding data sets) is calculated and presented on the y-axis. The x-axis of the data set view represents the data sets. The average of the values per data set (F-measures of the corresponding clustering algorithm) is calculated and depicted on the y-axis.

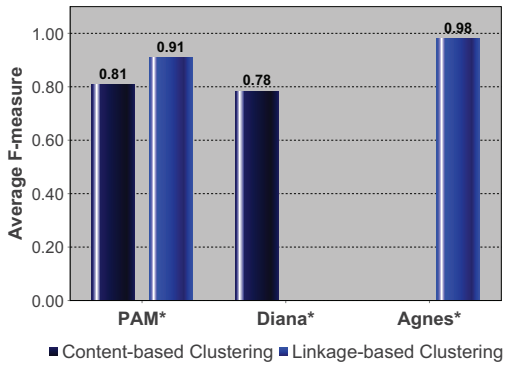


**Figure 6.26.:** F-measure: Clustering algorithm view

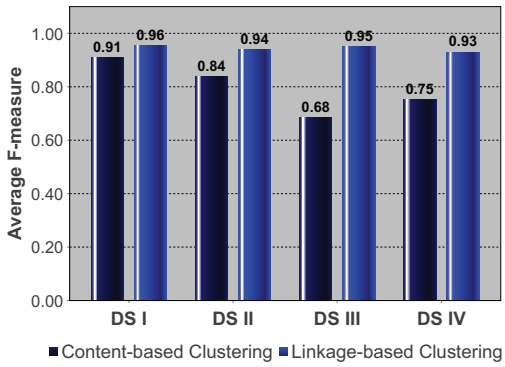


**Figure 6.27.:** F-measure: Data set view *(DS = Data Set)*

The summing up of the results show, that link-based clustering has the best results with both clustering algorithms (Figure 6.26). Contrary to the clustering without dimension reduction, PAM has better results than Diana in the context of content-based clustering. Consequently, the dimension reduction methods enhance the results of PAM very well. As expected, Agnes in combination with link-based clustering has the best overall clustering quality.
Each data set is clustered better by using link-based clustering (Figure 6.27). The results are nearly optimal. The content-based clustering results are different. The best results are given by data set I (including single topics). Data set III (including topics of a main topic) has the poorest results on average.
Like in the first analysis phase (Section 6.1, p. 98), link-based clustering achieved a better

clustering quality than content-based clustering. The use of dimension reduction methods enhances the results additionally. The results are enhanced in context of content-based clustering as well, however content-based clustering faces the problem of clustering of similar topics. The best results are achieved with link-based clustering in combination with a feature selection of 90% and the agglomerative hierarchical clustering algorithm Agnes. PAM has good results as well, but needs additionally feature transformation.

### 6.4.3. Internal Quality Analysis

In this section the internal quality of the clustering is analyzed by using the Dunn Index and SD Validity Index (described in Section 4.3.4, p. 71). Like in subsection above, the average results are shown. Because the results of the internal quality analysis are different to the external quality analysis, the results of the internal quality analysis are compared with the optimal solution. The * indicates that the clustering is optimized using dimension reduction.

Figure 6.28 (Dunn Index) and 6.30 (SD Validity Index) represent the average results of the clustering algorithms (clustering algorithm view). The x-axes represent the clustering algorithms and the y-axes sum up the data set results per clustering algorithm. Figure 6.29 (Dunn Index) and 6.31 (SD Validity Index) represent the average results of the data sets (data set view). The x-axes represent the data sets and the y-axes sum up the clustering algorithm results per data set. Based on these diagrams the internal quality of the clusters per clustering algorithm and data set can be compared and evaluated.
With the exception of the first data set, the Dunn Index corresponds to the F-measure (Figure 6.28 and 6.29). Figure 6.28 shows that the Dunn Index of link-based clustering is better than the Dunn Index of content-based clustering. Agnes (link-based clustering) has the best clustering results, PAM (link-based clustering) the second best. Diana (content-based clustering) has the poorest internal quality based on the Dunn Index. In the data set view (Figure 6.29) data set I and II (including single topics and overlapping topics) have the best Dunn Index. This means that the distance between clusters in these data sets is high (high distance) and the distance between documents in each cluster is small (high density). In context of content-based clustering the clusters of the third data set (including topics of a main topic) have the smallest density and distance. The topics of this data set are very similar. Since the topics of the fourth data set (including small topics) includes only five articles, the density and distance of the clusters is small as well. Using link-based clustering the topics can be differentiated better, because the Dunn Index is higher. For this reason, link-based clustering is more simple than content-based clustering.
The values of the SD Validity Index are different (Figure 6.30 and 6.31). Like the measures without dimension reduction, content-based clustering has a smaller SD Validity Index. This means that the internal quality of the content-based clustering is better. The internal quality depends on the structure of the data sets and on the clustering results. Corresponding to the SD Validity Index the content-based clustering forms good clusters, but compared to the F-measure the clusters are wrong. The reasons for these contradictory results are described in Section 6.1.2 (p. 100).
The results of the Dunn Index and SD Validity Index are different. At the same time, the results of the Dunn Index correspond to the results of the F-measure. Based on this index, the link-based clustering is better than the content-based clustering. The comparison of the data sets shows that the internal structure of the data sets is different. Data set I and II have

a data structure which is more suitable for clustering. Based on the Dunn Index, Agnes with link-based clustering achieved the best internal quality.

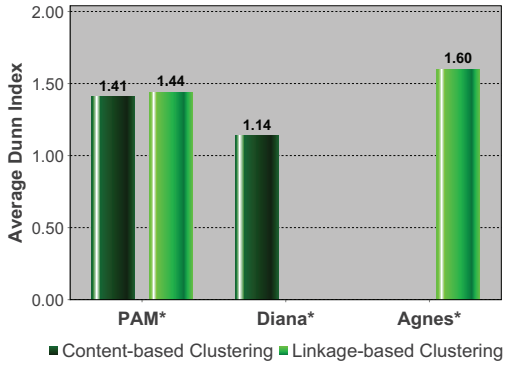


**Figure 6.28.:** Dunn Index: Clustering algorithm view

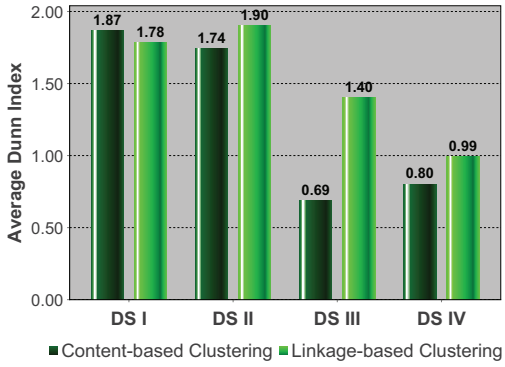


**Figure 6.29.:** Dunn Index: Data set view *(DS = Data Set)*

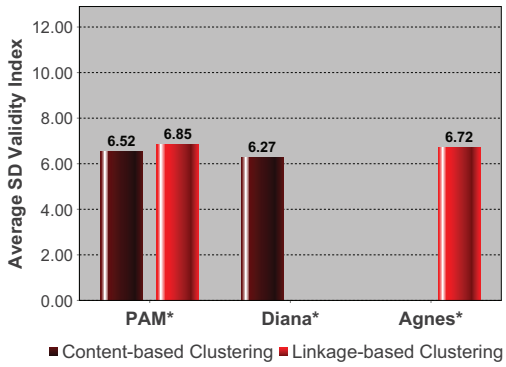


**Figure 6.30.:** SD Validity Index: Clustering algorithm view

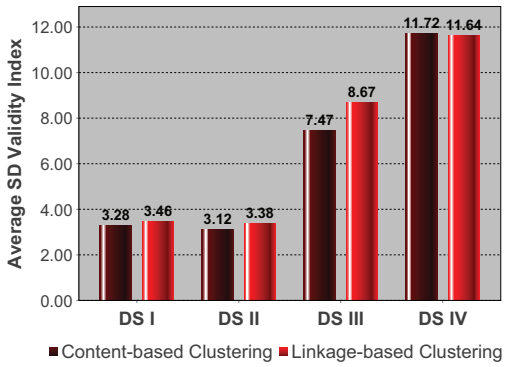


**Figure 6.31.:** SD Validity Index: Data set view *(DS = Data Set)*

To prove the validity of the Dunn Index and SD Validity Index, the optimal clustering solution of each data set is formed and the indices of these solutions are calculated. Figure 6.32, 6.33, 6.34 and 6.35 compare these results with the content-based and link-based clustering solutions. Figure 6.32 (Dunn Index) and 6.34 (SD Validity Index) compare the content-based clustering with the optimal solution and Figure 6.33 (Dunn Index) and 6.35 (SD Validity Index) compare the link-based clustering with the optimal solution. The x-axes represent the clustering algorithms and the y-axes depict the average index values of each clustering algorithm.

The content-based clustering results in a higher Dunn Index than the optimal solution (Figure 6.32). This evidence showing that the data set structures of the content-based clustering are not suited for the clustering of topics. The link-based clustering approach is better suited. The Dunn Index of the link-based clustering is smaller than the optimal solution (Figure 6.33). However, the difference between the Dunn Index values is not significant, because link-based clustering achieved nearly optimal results.

The comparison of the SD Validity Index values produces the same results. Using content-based clustering (Figure 6.34), PAM has a better SD Validity Index than the optimal solution.

The clustering with Diana could be improved, because the SD Validity Index of the optimal solution is smaller than the SD Validity Index of Diana. Using link-based clustering (Figure 6.35), the SD Validity Index reflects the quality of the clustering results. Only a small difference is found between the link-based clustering and the optimal solution.
The figures show that the internal data structure of content-based clustering causes a wrong clustering. The optimal solutions have a worse Dunn index and SD Validity index than the computed solutions. The internal structure of the link-based clustering approach is more suitable for topic clustering. The optimal solutions have a slighty better Dunn index and SD Validity index than the computed solutions.

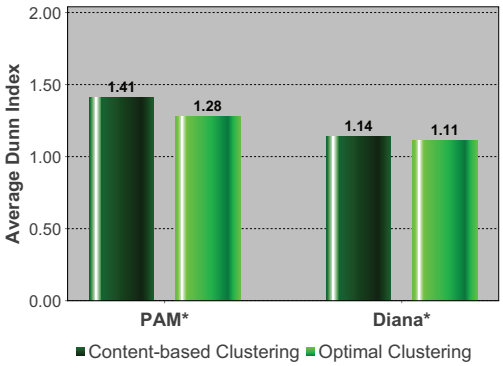


**Figure 6.32.:** Dunn Index:
Content-based vs. optimal clustering

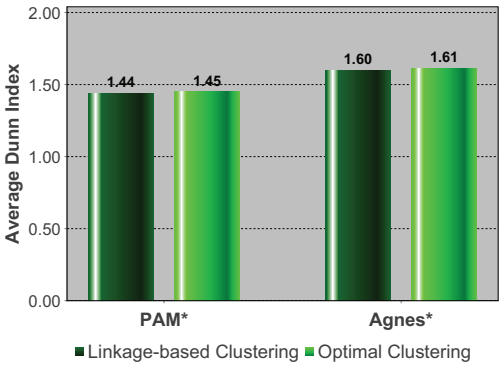


**Figure 6.33.:** Dunn Index:
Link-based vs. optimal clustering

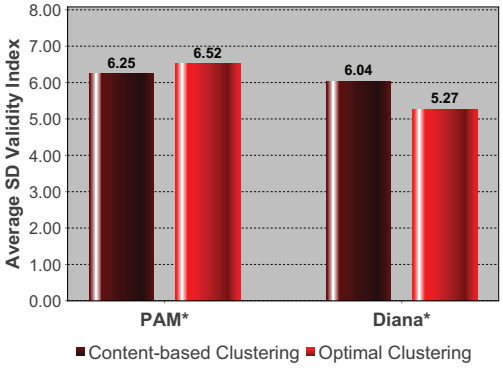


**Figure 6.34.:** SD Validity Index:
Content-based vs. optimal clustering

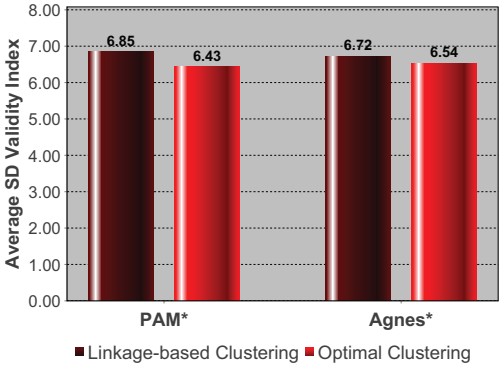


**Figure 6.35.:** SD Validity Index:
Link-based vs. optimal clustering

### 6.4.4. Performance Analysis

In this section the performance of the clustering is analyzed. The computing time is measured and the average results are calculated. The results are visualized in Figure 6.36 (clustering algorithm view) and Figure 6.37 (data set view). The x-axes represent the clustering algorithms (Figure 6.36) and data sets (Figure 6.37). The y-axis of Figure 6.36 visualizes the

average computing time of each clustering algorithm and the y-axis of Figure 6.37 visualizes the average computing time of each data set. The * indicates that the clustering is optimized using dimension reduction.

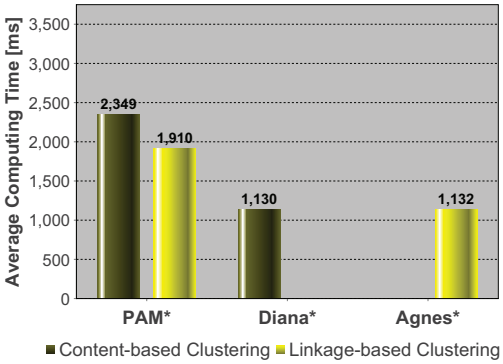


**Figure 6.36.:** Computing Time: Clustering algorithm view

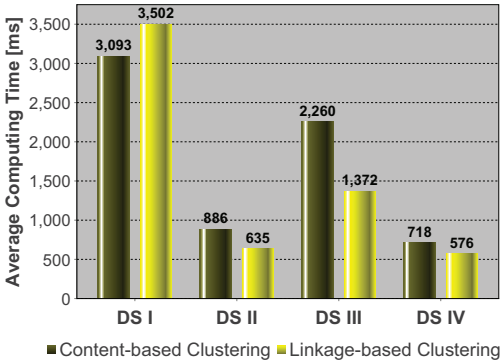


**Figure 6.37.:** Computing Time: Data set view *(DS = Data Set)*

The computing time shows that in most cases the link-based clustering approach has a better performance. PAM has the poorest performance (Figure 6.36), because the algorithm uses feature selection and feature transformation. The computing time of Agnes and Diana is less. Diana does not use feature selection and Agnes does not use feature transformation.
The computing time of the data sets depends on the size of the data sets (Figure 6.37) as well. Data set I (including single topics, 100 articles) and III (including topics of a main topic, 66 articles) have the highest computing time. The computing time of data set II (including overlapping topics, 50 articles) and IV (including small topics, 30 articles) is least. Only for data set I (including single topics) the content-based clustering requires less computing time than the link-based clustering.
The computing time depends on the data set size, the selected clustering approach and the selected dimension reduction methods. The use of dimension reduction methods increases the clustering results, but the computing time as well. The computing time of the link-based clustering approach is less, because the corresponding term document matrix is smaller. Because the clustering quality of link-based clustering is better, the term document matrix includes fewer unimportant terms.

### 6.4.5. Wiki-link Network Analysis

Although link-based clustering has good results, the optimal clustering is only achieved with the data set III (including topics of a main topic). Therefore different reasons are possible. The reasons for this behavior are explored in this section. Especially for this Wiki-link networks are used.Figure 6.38 and 6.39 visualize the topics of all data sets in Wiki-Link networks (described in Section 3.2.1, p. 16). Articles are represented by nodes. An edge between two article-nodes describes a link between the articles. Figure 6.38 groups the articles to their topics and gives an overview about the linkage between the topics. Figure 6.39 visualizes the articles. Articles of one topic are near together. In each network, the topics are highlighted by a color.

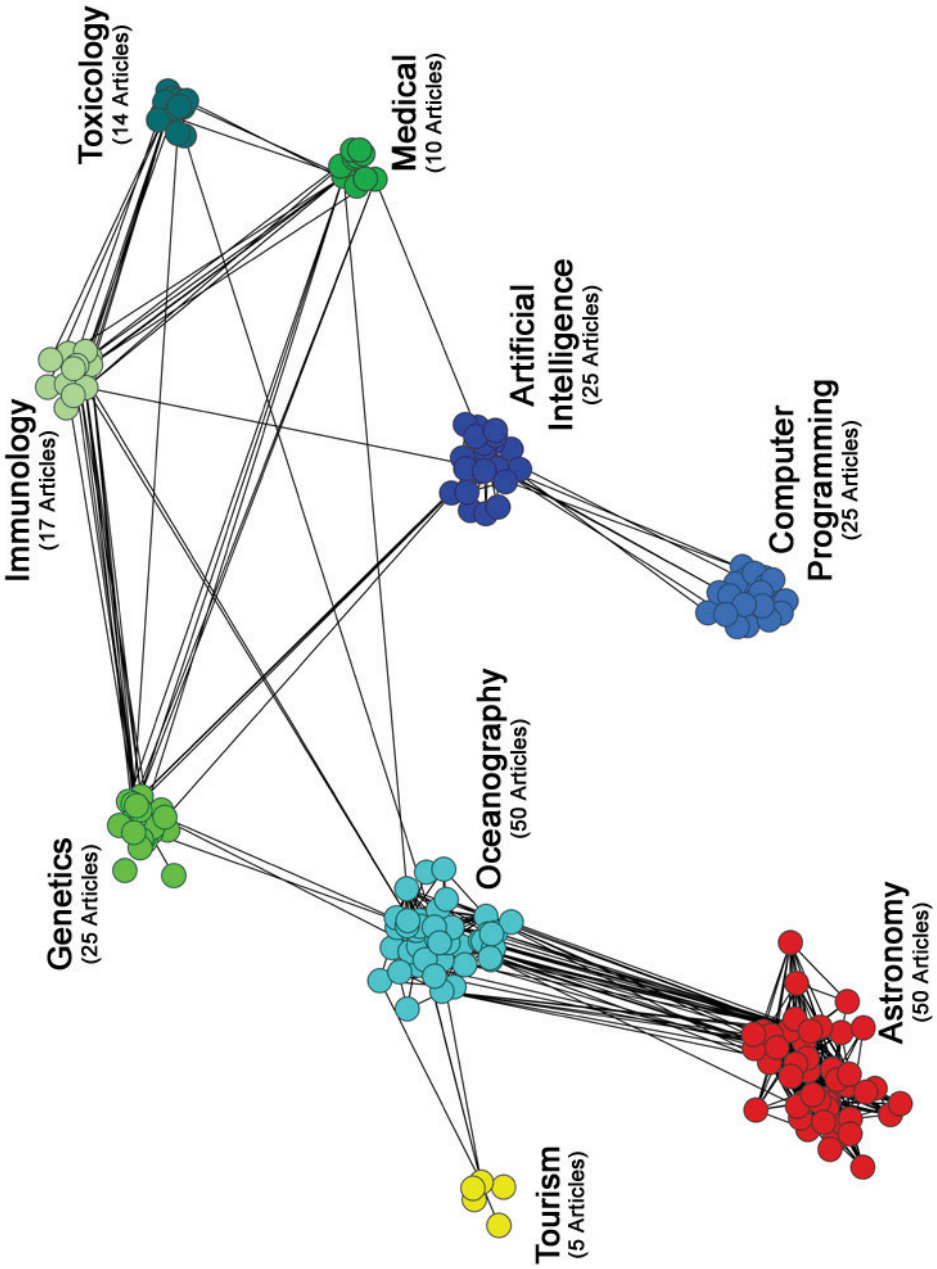


**Figure 6.38.:** Wiki Link Network - Topics, *created with SONIVIS*

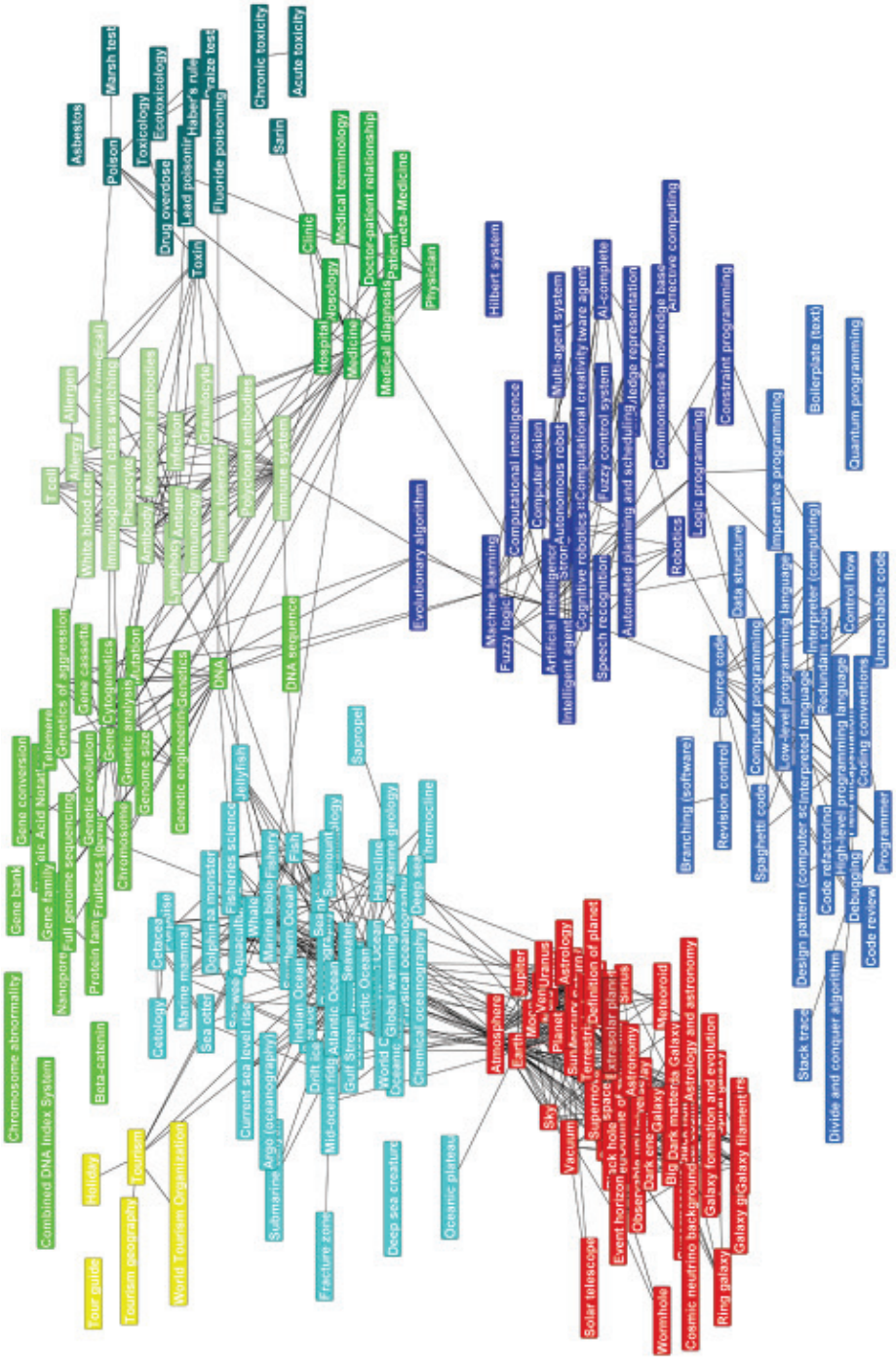


**Figure 6.39.:** Wiki Link Network, *created with SONIVIS*

Figure 6.38 shows that no topic remains separate. There are a lot of links between the topics. These links could impede the clustering. As expected there are links between the medical topics *Genetics*, *Immunology*, *Toxicology* and *Medical* (data set III, including topics of a main topic) and the topics *Artificial Intelligence* and *Computer Programming* (data set II, including overlapping topics). Unexpectedly there are a lot of links between the two topics of the first data set *Astronomy* and *Oceanography* (including single topics) and the topics *Oceanography* and *Artificial Intelligence* have links to the medical topics.
Figure 6.39 highlights the dependencies between the topics as well. Articles like *DNA*, *Tourism*, *Atmosphere* and *Earth* are bridges between different topics. The networks remember on small world networks, because they have high-connected subgraphs with a few edges between the subgraphs. There are articles which are not linked, such as *Beta-catenin*, *Deep sea creature*, *Hilbert system* and *Quantum programming*. Furthermore there are articles which are not linked to their topic, but to other topics, such as *Oceanic plateau* and *Sarin*.
Link-based clustering has better results than content-based clustering, because the terms which are used by link-based clustering reflect better the topic dependencies between articles. Consequently the important terms of an article have to be in the titles of outgoing links. That can be a disadvantage as well. Articles with few links or with links to articles which belong to a different topic, worsen the results. Furthermore there are articles which have no links and articles which are not linked to their topics, but to other topics. However it has to be considered that the Wiki-links networks show only a subset of the used links. The link-based clustering uses links which are not inside the Wiki-link networks as well. A dependency between two articles may also exists, if the articles are connected via a third article. Because all outgoing links of an article are considered, this third article hasn't been part of the data set.

In the following figures the results of the best clustering algorithm of the content-based and link-based clustering approach are visualized in Wiki-link networks. Agnes is the best clustering algorithm for link-based clustering and PAM delivers the best results in the context of content-based clustering. Figure 6.40 (content-based clustering) and Figure 6.41 (link-based clustering) visualize the results of the the first data set (including single topics). The results of the second data set (including overlapping topics) are visualized in Figure 6.42 (content-based clustering) and Figure 6.43 (link-based clustering). Figure 6.44 (content-based clustering) and Figure 6.45 (link-based clustering) includes the Wiki-link networks of the third data set (including topics of a main topic) and Figure 6.46 (content-based clustering) and Figure 6.47 (link-based clustering) includes the Wiki-links networks of data set IV (including small topics). Each cluster is visualized by a color. The optimal clustering solutions are visualized by putting articles of one cluster near together. The * indicates that the clustering is optimized using dimension reduction techniques.
The first Wiki-link network (Figure 6.40, content-based clustering with PAM) shows that a lot of articles of the topic *Astronomy* are assigned to the cluster of *Oceanography*. Many of these articles describe planets (*Earth*, *Moon*, *Uranus*, *Mars*, *Venus* and *Mercury (planet)*). It could be possible that the articles about the planets are closely related to the article about the *Earth*. The article about the *Earth* is a bridge between the two topics. Using link-based clustering for data set I (Figure 6.41), three articles are clustered wrong: *Ring Galaxy*, *Sirius* and *Earth*. The article *Earth* builds a bridge between the two topics. A correct clustering of this article is very difficult. *Ring Galaxy* seems to be a very young article (see Appendix E) with only a few revisions (48) and links (8). The number of terms (69) is very small as well. There is not enough information for clustering the article correctly. The article *Sirius* is not young. There are a lot of revisions (1556), links (299) and terms (962). A look into

the article text shows that the article includes a section about *Observational history*. Because of this section, there are a lot of links which do not belong to the topic *Astronomy*, such as *Ancient Egypt*, *Middle Kingdom*, *Nile*, *Hawaii*, *Marquesas Islands* and *New Zealand*.
Since the second data set includes overlapping topics, a lot of articles are clustered wrongly using the content-based clustering approach (Figure 6.42). A lot of articles of the topic *Computer Programming* are assigned to the topic *Artificial Intelligence*. Because the articles are very similar, the cluster quality of content-based clustering is not high. While using link-based clustering (Figure 6.43), only one article is clustered wrongly. The article *Speech recognition* is assigned to the topic *Computer Programming*. A look into the article text shows that the article contains only a few links to the topic *Artificial Intelligence*.
Data set III includes the most similar topics. In this data set the medical topics *Genetics*, *Immunology*, *Toxicology* and *Medical* have to be clustered. For this reason the content-based clustering is very poor (Figure 6.44). The clusters of *Medical* and *Toxicology* includes many articles which belong to other topics. Although the topics are very similar, the third data set was clustered optimal while using link-based clustering (Figure 6.45).
The last data set includes five small topics. Using content-based clustering (Figure 6.46) only two clusters are correct (*Computer Programming* and *Tourism*). The other clusters represent topics which are similar based on the content (*Genetics*, *Toxicology*, *Astronomy* and *Oceanography*). Consequently the articles are grouped wrongly. For example the article *Gene family* is assigned to the cluster of the topic *Astronomy*, because the article is very short (74 terms). Using link-based clustering (Figure 6.47) only one article was clustered wrongly. The article *Tour Guide* is assigned to the topic *Genetics*. The article is very short (101 terms) with only a few links (26 links). The Wiki-link network shows that there are no links between the article and the other articles of this topic. It has the appearance that there are no dependencies via third articles as well.
In summary the link-based clustering approach reflects the topics of a data set better. But the quality of link-based clustering depends on the quality of the linkage. The weakness of link-based clustering are small articles with only a few links. Articles with a large amount of links can be a problem as well. If an article has a lot of links, it becomes similar to content-based clustering. Furthermore there are articles which belong to more than one topic, such as *Earth*. The use of overlap clustering could solve this problem (described in Section 4.3.4, p. 63). Another problem are links to articles where titles consist of several words, such as *DNA sequence*, *Chronic toxicology* and *Acute toxicology*. In the actual link-based approach *Chronic toxicology* and *Acute toxicology* are different terms. It is possible that the splitting of the titles, called Tokenization (described in Section 4.3.1, p. 36), into terms improves the cluster quality. But the splitting of the titles can worsen the results as well. For example the splitting of the article title *Gene family* causes that now the word *family* belongs to an article about genetic.

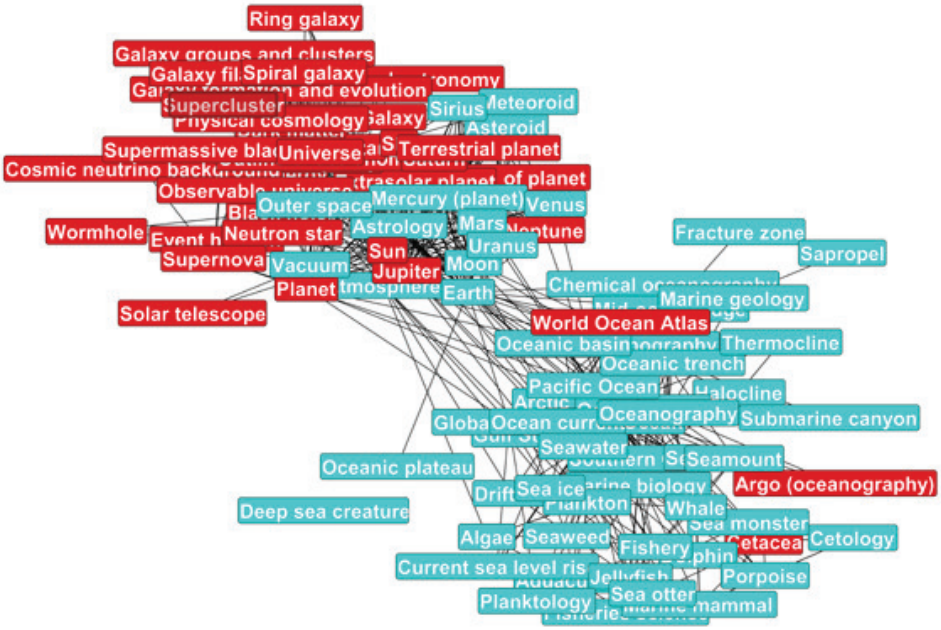


**Figure 6.40.:** Wiki-link network of data set I (PAM*, content-based clustering)

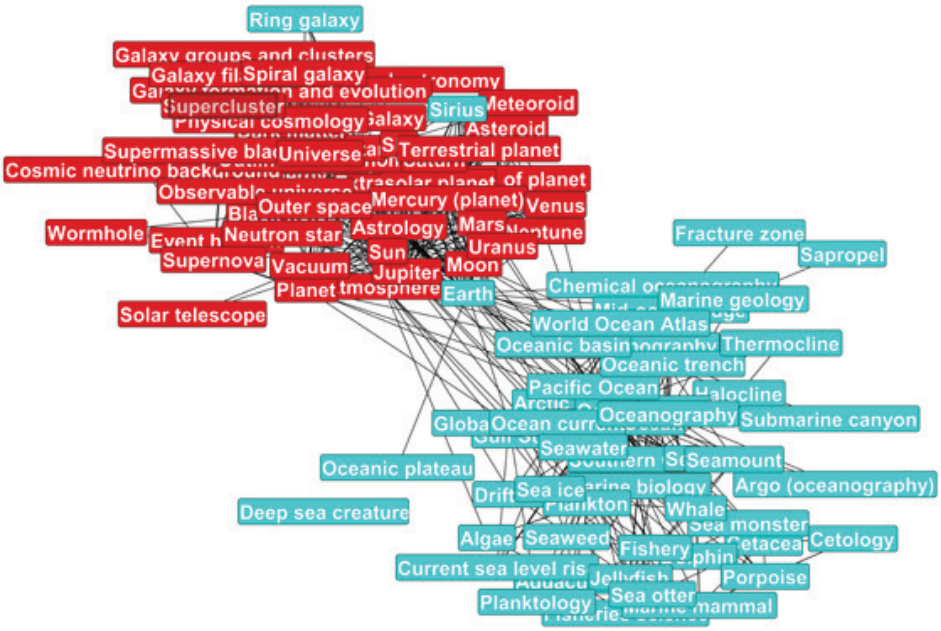


**Figure 6.41.:** Wiki-link network of data set I (Agnes*, link-based clustering)

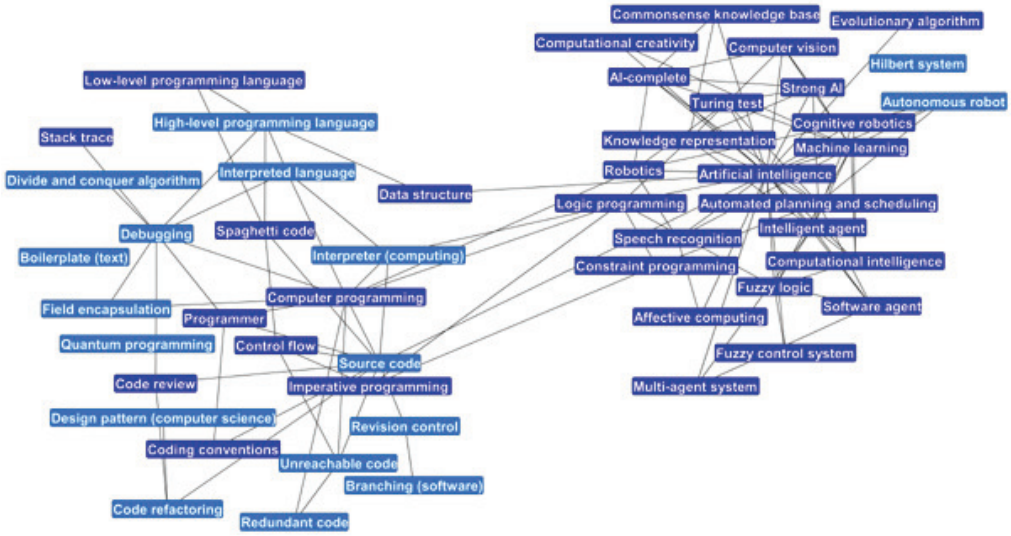


**Figure 6.42.:** Wiki-link network of data set II (PAM*, content-based clustering)

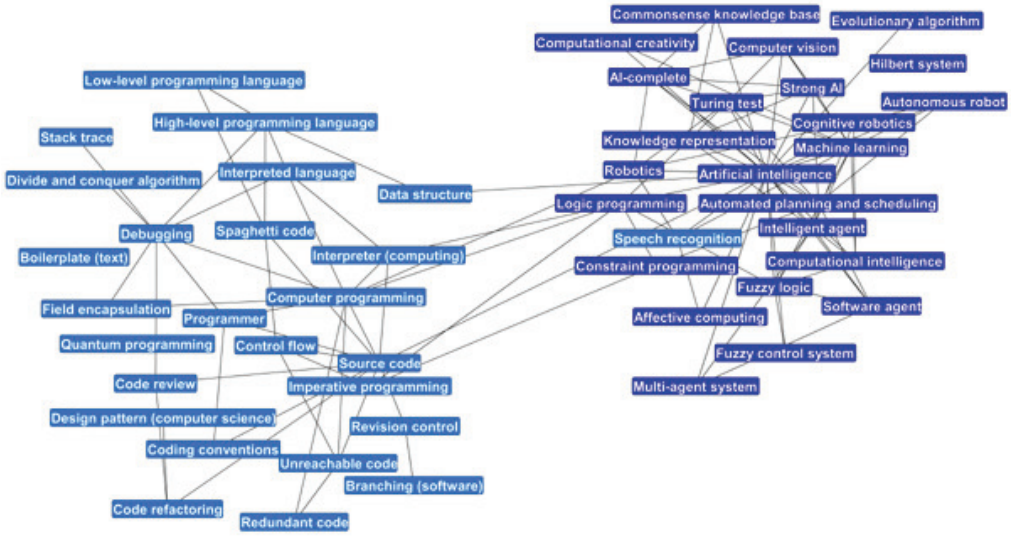


**Figure 6.43.:** Wiki-link network of data set II (Agnes*, link-based clustering)

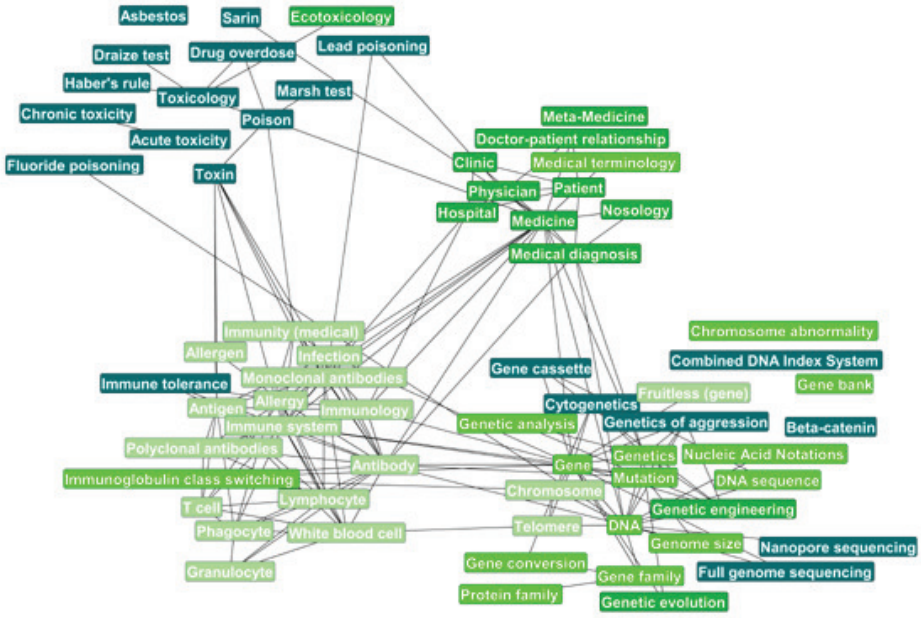


**Figure 6.44.:** Wiki-link network of data set III (PAM*, content-based clustering)

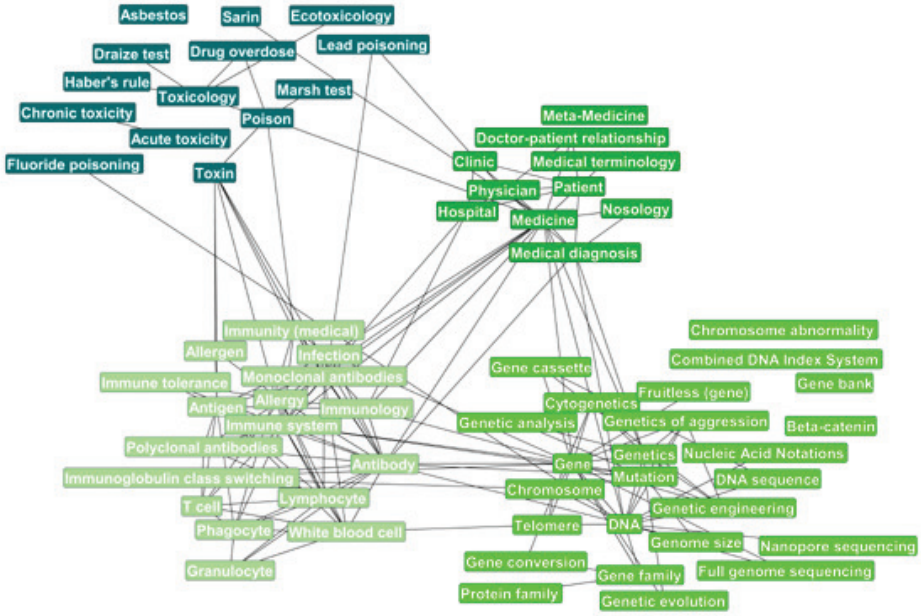


**Figure 6.45.:** Wiki-link network of data set III (Agnes*, link-based clustering)

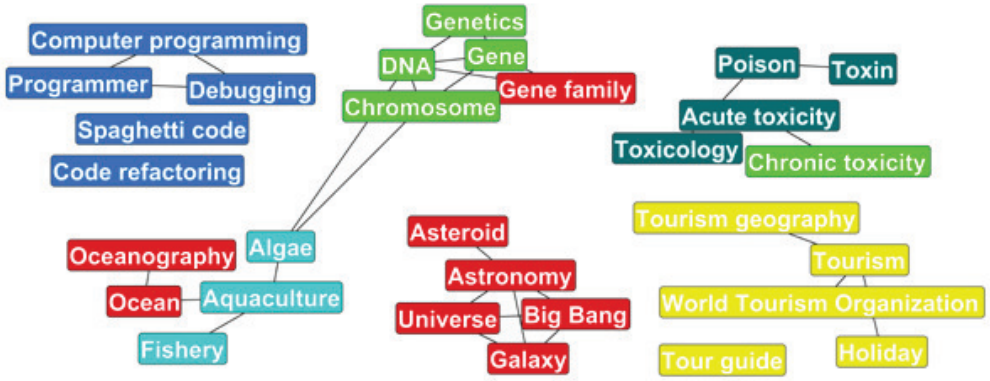


**Figure 6.46.:** Wiki Link Network of Data Set IV (PAM*, content-based clustering)

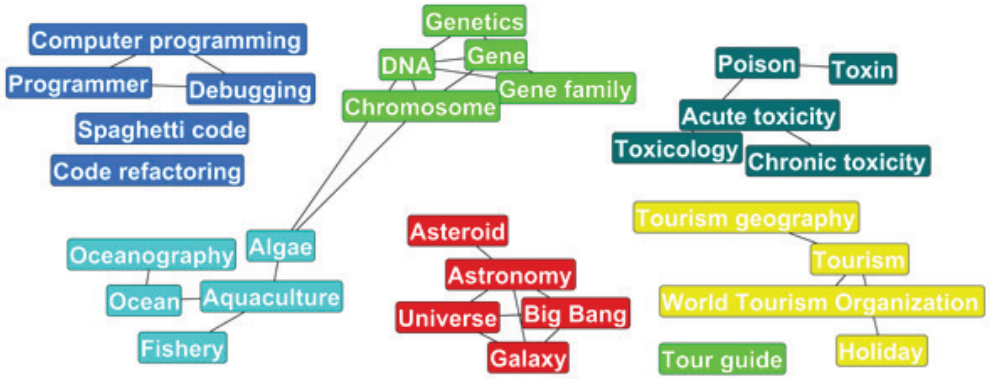


**Figure 6.47.:** Wiki-link network of data set IV (Agnes*, link-based clustering)

### 6.4.6. Summary

In this section the link-based clustering approach is compared with the content-based clustering approach while using dimension reduction techniques. The analysis shows that the external quality (F-measure) and performance (computing time) of the clustering depend on the clustering algorithm, the dimension reduction methods, the clustering approach and the structure of the data (similarity of the topics, size of the articles, size of the topics).
Like in the first section (Section 6.1, p. 98), the credibility of the internal quality results is criticized, because the results of the internal quality analysis (Dunn Index and SD Validity Index) are different to the results of the external quality analysis (F-measure). The results of the Dunn Index are similar to that of the F-measures, but the results of the SD Validity Index are contrary. The reasons for these differences are analyzed as well. The Dunn Index and

SD Validity Index were compared with the optimal clustering solution and the comparison suggested that the data structure of the content-based clustering approach is not suitable. The optimal solutions have a lower Dunn index and SD Validity index than the computed solutions. Because of this, the internal structure causes a wrong clustering concerning the topics of the data sets.

| | **F-measure** $FM$ | **Computing time [s]** $CT$ | **Clustering quality** $(\frac{1}{1+CT^{0.25}})^{0.5} \cdot FM^{0.5}$ |
|---|---|---|---|
| ***Content-based Clustering*** | | | |
| PAM | 0.643 | 0.730 | 0.578 |
| PAM* | 0.807 | 2.349 | 0.600 |
| Diana | 0.766 | 0.737 | 0.631 |
| Diana* | 0.784 | 1.130 | 0.621 |
| ***Link-based Clustering*** | | | |
| PAM | 0.711 | 0.578 | 0.616 |
| PAM* | 0.908 | 1.910 | 0.646 |
| Agnes | 0.866 | 0.569 | 0.681 |
| Agnes* | 0.979 | 1.132 | 0.694 |

**Table 6.5.:** Comparing the clustering algorithms
*(* = clustering is optimized using dimension reduction)*

The measurements of this analysis phase show that dimension reduction improves the external quality of the clustering. But the definition of suitable dimension reduction parameters is not easy, because this depends on the structure of the data (data set and clustering approach). The parameters can be chosen such that the results of some data sets are improved, but at the same time the results of other data sets get worse. Furthermore the use of dimension reduction techniques causes a deterioration of the performance, because the computing time increases. Table 6.5 evaluates the clustering configurations based on the external quality analysis (F-measure) and the performance analysis (computing time). Thereby the clustering algorithm with and without dimension reduction are listed. The computing time in seconds is combined with the F-measure value. The combination of the measures shows that dimension reduction sometimes doesn't enhance the clustering quality, because the increase of the computing time is very high (see Diana with content-based clustering). However the computing time is not high in absolute terms. Nevertheless the influence of the computing time for larger data sets should be examined, because the computing time depends on the size of the data set.
The Table 6.5 clarify further that link-based clustering (with and without dimension reduction) has a better clustering quality than content-based clustering. Also the internal quality based on the Dunn Index is higher. The term document matrix of the link-based clustering approach is smaller and the computations need less time. Consequently the term document matrix has to be included the more important terms of the articles and the terms which reflect the topics more precisely. In spite of there are the use of dimension reduction methods, content-based clustering has still a problem with the clustering of similar topics. Thus the dimension reduction methods removed not enough unimportant terms. The best clustering quality is achieved with the link-based clustering approach in combination with Agnes and a

feature selection of 90%.
Although link-based clustering obtains adequate results, the optimal clustering solutions are not achieved. The quality of link-based clustering depends on the quality of the linkage. Young or small articles, articles whose content do not correlate to the article title and general articles may cause a wrong clustering. Because small articles include only few links (e.g. *Tour Guide*), not enough information exists for the clustering. Articles whose content do not correlate to the article title (e.g. *Sirius*) includes many links to other topics. General articles belong to more than one topic (e.g. *Earth*). Consequently it is difficult to assign these articles to one topic. The use of an overlap clustering approach could be helpful (described in Section 4.3.4, p. 63). Another problem are article titles which have more than one word, such as *DNA sequence*. The use of Tokenization (described in Section 4.3.1, p. 36), which splits the title into words could solve the problem.

# 7. Conclusion and Outlook

The objective of this work is to identify a method which allows the clustering of content elements in topics. In this context, two approaches were compared: the link-based clustering approach using the outgoing links of content elements, and the content-based approach using the terms of content elements. The objective of this work is summarized in a research question and a hypothesis about the advantages and disadvantages of these two approaches (see Chapter 2, p. 9). Wikis are used as example of social information space for the analyses. In this context selected Wiki articles were summarized to data sets.
The data sets were transformed into a generic data model first. Based on this data model, analyses which are independent from the social information space are possible. The data model is part of the SONIVIS tool. An explanation of the generic data model is given in Chapter 3 (p. 13).
After defining the data sets for the analyses, an evaluation of possible analysis methods is conducted in Chapter 4 (p. 23). In this chapter different text analysis methods are described. The base for topic detection is the clustering of the content elements. The resulted clusters represents the topics. Suitable clustering algorithms are selected. Furthermore it is possible to enhance results with the use of dimension reduction methods. Thus possible dimension reduction methods are selected as well. The selected analysis methods for topic detection are implemented as a metric into the SONIVIS tool. The metric returns the cluster ID for each content element. Furthermore the quality of the clusters is calculated and reported.
Based on the selected data sets and the implemented metric, the analyses are executed. An overview about the data sets and the methodology is given in Chapter 5 (p. 87). The results are evaluated with an external quality analysis based on the F-measure, an internal quality analysis based on the Dunn Index and SD Validity Index and a performance analysis based on the computing time. The results are described in Chapter 6 (p. 97). The analyses are separated into four phases. The first clustering phase detects suitable clustering algorithms and compares the link-based and content-based clustering approach without dimension reduction techniques. The clustering algorithms PAM, bisecting PAM, Agnes and Diana (described in Section 4.3.4, p. 63) are used. The next two phases search for suitable dimension reduction parameters. The feature selection method term weight thresholding (described in Section 4.3.3, p. 60) and the feature transformation method latent semantic analysis (described in Section 4.3.3, p. 48) are used. In the last phase the link-based and content-based clustering approach are compared with the use of dimension reduction techniques and the improvements through the use of dimension reduction methods are evaluated.

The analyses show that the clustering quality depends on the clustering algorithm, the dimension reduction methods, the selected clustering approach and the structure of the data set (similarity of the topics, size of the articles, size of the topics, etc.). The best results are achieved by using link-based clustering. The credibility of the internal quality results is criticized, because the results of the internal quality analysis (Dunn Index and SD Validity Index) are different from the results of the external quality analysis (F-measure).
The results of the first clustering phase (Section 6.1, p. 98) show that link-based clustering

achieves better results than content-based clustering, if no dimension reduction is used. The results of the content-based approach correspond to the works from Zhao et al. [243] and Steinbach et al. [204]. Zhao et al. compare divisive and agglomerative clustering algorithm and discover that divide clustering are better suitable for text clustering. Following in this work, the divise clustering algorithm Diana is better for content-based clustering than the agglomerative clustering algorithm Agnes. Steinbach et al. compare agglomerative hierarchical clustering, k-means and bisecting k-means. The results show that standard k-means generates better clusters than agglomerative hierarchical clustering, and bisecting k-means is the best clustering technique. The result of this work shows that PAM, which is a variant of k-means, is better than the agglomerative clustering algorithm Agnes. Only the results of bisecting PAM are different to the work of Steinbach et al.. In this work, bisecting PAM produces the least efficiency concerning the external cluster quality (F-measure) and performance (computing time).
The results of link-based clustering are different. The difference between the clustering algorithms Agnes and Diana, corresponding to the link-based and the content-based clustering approaches is surprising. Agglomerative hierarchical clustering gets better results than divise hierarchical clustering while using link-based clustering. This fact indicates how different the underlying data structure is. The term document matrix which is based on the terms of a text is completely different when compared to the term document matrix which is based on the links of a text. The differences between the data structure of link-based and content-based clustering are additionally emphasized by the Dunn Index and SD Validity Index. The question arises, whether Agnes gets also good results by other data sets.
The next two clustering phases (Section 6.2, p. 105 and Section 6.3, p. 109) defined suitable parameters for the dimension reduction methods. The defined values are only appropriate for the selected data sets and clustering methods. For the use of other data sets and clustering methods, new parameters have to be determined. For further analyses and for the utilization of the topic detection methods in practice, other approaches for defining the dimension reduction parameters have to be used, because the selected approaches are very extensive and the change of the data set causes a re-defining of the parameters. The feature selection based on term weight thresholding works with percentage thresholds. The use of percentage thresholds is more suitable than the use of absolute thresholds. But based on the content-based clustering, the defined thresholds are different. Using link-based clustering a feature selection of 90% seems to be suitable. But the validity of this value have to be proven with further data sets. The definition of an appropriate value (number of dimensions) for feature transformation based on latent semantic analysis is more complex. In this work a value for each clustering approach, clustering algorithm and data set was defined. The parameter values for feature transformation differs, thus no default value can be defined. Unfortunately, based on the plot of the singular values no parameter value could be defined. Maybe the definition of a percentage value can be helpful.
In the last clustering phase (Section 6.4, p. 115) the improvement by the use of dimension reduction methods is evaluated and the link-based and content-based clustering approach are compared. In summary, dimension reduction enhances the cluster quality, but the computing time increases as well. Furthermore the parameters may cause that the results of some data sets are improved, but the results of other data sets are worsen. Consequently it will have to be weighed out which methods are really meaningful. The results correspond to the works from Tang et al. [209] [210]. Tang et al. compare latent semantic analysis, document frequency thresholding, mean TFIDF and further dimension reduction methods. Document frequency thresholding and mean TFIDF are similar to term weight thresholding. The mea-

surements show that latent semantic analysis, document frequency thresholding and mean TFIDF are very suitable dimension reduction methods for document clustering. The best dimension reduction method in this work is independent component analysis. Feature selection seems to be a better method for dimension reduction in the context of topic clustering. With a threshold of 90% the link-based clustering performs well and it is ensured that hardly any of the important terms will be deleted. Furthermore the execution of term weight thresholding isn't complex, because no further calculations have to be executed. Only the term weights which are smaller than the defined threshold have to be deleted. By contrast latent semantic analysis is more complex and has the problem of defining the best final number of dimensions. It should be examined in further analyses whether latent semantic analysis is a good method for topic clustering. Because independent component analysis seems to be a better feature transformation method, the quality of independent component analysis should be proven. Actually feature selection is executed first, followed from feature transformation. The quality of clustering should be proven by changing this order.
The comparison of link-based and content-based clustering shows that link-based clustering is more suitable for the detection of topics. The quality of the clustering is higher (external quality analysis) and the computing time is less (performance analysis). Furthermore link-based clustering only needs feature selection. Consequently the term document matrix of the link-based clustering approach is smaller and includes more of the important terms for topic detection. Even if content-based clustering uses dimension reduction methods, this approach still has a problem with the clustering of similar topics. The best results are achieved with link-based clustering combined with the agglomerative hierarchical clustering algorithm Agnes and a term weight threshold of 90%. PAM has good results as well, but needs additional feature transformation. Although link-based clustering obtains adequate results, the optimal clustering solutions are not achieved. The quality of link-based clustering depends on the quality of the linkage. Problems evolve from young or small articles, articles whose content do not correlate to the article title and general articles. These are articles with few links or links to other topics. The assignment of general articles to a topic is difficult, because a general article belongs to more than one topic. The use of an overlap clustering approach could be helpful.

Based on the analyses the research question of Chapter 2 (p. 9) can be answered. The results of this work show that the clusters of the link-based clustering approach reflects the topics better than the content-based clustering approach. Even the clustering of topics with a high similarity achieve good results. Consequently, the defined hypothesis is correct. Using the link-based approach, the important terms of an article are considered. The external cluster quality and the performance are better using the link-based clustering approach. In addition a preprocessing of the data (tokenization, stop word removal, stemming) is not required. The content-based clustering approach does not achieve the cluster quality of the link-based clustering. Although useless terms are removed by the stop word removal, a lot of unimportant terms still exists. This causes a noise in the data and the clustering worsen. A problem which arises is that the link-based clustering approach depends on the quality of the articles and thus on the quality of the linkage. Since the linkage of an article is made by users, the important terms are defined subjectively. This can lead to problems, as it is shown in the analyses.
The validation of the results should be proven with other data sets. For example the link-based clustering should be evaluated again with analyses on very large data sets including a lot of topics. The clustering algorithm Agnes achieves the best overall results, but the complexity of this algorithm is high. Consequently a more suitable clustering algorithm has to

be found. Maybe the clustering algorithm PAM achieves better results than the clustering algorithm Agnes using very large data sets. On the other side, further clustering algorithm and dimension reduction methods should be compared with the actual results. For example the independent component analysis, which has the best results in the work from Tang et al. [209] [210], should be implemented and compared with the actual methods. Another possibility is the change of the order of the dimension reduction methods. In these analyses, feature selection is executed before feature transformation.
Since the selection of suitable dimension reduction parameters is very complex in this work, better approaches have to be found. In addition the methods for the evaluation of the clustering results should be enhanced. This will be very important, if no better approach for the definition of appropriate dimension reduction parameters can be found. The external quality analysis requires the optimal clustering solution. This is unapplicable in practice. The internal quality analysis is applicable in practice, but the measures actually used (Dunn Index, SD Validity Index) produce wrong results. The performance analysis is suitable for the evaluation. But the performance analysis doesn't evaluate the correctness of the results. Consequently further methods for the evaluation of the cluster quality have to be found.
Since the optimal solution was not found with the link-based clustering as well, the topic detection has to be further improved. There are two approaches for improving the results: The enhancement of the methods and the enhancement of the underlying data.
The methods can be enhanced by the use of overlapping clusters to solve the problem with general articles (described in Section 4.3.4, p. 63). As described, further methods for clustering and dimension reduction, which are better than the actual methods, can be searched. Another possibility to improve the clustering is the tokenization of the article titles (described in Section 4.3.1, p. 36). In this case, the words of articles like *DNA sequence* and *Chronic toxicology* are separate terms for the clustering. The approaches can be enhanced by using NLP methods. The resolution of synonyms (synonym detection, described in Section 4.3.1, p. 39) and homonyms (word-sense disambiguation, described in Section 4.3.1, p. 39) and the identification of important word groups and phrases (multiword phrase grouping, described in Section 4.3.1, p. 39) are possibilities for extending the clustering. The extraction of terms can be extended by applying syntactical and semantical analyses such as Part-Of-Speech tagging (described in Section 4.3.1, p. 40) or parsing (described in Section 4.3.1, p. 39). With these methods the words are classified into nouns, verbs, adjectives, adverbs, conjunctions, foreign words, etc.
Furthermore it is possible to improve the underlying data. For example, the link-based clustering could consider the frequency of links as well. Maybe the extraction of subtitles from the text can enhance the clustering quality. Actually only the outgoing links of a content element are considered. The use of ingoing links or the use of the intersection of outgoing and ingoing links could improve the results. A combination of both clustering approaches seems to be a very interesting possibility for topic clustering. One possibility is, that only terms which are links will be considered (link-based), but the frequency of the link-term is the absolute frequency of the term in the text (content-based). Another possibility combines both approaches in another way. First a threshold has to be determined. This threshold defines the number of links, which a text should have in order to be included. If a text has fewer links, the content-based approach will be used. In the other case, only the links will be used for describing the article.
Another very important extension for the detection of topics is the definition of terms which describe each topic cluster. A possible approach is given by Siroker et al. [198].

The defined methods for topic detection can used for many Wiki analyses. It is possible to analyze Wikis on topic area, e.g. analyses of the author activity in or between topics or the identification of the main topics. The analysis of the author activity in a topic is helpful by the identification of topic specialists. By considering the development of a Wiki, the detection of new topics and their development is possible. Thus trends concerning the content development can be identified. Many Wikis are very large and therefore the analysis can be complex and confusing. The analysis on topic area reduces the complexity. The validity of the Wiki categorization can be proven as well. Furthermore it is possible to evaluate the validity of the Wiki-links. In this context, the articles of a topic cluster have to be linked.
One of the next steps could be the implementation of the described ideas for improvements. Furthermore it can be proven whether the link-based clustering approach achieved good results in other information spaces (blogs, news articles, online forums, etc.) as well. If this is the case, analyses on other social information spaces can be executed. Otherwise, the content-based clustering will have to be used or another possibility for topic detection may have to be developed. Since the methods are applied on a general data model, no changes at the implementation are necessary. Actually the implemented metric is applied on content elements. But it is possible to apply the clustering metric on actors as well. The results are clusters with actors, who work on same topics. Based on this approach, the identification of communities is possible.
The developed topic detection approaches can be used in practice as well. For example, for the detection of business topics in news or the detection of sentiments in social information spaces like Twitter. The main sentiments about movies or something else can be identified by clustering the sentiments based on their content. The detection of trends is a further practical use. In this context, the main topics in news, blogs etc. are identified for different periods of time. The results can be used for evaluating the influence of specific events and thus for marketing and public relation strategies.

# Part IV.

# Appendixes

# A. List of Abbreviations

AI ..... Artificial Intelligence
BE ..... Backward Elimination
BIC ..... Bayesian Information Criterion
BIRCH ..... Balanced Iterative Reducing and Clustering using Hierarchies
BSC ..... Bipartite spectral co-clustering
CACTUS ..... Categorical Clustering Using Summaries
CCA ..... Curvilinear Component Analysis
CDA ..... Curvilinear Distance Analysis
CFA ..... Confirmatory Factor Analysis
CL ..... Computational Linguistics
CLINK ..... Complete-link Hierarchical Clustering
CLTree ..... Clustering based on Decision Trees
CU ..... Category Utility
CURE ..... Clustering using Representatives
DA ..... Deterministic Annealing
DBSCAN ..... Density-Based Spatial Clustering of Applications with Noise
DBCLASD ..... Distribution-Based Clustering of Large Spatial Databases
DENCLUE ..... Density based Clustering
DF ..... Document Frequency
DIANA ..... Divisive Analysis
DM ..... Data Mining
DOC ..... Density-based Optimal Projective Clustering
DR ..... Dimension Reduction
EFA ..... Exploratory Factor Analysis
EM ..... Expectation-Maximization
ENCLUS ..... Entropy-based Clustering
ESA ..... Explicit Semantic Analysis
FDA ..... Fisher Discriminant Analysis
FS ..... Forward Selection
FSC ..... Fuzzy Subspace Clustering
FSSEM ..... Feature Subset Selection using EM clustering
GA ..... Genetic Search uses a genetic algorithm
GDILC ..... Grid-based Density-IsoLine Clustering
GP ..... Graph Partitioning
GTM ..... Generative Topographic Map
HMM ..... Hidden Markov Model
ICA ..... Independent Component Analysis
IE ..... Information Extraction
IG ..... Information Gain
InfoSpace ..... Social Information Space
IR ..... Information Retrieval

ISOMAP ..... Isometric Feature Mapping
KDA ..... Kernel Discriminant Analysis
KDD ..... Knowledge Discovery in Databases
KEA ..... Keyword Extraction Algorithm
LDA ..... Linear Discriminant Analysis
LDM ..... Lafon's Diffusion Maps
LLE ..... Local Linear Embedding
LS ..... Laplacian Score
LSA ..... Latent Semantic Analys60is
LSI ..... Latent Semantic Indexing
MAFIA ..... Merging of Adaptive Finite Intervals
mBM ..... Multivariate Bernoulli model
MDS ..... Multidimensional Scaling
MI ..... Mutual Information
mNM ..... Multinomial model
MWE ..... Multiword Expressions
NLP ..... Natural Language Processing
NMF ..... Non-negative Matrix Factorisation
OR ..... Odds Ratio
PART ..... Projective Adaptive Resonance Theory
PCA ..... Principal Component Analysis
PDDP ..... Principal Direction Divisive Partitioning
POS ..... Part-of-Speech
PPCA ..... Probabilistic Principal Component Analysis
PROCLUS ..... Projected Clusting
RM ..... Random Mapping
RSP ..... Repeated-String-Patterns
ROCK ..... Robust Clustering using Links
RP ..... Random Projection
SA ..... Simulated Annealing
SARS ..... Simulated Annealing using Random Sampling
SIS ..... Social Information Space
SLINK ..... Single-link Hierarchical Clustering
STING ..... Statistical Information Grid-based Clustering Method
SM ..... Semantic Mapping
SNA ..... Social Network Analysis
SOM ..... Self-Organizing Maps
SRM ..... Sparse Random Mapping
STC ..... Suffix Tree Clustering
SUBCAD ..... Subspace Clustering for High-dimensional Categorical Data
SVD ..... Singular Value Decomposition
TfV ..... Term Frequency Variance
TM ..... Text Mining
TS ..... Term Strength
TW ..... Term Weight
ORCLUS ..... Arbitrarily Oriented Projected Cluster Generation
vMF ..... von Mises-Fisher
VSM ..... Vector Space Model

VNS ..... Variable Neighborhood Search
WBSC ..... Word-based Soft Clustering
WSD ..... Word-Sense Disambiguation

# Glossary

**Anaphora resolution** Anaphora resolution is the process of resolving indirect references by using noun phrase, like *he* or *herself*., 39

**Antonym** Different words with an opposite meaning, such as *short* and *tall*., 27

**Artificial Intelligence** It is a research field devoted to developing programs that enable computers to display behavior that can (broadly) be characterized as intelligent., 83

**Bigram Proximity Matrix** Encodes a document as a matrix. The element $x_{ik}$ of the matrix is the number of times the i-th term appears before the j-th term., 45

**Center-based Clustering** By this algorithms each cluster is represented by a center., 63

**Chinese Whispers** A randomized graph-clustering algorithm which considered the neighbors for each node., 69

**Classification** Classification is a technology for grouping objects into categories., 75

**Clustering** Clustering group similar objects into clusters. Thus there are a high similarity between objects of the same group (homogeneity within the cluster) and a low similarity between objects of different groups (heterogeneity between the clusters)., 63

**Co-occurrence** These are words which often occur together in a text, such as *Los - Angeles*, *hard - disk* and *police - arrested*., 27

**Collocation** An expression of two or more words, which make sense together, such as *zero tolerance*., 27

**Coreference Resolution** Is the process of locating noun phrases which refer to a entity. The noun phrases describes the entity., 40

**Corpus** A collection of documents, e.g. articles, newspapers, web pages and eMails., 27

**Covariance matrix** A matrix including the correlations between variables., 51

**Data Mining** It is a research field to discover patterns and relationships in data that may be used to make valid predictions., 82

**Denoising** Denoising removes the noise from the data while preserving useful information., 48

**Density-based Clustering** Density-based Clustering uses connectivity and density functions for clustering. The resulted clusters are dense areas seperated by low-density areas., 63

**Dictionary** see Lexicon, 27

**Disjunct Clustering** Using Disjunct Clustering each object will be assigned to one cluster., 63

**Dissimilarity matrix** Based on a term document matrix it is possible to derive a dissimilarity matrix. Each column and row represents a document or term and the values describe the pairwise dissimilarity of the documents or terms., 44

**Entity Extraction** Is the process of locating and classifying elements in text, such as names of persons, organizations, locations, dates, times, etc., 40

**Fuzzy Clustering** Fuzzy Clustering extends the clustering to Overlap Clustering. Each data point is associated with every cluster using a membership function., 63

**Glossary** A list of difficult, technical or specialized terms with their definitions., 27

**Grammar** Grammar defines the structure of words (morphology) and sentences (syntax)., 27

**Graph-based Clustering** Graph-based Clustering apply clustering on a graph to partition them., 63

**Grid-based Clustering** Grid-based Clustering algorithms work with the value space of the data objects and not with the data objects like other clustering algorithms. By this approach a grid structure will be created and used for the clustering., 63

**Hidden Markov model** Statistical model for modeling a system with unobserved state., 29

**Hierarchical Clustering** Hierarchical clustering creates a hierarchy of clusters which can be represented in a tree diagram, called dendrogram., 63

**Hierarchical Clustering** This clustering techniques creates a hierarchy of clusters. The hierarchy can be represented in a dendrogram., 66

**Homonym** A word that has different meanings, such as *bank*., 27

**Indexing** Is the automatic extraction/detection of keywords in a text., 75

**Information Extraction** A research field which study the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources., 77

**Information Retrieval** A research field which study the finding of documents that satisfy some search criteria., 79

**K-means** A center-based clustering algorithm, at which each cluster is represented by a centroid and the objects of a cluster have a minimal distance to the centroid., 68

**Keyword** Significant or descriptive terms, which describes a document very well, 27

**Keyword Extraction** Is the automatic extraction/detection of keywords in a text., 75

**Latent Semantic Analysis** Revealing of the semantic information from a corpus by the transformation into a semantic space using singular value decomposition and denoising., 48

**Latent Semantic Indexing** See: Latent Semantic Analysis, 48

**Lemmatization** Lemmatization transform a word to its basic form, is called the lemma., 38

**Lexeme** A set of words that are different forms of the same base word., 27

**Lexicon** The set of all unique terms in the corpus., 27

**Linguistic** The science of the nature and structure of natural language, including phonetics, phonology, morphology, syntax and the fields of semantics., 27

**Mandelbrot-Zipfs law** Enhancement of Zipfs law for describing the dependency between rank and frequency., 29

**Matrix diagonalization** The decomposition of a square matrix into eigenvalues and eigenvectors., 33

**MediaWiki** MediaWiki is a Wiki implemenation. It is a open source software developing from the Wikimedia Foundation., 18

**Model-based Clustering** Model-based Clustering uses certain models for clusters and tries to find the best fit between the data set and the models., 63

**Morpheme** The smallest meaningful unit in a language., 27

**Morphology** Morphology is the study of the composition of words., 27

**Multidimensional Scaling** It transforms a dissimilarity matrix into a lower-dimensional representation whereby the relationships should be maintained. The method will be used frequently for visualization and approximation., 55

**Multiword Phrase Grouping** The detection and extraction of multiword expressions, like *add up*, *telephone box* or *Los Angeles*., 38

**N-gram model** Probabilistic model for modeling sequences of events., 29

**Natural Language Processing** It is a research field between linguistics and computer science which is concerned with the computational aspects of the human language., 79

**Ontology** An ontology is a formalization of a common understanding of words for an application or subject., 27

**Overlap Clustering** Using Overlap Clustering each object can be assigned to more than one cluster., 63

**Parsing** The syntactic analysis of the structure of a sentence according to a certain grammar theory. This allows the extraction of the subject, object, etc., 39

**Part-of-Speech Tagging** Part-of-Speech Tagging (POS-Tagging) is the grammatical tagging of a text. It determines the part of speech of each word., 40

**Phrase** A group of two or more linked words that functions as a single unit in the syntax of a sentence., 27

**Polysem** A word that has different meanings but they are similar, such as *wood*., 27

**Principal Components Analysis** Reduces the dimensions of a data set into a lower dimensional space, in which the main part of the data variance is. The lower dimensional space is defined by eigenvectors, called principal components., 51

**Reference corpus** A collection of documents for comparative purposes., 27

**Relation Extraction** Is the process of locating and classifying of relationships between entities. A relation can be defined over two or more entities, e.g. *is employee of* or *capital of*., 40

**Search-based Clustering** Compared to many other clustering algorithm, like fuzzy k-means, fuzzy c-means and hill climbing, Search-based Clustering algorithms find a globally optimal solution by exploring the solution space., 63

**Semantics** The study of meanings of words, phrases, sentences and texts. It is important for understanding language in social contexts., 27

**Sentence** The largest independent unit of grammar, that expresses a complete idea. Conventionally, a sentence includes a subject and a verb. It begins with a capital letter and concludes with an end mark, like a period, an exclamation point or a question mark., 27

**Significant Co-occurrences** Significant co-occurrences are terms those arise very often together., 75

**Singular Value Decomposition** Factors a matrix $A$ into 3 matrices $A = TSD^T$, which allow a simple matrix approximation for example., 33

**Stemming** The reduction of words to its root or stem by removing suffixes and prefixes., 37

**Stop word removal** The removal of stop words from a text document., 36

**Stop words** Words, which do not contribute meaningful content to a text document, like *the, and, but, or*., 36

**Subspace Clustering** By this approach the clusters are embedded in subspaces of the data set. The subspaces have their own dimensions. Compared to clustering with dimension reduction no information will be lost., 63

**Synonym** Different words with an identical or similar meaning, such as *bucket* and *pail*., 27

**Synonym Detection** Is the process of the detection of synonymous terms and the following conflation., 39

**Syntax** The rules of syntax describe, how words in phrases and sentences may be combined., 27

**Table Model** Model for saving the terms of documents in tables, e.g. database tables., 43

**Taxonomy** Hierarchical structure of words. This type of classification points out similarities between words., 27

**Technical Terminology** All technical terms of the field, which are generally usual., 27

**Term** A word or a phrase., 27

**Term document matrix** The documents of a corpus can be saved in a term document matrix. In this matrix each row represents a term and each column represents a document. An element of the matrix contains the weight of the i-th term in the j-th document., 44

**Term Weighting** Is the weighting of a term by a specific function., 42

**Text Chunking** Text chunking is the dividing of sentences into non-overlapping phrases., 40

**Text Mining** Is a natural language extension of the data mining research field for discovering of new, previously unknown information., 80

**Text Summarization** A method for shortening/summarizing of text, e.g. the automatic creation of an abstract., 75

**Thesaurus** A thesaurus is a dictionary which contains the semantic relations for words, e.g. synonyms, sometimes antonyms and contrasting words., 27

**Token** see Term, 27

**Tokenization** The process of splitting a text document into meaningful units., 36

**Topic Detection** The extraction/detection of topics in a document or a corpus., 75

**Topic map** A collection of topics, their relations and information sources, which are relevant for the topics. It defines a model of knowledge structures. In this model, the knowledge of a topic are collected., 27

**Vector Space Model** The vector space model allows the presentation of documents and search queries as vectors., 44

**Visualization** Methods for a visual representation of information, data or knowledge., 75

**Wiki** A Wiki is a website which can be read and edited by everyone. There is no pre-defined structure. The structure will be build by a collaborative process of the users., 18

**Wikipedia** A Wikipedia Wiki is an internet encyclopedia which is generating cooperatively by many users., 18

**Word** The combination of morphemes., 27

**Word-Sense Disambiguation** Word Sense Disambiguation (WSD) tries to resolve the ambiguity of terms., 39

**Zipfs law** Defines that the frequency of any word is inversely proportional to its rank., 29

# B. List of Figures


1.1 Overview of this work . . . . . 8

3.1 UML class diagram of an universal actor content-element model . . . . . 14
3.2 Model of the social information space analysis . . . . . 15

4.1 Research fields . . . . . 26
4.2 Text analysis: fundamentals . . . . . 27
4.3 Structural model of language . . . . . 28
4.4 Zipfs Law and Mandelbrot-Zipfs Law *(Alice's Adventures in Wonderland)* . . 32
4.5 Text analysis: methods . . . . . 35
4.6 Important analysis methods for text mining . . . . . 36
4.7 Latent semantic analysis 2D-plot . . . . . 51
4.8 Principal components analysis 2D-plot . . . . . 55
4.9 Metric MDS . . . . . 58
4.10 Non-metric MDS . . . . . 58
4.11 Isometric feature mapping 2D-plot . . . . . 59
4.12 Example: hierarchical agglomerative clustering . . . . . 68
4.13 Example: Chinese whispers . . . . . 70
4.14 Word cloud example . . . . . 75
4.15 Text analysis: research fields and applications . . . . . 77
4.16 Information extraction process . . . . . 78
4.17 Information retrieval process . . . . . 79
4.18 Text mining process . . . . . 81
4.19 Data mining process . . . . . 82

5.1 Preprocessing the data sets . . . . . 91
5.2 Topic detection process for analyses . . . . . 95

6.1 F-measure: Content-based clustering . . . . . 98
6.2 F-measure: Link-based clustering . . . . . 98
6.3 F-measure: Clustering algorithm view . . . . . 99
6.4 F-measure: Data set view . . . . . 99
6.5 Dunn Index: Clustering algorithm view . . . . . 101
6.6 Dunn Index: Data set view . . . . . 101
6.7 SD Validity Index: Clustering algorithm view . . . . . 101
6.8 SD Validity Index: Data set view . . . . . 101
6.9 Computing Time: Clustering algorithm view . . . . . 102
6.10 Computing Time: Data set view . . . . . 102
6.11 F-measure: Content-based clustering, Diana . . . . . 106
6.12 F-measure: Content-based clustering, PAM . . . . . 106
6.13 F-measure: Link-based clustering, Agnes . . . . . 107

6.14 F-measure: Link-based clustering, PAM . . . . . 107
6.15 Singular Values: Content-based clustering, no feature selection . . . . . 110
6.16 Singular Values: Content-based clustering, 50% feature selection . . . . . 110
6.17 Singular Values: Link-based clustering, 90% feature selection . . . . . 110
6.18 F-measure: Content-based clustering, Diana . . . . . 112
6.19 F-measure: Content-based clustering, PAM . . . . . 112
6.20 F-measure: Link-based clustering, Agnes . . . . . 113
6.21 F-measure: Link-based clustering, PAM . . . . . 113
6.22 F-measure: Clustering improvement through dimension reduction . . . . . 116
6.23 Computing Time: Time increase through dimension reduction . . . . . 116
6.24 F-measure: Content-based clustering . . . . . 118
6.25 F-measure: Link-based clustering . . . . . 118
6.26 F-measure: Clustering algorithm view . . . . . 118
6.27 F-measure: Data set view . . . . . 118
6.28 Dunn Index: Clustering algorithm view . . . . . 120
6.29 Dunn Index: Data set view . . . . . 120
6.30 SD Validity Index: Clustering algorithm view . . . . . 120
6.31 SD Validity Index: Data set view . . . . . 120
6.32 Dunn Index: Content-based vs. optimal clustering . . . . . 121
6.33 Dunn Index: Link-based vs. optimal clustering . . . . . 121
6.34 SD Validity Index: Content-based vs. optimal clustering . . . . . 121
6.35 SD Validity Index: Link-based vs. optimal clustering . . . . . 121
6.36 Computing Time: Clustering algorithm view . . . . . 122
6.37 Computing Time: Data set view . . . . . 122
6.38 Wiki Link Network - Topics . . . . . 123
6.39 Wiki Link Network . . . . . 124
6.40 Wiki-link network of data set I (PAM*, content-based clustering) . . . . . 127
6.41 Wiki-link network of data set I (Agnes*, link-based clustering) . . . . . 127
6.42 Wiki-link network of data set II (PAM*, content-based clustering) . . . . . 128
6.43 Wiki-link network of data set II (Agnes*, link-based clustering) . . . . . 128
6.44 Wiki-link network of data set III (PAM*, content-based clustering) . . . . . 129
6.45 Wiki-link network of data set III (Agnes*, link-based clustering) . . . . . 129
6.46 Wiki-link network of data set IV (PAM*, content-based clustering) . . . . . 130
6.47 Wiki-link network of data set IV (Agnes*, link-based clustering) . . . . . 130

# C. List of Tables


3.1 Metrics for the analysis of networks . . . 16
3.2 Metrics for the analysis of social information spaces . . . 17
3.3 Metrics for the analysis of Wikis . . . 17
3.4 Size and activity of MediaWiki projects (2009/02/04) . . . 18
3.5 WikiMedia namespaces (2009/06/17) . . . 20
3.6 Extraction and Transformation of wiki data into the data model . . . 21

4.1 Publicity of research fields (estimated) . . . 23
4.2 Differences between research fields . . . 24
4.3 Perspectives of text mining definitions . . . 24
4.4 Rank table . . . 31
4.5 Example of a document table . . . 43
4.6 Example of a sentence table . . . 43
4.7 Example of a term table . . . 44
4.8 Example of a co-occurrence table . . . 44
4.9 Example of a document-sentence inverse table . . . 44
4.10 Example of a sentence-term inverse table . . . 44
4.11 Example of a term document matrix . . . 45
4.12 Example of a bigram proximity matrix . . . 45
4.13 Dimension reduction techniques . . . 46
4.14 Feature transformation techniques . . . 47
4.15 Feature selection techniques . . . 47
4.16 Term document matrix for LSA . . . 49
4.17 Kruskal STRESS experiences . . . 57
4.18 Evaluation of dimension reduction methods . . . 61
4.19 Classification of clustering techniques . . . 64
4.20 Example: k-means . . . 69
4.21 Evaluation of clustering methods . . . 73

5.1 Integration of text mining into SONIVIS . . . 87
5.2 Statistics of the English Wikipedia . . . 88
5.3 Properties of data set I . . . 89
5.4 Properties of data set II . . . 89
5.5 Properties of data set III . . . 89
5.6 Properties of data set IV . . . 90
5.7 Term Transformer . . . 90
5.8 Selection of analysis methods . . . 93
5.9 Clustering evaluation measures . . . 94
5.10 Properties of clustering phases . . . 96

6.1 Selection of clustering algorithms . . . 104

6.2 Top term weight thresholds . . . 107
6.3 Selection of term weight thresholds . . . 108
6.4 Selection of optimal number of dimensions . . . 114
6.5 Comparing the clustering algorithms . . . 131

F.1 F-measure: Content-based clustering . . . 181
F.2 Dunn Index: Content-based clustering . . . 181
F.3 SD Validity Index: Content-based clustering . . . 181
F.4 Computing Time: Content-based clustering . . . 181
F.5 F-measure: Link-based clustering . . . 182
F.6 Dunn Index: Link-based clustering . . . 182
F.7 SD Validity Index: Link-based clustering . . . 182
F.8 Computing Time: Link-based clustering . . . 182
F.9 F-measure: Content-based clustering, Diana . . . 183
F.10 F-measure: Content-based clustering, PAM . . . 183
F.11 F-measure: Link-based clustering, Agnes . . . 183
F.12 F-measure: Link-based clustering, PAM . . . 183
F.13 Singular values: Content-based clustering, no feature selection . . . 184
F.14 Singular values: Content-based clustering, 50% feature selection . . . 184
F.15 Singular values: Link-based clustering, 90% feature selection . . . 185
F.16 F-measure: Content-based clustering, Diana . . . 185
F.17 F-measure: Content-based clustering, PAM . . . 185
F.18 F-measure: Link-based clustering, Agnes . . . 186
F.19 F-measure: Link-based clustering, PAM . . . 186
F.20 F-measure: Content-based clustering . . . 187
F.21 Dunn Index: Content-based clustering . . . 187
F.22 SD Validity Index: Content-based clustering . . . 187
F.23 Computing Time: Content-based clustering . . . 187
F.24 F-measure: Link-based clustering . . . 187
F.25 Dunn Index: Link-based clustering . . . 188
F.26 SD Validity Index: Link-based clustering . . . 188
F.27 Computing Time: Link-based clustering . . . 188
F.28 Dunn Index: Optimal solution for content-based clustering . . . 188
F.29 SD Validity Index: Optimal solution for content-based clustering . . . 188
F.30 Dunn Index: Optimal solution for link-based clustering . . . 189
F.31 SD Validity Index: Optimal solution for link-based clustering . . . 189

# D. Bibliography


[1] ACE. Annotation guidelines for entity detection and tracking (edt). http://www.ldc.upenn.edu/Projects/ACE/, 2004.

[2] Sisay F. Adafre and Maarten de Rijke. Discovering missing links in wikipedia. In *Proceedings of the 3rd international workshop on Link discovery (LinkKDD-05)*, pages 90–97, New York, NY, USA, 2006. ACM.

[3] G. W. Adamson and J. Boreham. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Information Processing & Management*, 10(7-8):253–260, 1974.

[4] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD Record ACM Special Interest Group on Management of Data*, page 94105, New York, NY, USA, 1998. ACM.

[5] Musawir Ali. Tensors and component analysis. Talk. http://graphics.cs.ucf.edu/presentation/tensor.ppt.

[6] James Allan. Building hypertext using information retrieval. *Information Processing and Management*, 33(2):145–159, 1997.

[7] M.R. Anderberg. *Cluster analysis for applications*. Monographs and Textbooks on Probability and Mathematical Statistics. Academic press, Inc., New York, NY, USA, 1973.

[8] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, May 1999.

[9] Mukund Balasubramanian and Eric L. Schwartz. The isomap algorithm and topological stability. *Science*, 295(5552):7, January 2002.

[10] Somnath Banerjee. Boosting inductive transfer for text classification using wikipedia. In *Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA-07)*, pages 148–153, Washington, DC, USA, 2007. IEEE Computer Society.

[11] Somnath Banerjee, Krishnan Ramanathan, and Ajay Gupta. Clustering short texts using wikipedia. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIRIR-07)*, pages 787–788, New York, NY, USA, 2007. ACM.

[12] Daniel Barbará, Yi Li, and Julia Couto. Coolcat: An entropy-based algorithm for categorical clustering. In *Proceedings of the eleventh international conference on Information and knowledge management (CIKM-02)*, pages 582–589, New York, NY, USA, 2002. ACM.

[13] Ken Barker and Nadia Cornacchia. Using noun phrase heads to extract document keyphrases. In *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, pages 40–52, 2000.

[14] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.

[15] S. Bay and M. Pazzani. Detecting change in categorical data: Mining contrast sets. In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD-99)*, page 302306, New York, NY, USA, 1999. ACM.

[16] Michael W. Berry and Murray Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval (Software, Environments, Tools)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2 edition, 2005.

[17] Michael W. Berry and Malu Castellanos. *Survey of Text Mining II: Clustering, Classification, and Retrieval*. Springer, 2008.

[18] J. C. Bezdek. *Fuzzy Mathematics in Pattern Classification*. PhD thesis, Applied Math. Center, Cornell University, Ithaca, 1973.

[19] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[20] Chris Biemann. Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs: the Second Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80, New York, NY, USA, 2006. Association for Computational Linguistics (ACL).

[21] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the 5th conference on Applied natural language processing*, pages 194–201, San Francisco, CA, USA, 1997. Morgan Kaufmann.

[22] Christopher M. Bishop and Christopher K. I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

[23] L. Bobrowski and J. Bezdek. c-means clustering with the l1 and l∞ norms. *IEEE Transactions on Systems (Man and Cybernetics)*, 21(3):545–554, 1991.

[24] Hans Hermann Bock. *Automatische Klassifikation*. Vandenhoeck and Ruprecht, Gttingen, 1974.

[25] Daniel Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, 1998.

[26] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, October 2001.

[27] Kenneth Bryan. Biclustering of expression data using simulated annealing. In *Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems (CBMS-05)*, pages 383–388, Washington, DC, USA, 2005. IEEE Computer Society.

[28] Chris Buckley, Gerard Salton, and James Allan. The smart information retrieval project. In *Workshop on Human Language Technology (HLT-93)*, pages 392–392, Morristown, NJ, USA, 1993. Association for Computational Linguistics (ACL).

[29] A. C. Cadavid, J. K. Lawrence, and A. Ruzmaikin. Principal components and independent component analysis of solar and space data. *Solar Physics*, 248:247–261, April 2007.

[30] Mary E. Califf. Relational learning techniques for natural language information extraction. http://citeseer.comp.nus.edu.sg/70265.html, 1998.

[31] Claire Cardie. Empirical methods in information extraction. *AI magazine*, 18(4):65–79, 1997.

[32] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in applied Sciences*, 1(4):300–307, May 2007.

[33] P. Cheeseman, J. Kelly, M. Self, J. Stutz., W. Taylor, and D. Freeman. Autoclass: A bayesian classification system. In *Proceedings of the international conference on Machine Learning (ICML-88)*, page 5464, San Mateo, CA; USA, 1988. Morgan Kaufmann.

[34] Peter Cheeseman and John Stutz. Bayesian classification (autoclass): Theory and results. In Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhrais Smyth, and Ramasamy Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI, 1996.

[35] C. Cheng, A. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining*, page 8493, New York, NY, USA, 1999. ACM.

[36] Nancy A. Chinchor. Overview of muc-7/met-2. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, April 1997.

[37] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler. Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems. *Multiscale Modeling & Simulation*, 7(2):842–864, 2008.

[38] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. In *Proceedings of the National Academy of Sciences*, volume 102, pages 7426–7431, May 2005.

[39] Two Crows Corporation. *Introduction to Data Mining and Knowledge Discovery*. Two Crows Corporation, third edition, 1999.

[40] Renato Fernandes Corrêa and Teresa Bernarda Ludermir. Dimensionality reduction by semantic mapping in text categorization. In *Proceedings of VIIIth Brazilian Symposium on Neural Networks*, volume 1, pages 1032–1037, Silver Spring, MD, USA, 2004. IEEE Computer Society.

[41] Anna B. Costello and Jason W. Osborne. Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis. *Practical Assessment, Research & Evaluation (PARE)*, 10(7), July 2005.

[42] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall, 2 edition, 2001.

[43] David Crystal. *The Cambridge Encyclopedia of Language*. Cambridge University Press, 1987.

[44] Gaoying Cui, Qin Lu, Wenjie Li, and Yirong Chen. Corpus exploitation from wikipedia for ontology construction. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the 6th International Language Resources and Evaluation (LREC-08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[45] Padraig Cunningham. Dimension reduction. Technical report, University College Dublin, August 2007.

[46] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR-92)*, pages 318–329, New York, NY, USA, 1992. ACM.

[47] Manoranjan Dash, Huan Liu, and Xiaowei Xu. 1 + 1 ¿ 2: Merging distance and density based clustering. In *Proceedings of the 7th International Conference on database systems for advanced applications*, pages 32–39, Hong Kong, 2001. IEEE Computer Society.

[48] J.L. Dawson. Suffix removal for word conflation. *Bulletin of the Association for Literary and Linguistic Computing*, 2(3):33–46, 1974.

[49] Jamie DeCoster. Overview of factor analysis. http://www.stat-help.com/factor.pdf, August 1998.

[50] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[51] P. Demartines and J. Herault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1):148–154, 1997.

[52] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm - series b (methodological). *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[53] Mark Devaney and Ashwin Ram. Efficient feature selection in conceptual clustering. In Douglas H. Fisher, editor, *In Proceedings of the Fourteenth International Conference on Machine Learning ICML*, pages 92–97. Morgan Kaufmann, 1997.

[54] Inderjit S. Dhillon. Coclustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD-01)*, pages 269–274. ACM, 2001.

[55] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

[56] Jochen Dörre, Peter Gerstl, and Roland Seiffert. Text mining: finding nuggets in mountains of textual data. In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD-99)*, pages 398–401, New York, NY, USA, 1999. ACM.

[57] J.G. Dy and C.E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, August 2004.

[58] Andreas Erber. Analysis of wikilink networks with sonivis:tool. Ongoing work, http://www.sonivis.org/wiki/index.php/User:Andreas, 2009.

[59] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd international conference on knowledge discovery and data mining*, page 226231, Portland, OR, 1996. AAAI.

[60] Usama Fayyad, Gregory Piatetsky-shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

[61] R. Feldman and I. Dagan. Knowledge discovery in textual databases (kdt). In *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 112–117, 1995.

[62] Ronen Feldman and James Sanger. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, Cambridge, MA, USA, December 2006.

[63] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[64] Christopher Fox. A stop list for general text. *SIGIR Forum*, 24(1-2):19–21, 1989.

[65] William B. Frakes and Ricardo A. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, June 1992.

[66] K. Fukunaga and D.R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, 20(2):176–183, 1971.

[67] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 6–12, Hyderabad, India, 2007.

[68] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics (SIAM), May 2007.

[69] V. Ganti, J. Gehrke, and R. Ramakrishnan. Cactus: Clustering categorical data using summaries. In *Proceedings of the 5th ACM SIGKDD international conference on knowledge discovery and data mining*, page 7383, New York, NY, USA, 1999. ACM.

[70] I. Gath and A. B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):773–780, 1989.

[71] Barry Glasgow, Alan Mandell, Dan Binney, Lila Ghemri, and David Fisher. Mita: An information extraction approach to analysis of free-form text in life insurance applications. In *AAAI/IAAI*, pages 992–999, 1997.

[72] F. Glover. Tabu search - part 1. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[73] F. Glover. Tabu search - part 2. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[74] Sebastian Göser. Inhaltsbasiertes information retrieval: Die textmining-technologie. *LDV Forum*, 14(1):48–52, 1997.

[75] J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:325–338, 1966.

[76] P.E. Greenwood and M.S. Nikulin. A guide to chi-squared testing. *Computational Statistics & Data Analysis*, 23(4):565–566, February 1997.

[77] Ralph Grishman. *Computational Linguistics An Introduction*. Cambridge University Press, 1986.

[78] Ralph Grishman and Beth Sundheim. Message understanding conference - 6: A brief history. In *International Conference on Computational Linguistics*, 1996.

[79] Patrick J.F. Groenen and Michel van de Velden. Multidimensional scaling. Technical report, Erasmus University Rotterdam, Econometric Institute, April 2004.

[80] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data (SIGMOD-98)*, pages 73–84, New York, NY, USA, 1998. ACM.

[81] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: a robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366, 2000.

[82] D. E. Gustafson and W. C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proceedings of the IEEE Conference on Decision and Control*, pages 761–776, San Diego, CA,USA, 1979.

[83] Richard D. Hackathorn. *Web farming for the data warehouse*. Morgan Kaufmann, San Francisco, CA, USA, 1999.

[84] Karin Haenelt. hnlichkeitsmae fr vektoren. Course material, February 2007.

[85] Margaret A. Hafer and Stephen F. Weiss. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10(11-12):371–385, 1974.

[86] Udo Hahn and Klemens Schnattinger. Towards text knowledge engineering. In *AAAI/IAAI*, pages 524–531, 1998.

[87] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17:107–145, 2001.

[88] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, September 2000.

[89] Andreas Handl. Materials to the book "multivariate analysemethoden: Theorie und praxis multivariater verfahren unter besonderer bercksichtigung von s-plus". University of Bielefeld, 2002. Chapter: Mehrdimensionale Skalierung.

[90] P. Hansen and N. Mladenovic. J-means: A new local search heuristic for minimum sum of squares clustering. *Pattern Recognition*, 34(2):405413, 2001.

[91] P. Hansen and N. Mladenovic. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449467, 2001.

[92] Karanikas Haralambos and Babies Theodulidis. Knowledge discovery in text and text mining software. CRIM/UMIST UK, 2002.

[93] Y. Hassan-Montero and V. Herrero-Solana. Improving tag-clouds as visual information retrieval interfaces. In *International Conference on Multidisciplinary Information Sciences and Technologies*, Merida, Spain, October 2006.

[94] Bin He and Yongzheng Zhang. Clustering documents in large text corpora. Technical report, Faculty of Computer Science, Dalhousie University, 2003.

[95] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *NIPS*. MIT Press, 2005.

[96] Marti A. Hearst. Untangling text data mining. School of Information Management & Systems University of California, Berkeley, 1999.

[97] Marti A. Hearst. What is text mining? http://people.ischool.berkeley.edu/~hearst/text-mining.html, October 2003.

[98] Gerhard Heyer, Uwe Quasthoff, and Thomas Wittig. *Text Mining: Wissensrohstoff Text*. W3L GmbH, 2006.

[99] Alexander Hinneburg and Daniel A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of 4th International Conference on Knowledge Discovery and Data Mining*, page 5865, New York, NY, USA, 1998. AAAI.

[100] Alexander Hinneburg and Daniel A. Keim. Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *Proceedings of 25th International Conference on Very Large Data Bases (VLDB-99)*, pages 506–517, San Francisco, CA, USA, 1999. Morgan Kaufmann.

[101] Hajo Hippner and Ren Rentzmann. Text mining. *Informatik Spektrum*, 29(4):287–290, 2006.

[102] Hajo Hippner and Klaus D. Wilde. Der prozess des data mining im marketing. Lecture: Analytisches CRM: Prozess und Methoden, 2009.

[103] Richard D. Holowczak and Nabil R. Adam. Information extraction based multiple-category document classification for the global legal information applications. In *National Conference on Artificial Intelligence (AAAI)*, 1997.

[104] Andreas Hotho, Andreas Nrnberger, and Gerhard Paa. A brief survey of text mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 20(1):19–62, May 2005.

[105] Kien A. Hua, S. D. Lang, and Wen K. Lee. A decomposition-based simulated annealing technique for data clustering. In *Proceedings of the thirteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (PODS-94)*, pages 117–128, New York, NY, USA, 1994. ACM.

[106] Shiping Huang, Matthew O. Ward, and Elke A. Rundensteiner. Exploration of dimensionality reduction for text visualization. In *Proceedings of the Coordinated and Multiple Views in Exploratory Visualization (CMV-05)*, pages 63–74, Washington, DC, USA, 2005. IEEE Computer Society.

[107] Z. Huang and M. Ng. A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, 7(4):446452, 1999.

[108] Zhexue Huang. A fast clustering algorithm to cluster very large categorical data sets in data mining. In *SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 1–8. ACM, 1997.

[109] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining Knowledge Discovery*, 2(3):283–304, 1998.

[110] Masahiro Ito, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. Association thesaurus construction methods based on link co-occurrence analysis for wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM-08)*, pages 817–826, New York, NY, USA, 2008. ACM.

[111] Paul S. Jacobs. *Text-based intelligent systems: current research and practice in information extraction and retrieval*, chapter Introduction: text power and intelligent systems, pages 1–8. L. Erlbaum Associates Inc., 1992.

[112] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Upper Saddle River, NJ, USA, 1988.

[113] Anuradha Jambunathan and Marco Ronchetti. Exploiting the collective intelligence contained in wikipedia to automatically describe the content of a document. In *Proceedings of the Workshop on Collective Intelligence at the Third Asian Semantic Web Conference*, pages 209–216, Bangkok, Thailand, December 2006.

[114] Helmut Jarosch. *Information Retrieval und Künstliche Intelligenz*, volume 1. Auflage. Deutscher Universitätsverlag, 2007.

[115] M. Jeon, H. Park, and J. B. Rosen. Dimension reduction based on centroids and least squares for efficient processing of text data. Technical report, University of Minnesota, Minneapolis, MN, USA, February 2001. Technical Report MN TR 01-010.

[116] T. Joachims and E. Leopold. Themenheft: Text-mining. vorwort der herausgeber. *Knstliche Intelligenz*, 2002.

[117] W. B. Johnson and J. Lindenstrauss. Extensions of lipshitz mapping into hilbert space. In *Conference in modern analysis and probability*, volume 26, pages 189–206. American Mathematical Society, Contemporary Mathematics, 1984.

[118] I. T. Jolliffe. *Principal Component Analysis*. Springer, 1986.

[119] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice Hall, 2008.

[120] Nandakishore Kambhatla and Todd K. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1493–1516, October 1997.

[121] P. Kanerva, J. Kristoferson, and A. Holst. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, pages 103–106, 2000.

[122] Anne Kao and Steve Poteet. Text mining and natural language processing: Introduction for the special issue. *SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining*, 7(1):1–2, 2005.

[123] George Karypis. Cluto a clustering toolkit. Technical report, Department of Computer Science, University of Minnesota, 2002.

[124] George Karypis, Eui, and Vipin K. News. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.

[125] Samuel Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proceedings of the International Joint Conference on Neural Networks IJCNN98*, volume 1, pages 413–418. Piscataway, NJ: IEEE, 1998.

[126] Laura Kassner, Vivi Nastase, and Michael Strube. Acquiring a taxonomy from the german wikipedia. In *Proceedings of the 6th International Language Resources and Evaluation (LREC-08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[127] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley & Sons, 1990.

[128] Frank C. Keil and Robert A. Wilson, editors. *The MIT encyclopedia of the cognitive sciences*. MIT Press, Cambridge, Massachusetts, 1999.

[129] Daniel Kinzler. *Automatischer Aufbau eines multilingualen Thesaurus durch Extraktion semantischer und lexikalischer Relationen aus der Wikipedia*. Diploma thesis, Department of Computer Science, Faculty of Mathematics and Computer Science, University of Leipzig, Leipzig, Germany, May 2008.

[130] Karsten Klüger. *Automatische Extraktion von Schlüsselwörtern aus Text*. Diploma thesis, MediaSystems, Faculty of Media, Bauhaus-University Weimar, Germany, June 2006.

[131] Yves Kodratoff. Knowledge discovery in texts: A definition, and applications. In *Proceedings of the 11th International Symposium on Foundations of Intelligent Systems (ISMIS-99)*, pages 16–29. Springer, 1999.

[132] Yves Kodratoff. Knowledge discovery in texts: A definition, and applications. In *Proceedings of the 11th International Symposium on Foundations of Intelligent Systems (ISMIS-99)*, pages 16–29. Springer, 1999.

[133] Teuvo Kohonen. *Self-Organizing Maps*. Springer, December 2000.

[134] Daphne Koller and Mehran Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292, 1996.

[135] R. Kosala and H. Blockeel. Mining knowledge from text using information extraction. *SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining*, 2(1):1–15, 2000.

[136] Ferenc Kovacs, Csaba Legany, and Attila Babos. Cluster validity measurement techniques. Technical report, Budapest University of Technology and Economics, Department of Automation and Applied Informatics, Budapest, Hungary, 2002.

[137] K. Krishna and M. Narasimha. Genetic k-means algorithm. *IEEE Transactions on Systems (Man and Cybernetics)*, 29(3):433439, 1999.

[138] Robert Krovetz. Viewing morphology as an inference process. In *SIGIR-93*, pages 191–202. ACM, 1993.

[139] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:127, 1964.

[140] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:115129, 1964.

[141] J. B. Kruskal and M. Wish. Multidimensional scaling. *Sage Publications*, 1977.

[142] Stephen Langdell. Examples of the use of data mining in financial applications. *Financial Engineering News*, April 2002.

[143] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD-99)*, pages 16–22, New York, NY, USA, 1999. ACM.

[144] Yann Le Cun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems II*, pages 598–605. Morgan Kaufmann, 1990.

[145] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[146] John Aldo Lee, Amaury Lendasse, and Michel Verleysen. Curvilinear distance analysis versus isomap. In *Proceedings of the 10th European Symposium on Artificial Neural Networks (ESANN-02)*, pages 185–192, 2000.

[147] Yanjun Li and Soon M. Chung. Parallel bisecting k-means with prediction clustering algorithm. *The Journal of Supercomputing*, 39(1):19–37, 2007.

[148] King-Ip Lin and Ravikumar Kondadadi. A word-based soft clustering algorithm for documents. In *Computers and Their Applications*, pages 391–394. ISCA, 2001. Proceedings of the 16th International Conference Computers and Their Applications (ISCA-01).

[149] Paul B. Losiewicz, Douglas W. Oard, and Ronald N. Kostoff. Textual data mining to support science and technology management. *J. Intell. Inf. Syst.*, 15(2):99–119, 2000.

[150] J. Loughrey and P. Cunningham. Using early-stopping to avoid overfitting in wrapper-based feature subset selection employing stochastic search. In M. Petridis, editor, *Proceedings of the 10th UK Workshop on Case-Based Reasoning*, pages 3–10. CMS Press, 2005.

[151] J. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

[152] H. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(92):159165, 1958.

[153] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1979.

[154] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, July 2008.

[155] K. V. Mardia. Statistics of directional data - series b. *Journal of the Royal Statistical Society*, 37(3):349–393, October 1975.

[156] Angel R. Martinez. A framework for the representation of semantics. Ph.D. thesis, George Mason University, 2002.

[157] Wendy L. Martinez and Angel R. Martinez. *Exploratory Data Analysis with MATLAB (Computer Science and Data Analysis)*. Chapman & Hall / CRC Press, November 2004.

[158] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(1):157–169, 2003.

[159] A. Mccallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, pages 41–48. AAAI, 1998.

[160] John McCarthy. What is artificial intelligence? http://cogprints.org/412/, January 1998.

[161] Alexander Mehler and Christian Wolff. *Themenschwerpunkt Text mining*. GLDV, 2005.

[162] Marina Meila and David Heckerman. An experimental comparison of model-based clustering methods. *Journal of Machine Learning Research*, 42(1-2):9–29, 2001.

[163] Rada Mihalcea. Using wikipedia for automatic word sense disambiguation. In *North American Chapter of the Association for Computational Linguistics (NAACL-07)*, pages 196–203. Association for Computational Linguistics (ACL), 2007.

[164] Dunja Mladenic. Feature subset selection in text-learning. In C. Nedellec and C. Rouveirol, editors, *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, volume 1398, pages 95–100, London, UK, 1998. Springer.

[165] Raymond J. Mooney and Razvan Bunescu. Mining knowledge from text using information extraction. *SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining*, pages 3–10, June 2005.

[166] Steven A. Morris and Gary G. Yen. Crossmaps: Visualization of overlapping relationships in collections of journal papers. *Proceedings of the National Academy of Sciences*, 101(1):5291–5296, April 2004.

[167] MUC-5. *Proceedings of the 5th Message Understanding Conference*. Morgan Kaufmann, 1993.

[168] Claudia Müller, Benedikt Meuthrath, and Anne Baumgrass. Analyzing wiki-based networks to improve knowledge processes in organizations. *Journal of Universal Computer Science*, 14(4):526–545, 2008.

[169] Claudia Müller, Benedikt Meuthrath, and Sabina Jeschke. Defining a universal actor content-element model for exploring social and information networks considering the temporal dynamic. In *Advances in Social Networks Analysis and Mining (ASONAM-09)*, pages 524–531, 2009.

[170] H. Nagesh, S. Goil, and A. Choudhary. A scalable parallel subspace clustering algorithm for massive data sets. In *2000 International Conference on Parallel Processing (ICPP 00)*, page 477486, Washington, Brussels, Tokyo, 2000. IEEE.

[171] Un Yong Nahm and Raymond J. Mooney. Text mining with information extraction. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.5941, 2002.

[172] K. Nakayama, T. Hara, and S. Nishio. Wikipedia mining for an association web thesaurus construction. In *Proceedings of IEEE International Conference on Web Information Systems Engineering*, pages 322–334, 2007.

[173] Jana Novovicová, Antonín Malík, and Pavel Pudil. Feature selection using improved mutual information for text classification. In A. L. N. Fred, T. Caelli, R. P. W. Duin, A. C. Campilho, and D. de Ridder, editors, *SSPR/SPR*, volume 3138, pages 1010–1017. Springer, 2004.

[174] C.D. Paice. Another stemmer. *SIGIR Forum*, 24:56–61, 1990.

[175] Andreas Plank. Grafiken und statistik in r. University FU Berlin, March 2008.

[176] Joel Plisson, Dunja Mladenic, Nada Lavrac, and Tomaz Erjavec. A lemmatization web service based on machine learning techniques. In *2nd Language & Technology Conference (LTC)*, 2005.

[177] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[178] C. Procopiuc, M. Jones, P. Agarwal, and T. Murali. A monte carlo algorithm for fast projective clustering. In *Proceedings of the 2002 ACM SIGMOD international conference on management of data*, page 418427, New York, NY, USA, 2002. ACM.

[179] Ross J. Quinlan. *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, January 1993.

[180] M. Rajman and R. Besoncon. Text mining - knowledge extraction from unstructured textual data. In *Proceedings of the 6th Conference of International Federation of Classification Societies (IFCS 1998)*, Rome, Italy, July 1998.

[181] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[182] Sam T. Roweis and Lawrence K. Saul. An introduction to locally linear embedding. http://www.cs.toronto.edu/~roweis/lle/, 2001.

[183] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2 edition, December 2002.

[184] Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for nlp. In *3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002)*, pages 1–15, 2001.

[185] Gerard Salton, editor. *Automatic Information Organization and Retrieval*. New York, McGraw-Hill, 1968.

[186] Gerard Salton. *Automatic text processing : the transformation, analysis and retrieval of information by computer*. Addison-Wesley, Reading, Mass., 1989.

[187] Gerard Salton, James Allan, and Chris Buckley. Automatic structuring and retrieval of large text files. *Communications of the ACM*, 37(2):97–108, 1994.

[188] J. W. Sammon. A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*, 19:401409, 1969.

[189] Antonio Sanfilippo and Ex Sharp. Preliminary recommendations on lexical semantic encoding - final report. EAGLES project, http://www.ilc.cnr.it/EAGLES96/rep2/, 1999.

[190] Sunita Sarawagi. Information extraction. *Foundations and Trends in Databases*, pages 261–377, 2008.

[191] E. Schikuta. Grid-clustering: An efficient hierarchical clustering method for very large data sets. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 2, page 101105, Vienna, Austria, 1996. IEEE.

[192] Peter Schonhofen. Identifying document topics using the wikipedia category network. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI-06)*, pages 456–462, Washington, DC, USA, 2006. IEEE Computer Society.

[193] Jeffrey W. Seifert. Data mining: An overview. CRS Report RL31798, Congressional Research Service and The Library of Congress, 2004.

[194] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, October 1948.

[195] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. Wavecluster: a wavelet-based clustering approach for spatial data in very large databases. *The VLDB Journal*, 8(3-4):289–304, 2000.

[196] Jonathon Shlens. A tutorial on principal component analysis. Technical report, Systems Neurobiology Laboratory, Salk Insitute for Biological Studies, December 2005.

[197] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[198] Dan Siroker and Steve Miller. Topical clustering, summarization, and visualization. http://nlp.stanford.edu/courses/cs224n/2003/fp/millersj/cs224nfp.pdf, 2008.

[199] Lindsay I. Smith. A tutorial on principal component analysis. http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf, February 2002.

[200] Stephen Soderland, David Aronow, David Fisher, Jonathan Aseltine, and Wendy Lehnert. Machine learning of textanalysis rules for clinical records. University of Massachusetts, Amherst MA, Harvard Community Health Plan, Brookline, 1995.

[201] Jeffrey Solka. Text data mining: Theory and methods. http://arxiv.org/abs/0807.2569v1, July 2008.

[202] Henry Stark and John W. Woods. *Probability, random processes, and estimation theory for engineers*. Prentice-Hall, Upper Saddle River, NJ, USA, 1986.

[203] Benno Stein, Sven zu Eissen Meyer, and Frank Wibrock. On cluster validity and the information need of users. In *Proceedings of the 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA-03)*, Benalmadena, Spain, 2003.

[204] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. Technical report, University of Minnesota, 2000.

[205] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. In *Proceedings of the 17th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search (AAAI-00)*, pages 58–64, Austin, Texas, USA, July 2000. AAAI.

[206] Fabian Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge - unifying wordnet and wikipedia. In *Proceedings of the 16th International World Wide Web Conference (WWW-07)*, pages 697–706, Banff, Canada, 2007. ACM.

[207] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd international conference on Machine learning ICML 06*, pages 905–912, New York, NY, USA, 2006. ACM.

[208] Z. Syed, T. Finin, and A. Joshi. Wikipedia as an ontology for describing documents. In *Proceedings of the Second International Conference on Weblogs and Social Media*. AAAI Press, March 2008.

[209] Bin Tang, Michael Shepherd, Malcolm Heywood, and Xiao Luo. Comparing dimension reduction techniques for document clustering. In Balazs Kegl and Guy Lapalme, editors, *The 18th Canadian Conference on Artificial Intelligence*, volume 3501 of *Lecture Notes in Computer Science*, pages 292–296, Victoria, BC, Canada, May 2005. Springer.

[210] Bin Tang, Michael Shepherd, Evangelos Milios, and Malcolm Heywood. Comparing and combining dimension reduction techniques for efficient text clustering. In *Proceedings International Workshop on Feature Selection for Data Mining: Interfacing Machine Learning and Statistics*, In conjunction with 2005 SIAM International Conference on Data Mining, pages 17–26, Newport Beach, CA, USA, April 2005.

[211] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

[212] Richmond Thomason. Logic and artificial intelligence. Stanford Encyclopedia of Phylosophy, May 2008.

[213] Michael E. Tipping and Chris M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[214] Richard M. Tong. Knowledge-based techniques for information retrieval. *International Journal of Intelligent Systems*, March 1989.

[215] Warren S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17:401419, 1952.

[216] Warren S. Torgerson. *Theory and Methods of Scaling*. Wiley & Sons, London, 1958.

[217] Yuen-Hsien Tseng. Multilingual keyword extraction for term suggestion. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR-98)*, pages 377–378, Melbourne, Australia, 1998. ACM.

[218] David G. Underhill, Luke McDowell, David J. Marchette, and Jeffrey L. Solka. Enhancing text analysis via dimensionality reduction. In *IRI*, pages 348–353. IEEE Systems, Man, and Cybernetics Society, 2007.

[219] Hans Uszkoreit. What is computational linguistics? http://www.coli.uni-saarland.de/~hansu/what{_}is{_}cl.html, 1996.

[220] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 2 edition, 1979.

[221] Wei Wang, Jiong Yang, and Richard R. Muntz. Sting: A statistical information grid approach to spatial data mining. In Matthias Jarke, Michael J. Carey, Klaus R. Dittrich, Frederick H. Lochovsky, Pericles Loucopoulos, and Manfred A. Jeusfeld, editors, *Proceedings of 23rd International Conference on Very Large Data Bases (VLDB-97)*, pages 186–195, Athens, Greece, August 1997. Morgan Kaufmann.

[222] J. Ward Jr. and M. Hook. Application of a hierarchical grouping procedure to a problem of grouping profiles. *Educational and Psychological Measurement*, 23(1):6981, 1963.

[223] W. John Wilbur and Karl Sirotkin. The automatic identification of stop words. *Journal of Information Science*, 18(1):45–55, 1992.

[224] Yorick Wilks. Information extraction as a core language technology. In *Lecture Notes in Computer Science, chapter In M-T. Pazienza (ed.), Information Extraction*, pages 14–18. Springer, 1997.

[225] Peter Willett. Similarity coefficients and weighting functions for automatic document classification: an empirical comparison. *International Classification*, 10:138–142, 1983.

[226] René Witte and Jutta Mülle, editors. *Text Mining: Wissensgewinnung aus natürlichsprachigen Dokumenten*. Universität Karlsruhe, Fakultät für Informatik, Institut für Programmstrukturen und Datenorganisation (IPD), 2006.

[227] Ian H. Witten and Eibe Frank. *Data mining : practical machine learning tools and techniques*. Elsevier, Morgan Kaufmann, second edition, 2005.

[228] Ian H. Witten, Gordon Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the 4th ACM Conference on Digital Libraries (DL-99)*, pages 254–255, Berkeley, California, USA, 1999. ACM.

[229] Lior Wolf and Amnon Shashua. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *Journal of Machine Learning Research*, 6:1855–1887, 2005.

[230] F. Wu and D. S. Weld. Autonomously semantifying wikipedia. In *Proceedings of the 16th ACM conference on Conference on information and knowledge management*, pages 41–50. ACM New York, NY, USA, 2007.

[231] S. Wu and P. A. Flach. Feature selection with labelled and unlabelled data. In M. Bohanec, B. Kasek, N. Lavrac, and D. Mladenic, editors, *Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning (ECML/PKDD-02)*, pages 156–167. University of Helsinki, August 2002.

[232] X. Xu, M. Ester, H. Kriegel, and J. Sander. A distribution-based clustering algorithm for mining in large spatial databases. In *Proceedings of the fourteenth international conference on data engineering (ICDE 98)*, page 324331, Orlando, FL, 1998. IEEE Computer Society.

[233] Yiming Yang. Noise reduction in a statistical approach to text categorization. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR-95)*, pages 256–263, New York, NY, USA, 1995. ACM.

[234] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML-97)*, pages 412–420, San Francisco, CA, USA, May 1997. Morgan Kaufmann.

[235] Yiming Yang and W. John Wilbur. Using corpus statistics to remove redundant words in text categorization. *Journal of the American Society for Information Science and Technology (JASIST)*, 47(5):357–369, 1996.

[236] Illhoi Yoo. Biomedical ontologies for medline text mining. Dept. of Health Management and Informatics School of Medicine, Seminar, March 2006.

[237] Shipeng Yu, Kai Yu, Volker Tresp, Hans-Peter Kriegel, and Mingrui Wu. Supervised probabilistic principal component analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD-06)*, pages 464–473, New York, NY, USA, 2006. ACM.

[238] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[239] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR-98)*, pages 46–54, New York, NY, USA, 1998. ACM.

[240] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 103–114, Montreal, Quebec, Canada, June 1996. ACM.

[241] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.

[242] Y. Zhao and J. Song. Gdilc: A grid-based density-isoline clustering algorithm. In *Proceedings of the international conferences on info-tech and info-net*, volume 3, page 140145, Beijing, 2001. IEEE.

[243] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the 11th international conference on Information and knowledge management (CIKM-02)*, pages 515–524. ACM, 2002.

[244] Shi Zhong and Joydeep Ghosh. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3):374–384, January 2005.

[245] Xiaojin Zhu. Latent topic models. Lecture: Advanced Natural Language Processing, 2009. http://pages.cs.wisc.edu/~jerryzhu/cs769/LDA.pdf.

[246] George Kingsley Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison Wesley, 1949.

# E. Topics of Data Sets

## E.1. Data Set I

### Topic: Astronomy

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Andromeda Galaxy | 1253 | 261 | 228 | 651 |
| Asteroid | 2681 | 634 | 346 | 1000 |
| Astrology | 5600 | 807 | 354 | 1106 |
| Astrology and astronomy | 244 | 49 | 121 | 545 |
| Astronomy | 3582 | 976 | 526 | 1224 |
| Atmosphere | 1004 | 318 | 208 | 384 |
| Big Bang | 4970 | 996 | 364 | 1039 |
| Black hole | 3834 | 1047 | 375 | 1242 |
| Comet | 3010 | 761 | 389 | 1136 |
| Cosmic neutrino background | 54 | 9 | 41 | 166 |
| Dark energy | 1101 | 283 | 174 | 707 |
| Dark matter | 2377 | 671 | 283 | 823 |
| Definition of planet | 2326 | 169 | 243 | 1099 |
| Earth | 5924 | 1505 | 562 | 1415 |
| Event horizon | 379 | 113 | 106 | 325 |
| Extrasolar planet | 1208 | 150 | 350 | 965 |
| Galaxy | 1557 | 376 | 354 | 935 |
| Galaxy filament | 142 | 19 | 84 | 261 |
| Galaxy formation and evolution | 293 | 56 | 88 | 475 |
| Galaxy groups and clusters | 206 | 50 | 99 | 290 |
| Jupiter | 1017 | 240 | 448 | 1165 |
| Mars | 507 | 100 | 526 | 1626 |
| Mercury (planet) | 1199 | 274 | 410 | 1143 |
| Meteoroid | 1236 | 379 | 144 | 608 |
| Milky Way | 3273 | 887 | 367 | 948 |
| Moon | 6877 | 1486 | 502 | 1422 |
| Neptune | 5177 | 1296 | 345 | 1085 |
| Neutron star | 1047 | 299 | 216 | 654 |
| Observable universe | 494 | 76 | 104 | 410 |
| Outer space | 1410 | 393 | 228 | 620 |
| Outline of astronomy | 121 | 6 | 218 | 417 |
| Physical cosmology | 534 | 127 | 232 | 679 |
| Planet | 2733 | 551 | 551 | 1259 |
| Ring galaxy | 45 | 8 | 37 | 69 |
| Saturn | 4817 | 1228 | 322 | 966 |

| | | | | |
|---|---|---|---|---|
| Sirius | 1556 | 301 | 299 | 962 |
| Sky | 1880 | 538 | 131 | 184 |
| Solar telescope | 72 | 11 | 48 | 145 |
| Spiral galaxy | 474 | 123 | 134 | 378 |
| Star | 1741 | 466 | 537 | 1179 |
| Sun | 3062 | 824 | 647 | 1467 |
| Supercluster | 172 | 50 | 91 | 175 |
| Supermassive black hole | 670 | 236 | 103 | 345 |
| Supernova | 2898 | 734 | 293 | 855 |
| Terrestrial planet | 442 | 120 | 144 | 367 |
| Universe | 4359 | 1208 | 551 | 1340 |
| Uranus | 5378 | 1204 | 380 | 1162 |
| Vacuum | 1194 | 269 | 357 | 1125 |
| Venus | 5080 | 1361 | 487 | 1415 |
| Wormhole | 1588 | 551 | 146 | 563 |
| ∅ | 2055.96 | 491.92 | 285.86 | 810.42 |

### Topic: Oceanography

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Algae | 2598 | 509 | 289 | 870 |
| Aquaculture | 975 | 267 | 196 | 819 |
| Arctic Ocean | 937 | 244 | 381 | 859 |
| Argo (oceanography) | 126 | 15 | 42 | 301 |
| Atlantic Ocean | 2388 | 705 | 404 | 961 |
| Cetacea | 645 | 169 | 288 | 685 |
| Cetology | 125 | 19 | 67 | 332 |
| Chemical oceanography | 55 | 10 | 36 | 86 |
| Current sea level rise | 1404 | 264 | 176 | 904 |
| Deep sea | 284 | 100 | 76 | 381 |
| Deep sea creature | 80 | 18 | 34 | 319 |
| Dolphin | 2885 | 642 | 335 | 1267 |
| Drift ice | 89 | 9 | 52 | 154 |
| Fish | 3799 | 998 | 518 | 1490 |
| Fisheries science | 71 | 4 | 80 | 241 |
| Fishery | 395 | 69 | 101 | 260 |
| Fracture zone | 28 | 11 | 10 | 50 |
| Global warming | 7489 | 1402 | 339 | 986 |
| Gulf Stream | 786 | 207 | 136 | 472 |
| Halocline | 72 | 11 | 31 | 113 |
| Indian Ocean | 1330 | 381 | 299 | 634 |
| Jellyfish | 3791 | 1241 | 270 | 1090 |
| Marine biology | 2501 | 762 | 288 | 595 |
| Marine geology | 42 | 8 | 50 | 167 |

| | | | | |
|---|---|---|---|---|
| Marine mammal | 278 | 75 | 86 | 277 |
| Mid-ocean ridge | 617 | 193 | 104 | 304 |
| Ocean | 2892 | 842 | 302 | 686 |
| Ocean current | 911 | 259 | 151 | 314 |
| Oceanic basin | 111 | 24 | 62 | 127 |
| Oceanic plateau | 47 | 4 | 39 | 121 |
| Oceanic trench | 561 | 145 | 187 | 776 |
| Oceanography | 1541 | 409 | 192 | 537 |
| Pacific Ocean | 2421 | 744 | 440 | 839 |
| Physical oceanography | 389 | 48 | 195 | 677 |
| Planktology | 41 | 4 | 30 | 79 |
| Plankton | 1065 | 326 | 200 | 426 |
| Porpoise | 537 | 154 | 80 | 298 |
| Sapropel | 21 | 2 | 16 | 99 |
| Sea | 948 | 292 | 268 | 268 |
| Sea ice | 267 | 57 | 81 | 339 |
| Sea monster | 631 | 184 | 106 | 395 |
| Sea otter | 2610 | 518 | 269 | 1340 |
| Seamount | 199 | 53 | 76 | 378 |
| Seawater | 852 | 251 | 106 | 406 |
| Seaweed | 887 | 253 | 130 | 447 |
| Southern Ocean | 840 | 229 | 212 | 567 |
| Submarine canyon | 76 | 19 | 66 | 170 |
| Thermocline | 134 | 31 | 56 | 280 |
| Whale | 3089 | 872 | 193 | 791 |
| World Ocean Atlas | 30 | 4 | 55 | 116 |
| ∅ | 1097.80 | 281.14 | 164.00 | 501.86 |

## E.2. Data Set II

### Topic: Artificial Intelligence

| | **#Revs** | **#Users** | **#Links** | **#Terms** |
|---|---|---|---|---|
| Affective computing | 188 | 24 | 42 | 227 |
| AI-complete | 65 | 15 | 43 | 272 |
| Artificial intelligence | 3364 | 903 | 506 | 1305 |
| Automated planning and scheduling | 51 | 11 | 44 | 153 |
| Autonomous robot | 248 | 66 | 56 | 458 |
| Cognitive robotics | 83 | 18 | 30 | 220 |
| Commonsense knowledge base | 35 | 2 | 27 | 97 |
| Computational creativity | 53 | 14 | 91 | 826 |
| Computational intelligence | 138 | 38 | 46 | 113 |
| Computer vision | 686 | 161 | 78 | 508 |
| Constraint programming | 149 | 55 | 60 | 208 |

| | | | | |
|---|---|---|---|---|
| Evolutionary algorithm | 235 | 81 | 86 | 350 |
| Fuzzy control system | 131 | 32 | 59 | 618 |
| Fuzzy logic | 919 | 270 | 179 | 594 |
| Hilbert system | 186 | 9 | 49 | 211 |
| Intelligent agent | 250 | 43 | 78 | 306 |
| Knowledge representation | 321 | 88 | 136 | 496 |
| Logic programming | 387 | 68 | 122 | 615 |
| Machine learning | 524 | 165 | 106 | 338 |
| Multi-agent system | 280 | 92 | 48 | 326 |
| Robotics | 986 | 282 | 290 | 1191 |
| Software agent | 260 | 83 | 89 | 514 |
| Speech recognition | 1157 | 363 | 131 | 739 |
| Strong AI | 553 | 119 | 139 | 753 |
| Turing test | 934 | 249 | 192 | 987 |
| ∅ | 487.32 | 130.04 | 109.08 | 497.00 |

### Topic: Computer Programming

| | **#Revs** | **#Users** | **#Links** | **#Terms** |
|---|---|---|---|---|
| Boilerplate (text) | 140 | 38 | 31 | 200 |
| Branching (software) | 34 | 7 | 18 | 100 |
| Code refactoring | 447 | 128 | 123 | 348 |
| Code review | 176 | 64 | 39 | 176 |
| Coding conventions | 57 | 4 | 52 | 249 |
| Computer programming | 1766 | 546 | 178 | 588 |
| Control flow | 466 | 84 | 132 | 484 |
| Data structure | 513 | 144 | 86 | 188 |
| Debugging | 382 | 90 | 73 | 404 |
| Design pattern (computer science) | 1090 | 351 | 171 | 544 |
| Divide and conquer algorithm | 267 | 55 | 90 | 419 |
| Field encapsulation | 30 | 8 | 16 | 122 |
| High-level programming language | 321 | 98 | 70 | 241 |
| Imperative programming | 218 | 63 | 108 | 298 |
| Interpreted language | 231 | 61 | 95 | 207 |
| Interpreter (computing) | 313 | 81 | 85 | 241 |
| Low-level programming language | 161 | 49 | 36 | 119 |
| Programmer | 606 | 181 | 130 | 474 |
| Quantum programming | 46 | 10 | 33 | 213 |
| Redundant code | 48 | 5 | 17 | 60 |
| Revision control | 579 | 194 | 124 | 479 |
| Source code | 523 | 167 | 121 | 351 |
| Spaghetti code | 182 | 54 | 64 | 233 |
| Stack trace | 31 | 4 | 9 | 50 |
| Unreachable code | 75 | 18 | 26 | 143 |

| | | | | |
|---|---|---|---|---|
| ∅ | 348.08 | 100.16 | 77.08 | 277.24 |

## E.3. Data Set II

### Topic: Genetics

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Beta-catenin | 45 | 7 | 36 | 203 |
| Chromosome | 2252 | 758 | 297 | 807 |
| Chromosome abnormality | 127 | 41 | 38 | 128 |
| Combined DNA Index System | 123 | 30 | 38 | 300 |
| Cytogenetics | 164 | 37 | 106 | 583 |
| DNA | 4469 | 863 | 453 | 1165 |
| DNA sequence | 109 | 22 | 43 | 112 |
| fruitless (gene) | 60 | 7 | 36 | 190 |
| Full genome sequencing | 301 | 7 | 89 | 470 |
| Gene | 2116 | 497 | 329 | 881 |
| Gene bank | 51 | 14 | 20 | 64 |
| Gene cassette | 21 | 3 | 23 | 77 |
| Gene conversion | 68 | 21 | 21 | 132 |
| Gene family | 33 | 5 | 22 | 74 |
| Genetic analysis | 21 | 1 | 16 | 48 |
| Genetic engineering | 4167 | 1323 | 133 | 428 |
| Genetic evolution | 13 | 3 | 15 | 73 |
| Genetics | 2778 | 734 | 323 | 925 |
| Genetics of aggression | 86 | 9 | 76 | 409 |
| Genome size | 42 | 11 | 75 | 315 |
| Mutation | 1569 | 477 | 184 | 570 |
| Nanopore sequencing | 27 | 3 | 18 | 198 |
| Nucleic Acid Notations | 28 | 0 | 6 | 281 |
| Protein family | 52 | 10 | 54 | 259 |
| Telomere | 472 | 163 | 154 | 814 |
| ∅ | 767.76 | 201.84 | 104.20 | 380.24 |

### Topic: Immunology

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Allergen | 173 | 43 | 137 | 251 |
| Allergy | 1601 | 406 | 313 | 971 |
| Antibody | 1505 | 307 | 313 | 879 |

| | | | | |
|---|---|---|---|---|
| Antigen | 438 | 157 | 99 | 216 |
| Granulocyte | 128 | 23 | 117 | 353 |
| Immune system | 2874 | 740 | 439 | 1088 |
| Immune tolerance | 83 | 18 | 34 | 227 |
| Immunity (medical) | 195 | 38 | 136 | 543 |
| Immunoglobulin class switching | 37 | 11 | 41 | 138 |
| Immunology | 343 | 103 | 127 | 639 |
| Infection | 435 | 152 | 90 | 226 |
| Lymphocyte | 357 | 112 | 128 | 308 |
| Monoclonal antibodies | 359 | 101 | 153 | 654 |
| Phagocyte | 1052 | 75 | 252 | 948 |
| Polyclonal antibodies | 88 | 29 | 44 | 476 |
| T cell | 574 | 179 | 129 | 408 |
| White blood cell | 1490 | 542 | 159 | 398 |
| ∅ | 690.12 | 178.59 | 159.47 | 513.12 |

### Topic: Medical

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Clinic | 145 | 24 | 63 | 242 |
| Doctor-patient relationship | 32 | 5 | 23 | 316 |
| Hospital | 1030 | 242 | 301 | 925 |
| Medical diagnosis | 356 | 79 | 166 | 524 |
| Medical terminology | 158 | 33 | 37 | 251 |
| Medicine | 2965 | 735 | 509 | 1330 |
| Meta-Medicine | 59 | 11 | 16 | 326 |
| Nosology | 69 | 10 | 52 | 126 |
| Patient | 185 | 33 | 65 | 148 |
| Physician | 1563 | 385 | 198 | 564 |
| ∅ | 656.20 | 155.70 | 143.00 | 475.20 |

### Topic: Toxicology

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Acute toxicity | 29 | 8 | 26 | 77 |
| Asbestos | 2031 | 570 | 266 | 1430 |
| Chronic toxicity | 22 | 9 | 10 | 48 |
| Draize test | 222 | 34 | 57 | 522 |
| Drug overdose | 431 | 151 | 149 | 462 |
| Ecotoxicology | 63 | 9 | 31 | 178 |
| Fluoride poisoning | 201 | 37 | 82 | 457 |

| | | | | |
|---|---|---|---|---|
| Haber's rule | 14 | 1 | 20 | 65 |
| Lead poisoning | 876 | 249 | 229 | 1119 |
| Marsh test | 82 | 12 | 46 | 272 |
| Poison | 942 | 247 | 237 | 554 |
| Sarin | 541 | 171 | 169 | 580 |
| Toxicology | 531 | 149 | 90 | 259 |
| Toxin | 427 | 105 | 117 | 295 |
| ∅ | 458.00 | 125.14 | 109.21 | 451.29 |

## E.4. Data Set IV

### Topic: Astronomy [5]

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Asteroid | 2681 | 634 | 346 | 1000 |
| Astronomy | 3582 | 976 | 526 | 1224 |
| Big Bang | 4970 | 996 | 364 | 1039 |
| Galaxy | 1557 | 376 | 354 | 935 |
| Universe | 4359 | 1208 | 551 | 1340 |
| ∅ | 3429.80 | 838.00 | 428.20 | 1107.60 |

### Topic: Genetics [5]

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Chromosome | 2252 | 758 | 297 | 807 |
| DNA | 4469 | 863 | 453 | 1165 |
| Gene | 2116 | 497 | 329 | 881 |
| Gene family | 33 | 5 | 22 | 74 |
| Genetics | 2778 | 734 | 323 | 925 |
| ∅ | 2329.60 | 571.40 | 284.80 | 770.40 |

### Topic: Oceanography [5]

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Algae | 2598 | 509 | 289 | 870 |
| Aquaculture | 975 | 267 | 196 | 819 |
| Fishery | 395 | 69 | 101 | 260 |

| | | | | |
|---|---|---|---|---|
| Ocean | 2892 | 842 | 302 | 686 |
| Oceanography | 1541 | 409 | 192 | 537 |
| ∅ | 1680.20 | 419.20 | 216.00 | 634.40 |

### Topic: Computer Programming [5]

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Code refactoring | 447 | 128 | 123 | 348 |
| Computer programming | 1766 | 546 | 178 | 588 |
| Debugging | 382 | 90 | 73 | 404 |
| Programmer | 606 | 181 | 130 | 474 |
| Spaghetti code | 182 | 54 | 64 | 233 |
| ∅ | 676.60 | 199.80 | 113.60 | 409.40 |

### Topic: Tourism [5]

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Holiday | 2088 | 616 | 153 | 340 |
| Tour guide | 225 | 54 | 26 | 101 |
| Tourism | 3198 | 893 | 331 | 875 |
| Tourism geography | 74 | 16 | 31 | 140 |
| World Tourism Organization | 169 | 32 | 47 | 327 |
| ∅ | 1150.80 | 322.20 | 117.60 | 356.60 |

### Topic: Toxicology [5]

| | #Revs | #Users | #Links | #Terms |
|---|---|---|---|---|
| Acute toxicity | 29 | 8 | 26 | 77 |
| Chronic toxicity | 22 | 9 | 10 | 48 |
| Poison | 942 | 247 | 237 | 554 |
| Toxicology | 531 | 149 | 90 | 259 |
| Toxin | 427 | 105 | 117 | 295 |
| ∅ | 390.20 | 103.60 | 96.00 | 246.60 |

# F. Analyses Results

## F.1. Clustering Case 1 - Clustering Algorithm

| | PAM | bisPAM | Agnes | Diana | ∅ |
|---|---|---|---|---|---|
| Data Set I | 0.518 | 0.518 | 0.355 | 0.980 | 0.593 |
| Data Set II | 0.626 | 0.626 | 0.376 | 0.920 | 0.637 |
| Data Set III | 0.530 | 0.383 | 0.164 | 0.564 | 0.410 |
| Data Set IV | 0.897 | 0.533 | 0.625 | 0.598 | 0.663 |
| ∅ | 0.643 | 0.515 | 0.380 | 0.766 | 0.5760 |

**Table F.1.:** F-measure: Content-based clustering

| | PAM | bisPAM | Agnes | Diana | ∅ |
|---|---|---|---|---|---|
| Data Set I | 1.99 | 1.99 | 1.54 | 1.93 | 1.862 |
| Data Set II | 1.79 | 1.79 | 1.45 | 1.72 | 1.688 |
| Data Set III | 1.44 | 1.61 | 0.7 | 1.27 | 1.253 |
| Data Set IV | 1.19 | 0.88 | 0.75 | 0.95 | 0.943 |
| ∅ | 1.601 | 1.567 | 1.109 | 1.468 | 1.436 |

**Table F.2.:** Dunn Index: Content-based clustering

| | PAM | bisPAM | Agnes | Diana | ∅ |
|---|---|---|---|---|---|
| Data Set I | 3.29 | 3.29 | 1.22 | 3.27 | 2.768 |
| Data Set II | 3.2 | 3.2 | 1.38 | 3.12 | 2.725 |
| Data Set III | 7.74 | 8.48 | 1.54 | 6.93 | 6.175 |
| Data Set IV | 11.78 | 8.45 | 6.74 | 9.35 | 9.083 |
| ∅ | 6.505 | 5.857 | 2.720 | 5.669 | 5.188 |

**Table F.3.:** SD Validity Index: Content-based clustering

| | PAM | bisPAM | Agnes | Diana | ∅ |
|---|---|---|---|---|---|
| Data Set I | 1293.840 | 2133.777 | 1278.988 | 1264.008 | 1492.653 |
| Data Set II | 487.816 | 937.971 | 465.899 | 429.096 | 580.196 |
| Data Set III | 685.785 | 2050.063 | 737.135 | 750.041 | 1055.756 |
| Data Set IV | 451.773 | 2062.323 | 509.853 | 503.363 | 881.828 |
| ∅ | 729.803 | 1796.033 | 747.969 | 736.627 | 1002.608 |

**Table F.4.:** Computing Time: Content-based clustering

| | PAM | bisPAM | Agnes | Diana | ∅ |
|---|---|---|---|---|---|
| Data Set I | 0.725 | 0.960 | 0.970 | 0.366 | 0.755 |
| Data Set II | 0.670 | 0.670 | 0.980 | 0.556 | 0.719 |
| Data Set III | 0.535 | 0.551 | 0.673 | 0.554 | 0.578 |
| Data Set IV | 0.912 | 0.543 | 0.841 | 0.704 | 0.750 |
| ∅ | 0.711 | 0.681 | 0.866 | 0.545 | 0.701 |

**Table F.5.:** F-measure: Link-based clustering

| | PAM | bisPAM | Agnes | Diana | ∅ |
|---|---|---|---|---|---|
| Data Set I | 1.86 | 1.86 | 1.84 | 1.82 | 1.843 |
| Data Set II | 1.76 | 1.76 | 1.87 | 1.63 | 1.755 |
| Data Set III | 1.8 | 1.37 | 1.39 | 1.55 | 1.528 |
| Data Set IV | 1.02 | 0.85 | 0.76 | 0.93 | 0.889 |
| ∅ | 1.610 | 1.460 | 1.464 | 1.481 | 1.504 |

**Table F.6.:** Dunn Index: Link-based clustering

| | PAM | bisPAM | Agnes | Diana | ∅ |
|---|---|---|---|---|---|
| Data Set I | 3.23 | 3.23 | 3.24 | 4.37 | 3.520 |
| Data Set II | 3.25 | 3.25 | 3.28 | 4.94 | 3.680 |
| Data Set III | 7.91 | 7.79 | 6.52 | 11.27 | 8.370 |
| Data Set IV | 11.97 | 8.37 | 9.85 | 8.01 | 9.549 |
| ∅ | 6.589 | 5.662 | 5.721 | 7.147 | 6.280 |

**Table F.7.:** SD Validity Index: Link-based clustering

| | PAM | bisPAM | Agnes | Diana | ∅ |
|---|---|---|---|---|---|
| Data Set I | 1154.210 | 2023.773 | 1147.287 | 1155.937 | 1370.302 |
| Data Set II | 282.399 | 774.602 | 309.310 | 312.428 | 419.685 |
| Data Set III | 528.102 | 1800.646 | 465.871 | 442.660 | 809.320 |
| Data Set IV | 349.029 | 1825.790 | 352.498 | 355.586 | 720.726 |
| ∅ | 578.435 | 1606.203 | 568.741 | 566.653 | 830.008 |

**Table F.8.:** Computing Time: Link-based clustering

## F.2. Clustering Case 2 - Feature Selection

| | 100% | 90% | 80% | 70% | 60% | 50% | 40% | 30% | 20% | 10% |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.98 | 0.84 | 0.84 | 0.66 | 0.66 | 0.66 | 0.42 | 0.38 | 0.38 | 0.36 |
| Data Set II | 0.92 | 0.71 | 0.71 | 0.43 | 0.45 | 0.45 | 0.45 | 0.49 | 0.38 | 0.38 |
| Data Set III | 0.56 | 0.56 | 0.44 | 0.46 | 0.45 | 0.36 | 0.28 | 0.33 | 0.25 | 0.25 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set IV | 0.60 | 0.60 | 0.60 | 0.84 | 0.60 | 0.75 | 0.46 | 0.36 | 0.28 | 0.28 |
| Average | 0.77 | 0.68 | 0.65 | 0.60 | 0.54 | 0.56 | 0.40 | 0.39 | 0.32 | 0.31 |

**Table F.9.:** F-measure: Content-based clustering, Diana

| | **100%** | **90%** | **80%** | **70%** | **60%** | **50%** | **40%** | **30%** | **20%** | **10%** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.52 | 0.52 | 0.52 | 0.51 | 0.53 | 0.76 | 0.77 | 0.77 | 0.70 | 0.36 |
| Data Set II | 0.63 | 0.63 | 0.63 | 0.63 | 0.65 | 0.62 | 0.57 | 0.48 | 0.38 | 0.38 |
| Data Set III | 0.53 | 0.53 | 0.53 | 0.53 | 0.49 | 0.46 | 0.44 | 0.45 | 0.17 | 0.17 |
| Data Set IV | 0.90 | 0.90 | 0.90 | 0.90 | 0.87 | 0.87 | 0.66 | 0.56 | 0.33 | 0.33 |
| Average | 0.64 | 0.64 | 0.64 | 0.64 | 0.63 | 0.68 | 0.61 | 0.57 | 0.39 | 0.31 |

**Table F.10.:** F-measure: Content-based clustering, PAM

| | **100%** | **90%** | **80%** | **70%** | **60%** | **50%** | **40%** | **30%** | **20%** | **10%** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.97 | 0.97 | 0.98 | 0.36 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Data Set II | 0.98 | 0.98 | 0.96 | 0.96 | 0.38 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Data Set III | 0.67 | 1.00 | 0.20 | 0.20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Data Set IV | 0.84 | 0.97 | 0.84 | 0.28 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Average | 0.87 | 0.98 | 0.74 | 0.45 | 0.19 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

**Table F.11.:** F-measure: Link-based clustering, Agnes

| | **100%** | **90%** | **80%** | **70%** | **60%** | **50%** | **40%** | **30%** | **20%** | **10%** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.73 | 0.93 | 0.87 | 0.61 | 0.61 | 0.36 | 0.36 | 0.36 | 0.36 | 0.36 |
| Data Set II | 0.67 | 0.49 | 0.49 | 0.45 | 0.38 | 0.38 | 0.38 | 0.38 | 0.38 | 0.38 |
| Data Set III | 0.54 | 0.55 | 0.50 | 0.46 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 | 0.17 |
| Data Set IV | 0.91 | 0.89 | 0.87 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 | 0.33 |
| Average | 0.71 | 0.72 | 0.68 | 0.47 | 0.37 | 0.31 | 0.31 | 0.31 | 0.31 | 0.31 |

**Table F.12.:** F-measure: Link-based clustering, PAM

## F.3. Clustering Case 3 - Feature Transformation

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 44.58 | 37.06 | 33.84 | 33.02 | 31.88 | 31.41 | 31.24 | 31.01 | 30.90 | 30.68 |
| Data Set II | 35.38 | 32.11 | 27.95 | 27.23 | 23.65 | 22.41 | 20.66 | 20.49 | 19.22 | 18.77 |
| Data Set III | 40.89 | 33.84 | 32.84 | 32.13 | 28.68 | 25.66 | 25.09 | 24.81 | 24.55 | 24.47 |
| Data Set IV | 31.62 | 27.76 | 27.61 | 26.49 | 26.27 | 25.74 | 25.41 | 23.77 | 23.21 | 22.51 |

| | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** | **55** | **60** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 27.68 | 26.27 | 24.94 | 22.97 | 21.46 | 20.46 | 18.33 | 17.28 | 16.22 | 15.63 |
| Data Set II | 16.69 | 15.62 | 13.52 | 11.03 | 9.77 | 8.61 | 6.95 | 3.88 | | |
| Data Set III | 21.51 | 19.27 | 17.92 | 15.26 | 13.88 | 12.71 | 12.08 | 10.46 | 8.73 | 6.41 |

| | | | | |
|---|---|---|---|---|
| Data Set IV | 18.62 | 14.69 | 10.53 | 3.53 |

| | **65** | **66** | **70** | **75** | **80** | **85** | **90** | **95** | **100** | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 14.37 | 13.06 | 11.97 | 11.32 | 9.78 | 8.04 | 6.89 | 4.74 | | |
| Data Set II | | | | | | | | | | |
| Data Set III | 3.22 | | | | | | | | | |
| Data Set IV | | | | | | | | | | |

**Table F.13.:** Singular values: Content-based clustering, no feature selection

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 44.58 | 37.06 | 33.83 | 33.02 | 31.86 | 31.41 | 31.23 | 31.01 | 30.90 | 30.68 |
| Data Set II | 35.36 | 32.09 | 27.93 | 27.22 | 23.64 | 22.39 | 20.64 | 20.48 | 19.21 | 18.76 |
| Date Set III | 40.87 | 33.79 | 32.84 | 32.11 | 28.64 | 25.57 | 25.06 | 24.79 | 24.54 | 24.46 |
| Data Set IV | 31.52 | 27.68 | 27.54 | 26.38 | 26.23 | 25.67 | 25.39 | 23.72 | 23.13 | 22.47 |

| | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** | **55** | **60** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 27.68 | 26.27 | 24.93 | 22.96 | 21.45 | 20.45 | 18.32 | 17.28 | 16.22 | 15.62 |
| Data Set II | 16.69 | 15.61 | 13.51 | 11.02 | 9.76 | 8.59 | 6.95 | 3.87 | | |
| Date Set III | 21.50 | 19.25 | 17.91 | 15.24 | 13.86 | 12.70 | 12.06 | 10.45 | 8.73 | 6.41 |
| Data Set IV | 18.58 | 14.67 | 10.51 | 3.50 | | | | | | |

| | **65** | **66** | **70** | **75** | **80** | **85** | **90** | **95** | **100** | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 14.37 | 13.06 | 11.96 | 11.32 | 9.78 | 8.04 | 6.88 | 2.95 | | |
| Data Set II | | | | | | | | | | |
| Date Set III | 3.21 | | | | | | | | | |
| Data Set IV | | | | | | | | | | |

**Table F.14.:** Singular values: Content-based clustering, 50% feature selection

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 41.04 | 37.12 | 36.36 | 35.82 | 35.49 | 34.90 | 34.54 | 34.36 | 32.53 | 32.21 |
| Data Set II | 31.20 | 26.27 | 19.99 | 19.81 | 19.70 | 18.45 | 18.19 | 17.62 | 16.98 | 16.77 |
| Data Set III | 34.68 | 29.28 | 28.50 | 27.92 | 27.10 | 25.52 | 25.07 | 24.06 | 23.96 | 23.79 |
| Data Set IV | 29.97 | 28.38 | 26.00 | 25.38 | 25.03 | 23.66 | 22.76 | 22.68 | 22.35 | 22.19 |

| | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** | **55** | **60** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 30.89 | 29.73 | 28.33 | 27.12 | 25.69 | 23.58 | 21.73 | 20.36 | 18.76 | 17.61 |
| Data Set II | 14.68 | 13.58 | 12.68 | 10.82 | 9.22 | 8.18 | 5.97 | 3.87 | | |
| Data Set III | 20.37 | 18.31 | 15.96 | 14.50 | 13.61 | 11.32 | 10.04 | 8.57 | 7.33 | 5.44 |
| Data Set IV | 17.83 | 14.01 | 9.22 | 3.50 | | | | | | |

| | **65** | **66** | **70** | **75** | **80** | **85** | **90** | **95** | **100** | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 15.64 | 14.75 | 13.69 | 12.41 | 10.98 | 9.72 | 7.58 | 4.51 | | |
| Data Set II | | | | | | | | | | |
| Data Set III | 3.81 | | | | | | | | | |
| Data Set IV | | | | | | | | | | |

**Table F.15.:** Singular values: Link-based clustering, 90% feature selection

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.27 | 0.58 | 0.49 | 0.52 | 0.75 | 0.58 | 0.44 | 0.78 | 0.44 | 0.51 |
| Data Set II | 0.48 | 0.54 | 0.46 | 0.54 | 0.60 | 0.60 | 0.45 | 0.57 | 0.60 | 0.50 |
| Data Set III | 0.27 | 0.35 | 0.28 | 0.28 | 0.58 | 0.53 | 0.56 | 0.56 | 0.60 | 0.55 |
| Data Set IV | 0.37 | 0.35 | 0.56 | 0.43 | 0.63 | 0.52 | 0.39 | 0.50 | 0.32 | 0.35 |

| | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** | **55** | **60** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.85 | 0.79 | 0.93 | 0.77 | 0.78 | 0.75 | 0.75 | 0.81 | 0.64 | 0.74 |
| Data Set II | 0.80 | 0.66 | 0.45 | 0.84 | 0.84 | 0.49 | 0.49 | 0.92 | | |
| Data Set III | 0.52 | 0.33 | 0.40 | 0.43 | 0.55 | 0.32 | 0.32 | 0.40 | 0.50 | 0.30 |
| Data Set IV | 0.38 | 0.41 | 0.45 | 0.60 | | | | | | |

| | **65** | **66** | **70** | **75** | **80** | **85** | **90** | **95** | **100** | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.72 | | 0.83 | 0.86 | 0.38 | 0.30 | 0.32 | 0.29 | 0.98 | |
| Data Set II | | | | | | | | | | |
| Data Set III | 0.43 | 0.56 | | | | | | | | |
| Data Set IV | | | | | | | | | | |

**Table F.16.:** F-measure: Content-based clustering, Diana

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.57 | 0.57 | 0.54 | 0.60 | 0.83 | 0.73 | 0.56 | 0.72 | 0.69 | 0.68 |
| Data Set II | 0.66 | 0.50 | 0.50 | 0.54 | 0.46 | 0.48 | 0.31 | 0.50 | 0.54 | 0.61 |
| Data Set III | 0.32 | 0.38 | 0.41 | 0.37 | 0.60 | 0.77 | 0.71 | 0.66 | 0.73 | 0.59 |
| Data Set IV | 0.41 | 0.42 | 0.48 | 0.55 | 0.59 | 0.44 | 0.50 | 0.52 | 0.45 | 0.49 |

| | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** | **55** | **60** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.82 | 0.69 | 0.75 | 0.76 | 0.80 | 0.84 | 0.79 | 0.77 | 0.74 | 0.67 |
| Data Set II | 0.74 | 0.66 | 0.74 | 0.68 | 0.60 | 0.75 | 0.56 | 0.62 | | |
| Data Set III | 0.59 | 0.47 | 0.51 | 0.41 | 0.46 | 0.49 | 0.51 | 0.43 | 0.38 | 0.28 |
| Data Set IV | 0.59 | 0.70 | 0.48 | 0.87 | | | | | | |

| | **65** | **66** | **70** | **75** | **80** | **85** | **90** | **95** | **100** | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.68 | | 0.59 | 0.60 | 0.56 | 0.65 | 0.60 | 0.59 | 0.76 | |
| Data Set II | | | | | | | | | | |
| Data Set III | 0.39 | 0.53 | | | | | | | | |
| Data Set IV | | | | | | | | | | |

**Table F.17.:** F-measure: Content-based clustering, PAM

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.36 | 0.73 | 0.71 | 0.66 | 0.67 | 0.88 | 0.35 | 0.35 | 0.92 | 0.35 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set II | 0.38 | 0.54 | 0.46 | 0.49 | 0.76 | 0.48 | 0.37 | 0.37 | 0.37 | 0.82 |
| Data Set III | 0.21 | 0.48 | 0.65 | 0.66 | 0.62 | 0.55 | 0.63 | 0.43 | 0.63 | 0.64 |
| Data Set IV | 0.30 | 0.33 | 0.41 | 0.39 | 0.35 | 0.50 | 0.47 | 0.48 | 0.39 | 0.43 |

| | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** | **55** | **60** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.36 | 0.36 | 0.36 | 0.36 | 0.36 | 0.36 | 0.36 | 0.36 | 0.36 | 0.36 |
| Data Set II | 0.38 | 0.38 | 0.38 | 0.38 | 0.38 | 0.38 | 0.38 | 0.98 | | |
| Data Set III | 0.57 | 0.44 | 0.20 | 0.17 | 0.17 | 0.22 | 0.22 | 0.22 | 0.26 | 0.37 |
| Data Set IV | 0.49 | 0.36 | 0.73 | 0.97 | | | | | | |

| | **65** | **66** | **70** | **75** | **80** | **85** | **90** | **95** | **100** | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.36 | | 0.40 | 0.40 | 0.42 | 0.36 | 0.36 | 0.36 | 0.97 | |
| Data Set II | | | | | | | | | | |
| Data Set III | 1.00 | 1.00 | | | | | | | | |
| Data Set IV | | | | | | | | | | |

**Table F.18.:** F-measure: Link-based clustering, Agnes

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.54 | 0.76 | 0.70 | 0.72 | 0.66 | 0.90 | 0.92 | 0.94 | 0.90 | 0.93 |
| Data Set II | 0.61 | 0.58 | 0.76 | 0.78 | 0.76 | 0.76 | 0.82 | 0.80 | 0.84 | 0.82 |
| Data Set III | 0.33 | 0.46 | 0.71 | 0.70 | 0.76 | 0.79 | 0.79 | 0.59 | 0.85 | 0.82 |
| Data Set IV | 0.32 | 0.34 | 0.42 | 0.46 | 0.43 | 0.50 | 0.49 | 0.61 | 0.53 | 0.49 |

| | **15** | **20** | **25** | **30** | **35** | **40** | **45** | **50** | **55** | **60** |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.92 | 0.84 | 0.83 | 0.89 | 0.88 | 0.87 | 0.83 | 0.76 | 0.82 | 0.80 |
| Data Set II | 0.90 | 0.86 | 0.82 | 0.78 | 0.80 | 0.72 | 0.29 | 0.49 | | |
| Data Set III | 0.90 | 0.87 | 0.77 | 0.59 | 0.61 | 0.79 | 0.76 | 0.73 | 0.56 | 0.42 |
| Data Set IV | 0.43 | 0.68 | 0.86 | 0.89 | | | | | | |

| | **65** | **66** | **70** | **75** | **80** | **85** | **90** | **95** | **100** | |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Set I | 0.75 | | 0.76 | 0.73 | 0.72 | 0.71 | 0.53 | 0.57 | 0.93 | |
| Data Set II | | | | | | | | | | |
| Data Set III | 0.55 | 0.55 | | | | | | | | |
| Data Set IV | | | | | | | | | | |

**Table F.19.:** F-measure: Link-based clustering, PAM

## F.4. Clustering Case 4 - Content-based vs. Link-based Clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 0.838 | 0.980 | 0.909 |
| Data Set II | 0.754 | 0.920 | 0.837 |
| Data Set III | 0.767 | 0.602 | 0.685 |

| | | | |
|---|---|---|---|
| Data Set IV | 0.868 | 0.634 | 0.751 |
| ∅ | 0.807 | 0.784 | 0.795 |

**Table F.20.:** F-measure: Content-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 1.803 | 1.930 | 1.866 |
| Data Set II | 1.756 | 1.724 | 1.740 |
| Data Set III | 0.804 | 0.575 | 0.690 |
| Data Set IV | 1.278 | 0.325 | 0.802 |
| ∅ | 1.410 | 1.139 | 1.274 |

**Table F.21.:** Dunn Index: Content-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 3.284 | 3.269 | 3.277 |
| Data Set II | 3.112 | 3.120 | 3.116 |
| Data Set III | 7.872 | 7.059 | 7.466 |
| Data Set IV | 11.814 | 11.620 | 11.717 |
| ∅ | 6.521 | 6.267 | 6.394 |

**Table F.22.:** SD Validity Index: Content-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 4922.603 | 1264.008 | 3093.305 |
| Data Set II | 1343.771 | 429.096 | 886.434 |
| Data Set III | 2401.395 | 2118.981 | 2260.188 |
| Data Set IV | 726.436 | 708.772 | 717.604 |
| ∅ | 2348.551 | 1130.214 | 1739.383 |

**Table F.23.:** Computing Time: Content-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 0.940 | 0.970 | 0.955 |
| Data Set II | 0.899 | 0.980 | 0.939 |
| Data Set III | 0.899 | 1.000 | 0.949 |
| Data Set IV | 0.894 | 0.966 | 0.930 |
| ∅ | 0.908 | 0.979 | 0.944 |

**Table F.24.:** F-measure: Link-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 1.735 | 1.835 | 1.785 |
| Data Set II | 1.931 | 1.872 | 1.901 |
| Data Set III | 1.136 | 1.669 | 1.402 |

| | | | |
|---|---|---|---|
| Data Set IV | 0.964 | 1.024 | 0.994 |
| ∅ | 1.441 | 1.600 | 1.521 |

**Table F.25.:** Dunn Index: Link-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 3.681 | 3.240 | 3.460 |
| Data Set II | 3.488 | 3.280 | 3.384 |
| Data Set III | 8.899 | 8.441 | 8.670 |
| Data Set IV | 11.338 | 11.935 | 11.636 |
| ∅ | 6.852 | 6.724 | 6.788 |

**Table F.26.:** SD Validity Index: Link-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 4412.953 | 2590.972 | 3501.962 |
| Data Set II | 744.540 | 525.375 | 634.958 |
| Data Set III | 1921.916 | 821.437 | 1371.677 |
| Data Set IV | 561.464 | 591.048 | 576.256 |
| ∅ | 1910.218 | 1132.208 | 1521.213 |

**Table F.27.:** Computing Time: Link-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 1.600 | 1.829 | 1.715 |
| Data Set II | 1.900 | 1.922 | 1.911 |
| Data Set III | 0.445 | 0.614 | 0.530 |
| Data Set IV | 1.177 | 0.087 | 0.632 |
| ∅ | 1.280 | 1.113 | 1.197 |

**Table F.28.:** Dunn Index: Optimal solution for content-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 3.322 | 3.271 | 3.296 |
| Data Set II | 3.220 | 3.213 | 3.217 |
| Data Set III | 6.403 | 6.511 | 6.457 |
| Data Set IV | 12.043 | 11.168 | 11.606 |
| ∅ | 6.247 | 6.041 | 6.144 |

**Table F.29.:** SD Validity Index: Optimal solution for content-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 1.93 | 1.89 | 1.912 |
| Data Set II | 1.87 | 1.95 | 1.911 |
| Data Set III | 0.89 | 1.51 | 1.197 |

| | | | |
|---|---|---|---|
| Data Set IV | 1.11 | 1.11 | 1.109 |
| ∅ | 1.450 | 1.614 | 1.532 |

**Table F.30.:** Dunn Index: Optimal solution for link-based clustering

| | **PAM*** | **Diana*** | ∅ |
|---|---|---|---|
| Data Set I | 3.77 | 3.23 | 3.500 |
| Data Set II | 3.43 | 3.28 | 3.355 |
| Data Set III | 6.5 | 7.6 | 7.049 |
| Data Set IV | 12.03 | 12.03 | 12.033 |
| ∅ | 6.432 | 6.537 | 6.484 |

**Table F.31.:** SD Validity Index: Optimal solution for link-based clustering

# G. German Summary

Das Ziel dieser Arbeit ist die Identifizierung einer Methode, die es ermöglicht *Content Elemente* (z.B. Wiki Artikel, Blogeinträge,...) eines sozialen Informationsraumes zu Themen zusammenzufassen. In diesem Zusammenhang werden zwei Ansätze miteinander verglichen. Der link-basierte Ansatz benutzt die Links eines *Content Elements* für seine Beschreibung. Der inhalts-basierte Ansatz verwendet die Terme eines *Content Elementes*. Das Ziel dieser Arbeit ist in einer Forschungsfrage in Kapitel 2 (p. 9) zusammengefasst. Der sozialen Informationsraum für die Überprüfung der Ansätze sind Wikis. Dabei werden zunächst Wiki Artikel zu Datensätzen zusammengefasst. Innerhalb dieser Datensätze befinden sich verschiedene Themen, die durch die beiden Ansätze identifiziert werden sollen.
Zunächst werden die Datensätze in ein allgemeines Datenmodell überführt. Basierend auf diesem Datenmodell können Methoden, unabhängig von einen sozialen Informationsraum implementiert werden. Das Datenmodell ist Teil des SONIVIS Tools[1]. Eine Beschreibung des Datenmodells befindet sich in Kapitel 3 (p. 13).
Nachdem die Daten für die Analyse feststehen, ist es erforderlich mögliche Analysemethoden für die Identifizierung von Themen zu bestimmen. Dafür werden in Kapitel 4 (p. 23) verschiedene Textanalyse-Methoden vorgestellt und bewertet. Die Basis für die Bestimmung von Themen bildet das Clustering. Dabei repräsentieren die entstehenden Cluster die Themen. Demzufolge ist es notwendig geeignete Clusteringalgorithmen auszuwählen. Außerdem ist es möglich die Ergebnisse durch die Anwendung von Dimensionsreduzierungsverfahren zu verbessern. Somit ist es auch erforderlich geeignete Algorithmen für die Reduzierung der Dimensionen zu bestimmen. Die ausgewählten Methoden wurden in Form einer Metrik in das SONIVIS Tool implementiert. Die Metrik gibt die Cluster ID für jedes *Content Element* zurück. Zusätzlich wird die interne Qualität der Cluster berechnet und ausgegeben.
Basierend auf den ausgewählten Datensets und der implementierten Metrik, werden Analysen für die Bewertung der Ansätze durchgeführt. Eine Übersicht über die Datensets und die angewandte Methodik befindet sich in Kapitel 5 (p. 87). Die Ergebnisse wurden mit einer externen Qualitätsanalyse (F-measure), einer internen Qualitätsanalyse (Dunn Index, SD Validity Index) und einer Performanceanalyse (Berechnungszeit) bewertet. In Kapitel 6 (p. 97) sind die verschiedenen Analysen und deren Ergebnisse beschrieben. Die Analysen sind in 4 Phasen unterteilt. Die erste Phase bestimmt geeignete Clusteringalgorithmen und vergleicht den link-basierten und inhalts-basierten Clusteringansatz ohne Dimensionsreduzierung. Die Clusteringalgorithmen PAM, bisecting PAM, Agnes and Diana (beschrieben in Abschnitt 4.3.4, p. 63) wurden innerhalb dieser Phase benutzt. Die nächsten zwei Phasen beschäftigen sich mit der Auswahl geeigneter Parameter für die Dimensionsreduzierung. Für die Dimensionsreduzierung wurden die *Feature Selection* Methode *Term Weight Thresholding* (beschrieben in Abschnitt 4.3.3, p. 60) und die *Feature Transformation* Methode *Latent Semantic Analysis* (beschrieben in Abschnitt 4.3.3, p. 48) ausgewählt. Die letzte Phase vergleicht den link-basierten und inhalts-basierten Clusteringansatz unter Benutzung der Dimensionsreduzierungsmethoden und bewertet die Verbesserungen, die durch die Anwendung der Methoden für die Dimensionsreduzierungs erreicht wurden.

Die Analysen zeigen das die Qualität der Cluster von dem Clusteringalgorithmus, der Dimensionsreduzierungsmethode, dem ausgewählten Clusteringansatz und der Struktur der Daten (Ähnlichkeit der Themen, Größe der Artikel, Größe der Themen, etc.) abhängt. Die besten Ergebnisse wurden mit dem link-basierten Clusteringansatz erreicht. Die Glaubhaftigkeit der internen Qualitätsanalyse ist zweifelhaft, da sich die Ergebnisse der internen Qualitätsanalyse (Dunn Index and SD Validity Index)

[1] SONIVIS ist eine Open Source Software für die Analyse und die Visualisierung von virtuellen Informationsräumen. http://www.sonivis.org/

von den Ergebnissen der externen Qualitätsanalyse (F-measure) unterscheiden.
Die Ergebnisse der ersten Phase (Abschnitt 6.1, p. 98) zeigen, dass der link-basierte Clusteringansatz bessere Ergebnisse erzielt als der inhalts-basierte Clusteringansatz, wenn keine Dimensionsreduzierung vorgenommen wird. Die Ergebnisse des inhaltsbasierten Clusteringansatzes entsprechen Arbeiten von Zhao et al. [243] and Steinbach et al. [204]. Zhao et al. vergleicht divisive und agglomerative Clusteringalgorithmen und entdeckt, dass sich divisives Clustering mehr für das Clustering von Texten eignet. Innerhalb dieser Arbeit ist der divise Clusteringalgorithmus Diana besser für das inhalts-basierte Clustering geeignet als der agglomerative Clusteringalgorithmus Agnes. Steinbach et al. vergleicht agglomeratives hierarchisches Clustering, k-means und bisecting k-means. Die Ergebnisse zeigen, dass k-means bessere Cluster generiert als agglomeratives hierarchisches Clustering und bisecting k-means die beste Clusteringtechnik ist. Die Ergebnisse dieser Arbeit zeigen, dass PAM (eine Variante von k-means) besser ist als agglomeratives hierarchisches Clustering. Nur die Ergebnisse von bisecting PAM sind anders als in der Arbeit von Steinbach et al.. Hier ist bisecting PAM der schlechteste Clusteringalgorithmus bezüglich der externen Qualitätsanalyse (F-measure) und der Performanceanalyse (Berechnungszeit).
Die Ergebnisse des link-basierten Clusterings sind anders. Sehr überraschend ist der starke Unterschied zwischen den Clusteringalgorithmen Agnes und Diana bezüglich des link-basierten und inhaltsbasierten Clusterings. Bei der Benutzung von link-basiertem Clustering erreicht agglomeratives hierarchisches Clustering bessere Ergebnisse als divises hierarchisches Clustering. Dieser Fakt verdeutlicht, wie unterschiedlich die Struktur der Daten bei dem link-basierten und inhalts-basierten Ansatz ist. Die Term Dokument Matrix des inhalts-basierten Ansatzes (basierend auf den Termen des Textes) ist völlig anders als die Term Dokument Matrix des link-basierten Ansatzes (basierend auf den Links eines Textes). Der Unterschied der Datensätze wird zusätzlich durch die interne Qualitätsanalyse bestärkt. Die Frage ist nun, ob Agnes auch bei anderen Datensets gute Ergebnisse erzielt.
Die nächsten zwei Phasen (Abschnitt 6.2, p. 105 und Abschnitt 6.3, p. 109) bestimmen geeignete Werte für die Dimensionsreduzierungsmethoden. Die gewählten Werte sind nur für die ausgewählten Datensets und Clusteringmethoden geeignet. Wenn ein anderes Datenset oder eine andere Clusteringmethode benutzt werden, müssen die Werte neu bestimmt werden. Für weitere Analysen und die Anwendung der Themenidentifizierungs-Methoden in der Praxis sollten andere Ansätze für die Bestimmung der Dimensionsreduzierungs-Parameter benutzt werden. Die hier benutzten Ansätze sind komplex und eine Änderung des Datensets verursacht eine Neubestimmung der Parameter. *Feature Selection* mit *Term Weight Thresholding* arbeitet mit einem prozentualen Grenzwert. Die Benutzung von prozentualen Grenzwerten ist besser als die Benutzung von absoluten Grenzwerten. Aber auf Basis des inhalts-basierten Clusterings sind die prozentualen Grenzwerte verschieden. Wenn der link-basierte Ansatz gewählt wird, scheint ein Grenzwert von 90% geeignet zu sein. Aber die Gültigkeit dieses Grenzwertes muss mit weiteren Datensets überprüft werden. Die Bestimmung von passenden Werten (Anzahl der Dimensionen) für *Feature Transformation* mit *Latent Semantic Analysis* ist komplexer. In dieser Arbeit wurde ein Wert für jeden Clusteringansatz, Clusteringalgorithmus und für jedes Datenset definiert. Da die bestimmten Werte sehr unterschiedlich sind, kann kein Standardwert definiert werden. Bedauerlicherweise konnte auch kein Wert mit Hilfe der *Singular Values* bestimmt werden. Eventuell ist auch hier eine prozentuale Angabe hilfreich.
In der letzten Phase (Abschnitt 6.4, p. 115) wurden die Verbesserungen durch die Benutzung der Dimensionsreduzierung bewertet und der link-basierte mit dem inhalts-basierten Ansatz verglichen. Im Allgemeinen verbessert die Dimensionsreduzierung die Clusterqualität, aber die Berechnungszeit steigt ebenfalls an. Außerdem können die gewählten Parameter bewirken, dass die Ergebnisse von einigen Datensets verbessert werden, aber die Ergebnisse von anderen Datensets verschlechtern sich. Somit muss abgewogen werden, ob die Benutzung der Methoden sinnvoll ist. Die Ergebnisse entsprechen Arbeiten von Tang et al. [209] [210]. Tang et al. vergleicht *Latent Semantic Analysis*, *Document Frequency Thresholding*, *mean TFIDF* und weitere Dimensionsreduzierungsmethoden. Die Methoden *Document Frequency Thresholding* und *mean TFIDF* sind ähnlich der Methode *Term Weight Thresholding*. Die Messungen zeigen, dass *Latent Semantic Analysis*, *Document Frequency Thresholding* und *mean TFIDF* geeignete Methoden für das Clustering von Dokumenten sind. Die beste Dimensionsreduzierungsmethode in dieser Arbeit ist die *Independent Component Analysis*. Im Zusammenhang mit dem Themenclustering scheint *Feature Selection* besser geeignet zu sein als *Fea-*

*ture Transformation*. Mit einem Grenzwert von 90% clustert der link-basierte Ansatz sehr gut und es ist sichergestellt, dass nicht so viele wichtige Terme entfernt werden. Außerdem ist *Term Weight Thresholding* nicht komplex, da keine weiteren Berechnungen ausgeführt werden müssen. Nur die Termgewichte die kleiner sind als der definierte Grenzwert müssen entfernt werden. Im Gegensatz dazu ist die *Latent Semantic Analysis* komplexer und es ist schwieriger eine geeignete Anzahl von Dimensionen zu bestimmen. Es sollte in weiteren Analysen herausgefunden werden, ob *Latent Semantic Analysis* eine gute Methode für das Clustering von Themen ist. Außerdem sollte die Qualität der *Independent Component Analysis* überprüft werden, da sie in den Arbeiten von Tang et al. [209] [210] bessere Ergebnisse lieferte. Es kann außerdem überprüft werden, ob die Ergebnisse sich verbessern, wenn zuerst *Feature Transformation* und anschließend *Feature Selection* benutzt wird. In dieser Arbeit wird zuerst *Feature Selection* und dann *Feature Transformation* ausgeführt.
Der Vergleich des link-basierten und inhalts-basierten Clusterings zeigt, dass link-basiertes Clustering mehr für die Bestimmung der Themen geeignet ist. Die Clusteringqualität ist höher (externe Qualitätsanalyse) und die Berechnungszeit ist geringer (Performanceanalyse). Außerdem benötigt der link-basierte Clusteringansatz mit Agnes nur *Feature Selection*. Demzufolge beinhaltet die Term Dokument Matrix von dem link-basierten Ansatz weniger unwichtige Terme und sie ist zudem kleiner, da nur die Links berücksichtigt werden. Auch wenn das inhalts-basiertes Clustering die Dimensionsreduzierung benutzt, hat es immer noch Probleme mit dem Clustering von ähnlichen Themen. Die besten Ergebnisse wurden mit dem link-basierten Clustering in Kombination mit dem agglomerativem Clusteringalgorithmus Agnes und einem Grenzwert beim *Term Weight Thresholding* von 90% erreicht. PAM erreicht auch gute Ergebnisse, benötigt aber zusätzlich *Feature Transformation*. Obwohl link-basiertes Clustering gute Ergebnisse erreicht, wurde die optimale Clusteringlösung nicht erzielt. Die Qualität des link-basierten Clusterings hängt von der Qualität der Verlinkungen ab. Probleme entstehen bei jungen oder zu kleinen Artikeln, Artikeln deren Inhalt nicht mit dem Titel übereinstimmt oder allgemeinen Artikeln. Diese Artikel haben nur wenige Links oder Links zu anderen Themen. Die Zuweisung eines allgemeinen Artikels zu einem Thema ist schwierig, da allgemeine Artikel zu mehreren Themen gehören (z.B. *Erde*). Die Benutzung des *Overlap Clustering* könnte das Problem lösen.

Mit Hilfe der Analysen kann die Forschungsfrage aus dem Kapitel 2 (p. 9) beantwortet werden. Die Ergebnisse dieser Arbeit zeigen, dass sich link-basiertes Clustering eher für die Identifizierung von Themen eignet. Sogar das Clustering von sehr ähnlichen Themen erreicht gute Ergebnisse. Somit werden bei dem link-basierten Ansatz weniger unwichtige Terme berücksichtigt. Die externe Clusterqualität und die Performance ist besser und die Vorverarbeitung der Daten (*Tokenization*, *Stop word Removal*, *Stemming*) ist nicht erforderlich. Der inhalts-basierte Ansatz erreicht nicht die Clusterqualität des link-basierten Ansatzes. Obwohl unwichtige Terme durch das *Stop word Removal* entfernt wurden, befinden sich immer noch sehr viele unwichtige Terme in der Term Dokument Matrix. Das verursacht ein *Rauschen* in den Daten und das Clustering verschlechtert sich. Ein Problem des link-basierten Clustering ist, dass die Qualität der Ergebnisse von der Qualität der Artikel und somit von der Qualität der Verlinkungen abhängt. Die Verlinkungen wurden von Autoren angefertigt. Demzufolge wurden die wichtigen Terme eines Textes subjektiv bestimmt. Das kann zu Problemen führen, wie die Analysen dieser Arbeit zeigen.
Die Gültigkeit der Ergebnisse sollte mit anderen Datensets weiter untersucht werden. Die Qualität des link-basierten Ansatzes sollte zum Beispiel mit sehr großen Datensets, die viele Themen beinhalten, weiter überprüft werden. Auf der anderen Seite sollte überprüft werden, ob sich weitere Clusteringalgorithmen und Dimensionsreduzierungverfahren für die Identifizierung von Themen eignen. Wie zum Beispiel die *Independent Component Analysis*, die auch in der Arbeit von Tang et al. [209] [210] die besten Ergebnisse erzielte. Auch der Einfluss der Reihenfolge der Dimensionsreduzierungsmethoden sollte überprüft werden. In dieser Arbeit wurde *Feature Selection* vor *Feature Transformation* ausgeführt.
Da in dieser Arbeit die Bestimmung geeigneter Dimensionsreduzierungs-Parameter sehr komplex ist, sollten bessere Ansätze gefunden werden. Zusätzlich sollten bessere Methoden für die Evaluierung der Ergebnisse gefunden werden. Das ist auch sehr wichtig, falls kein besserer Ansatz für die Definition von geeigneten Dimensionsreduzierungs-Parametern gefunden wird. Die externe Qualitätsanalyse

benötigt die optimale Clusterlösung für ihre Bewertung. Das ist in der Praxis nicht anwendbar. Die interne Qualitätsanalyse ist in der Praxis anwendbar, aber gibt bei den aktuellen Messungen falsche Ergebnisse zurück. Die Performanceanalyse eignet sich für die Bewertung der Ergebnisse, aber sie bewertet nicht die Korrektheit der Resultate. Somit müssen weitere Methoden für die Evaluierung der Clusterqualität gefunden werden.
Da auch mit dem link-basierten Ansatz ein optimales Clustering nicht erreicht wurde, muss die Themenidentifizierung weiter verbessert werden. Es gibt zwei Ansätze um die Ergebnisse weiter zu verbessern: Die Optimierung der Methoden und die Aufwertung der Daten.
Um das Problem mit allgemeinen Artikeln zu lösen, kann *Overlap Clustering* verwendet werden (beschrieben in Kapitel 4.3.4, p. 63). Wie beschrieben, ist es mglich nach besseren Methoden für das Clustering und die Dimensionsreduzierung zu suchen. Eine weitere Möglichkeit, um das Clustering zu verbessern, ist die Tokenisierung der Artikeltitel (beschrieben in Abschnitt 4.3.1, p. 36). In diesem Fall werden Wörter wie *DNA sequence* und *Chronic toxicology* als eigene Terme bei dem Clustering behandelt. Die Ansätze können weiterhin durch die Verwendung von NLP Methoden verbessert werden. Die Auflösung von Synonymen (*Synonym Detection*, beschrieben in Abschnitt 4.3.1, p. 39) und Homonymen (*Word-sense Disambiguation*, beschrieben in Abschnitt 4.3.1, p. 39) und die Identifizierung von wichtigen Wortgruppen und Phrasen (*Multiword Phrase Grouping*, beschrieben in Abschnitt 4.3.1, p. 39) sind Möglichkeiten für die Verbesserung des Clusterings. Die Extrahierung der Terme kann durch syntaktische und semantische Analysen, wie zum Beispiel *Part-Of-Speech* (beschrieben in Abschnitt 4.3.1, p. 40) oder *Parsing* (beschrieben in Abschnitt 4.3.1, p. 39), erweitert werden. Mit diesen Methoden werden die Wörter in Substantive, Verben, Adjektive, usw. unterteilt.
Weiterhin ist es möglich die zugrunde liegenden Daten weiter aufzuwerten. So könnte zum Beispiel der link-basierte Ansatz auch die Häufigkeit einer Verlinkung berücksichtigen. Eventuell ist die Extrahierung von Untertiteln aus dem Text hilfreich. Momentan werden nur die ausgehenden Links eines Textes berücksichtigt. Die Benutzung der eingehenden Links oder die Benutzung des Durchschnittes von ausgehenden und eingehenden Links könnte die Ergebnisse weiter verbessern. Eine Kombination von beiden Clusteringansätzen ist eine weitere interessante Möglichkeit für die Identifizierung von Themen. Wie zum Beispiel die Definition eines Grenzwertes. Dieser Grenzwert bestimmt die Anzahl der Links, die ein Text beinhalten muss. Wenn der Text weniger Links hat, wird der inhalts-basierte Ansatz gewählt. In dem anderen Fall, werden nur die Links für das Clustering benutzt. Eine weitere Herangehensweise benutzt nur die Terme, die als Links im Text auftreten (link-basierter Ansatz), aber die Häufigkeit eines solchen Termes ist die absolute Häufigkeit des Termes in dem Text (inhaltsbasierter Ansatz).
Eine weitere sehr wichtige Erweiterung für die Identifizierung von Themen ist die Bestimmung von Termen, die die das Thema eines Cluster beschreiben. Ein möglicher Ansatz ist durch Siroker et al. [198] gegeben.

Die definierten Methoden für die Identifizierung von Themen können für viele Wiki Analysen genutzt werden. Es ist nun möglich ein Wiki auf Themenebene zu analysieren. So ist es zum Beispiel möglich die Autorenaktivität in oder zwischen Themen zu analysieren oder die Hauptthemen eines Wikis zu identifizieren. Die Analyse der Autorenaktivität in einem Thema ist hilfreich, um die Spezialisten eines Themas zu identifizieren. Wenn die Entwicklung eines Wikis betrachtet wird, ist die Identifizierung von neuen Themen und die Betrachtung der Themenentwicklung möglich. Somit können Trends bezüglich der inhaltlichen Entwicklung bestimmt werden. Viele Wikis sind sehr groß und demzufolge kann eine Analyse sehr komplex und unübersichtlich werden. Die Analyse auf Themenebene reduziert die Komplexität. Die Gültigkeit der Kategorisierung der Artikel in einem Wiki kann ebenfalls überprüft werden. Außerdem ist es möglich die Gültigkeit der Verlinkung zu überprüfen.
Einer der nächsten Schritt bezüglich dieser Arbeit sollte die Implementierung der beschrieben Verbesserungsvorschläge sein. Weiterhin sollte überprüft werden, ob der link-basierte Ansatz auch in anderen sozialen Informationsräumen (Blogs, Nachrichtenartikel, Online Foren, etc.) bessere Ergebnisse liefert. Wenn das der Fall ist, können Analysen auf anderen Informationsräumen durchgeführt werden. Sollte dies nicht der Fall sein, muss das inhalts-basierte Clustering benutzt werden oder es muss eine andere Möglichkeit für die Identifizierung der Themen gefunden werden. Da die Meth-

oden auf einem allgemeinen Datenmodell basieren, sind keine Änderungen in der Implementierung notwendig. Momentan wird die implementierte Metrik auf *Content Elemente* angewendet. Es ist aber auch möglich, die Metrik auf *Akteure* anzuwenden. Die Ergebnisse spiegeln Cluster mit Autoren, die an dem selben Thema gearbeitet haben, wieder. Mit diesem Ansatz ist die Identifizierung von Communities möglich.
Die entwickelten Methoden für die Themenidentifizierung können auch in der Praxis eingesetzt werden. Zum Beispiel für die Bestimmung von wichtigen Business-Themen in Zeitungsartikeln oder die Identifizierung von Meinung in sozialen Informationsräumen wie zum Beispiel Twitter. So können zum Beispiel die Hauptmeinungen über Filme oder politische Entscheidungen bestimmt werden. Die Identifizierung von Trends ist ein weiterer praktischer Nutzen. In diesem Zusammenhang werden Hauptthemen in Zeitungsartikeln, Blogs, usw. für verschiedene Zeitpunkte bestimmt. Die Ergebnisse können für die Bewertung von Einflüssen von bestimmten Ereignissen genutzt werden und somit für Marketing und Public Relation Strategien.