Technical Report

UCAM-CL-TR-802
ISSN 1476-2986

Number 802

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Latent semantic sentence clustering for multi-document summarization

Johanna Geiß

July 2011

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted April 2011 by the author for the degree of Doctor of Philosophy to the University of Cambridge, St. Edmund's College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Latent semantic sentence clustering for multi-document summarization

Johanna Geiß

Summary

This thesis investigates the applicability of Latent Semantic Analysis (LSA) to sentence clustering for Multi-Document Summarization (MDS). In contrast to more shallow approaches like measuring similarity of sentences by word overlap in a traditional vector space model, LSA takes word usage patterns into account. So far LSA has been successfully applied to different Information Retrieval (IR) tasks like information filtering and document classification (Dumais, 2004).

In the course of this research, different parameters essential to sentence clustering using a hierarchical agglomerative clustering algorithm (HAC) in general and in combination with LSA in particular are investigated. These parameters include, inter alia, information about the type of vocabulary, the size of the semantic space and the optimal numbers of dimensions to be used in LSA. These parameters have not previously been studied and evaluated in combination with sentence clustering (chapter 4).

This thesis also presents the first gold standard for sentence clustering in MDS. To be able to evaluate sentence clusterings directly and classify the influence of the different parameters on the quality of sentence clustering, an evaluation strategy is developed that includes gold standard comparison using different evaluation measures (chapter 5). Therefore the first compound gold standard for sentence clustering was created. Several human annotators were asked to group similar sentences into clusters following guidelines created for this purpose (section 5.4). The evaluation of the human generated clusterings revealed that the human annotators agreed on clustering sentences above chance. Analysis of the strategies adopted by the human annotators revealed two groups – hunters and gatherers – who differ clearly in the structure and size of the clusters they created (chapter 6).

On the basis of the evaluation strategy the parameters for sentence clustering and LSA are optimized (chapter 7). A final experiment in which the performance of LSA in sentence clustering for MDS is compared to the simple word matching approach of the traditional Vector Space Model (VSM) revealed that LSA produces better quality sentence clusters for MDS than VSM.

Acknowledgments

I would like to thank ...

... my supervisor Dr Simone Teufel for her valuable support and guidance throughout the last years.

... Dr Aurelie Herbelot, Jette Klein-Berning, Benjamin Becker, Ole Esleben, Dr. rer. nat. Johannes Fieres and Dorothea Weithoener for annotating the sentence sets. Without them this work would have not been possible.

... Andrew Ross for proofreading and valuable tips on the English language.

... my office partner and friend Katia Shutova, it would have been difficult without you.

... Stuart Moore for showing me the English seaside and that diving in UK can be fun.

... Dr Thomas Türck for his time whenever I needed someone to talk, for an unforgettable day of Christmas baking and proper German schnitzel.

... my friends in Germany for their friendship, support and skype calls.

... my parents, who always believed in me and supported me throughout my PhD journey.

... Benny for travelling back and forth, for enduring my moods and quirks and for just being there all the time.

Finally I would like to thank my two examiners Dr Kalina Bontcheva and Dr Stephen Clark for their thorough and friendly approach and their comments and suggestions.

This research was made possible by a Cambridge European Trust Bursary, a Charter Studentship from St Edmund's College, a Computer Lab Departmental Award and a EPSRC Doctoral Training Award.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> The beginning is the most important part of the work.
>
> PLATO

In this thesis I will examine redundancy identification for Multi-Document Summarization (MDS) using sentence clustering based on Latent Semantic Analysis (LSA) and its evaluation. The task of multi-document summarization is to create one summary for a group of documents that largely cover the same topic. Multi-document summarization is becoming increasingly important as the rapid development of the internet increases the amount of textual information available online.

The greatest challenge for MDS lies in identifying redundant information within a group of documents. To prevent repetition in a summary the MDS system needs to be able to detect similar and overlapping information and include it in the summary only once. On the other hand redundant information is a good measure of importance. Information that is given in most of the documents must be relevant to the topic and should be included in the summary, whereas information that is only present in one document might be omitted.

Sentence clustering can be used to find repeated information by grouping similar sentences together. There are different methods to identify similar sentences. Some systems use shallow techniques for detecting similarities in sentences, e.g., word or n-gram overlap. Others use deeper syntactic or semantic analysis. The resulting clusters represent subtopics of the document set, where one cluster ideally describes one subtopic. The clusters can be ranked for summary worthiness by distribution over documents. To avoid repetition, one sentence can be chosen or generated for each cluster to represent that subtopic in the summary.

Redundancy identification and sentence clustering is a central step in MDS. If redundant information is not detected, the summary will be repetitive and not very informative. If the most important subtopics of the document set are not identified, the summary will not properly

reflect the content of the source documents. Therefore it is necessary and useful to examine sentence clustering for MDS in detail.

In the course of this thesis I will examine the applicability of Latent Semantic Analysis (LSA) to sentence clustering. In contrast to more shallow approaches like measuring similarity of sentences by word overlap in a traditional vector space model, LSA takes word usage patterns into account. The analysis considers not only the similarity between the surface form of the sentences but also the underlying latent semantic structure within the sentences. At the same time this approach is largely language independent, except perhaps for a stop word list or a tokenizer, and is not reliant on deep syntactic and semantic analysis of the source documents. So far LSA has been successfully applied to Information Retrieval (IR) tasks such as information filtering and document classification (Dumais, 2004). However, its influence and capability for sentence clustering for MDS has not been thoroughly studied. In this research I will examine the influence of different parameters on sentence clustering using LSA and give an account of how the parameters can be optimized.

Sentence clustering has been used as an early step in automatic text summarization, but its functionality and performance have not been given sufficient attention. In most approaches the clustering results are not directly evaluated. Usually the quality of the sentence clusters are only evaluated indirectly by judging the quality of the generated summary. There is still no standard evaluation method for summarization and no consensus in the summarization community on how to evaluate a summary. The methods at hand are either superficial or expensive in time and resources and not easily repeatable. Another argument against indirect evaluation of clustering is that troubleshooting becomes more difficult. If a poor summary was created it is not clear which component, for example information extraction through clustering or summary generation (using, e.g., language regeneration) is responsible for the lack of quality. However, there is no gold standard for sentence clustering available to which the output of a clustering system could be compared. Therefore I develop a strategy to build a gold standard for sentence clustering in MDS. I design a set of guidelines and rules that are given to human annotators. I compare the clusterings created by the human judges and evaluate their inter-annotator agreement.

Another challenge is the comparison of sentence clusters to the gold standard. Numerous evaluation methods are available. Each of them focusses on different properties of clustering and has its own advantages. I describe and evaluate the most widely used and most promising measures.

The thesis is organized as follows: Chapter 2 introduces MDS, highlighting some MDS systems using sentence clustering and LSA. Chapter 3 gives an overview of the models and implementation used. Chapter 4 gives an account of the problems occurring when clustering sentences and the parameters that need to be considered when using LSA. Chapter 5 explains the evaluation strategy, introduces the concept of a gold standard evaluation, explains how to create a gold standard for sentence clustering, and compares different evaluation metrics. Chapter 6 presents the results of the comparison of human generated clusterings. The results

of the parameter optimization experiments are given in chapter 7. Chapter 8 compares the performance of LSA in sentence clustering for MDS with the baseline approach using VSM. Finally Chapter 9 summarizes the contributions of this thesis.

Chapter 2

Background and motivation

> Alles Gescheite ist schon gedacht worden.
> Man muss nur versuchen, es noch einmal zu denken.[1]
>
> *Wilhelm Meister's Wanderjahre*
> JOHANN WOLFGANG GOETHE

The advent of the internet and easy access to computers and the global network has led to an increasing amount of information, most of it textual. Nowadays anyone connected to the internet can access information about virtually anything from all over the world, at his or her fingertips. Anyone can contribute to the internet and not only consume but also produce information. The number of websites has grown from 100,000 in 1996 to over 270 million in January 2011[2]. This development is leading to information overload. To avoid drowning in information, the flow needs to be filtered and the content condensed. Automatic text summarization can help by providing shortened versions of texts. It is a technique where a computer program creates a shortened version of a text while preserving the content of the source. In the course of shortening no important information must be omitted and no information should be repeated. The summary contains the most important information from the original text. In general every automatic text summarization system involves three basic steps – analysis, processing and generation (see figure 2.1). In the first step the document(s) to be summarized are analysed, e.g., redundant information is identified. In the next step, processing, the information for the summary is selected, for example the most important clusters of redundant information are selected. During generation the actual text of the summary is generated, for example by including one sentence for each cluster in the summary.

Although all summarization systems have these three stages in common, different systems produce different summaries. There are numerous types of summaries with different charac-

[1] All intelligent thoughts have already been thought; what is necessary is only to try to think them again.

[2] `http://news.netcraft.com/archives/2011/01/12/january-2011-web-server-survey-4.html` last visited 14.02.2011

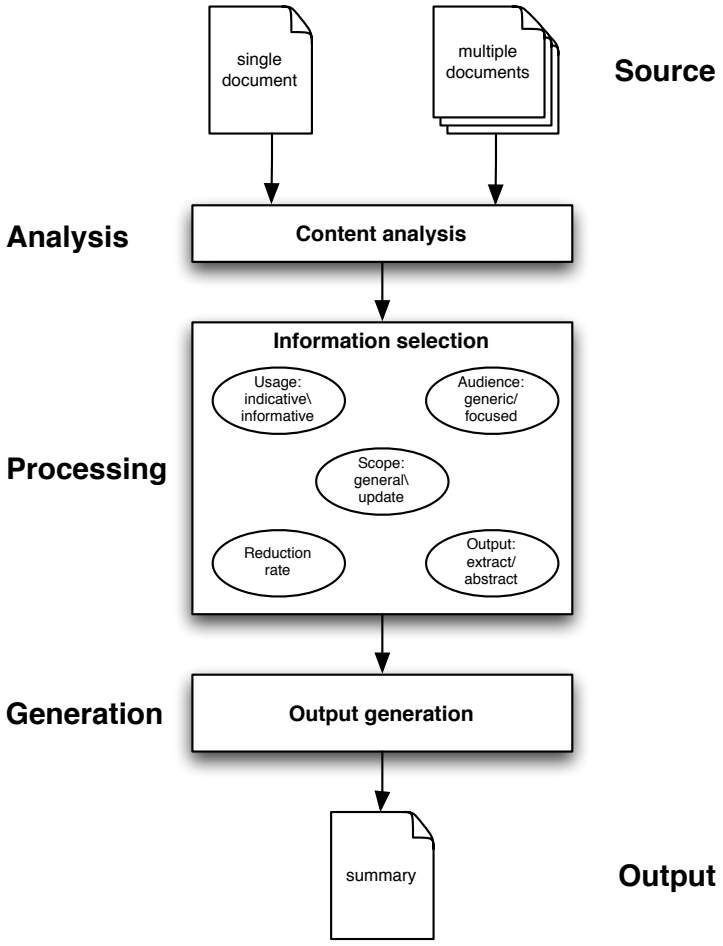


Figure 2.1: *Framework of an automatic text summarization system*

teristics (see figure 2.1). The first aspect is the *source* to be summarized; this can be a single document or multiple documents.

Another group of characteristics relates to the content. One such characteristic is the *audience*. The summary might be targeted to a generic audience, so that the summary reflects all major themes of the original text(s). If a query is provided, e.g., by a user, he or she is only interested in a specific aspect and the summary will be targeted to meet the information need. Another feature is the *scope* of a summary. A general or background summary will give all available information, assuming the reader has no prior knowledge of the topic, whereas in an update summary only the latest developments are considered. The produced summary can be indicative or informative. An indicative summary provides information about the topic of the text(s) and indicates what the text is about. It does not necessarily contain content from the source. An informative summary contains the main statements of the source. It can stand in place of the original document(s) whereas an indicative summary can only support the decision to read the original text(s) or not.

The last group of features determines the output of a summarization system. The *reduc-*

tion rate determines the length of the summary and the *output type* indicates if an extract or abstract should be created. An extract consists entirely of material from the source; usually sentences are selected from the source and pasted into a summary. An abstract on the other hand includes sentences that are not necessarily present in the source. For example new sentences are (re-)generated from the source text. If sentences that were not directly connected in the source are concatenated to form an extract, the produced summary might lack in readability, coherence and clarity, for example when anaphora are not resolved. Sentences in an abstract are (re-)generated from source sentences or phrases and the problems of readability and coherence are solved during generation. The creation of an extract is easier and more robust. The output always contains grammatical sentences as they are taken from the source directly. For generating an abstract more complex methods are required and the output might not always consist of grammatical sentences only. In this thesis I will concentrate on methods for creating informative and general summaries of multiple documents that are targeted at a general audience.

The internet offers access to many websites of newspapers or news agencies. Most news websites from the same region or country report about the same events and sometimes even for an extended period of time. For example when there is a major natural disaster all newspapers and news agencies publish reports about this event during the following days or weeks. This results in information overlap. Information is repeated, even within the same newspaper or agency. Therefore the amount of redundant information is growing and the reader often reads the same piece of information over and over again. Multi-document summarization can help to avoid repetition and to provide the reader with condensed non-redundant information. Thus users can quickly familiarize themselves with a topic that was previously described by a large cluster of documents.

2.1 Multi-document summarization

In MDS one summary is created for a set of documents. These document sets can contain articles from different sources, e.g., from different newspapers on different dates or documents retrieved by an information retrieval system in response to a query. The documents in a set are related to each other, with content related to the same event, person or topic. The Document Understanding Conferences (DUC) (DUC, 2007) provide such sets of documents. DUC was established in 2001 to offer a forum for summarization researchers to compare their methods and results on common test sets given standardized evaluation methods. DUC was annually held by NIST[3] from 2001 until 2008, when DUC became part of the Text Analysis Conference (TAC) as summarization track. Over the years different summarization tasks were addressed, for example single document summarization, multi-document summarization, and update summarization. Each year a set of document clusters and an evaluation scheme were provided. The document clusters (hereinafter referred to as DUC clusters) consist of documents from various

[3]National Institute of Standards and Technology `www.nist.gov` last visited 15 March 2011

newswires such as AP or New York Times. These articles were extracted from different corpora like TIPSTER (Harman and Liberman, 1993), TREC (TREC, 2011) and AQUAINT (Linguistic Data Consortium, 2002). The actual DUC clusters were created by NIST assessors who chose topics and selected 10 to 50 documents related to each topic.

The documents within a set overlap in information and thereby include redundant information. Irrelevant to single document summarization, information overlap is one of the biggest challenges to MDS systems. While repeated information is a good evidence for importance, this information should be included in a summary only once in order to avoid a repetitive summary. The idea behind this is that information that is repeated throughout a collection of articles about the same event must be important for the understanding and comprehension of the described episode. The task is to identify the information that is common to most of the articles and represent that piece of information only once in the summary. The problem is that almost certainly the documents in a set are written by different authors with different writing styles and vocabularies. So the content might be the same or very similar but the surface of the articles might be very different.

Different approaches to recognize redundancy in text are used. Goldstein et al. (2000) developed maximal marginal relevance for MDS (MMR-MD) for detecting redundant sentences when creating extraction based multi-document summaries in response to a user query. The MMR-MD is used to minimize redundancy and maximize relevance and diversity of a summary. First the texts to be summarized are segmented into passages (sentences, phrases or paragraphs). Passages are treated as bags of words and cosine similarities between passages and a query are calculated. The cosine similarity of a passage and a query is given by the cosine of the angle between two vectors describing the passage and the query (section 3.1.2). All passages with a cosine score below a certain threshold are discarded, that is to say all passages that are not relevant to the query are removed. Then the MMR-MD metric is applied to the remaining passages, determining those passages that are summary worthy, i.e., which are relevant to the query but different to other passages already selected for the summary. Similarity calculation for passages is here based on the cosine similarity in a traditional vector space. Thus the redundancy detection is based on word overlap. In the end the passages are combined following some summary cohesion criterion, e.g., ranking by relevance or time. A similar approach to redundancy removal is used in NeATS (Lin and Hovy, 2002), which combines well known techniques from single document summarization. Important content is determined by identifying key concepts in a document set and ranking the sentences of that document accordingly. The sentences are selected and filtered using the position of a sentence in a document, stigma words (words that usually cause discontinuity in summaries like conjunctions or pronouns) and MMR. The remaining sentences are ordered chronologically and, where necessary, the sentences are paired with an introductory sentence.

A similar approach is used in MEAD (Radev et al., 2004). The system also uses statistical methods to determine which sentences to include in a summary. The sentences are compared to a centroid vector of a document set. A centroid consists of words that represent a cluster

of documents. For each sentence a score is calculated using the similarity to the centroid, the position in the document, the word overlap with the first sentence in the document and a redundancy penalty that measures the word overlap with other sentences.

These three approaches use word overlap to measure redundancy after the summary worthy sentences were already identified. The statistical methods used are fast and robust. They are largely independent of language, domain and genre.

The system described in Barzilay and Elhadad (1997) incorporates different techniques. Common information is identified by building lexical chains using nouns and noun compounds. The documents are segmented and lexical chains are built within these segments by using WordNet relations. This approach enriches the surface form of a noun with meaning, by including a word and its sense in a specific sentence in a lexical chain. Each chain consists of similar words and the similarity of the words is only based on WordNet relations of the word senses. The chains are scored and for the strongest chains sentences are selected to be included in the summary. For scoring chains and selecting sentences for the strongest chains, the algorithm uses word count and word overlap. Since the scope of WordNet is limited and most of the entries are nouns, only part of the texts can be used for finding lexical chains.

Barzilay and McKeown (2005) uses a text-to-text generation approach. In order to create a coherent abstract they try to identify common clauses within sentences by aligning syntactic trees and combining these fragments into a new sentence. For this approach clusters of sentences are used as input.

The latter two approaches require deeper syntactic and semantic analysis than the other systems described. This makes them dependent on languages and maybe even domains. On the other hand the systems rely not only on the surface form of text but also on the meaning of words.

Sentence clustering is often used as a first step in MDS to identify redundant information. In sentence clustering, semantically similar sentences are grouped together. Sentences within a cluster overlap in information, but they do not have to be identical in meaning. In contrast to paraphrases, sentences in a cluster do not have to cover the same amount of information. For each cluster one sentence can be selected or as in Barzilay and McKeown (2005) be generated to be included in the summary. In the next section I introduce some systems using clustering and describe them in more detail.

2.1.1 Sentence clustering in summarization

For single document summarization Aliguliyev (2006) presents a summarization system based on sentence clustering. The sentences from a document are clustered so that the intra-cluster similarity is maximized while the inter-cluster similarity is minimized. For similarity calculation between the sentences the cosine measure is used and the sentences are represented as

vectors of features (here words). For each cluster a representative sentence is chosen by determining the proximity to other sentences in the cluster and to the cluster centre. The similarity of these sentences to the title is calculated and the sentences are ranked in descending order by their relevance score (the weighted sum of the two similarity measures). Sentences are added to the summary until the desired length of the summary is reached. In this approach the recognition of redundancy is based on word overlap. The similarity between sentences is calculated using the traditional vector space model. In this model two sentences are similar the more vector features (here words) they have in common.

Sentence clustering for multiple documents in described in Seno and Nunes (2008). The sentences of the documents are clustered using an incremental clustering algorithm. The first cluster is created by selecting the first sentence from the first document. For all following sentences it is decided if the sentence should belong to an existing cluster or if a new cluster is created. The decision is based on a similarity threshold. In the described study two similarity measures were tested: (i) word overlap and (ii) cosine measure. The word overlap metric determines the number of words a sentence and a cluster have in common in proportion to the total number of words in that sentence and that cluster. The best results were achieved when the threshold for world overlap was 0.2, i.e., a sentence was only assigned to a cluster if the word overlap score with that cluster was larger than 0.2. The second metric tested was the cosine similarity between a sentence and the centroid of a cluster. Here a centroid consists of terms that are representative for the cluster. Two methods for selecting the words for the centroid were introduced involving $tf\text{-}idf$ (term frequency-inverse document frequency) and $tf\text{-}isf$ (term frequency-inverse sentence frequency). The best sentence clusters were achieved using the $tf\text{-}idf$ centroid with a threshold of 0.3. The clusterings created by the system were evaluated on a corpus of 20 sets of news articles. A reference clustering corpus was created by the first author of that paper.

Marcu and Gerber (2001) present a system where non-overlapping elementary discourse units (*edus*) are assigned importance scores and are clustered using a C-Link clustering algorithm. They claim that large clusters with a high intra-cluster similarity can reliably represent key information of the documents. The clusters are ranked according to the position of *edus* in the documents, the importance of *edus*, and intra-cluster similarity. For each most important cluster one *edu* is selected for the summary, until the target summary length is reached. This approach takes discourse information into account. An *edu* is about the size of a clause. However the similarity calculation is again based on word overlap between the single units.

A different strategy is used in Hatzivassiloglou et al. (1999, 2001). The system described is called SIMFINDER. It serves as analysis component for a multi-document summarizer described in McKeown et al. (1999) that incorporates text reformulation for abstract generation (Barzilay, 2003). In SIMFINDER similarity between paragraphs (mostly containing one sentence) is determined using a range of linguistic features. These features include primitive features based on single words or simplex noun phrases and composite features that combine primitive features pairwise. The primitive features include word co-occurrence, matching noun

phrases, WordNet synonyms or shared proper nouns. Once the similarities between any two text passages are calculated, a non-hierarchical clustering algorithm is used to group similar text units into clusters. Each cluster is then represented by one sentence in the summary. MultiGen (Barzilay, 2003) analyses the sentences in each cluster and regenerates a sentence from the information common to all sentences in that cluster.

For the evaluation of SIMFINDER a set of 10,535 manually marked pairs of paragraphs was created. Two human annotators were asked to judge if the paragraphs contained common information. They were given the guideline that only paragraphs that describe the same object in the same way or in which the same object is acting the same are to be considered similar. They found significant disagreement between the judges but the annotators were able to resolve their differences. SIMFINDER found 36.6% of the similar paragraphs with 60.5% precision. Unfortunately the evaluation looked only at pairs of paragraphs and not at clusters. Due to the linguistic features, SIMFINDER is language dependent. Apart from the WordNet feature, the similarity calculation is again based on simple word overlap enriched with some syntactic analysis. The WordNet feature is limited to nouns and verbs present in WordNet. Although WordNet keeps growing it only partially describes the English language and the relationships between words.

The approaches described above did not evaluate the sentence clusters produced directly but evaluated only the resulting summaries. There are two problems with indirect cluster evaluation:

(i) There is no consensus in the summarization community of how best to evaluate a summary. The evaluation methods used are either superficial, like ROUGE (Lin, 2004), which counts overlapping units between two summaries, or expensive in time and resources, like Pyramids (Nenkova and Passonneau, 2004). Hence the results are rarely comparable.

(ii) A correct classification of sentences into clusters cannot be verified. If the resulting summary is of poor quality it is difficult to determine which component is responsible.

Thus a clearly laid out strategy for evaluating sentence clusterings for MDS is needed.

2.1.2 LSA in summarization

There are some approaches that incorporate LSA in their MDS systems. Steinberger and Krišťan (2007) developed an algorithm for single document summarization using LSA and apply it to the problem of MDS. Their system works as follows: first a term-by-sentence matrix (TSM) $\boldsymbol{A}$ is created where each row represents a term and each column represents a sentence. The cells of the matrix contain weighted term frequencies. Singular Value Decomposition (SVD) is applied to $\boldsymbol{A}$ which breaks down the original TSM into r base vectors which are linearly independent (for details see section 3.2). The result of SVD are three matrices $\boldsymbol{T}$, $\boldsymbol{S}$ and $\boldsymbol{D^T}$. These submatrices are used to calculate a ranking matrix $\boldsymbol{S^2 D^T}$, which is used to

rank the sentences of the document collection. For each sentence the length of the corresponding vector in the ranking matrix is calculated. The sentences with the highest scores ($length\ of\ vector$ divided by $(number\ of\ terms)^{0.4}$) are selected for the summary. To avoid redundancy only sentences that are not similar to sentences already in the extract, measured with the cosine similarity in the original term space, are added to it. The evaluation was done on 50 DUC 2005 clusters containing 1300 documents in total. The resulting extracts were evaluated with ROUGE. The MDS system scored better than 27 other systems and worse than 5. In this approach SVD helps to capture the relationships between words by analysing co-occurrence patterns. Thus terms and sentences can be grouped on a more semantic basis than on word overlap only. The authors claim that each singular vector (rows in $\boldsymbol{D^T}$) represents a salient and recurring word usage pattern. They assume that each word usage pattern describes one topic in the document collection. Thus the topics of a document collection are represented by the singular vectors and the magnitude of the singular value represents the degree of importance of the corresponding pattern within the document set. The authors assume that the sentence that best represents this pattern will have the largest index value within the singular vector. The importance and the coverage of the word usage patterns within the sentences are taken into account by calculating the length of the sentence vectors in $\boldsymbol{S^2 D^T}$. The authors seek to choose sentences for their summary that have the greatest combined weight across all important topics.

The problem is that in this approach the number of topics is linked to the number of singular values. In SVD the number of singular values and vectors is equal to the rank of the matrix, which is the number of linearly independent rows or columns of $\boldsymbol{A}$. To reveal the latent semantic structure and the themes in a document collection the dimensions are reduced (for details see section 3.2). Choosing the number of dimensions to keep also determines the number of topics. The unresolved problem is to determine the number of topics relevant for the document collection in advance, when the system claims to be fully automatic. The relation between word usage pattern and topic is debatable: is one word usage pattern equal to a topic of a document collection? In addition, even if there is a one-to-one relationship between patterns and topics, why are sentences selected that have the highest score over all topics? One could also choose one sentence for each topic. Also, only the resulting extracts were evaluated using ROUGE, where N-grams are compared to human written abstracts. A problem here is that human written abstracts are compared to extracts. ROUGE only takes matches of N-grams into account, but since humans created the abstracts, the content of the summaries compared can be similar but the words used can be very different. But evaluation of summaries is an open and very controversial problem. There was no evaluation of whether the sentences selected for the summary really do represent the most important topics of the document collection.

Another approach using sentence clustering and LSA is described in Bing et al. (2005). After a term-by-sentence matrix is built and factorized using SVD the sentences are compared pairwise. The sentences with the highest similarity are merged to a sentence cluster, called a *fake sentence*. This cluster and the rest of the sentences are used to create a new matrix and again (after applying SVD) all sentences are compared pairwise and so on. This process

is repeated until the predefined number of clusters is reached. For each cluster the centroid sentence is determined. The centroid sentences are then sorted and included in the summary. This approach was tested on a dataset consisting of 20 documents clusters of 7-9 news articles from various websites. Judges were asked to grade the 20 extracts created by the system using a score between 1 (bad summary) and 5 (good summary). 75% of the summaries were marked with a score of 3 or 4. In this system only matrix $\boldsymbol{D}$ is used to cluster the sentences. I think it is unreasonable that the singular values are not taken into account since they correspond to the importance of each concept within the document collection. It is very time and memory consuming to build a new matrix and perform SVD every time two sentences are joined. The *fake sentence* is longer than the other sentences, therefore will score higher similarity with other sentences and attract more sentences. Again only the summary was evaluated as human judges were asked to score the summary. The judges were not provided with other summaries, to which they could compare the automatically created abstracts. It is not clear which instructions the judges received or if they had access to the original documents.

The Embra system for DUC 2005 (Hachey et al., 2005) uses LSA to build a very large semantic space to derive a more robust representation of sentences. In this space not only the documents that are summarized are included but also other documents from the DUC 2005 and AQUAINT corpus. First a term-by-document matrix (TDM) is created, SVD is applied and the resulting submatrices are reduced to 100 dimensions. From the submatrices a sentence representation is built, where each sentence is represented by a vector that is the average of the vectors of the words the sentence contains. This sentence representation is then passed to an MMR-style algorithm, which determines relevancy and redundancy (see above). The sentences with the highest MMR score are selected for the summary. In contrast to Goldstein et al. (2000) here the redundancy calculation is not based on single word overlap but on the word usage patterns revealed by LSA. SVD was performed on a term by document matrix, but it was not evaluated how the size of a semantic space influences redundancy identification.

The approaches described here all lack the evaluation of the influence of LSA on detecting redundant information. Only indirect evaluation of the redundancy identification was carried out by evaluating the resulting summary. The influence of parameters like the number of dimensions (k), size of the semantic space or vocabulary included have not been properly analysed. However these parameters might have a great impact on the quality of redundancy identification. Once again in order to optimize these parameters a direct evaluation of the sentence clusters is required.

2.2 Motivation

Sentence clustering integrates well in a MDS system creating general or query-related abstracts. Given a set of related documents, the documents are split into smaller units like paragraphs, sentences, snippets and words. The units are clustered separately or simultaneously. The retrieved

clusters of different sized units represent the theme of the document set. Afterwards the clusters are handed over to a language generation system, e.g., MultiGen (Barzilay, 2003), which creates an abstract.

The prerequisite for an MDS system is one or more sets of documents that relate to the same topic, which in case of news articles can be, e.g., a person, an aspect of a person's life or an event like a natural disaster or a general election. For this thesis it is assumed that a document classifier is in place which groups documents into topic groups.

For an MDS system it is vital to find out what a given set of documents is about. The premise is that the documents in a set are about the same matter. The assumption is that the information that most often recurs in the documents is the main subject of the texts. But in different documents the same or similar pieces of information might be given by different sentences, wordings or expressions. Redundancy identification and removal is therefore a crucial step in MDS. As described in the previous sections, often redundancies are removed after the content for the summary has been selected (Goldstein et al., 2000; Lin and Hovy, 2002; Radev et al., 2004). This approach is not well suited for MDS. If the documents to be summarized overlap considerably, which is the optimal starting position for any MDS system, there will be many sentences that are very similar. Thus clustering the sentences first by theme is less costly in time and resources.

Clustering is a well established method for identifying redundancy (see section 2.1.1). It can also be used to rank clusters by their summary worthiness. The idea behind sentence clustering is to find repetitive and hence similar information. Information that is repeated throughout a collection of articles about the same event must be important for the understanding and comprehension of the described episode. To find redundant information within a document set, different surface forms of the same or very similar information are grouped together. Once these groups of text units are found they are ranked. The best clusters represent the topics of that document set. In many MDS approaches, the text units that are grouped into clusters are sentences. The similarity between sentences is often measured by the number of terms they share. To overcome the problems of different word forms and synonyms, stemming and WordNet (Fellbaum, 1998) are often used. These approaches are language dependent and rely on the coverage of WordNet.

Clustering using LSA is largely independent of the language, unlike approaches that require deeper linguistics analysis. It has also the advantage that the similarity estimation is not based on shallow methods like word matching. Incorporation of LSA takes underlying latent semantic structures in form of word usage patterns into account. Thus the problem of synonymy is avoided.

The first step towards multi-unit clustering is to examine the most obvious unit – the sentences. Sentences are reasonably easy to identify. There are several sentence boundary detectors available, e. g., RASP (Briscoe et al., 2006). A sentence is normally the largest grammatical unit. In traditional definitions a sentence is often described as a set of words expressing a com-

plete thought (Chalker and Weiner, 1994). In Peters (2004) the functions of a sentence are described as making statements, asking questions, uttering commands and voicing exclamations.

In this thesis the focus lies on summarizing multiple news articles. The advantage of the journalistic writing style is that journalists usually try to be explicit and precise. Wikipedia (2011b)[4] describes the news style as follows:

> "Journalistic prose is explicit and precise, and tries not to rely on jargon. As a rule, journalists will not use a long word when a short one will do. They use subject-verb-object construction and vivid, active prose [...]. They offer anecdotes, examples and metaphors, and they rarely depend on colorless generalizations or abstract ideas. News writers try to avoid using the same word more than once in a paragraph [..]."

Thus fragmentary sentences, exclamative or imperative sentences will remain the exception.

Alternatively other text units like words, paragraphs or snippets could be used for identifying redundant information in news articles. However they have some disadvantages over sentences. Single words are too small to be used as a clustering unit. The meaning of a word often depends on the context. One word alone does not cover a whole piece of information. On the other hand, paragraphs, which contain more than one sentence, are too large a unit for clustering in MDS. The sentences of a paragraph might contain sentences with slightly different topics or themes. In clustering for redundancy identification it is important that the units used cover a whole piece of information and ideally only one. However this problem of multiple themes might also occur with sentences. Desirable would be a unit smaller than a sentence that covers exactly one topic, thought or theme. Thadani and McKeown (2008) call the textual representation of a basic unit of information in a document a *nugget*, but in their experiments they use a concept-annotated corpus. Finding information nuggets in text requires stable and reliable information extraction techniques using deep semantic and syntactic analysis. Thus it is best to start with the best unit at hand – sentences.

I first look at the parameters that could influence the quality of sentence clusterings using LSA. These have not previously been evaluated in detail in relation to sentence clustering for MDS. First, basic indexing techniques like stemming or stop word removal are tested on IR corpora (section 4.2). I will then investigate how different index vocabularies influence sentence clustering. The index vocabulary determines the terms that form the basis on which sentences are compared and similarity between them is calculated. For IR it is often claimed that the nouns of a sentence carry the most meaning and that they are the distinctive features. I will examine different strategies to create index vocabularies and assess their influence on redundancy identification using sentence clustering (section 4.2.2).

[4] `http://en.wikipedia.org/wiki/News_style` last visited 2. March 2011 11:15

The first sentence cluster experiments are carried out on the Microsoft Research Paraphrase Corpus (Dolan et al., 2004; Dolan and Brockett, 2005). Here I test and evaluate the basic clustering algorithm parameters like linkage and distance metric. In more detail I will examine the clustering threshold t for the cophenetic distance. This threshold determines where a cluster tree produced by hierarchical agglomerative clustering (HAC) is cut to be split into flat clusters. This parameter might have a great impact on the quality of the clusterings.

Another important parameter I examine is the size of the clustering space, which is determined by the number of sentences included (see section 4.3). There are different hypotheses about how the size of the corpus might influence the structure and quality of the semantic space. One approach assumes that a large corpus or an additional background corpus leads to a more robust and accurate semantic space. A different strategy says that a general background corpus might bias term relations and that a smaller localized corpus is more sensitive to small effects. The influence of different sized corpora on sentence clustering for MDS has never been studied before. I will test three space options (local, extended and global) and how they affect sentence clustering.

One of the most important parameters for any LSA application is the number of remaining dimensions k. I also study how k influences the quality of sentence clustering and whether the optimal value for k depends on other parameters like t, cluster space size or vocabulary.

I will also compare the performance of LSA in sentence clustering with the simple word matching approach of VSM. This comparison will show if the usage of underlying latent semantic structures in a text helps to create better sentences clusters for redundancy identification.

A strong evaluation strategy is needed to evaluate the influence of different parameters on clustering quality and to compare different approaches. I decided to use a gold standard comparison strategy (see chapter 5 for details), where the clusterings created by my system BOSSE^{Clu} are compared to a ground truth. Since no gold standard was available for sentence clustering in MDS, I decided to create the first compound gold standard for sentence clustering with the help of human annotators. In this context, guidelines were designed to guide the annotators and help them to create consistent clusterings that comply with the same standards. In course of the creation of a gold standard for MDS, I will also study how humans cluster similar sentences and which strategies they use.

Also important for evaluation are the measures used. Since many different evaluation metrics are available, I will examine their properties and evaluate which measures are best suited for sentence clustering in MDS.

Chapter 3

Semantic spaces, algorithms and implementation

And now for something completely different.

MONTY PYTHON

In this chapter I describe the different semantic spaces, algorithms and the implementation used throughout this thesis. Section 3.1 introduces the standard Vector Space Model (VSM). Section 3.2 explains Latent Semantic Analysis (LSA) and how it incorporates Singular Value Decomposition (SVD) to reveal the latent semantic structure within a text collection. An overview of the implementation and structure of my sentence clustering system called BOSSE^{Clu} is given in section 3.4. Section 3.5 introduces the clustering algorithm.

3.1 Standard vector space model

The standard vector space model (hereinafter referred to as VSM) is a model for representing text in a vector space based on the bag of words approach. It was first presented as a model for Information Retrieval (IR) in Salton (1979) and was used in the System for the Mechanical Analysis and Retrieval of Text (SMART) information retrieval system (Salton, 1971b) (on which see Dubin (2004)).

In VSM, text units of a corpus are represented by vectors. Traditionally a whole document is used as a text unit, but any other text unit like paragraphs or sentences can be used just as well. Each dimension of a vector corresponds to a term that is present in the corpus. A term might be, e.g., a single word, n-gram or a phrase. If a term occurs in a document the value of that dimension is non-zero. Values can be binary ($1 \rightarrow$ term is present in the document, $0 \rightarrow$ term is not present in the document), frequencies of terms in the document, or term weights. A whole text corpus can then be represented by a term-by-document matrix $\mathbf{A}$. Consider the following example: a sample text corpus containing the following three sentences:

s_1 It's raining **cat**s and **dog**s.

s_2 A **man** is taking his **dog** for a walk.

s_3 A **man** takes his **cat** to the vet.

The terms **man**, **cat** and **dog** are used for indexing. The corpus can then be represented by the term-by-sentence matrix (TSM) **A** in table 3.1.

	s_1	s_2	s_3
cat	1	0	1
dog	1	1	0
man	0	1	1

Table 3.1: *TSM for a small sample corpus*

Figure 3.1 shows a graph of the vector space drawn by the terms. The sentences are represented as vectors in space.

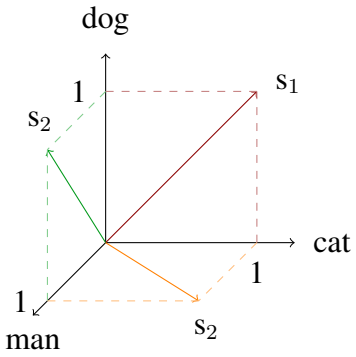


Figure 3.1: *Vector space for a small sample corpus*

The advantage of VSM is that within this model similarities between documents or a query and a document can be calculated.

3.1.1 Term weighting

The idea behind term weighting is to assign a weight to represent the importance of a term. The raw frequency of a term only states how often a term occurs in a document without measuring the importance of that term within the document or within the whole collection. Different weighting schemes are available. The most common and popular one is the tf-idf weighting scheme (Salton and McGill, 1986). It combines local and global weighting of a term. I will use a modified version called ntf-idf since earlier experiments showed that using this modified scheme led to a slight increase in retrieval performance than using the raw term frequency (tf).

Local term weighting For local term weighting I will use the normalized term frequency (ntf) (Haenelt, 2009). It measures the importance of a term within a document.

$$ntf_{i,m} = \frac{freq_{i,m}}{max_{j,m}} \tag{3.1}$$

The normalized term frequency $ntf_{i,m}$ is the fraction of the frequency $freq_{i,m}$ of term t_i in document D_m and the highest frequency $max_{j,m}$ of any term t_n in document D_m. This formula assign a higher weight to terms that occur often in a document.

Global term weighting (idf) The inverse document frequency (idf) (Spärck Jones, 1972) measures the importance of a term within the document collection.

$$idf_i = \log \frac{N}{n_i} \tag{3.2}$$

Here N is the number of all documents in the collection and n_i is the number of documents that term i occurs in. A term that occurs in every document of the collection gets a lower idf value. This reflects the fact that it is not as significant for the distinction between documents as terms that occur rarely throughout the document collection. For sentence clustering this scheme was adopted to use sentences instead of documents. Thus the global weighting scheme is renamed to isf – inverse sentence frequency.

This results in the ntf-isf weighting scheme:

$$w_{i,m} = ntf_{i,m} \times isf_i \tag{3.3}$$

where the weight w of a term i in a sentence m is defined by the product of the local weight of term i in sentence m and the global weight of term i. Dumais (1990) did some experiments on weighting schemes and LSI and concluded that using weighting has a positive effect on retrieval performance.

3.1.2 Similarity calculation

A very popular similarity measure is the cosine similarity. This measure is based on the angle α between two vectors in the VSM. The closer the vectors are to each other the more similar are the documents. The calculation of an angle between two vectors $\vec{a}$ and $\vec{b}$ can be derived from the Euclidean dot product:

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cdot \cos(\alpha) \tag{3.4}$$

This states that the product of two vectors is given by the product of their norms (in spatial terms, the length of the vector) multiplied by the cosine of the angle α between them. Given equation 3.4 the cosine similarity is therefore:

$$\cos(\alpha) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} \tag{3.5}$$

The values of $\cos(\alpha)$ can range from -1 for opposing vectors to 1 for identical vectors.

3.1.3 Disadvantages of VSM

Within the VSM only similarities between documents or between a query and documents can be calculated within one space. If terms were to be compared to each other another space would have to be drawn. In a term space, where the terms represent the dimensions, the terms are considered to be linearly independent, which means their relations to each other are not taken into account. Furthermore in the traditional vector space the similarity calculation is based only on word matching. Each dimensions of a vector corresponds to a term. Two documents with a similar topic but different vocabulary will not be placed next to each other. Only documents that overlap in vocabulary will be considered similar.

3.2 Latent Semantic Analysis

Latent Semantic Indexing (LSI) was developed as a special vector space approach to conceptual IR (Deerwester et al., 1990). It attempts to overcome two common problems of search engines – synonymy and polysemy. In the standard VSM (Salton, 1971b), the terms are assumed to be independent and thus term associations are ignored. By contrast LSI re-expresses a co-occurrence matrix in a new coordinate system. The idea is to uncover the latent semantic structure of a document collection, i.e., to find hidden relations between terms, sentences, documents or other text units. This is achieved by using high-order co-occurrence (Kontostathis and Pottenger, 2006). Unlike methods like VSM relying on literal word overlap for similarity calculation, LSA relies on "a derived semantic relatedness measure" (Foltz et al., 1998). This measure reflects the semantic similarity between words that are used in similar context, e.g., synonyms, antonyms, hyponyms or compounds.

The technique is called LSI when it is applied to IR otherwise it is called LSA

3.2.1 Latent Semantic Analysis: the basics

I will explain the functionality of LSA using an example from term similarity calculation. Consider table 3.2, which consists of 9 titles from technical reports from Deerwester et al. (1990). The data set can be represented by the term-by-document matrix $\mathbf{B}$ shown in table 3.3. Each column describes a document, each row a term. Each cell entry indicates the frequency of a term occurring in a document. This term-by-document matrix $\mathbf{B}$ can be used to calculate the similarity between terms. When calculating similarities between terms each term is represented as a vector in the Cartesian coordinate system for the standard vector space where the documents define the dimensions. To calculate the similarity of two terms in a vector space, the distance between the two term vectors can be measured using the cosine similarity measure (see section 3.1.2). A cosine of -1 implies that the compared vectors point into opposite directions whereas vectors that have the same direction receive a cosine of 1. The cosine function for two vectors

d1: **Human** machine **interface** for Lab ABC **computer** applications
d2: A **survey** of **user** opinion of **computer system response time**
d3: The **EPS user interface** management **system**
d4: **System** and **human system** engineering testing of **EPS**
d5: Relation of **user**-perceived **response time** to error measurement
d6: The generation of random, binary, unordered **trees**
d7: The intersection **graph** of paths in **trees**
d8: **Graph minors** IV: Widths of **trees** and well-quasi-ordering
d9: **Graph minors**: A **survey**

Table 3.2: *Sample data set from Deerwester et al. (1990). Underlined words occur in more than one title and are selected for indexing.*

	d1	d2	d3	d4	d5	d6	d7	d8	d9
computer	1	1	0	0	0	0	0	0	0
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
system	0	1	1	2	0	0	0	0	0
time	0	1	0	0	1	0	0	0	0
user	0	1	1	0	1	0	0	0	0
eps	0	0	1	1	0	0	0	0	0
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Table 3.3: *Term-by-document matrix* $\mathbf{B}$ *for the sample data set from table 3.2*

of a frequency matrix in standard vector space can only return values in the range $[0, 1]$, because in the traditional VSM all vectors lie within the positive quadrant. The values in any frequency matrix are always positive – a negative frequency count is not possible. A cosine of 0 for two vectors in standard vector space means that their distance is maximal which implies that they are maximally dissimilar. In contrast to VSM, values in an LSA matrix can be negative so the cosine of two vectors in a LSA space can have a value anywhere in the range of $[-1, 1]$.

Using the rows of the term-by-document matrix $\mathbf{B}$ shown in table 3.3 the cosine similarity for all term pairs in standard vector space were calculated and are given in table 3.4. A cell in that table gives the similarity between the two corresponding terms. The terms “system” and “EPS” (coloured in green in the tables below) co-occur three times in the sample data set and their cosine is 0.87, the highest measured in this example (apart from identical terms). The terms “human” and “user” (coloured in red in the tables below) have a cosine of 0, the lowest value possible in the standard VSM, as they never co-occur directly.

The standard VSM only takes direct co-occurrence hereinafter referred to as first order co-

	computer	human	interface	response	survey	system	time	user	eps	trees	graph	minors
computer	1.00	0.50	0.50	0.50	0.50	0.29	0.50	0.41	0.00	0.00	0.00	0.00
human	0.50	1.00	0.50	0.00	0.00	0.58	0.00	0.00	0.50	0.00	0.00	0.00
interface	0.50	0.50	1.00	0.00	0.00	0.29	0.00	0.41	0.50	0.00	0.00	0.00
response	0.50	0.00	0.00	1.00	0.50	0.29	1.00	0.82	0.00	0.00	0.00	0.00
survey	0.50	0.00	0.00	0.50	1.00	0.29	0.50	0.41	0.00	0.00	0.41	0.5
system	0.29	0.58	0.29	0.29	0.29	1.00	0.29	0.47	0.87	0.00	0.00	0.00
time	0.50	0.00	0.00	1.00	0.50	0.29	1.00	0.82	0.00	0.00	0.00	0.00
user	0.41	0.00	0.41	0.82	0.41	0.47	0.82	1.00	0.41	0.00	0.00	0.00
eps	0.00	0.50	0.50	0.00	0.00	0.87	0.00	0.41	1.00	0.00	0.00	0.00
trees	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.67	0.41
graph	0.00	0.00	0.00	0.00	0.41	0.00	0.00	0.00	0.00	0.67	1.00	0.82
minors	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.41	0.82	1.00

Table 3.4: *Cosine term similarity for the sample data set from table 3.2 in standard vector space.*

occurrence into account. LSA on the other hand takes more information into account and provides a better representation of term relations. Table 3.5 shows the cosine values for term-to-term similarities calculated in LSA space from the matrix $\mathbf{B}$ in table 3.3. Thus tables 3.4 and 3.5 show the similarity scores for the same data once calculated in standard vector space (table 3.4) and once in LSA space (table 3.5).

	computer	human	interface	response	survey	system	time	user	eps	trees	graph	minors
computer	1.00	0.21	0.85	0.99	0.85	0.30	0.99	0.99	0.25	-0.05	0.04	0.07
human	0.21	1.00	0.68	0.06	0.11	0.99	0.06	0.22	0.99	-0.01	-0.01	-0.02
interface	0.85	0.68	1.00	0.77	0.63	0.75	0.77	0.86	0.72	-0.15	-0.09	-0.06
response	0.99	0.06	0.77	1.00	0.83	0.16	1.00	0.99	0.11	-0.08	0.01	0.05
survey	0.85	0.11	0.63	0.83	1.00	0.20	0.83	0.82	0.15	0.48	0.56	0.59
system	0.30	0.99	0.75	0.16	0.2	1.00	0.16	0.31	0.99	0.005	0.003	0.003
time	0.99	0.06	0.77	1.00	0.83	0.16	1.00	0.99	0.11	-0.08	0.01	0.05
user	0.99	0.22	0.86	0.99	0.82	0.31	0.99	1.00	0.27	-0.09	-0.01	0.02
eps	0.25	0.99	0.72	0.11	0.15	0.99	0.11	0.27	1.00	-0.02	-0.02	-0.03
trees	-0.05	-0.01	-0.15	-0.08	0.48	0.005	-0.08	-0.09	-0.02	1.00	0.99	0.99
graph	0.04	-0.01	-0.09	0.01	0.56	0.00	0.01	-0.01	-0.02	0.99	1.00	0.99
minors	0.07	-0.02	-0.06	0.05	0.59	0.003	0.05	0.02	-0.03	0.99	0.99	1.0

Table 3.5: *Cosine term similarity for the sample data set from table 3.2 in LSA space*

In LSA co-occurrences of higher order are taken into account. For example “human” and “user” have a similarity value of 0.22 in the LSA space instead of 0 in the VSM. The relation between the terms are second-order via co-occurrences with “system”: “human” co-occurs

twice with "system" and "system" co-occurs twice with "user". Other co-occurrence chains are: "human-interface-user", "human-computer-user", "human-EPS-user".
An example of third order co-occurrence is the pair "trees" (coloured in blue) and "system". They receive a similarity value of 0.005 in the LSA space instead of 0 in standard VSM. Their co-occurrence chain is "trees-graph-survey-system". In the standard vector space, the two latter pairs receive a similarity value of 0 and are both marked as not similar. In LSA space these pairs are discriminated, they receive different cosine values based on their different co-occurrence patterns. LSA allows one to find more relationships between terms and to get a more differentiated view on the data set and its underlying relations. To calculate these underlying relations Singular Value Decomposition (SVD) is used.

3.2.2 Introduction to Singular Value Decomposition

Singular Value Decomposition (SVD) is a method from the field of linear algebra. The purpose of SVD is to diagonalize any $t \times d$ matrix $\mathbf{A}$. The diagonalization corresponds to a transition to a new coordinate system (Lang and Pucker, 1998). This transition brings forth the latent semantic structure of a document set.

To explain the effect of SVD, I will use an example from image compression, where SVD is used to optimize the relation between image quality and file size. Figure 3.2 shows pictures of a clown with different quality.

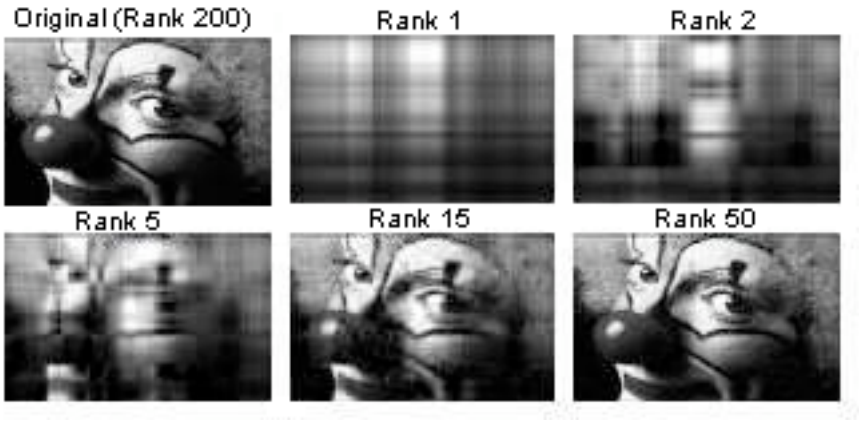


Figure 3.2: *SVD in image compression: View the $m \times n$ image as a matrix. The rank of this matrix was reduced from $r = 200$ to $k = 1, 2, 5, 15, 50$. Hardly any difference is visible between the rank $k = 50$ approximation of the image and the original, but the file size is reduced from $m \times n$ to $k(m + n)$. (Source: Persson (2007))*

Any picture can be seen as a matrix, where each cell contains a number that corresponds to a colour or a grayscale. The upper left image in figure 3.2 shows the original picture, a matrix of rank $r = 200$. The rank (r) of a matrix is the smaller of the number of linear independent rows and columns. SVD is used to reduce the rank and thereby the file size of the image. If the rank is reduced to $k = 1$ or $k = 2$ the image of the clown is not recognizable. The clown can

be recognized in the rank 15 ($k = 15$) approximation but the image is blurred. At rank $k = 50$ hardly any difference between the approximation and the original image can be detected, but the file size is reduced from $m \times n$ to $k(m + n)$.

The clown can be recognized because SVD emphasizes the most essential features and information while unimportant details are suppressed. On the highest level of abstraction (rank-1 approximation) only the very basic structure of the image is depicted. SVD ranks the features by importance for the image. By reducing the rank to k, only the first k features are kept.

One aims to find the optimal rank approximation where all and only the important information is shown. If the important features are not all captured the picture cannot be recognized, whereas if too many features are kept the data structure is unnecessarily large.

In the example in section 3.2.1 SVD finds concepts and relations between terms in the term-by-document matrix $\mathbf{B}$ and ranks them by importance. Only the k most important concepts are kept, and the term similarity is calculated on the basis of the reduced matrix. The benefit of this reduction to the optimal rank-k approximation is that the term similarity calculation is only based on the most characteristic features of the document collection at hand. The noise that *blurs* the *clear* view on the hidden relations is suppressed.

Mathematically speaking the characteristic concepts of a term-by-document matrix are its eigenvectors. First I will explain Eigenvalue Decomposition (EVD) for square matrices from which SVD for rectangular matrices is derived. The goal of EVD is to find eigenvectors $\vec{x}$ that point in the same direction as $\mathbf{A}x$, i.e., vectors that satisfy equation 3.6.

$$\mathbf{A}\vec{x} = \lambda\vec{x} \tag{3.6}$$

Here λ is an eigenvalue, which determines the scaling of the corresponding eigenvector $\vec{x}$. For example the eigenvectors for the following matrix $\mathbf{C}$ of rank 3 are $\vec{x}_1$, $\vec{x}_2$ and $\vec{x}_3$:

$$\mathbf{C} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 4 \end{pmatrix} \quad \Rightarrow \quad \underset{\lambda_1 = 9}{\vec{x}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}} \quad \underset{\lambda_2 = 4}{\vec{x}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}} \quad \underset{\lambda_3 = 2}{\vec{x}_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}}$$

In the case of a diagonal matrix the eigenvectors are the canonical unit vectors, i.e., the vectors spanning the coordinate system. Equation 3.6 can be solved by subtracting $\lambda\vec{x}$ to obtain:

$$(\mathbf{A} - \lambda\mathbf{I})\vec{x} = 0 \tag{3.7}$$

Here $\mathbf{I}$ is the unit matrix – a diagonal matrix where the main diagonal consists only of ones. If this equation has a non-trivial solution, then $\mathbf{A} - \lambda\mathbf{I}$ is not invertible, which means there is no $\mathbf{B}^{-1} = (\mathbf{A} - \lambda\mathbf{I})^{-1}$ that fulfils $\mathbf{B}^{-1}\mathbf{B} = \mathbf{B}\mathbf{B}^{-1} = \mathbf{I}$. From that it follows that the determinant of $(\mathbf{A} - \lambda\mathbf{I})$ has to be 0:

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0 \tag{3.8}$$

For a detailed derivation of this transformation see Strang (2003). With this equation eigenvalues λ can be calculated since $det(\mathbf{A} - \lambda\mathbf{I})$ will result in a polynomial of r^{th} order.

The procedure described here is called Eigenvalue Decomposition (EVD) since it can only be applied to certain classes of square matrices. In IR most term-by-document matrices are rectangular hence the generalization for rectangular matrices, Singular Value Decomposition (SVD), is used. SVD and EVD are related. EVD decomposes a square matrix $\mathbf{C}$ into two submatrices $\mathbf{Q}$ and $\mathbf{\Lambda}$ where $\mathbf{Q}$ represents the eigenvectors and the eigenvalues are listed in descending order in matrix $\mathbf{\Lambda}$:

$$\mathbf{C} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} \tag{3.9}$$

In contrast to a square matrix a rectangular matrix has two sets of eigenvectors, the right singular vectors and the left singular vectors. SVD decomposes any rectangular $t \times d$ matrix $\mathbf{A}$ into three submatrices $\mathbf{T}$, $\mathbf{S}$ and $\mathbf{D}$ (figure 3.3). The left singular vectors are represented by $\mathbf{T}$, the right singular vectors by $\mathbf{D}$.

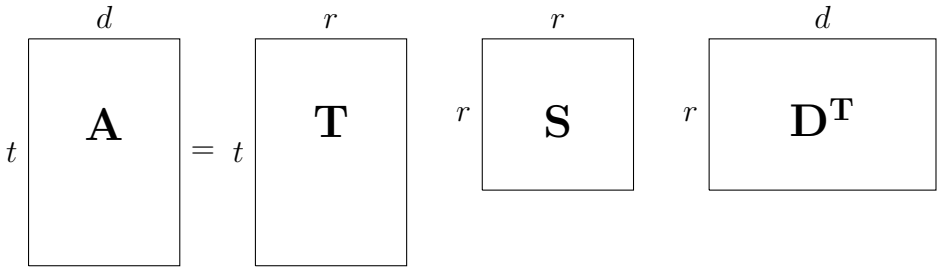


Figure 3.3: *Singular Value Decomposition: $\mathbf{A}$ is a $t \times d$ matrix, where t is the number of index terms, d the number of documents indexed, and r the rank of the matrix $\mathbf{A}$.*

Any rectangular matrix $\mathbf{A}$ is squared by multiplying it by $\mathbf{A}^T$. The eigenvectors of $\mathbf{A_T} = \mathbf{A}\mathbf{A}^T$ are the left singular vectors of $\mathbf{A}$ and the eigenvectors of $\mathbf{A_D} = \mathbf{A}^T\mathbf{A}$ are the right singular vectors of $\mathbf{A}$. The eigenvectors and the eigenvalues for these auxiliary matrices can be calculated by EVD as described above.

Singular values are the square roots of the common eigenvalues of $\mathbf{A_T}$ and $\mathbf{A_D}$ and are written in descending order in $\mathbf{S}$. The eigenvectors in $\mathbf{T}$ and $\mathbf{D}$ are ordered correspondingly.

Only when the term-by-document matrix is decomposed into these three submatrices is it possible to reduce the number of dimensions of the semantic space and thereby the number of concepts (or features). In that case only the first k singular values in $\mathbf{S}$ and the corresponding vectors in $\mathbf{T}$ and $\mathbf{D}$ are kept. This number of remaining dimensions (k) is a crucial value for the performance of any LSA based application. If too many dimensions are kept, the latent semantic structure cannot be revealed because the documents and words are not projected near enough to each other and too much noise is left. If k is too small then too many words and/or documents will be superimposed on one another, destroying the latent semantic structure.

The three sub-matrices $\mathbf{T_k}$, $\mathbf{S_k}$ and $\mathbf{D_k}$ describe the coordinate system, the semantic space

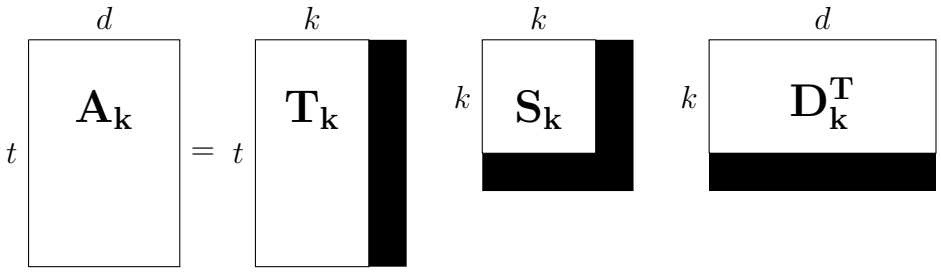


Figure 3.4: *Reduced Singular Value Decomposition:* $\mathbf{A}$ *is a* $t \times d$ *matrix, where* t *is the number of index terms,* d *the number of documents indexed,* r *the rank of the matrix* $\mathbf{A}$*, and* k *the number of dimensions kept.*

for a document collection. The derived concepts or topics of the document collection are depicted in $\mathbf{D_k}$, and the word distribution patterns in $\mathbf{T_k}$. In spatial terms the rows of the matrices $\mathbf{T_k}$ and $\mathbf{D_k}$ are the coordinates of points representing the terms and documents in reduced k dimensional space. The matrix $\mathbf{S}_k$ is used to rescale the axes in order to be able to compare different objects to each other.

Depending on the type of similarity calculation required, the submatrices are multiplied with $\mathbf{S}$. For term-to-term similarity calculation the vectors of the matrix $\mathbf{CT_k} = \mathbf{T_kS_k}$ are used. To compare documents with each other the distances between the vectors of the matrix $\mathbf{CD_k} = \mathbf{D_kS_k}$ are calculated. If the task is to compare a term to a document the vectors of the matrices $\mathbf{CTD1_k} = \mathbf{T_kS_k^{\frac{1}{2}}}$ and $\mathbf{CTD2_k} = \mathbf{D_kS_k^{\frac{1}{2}}}$ are used to calculate the cosine similarity. For my experiments I use the scaled space $\mathbf{CD_k} = \mathbf{D_kS_k}$. Since I explore the potential of LSA in the field of sentence clustering for MDS, I will call this space the clustering space. With each k (number of remaining dimensions) a different clustering space is formed.

Yu et al. (2002) summarizes the advantages of LSA as follows:

> "The SVD algorithm preserves as much information as possible about the relative distances between the document vectors, while collapsing them down into a much smaller set of dimensions. In this collapse, information is lost, and content words are superimposed on one another. Information loss sounds like a bad thing, but here it is a blessing. What we are losing is noise from our original term-document matrix, revealing similarities that were latent in the document collection. Similar things become more similar, while dissimilar things remain distinct. This reductive mapping is what gives LSI its seemingly intelligent behaviour of being able to correlate semantically related terms. We are really exploiting a property of natural language, namely that words with similar meaning tend to occur together."

In contrast to the VSM the dimensions of the vector, which represents a sentence, do not correspond to a term but rather to a concept or a word usage pattern. Thus the similarity calculation for sentence clustering using LSA is not only based on word matching but on latent semantic relations of the terms.

3.3 Data set

The data set that is used throughout the experiments in this thesis and that was given to human annotators was compiled from DUC sets. The document sets were created by DUC for different challenges such as single-/multi-document summarization, topic focused summarization, or update summaries. For my research I chose six document sets from various DUC from 2002, 2003, 2006 and 2007. DUC clusters are categorized in different types of clusters. I chose document sets from the following categories:

Single natural disaster Sets of documents that describe a single natural disaster event and were created within a seven-day window.

Single event Sets of documents that describe a single event in any domain and were created within a seven-day window.

Biography Sets of documents that present biographical information mainly about a single individual.

These categories were chosen because document sets from these groups can be summarized relatively easily by one generic summary. Other document sets that, e.g., describe multiple distinct events of a single type were not used for my research as they were designed for topic focused summarization or update summarization. From six DUC clusters of documents I extracted sets of sentences that met certain requirements and constraints.

Particularly the newer document clusters (e.g., from DUC 2006 and 2007) contain many documents and therefore many sentences. To build good sentence clusters, human annotators have to compare each sentence to each other sentence and maintain an overview of the topics within the documents. Because of human cognitive limitations the number of documents and sentences had to be reduced. For my experiments I defined a set of constraints for a sentence set:

1. A sentence set must consist of sentences from at least 5 and not more than 15 documents from one DUC document set.
2. A sentence set should consist of 150 – 200 sentences.
3. If a DUC set contains only 5 documents all of them are used to create the sentence set, even if that leads to more than 200 sentences.
4. If a DUC set contains more than 15 documents, only 15 documents are used for clustering even if the number of 150 sentences is not reached.

To obtain sentence sets that comply with these requirements, I designed an algorithm that takes the number of documents in a DUC set, the date of publishing, the number of documents published on the same day, and the number of sentences in a document into account. If a document

set includes articles published on the same day they were given preference, because they tend to be more similar to each other and have the same standard of knowledge, e.g., they do not tend to vary in numbers of casualties. Furthermore shorter documents (in terms of number of sentences) were favoured (for more details see section 3.4). The properties of the resulting sentence sets are listed in table 3.6.

Name	DUC	DUC id	Docs	Sent	Type	Topic
EgyptAir	2006	D0617H	9	191	single event	Crash of the EgyptAir Flight 990
Hubble	2002	d116i	6	199	single event	Launch of Hubble space telescope
Iran	2002	d103g	9	185	natural disaster	Earthquake in northern Iran in 1990
Rushdie	2007	D0712C	15	103	biography	"Death sentence" on Salman Rushdie proclaimed by Iran
Schulz	2003	d102a	5	248	biography	Death of Charles Schulz, creator of the Peanuts
Volcano	2002	d073b	5	162	natural disaster	Eruption of Mount Pinatubo

Table 3.6: *Details of sentence sets*

3.4 Implementation

For the experiments described in this thesis I used BOSSE^{Clu}. BOSSE, which is implemented in Python, was developed as part of my earlier work at the University of Heidelberg (Geiß, 2006) as a local search engine for Wikipedia articles using Latent Semantic Indexing (LSI). For the research described in this dissertation I adapted BOSSE in order to process DUC document sets and to create sentence clusterings from them. BOSSE was also extended to process other corpora like MEDLINE, Cranfield, and MSRPC. I changed the internal structure of BOSSE to make it more efficient in terms of time and memory. I introduced new classes and changed the way matrices and information about terms, sentences and documents are stored. I also added the VSM. A user calling BOSSE can specify whether the standard vector space or the LSA space should be used for IR or clustering. This makes it easier to compare LSA and VSM and assures that the same indexing and weighting schemes are used.

To distinguish the new version of BOSSE from the old, I refer to it as BOSSE^{Clu}. Figure 3.5 shows the system framework of BOSSE^{Clu}, which is composed of two parts: preparation and clustering.

3.4.1 Preparation

During preparation documents are read in and split into sentences. Sentence sets are created and indexed. Term-by-sentence matrices are built and SVD is performed.

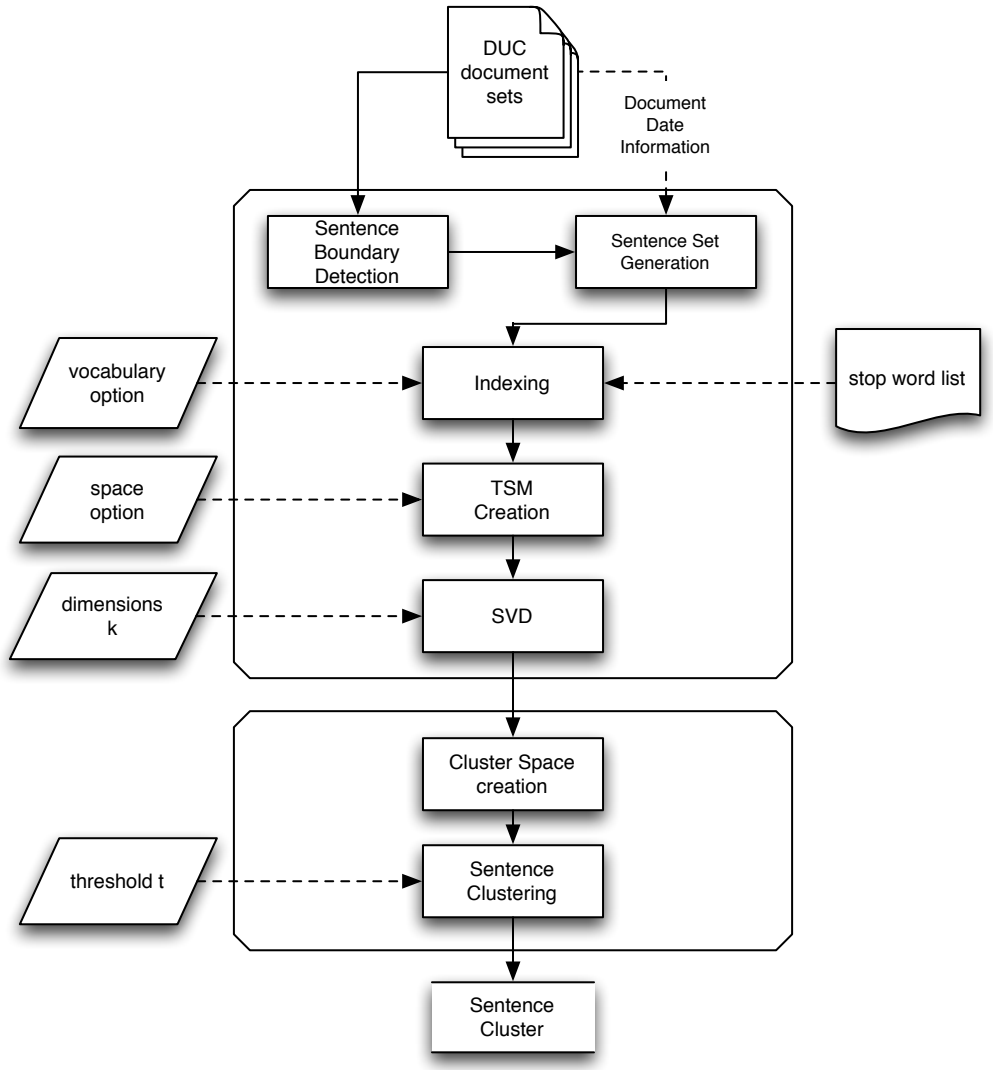


Figure 3.5: *System framework of* BOSSE^{Clu}

Sentence boundary detection

To split the documents of a document set into sentences, BOSSE^{Clu} uses the sentence boundary detector from RASP (Briscoe et al., 2006). After all documents from the sets have been segmented, sentence sets are generated.

Sentence set generation

Some constraints for a sentence set were defined (for details see section 3.3). To obtain sentence sets that comply to these specifications I designed an algorithm which I will outline here. The algorithm takes the date a document was published, the agency it was published by and the number of sentences it contains into account. First the documents from a set are sorted by date and then ranked. The top ranks are taken by documents that were published on the same

day. They are followed by articles published on consecutive days. There might be several blocks of consecutive days on which articles were published. From each of these blocks the shortest document for each agency is selected. The remaining documents are sorted by number of sentences and are added to the rank list. This approach ensures that as many different articles (from different agencies) as possible are present in a sentence set and that the articles have a lot of information in common. By giving preference to documents published on the same day or on consecutive days it is ensured that the authors had a similar knowledge base. Thus the proportion of redundant information is maximized. The sentences from the documents are added to the sentence sets according to the rank list. Before sentences are added, the algorithm checks whether, by including all the sentences from a document, the maximum number of documents or the maximum number of sentences in a sentence set is exceeded. Only complete articles are added to the sentence sets. Regardless of the number of sentences already in the sentence set, documents (more particularly their sentences) are added to the sentence set until the minimum number of documents is reached. The sentences in a set are then ordered by date before they are given to human annotators.

Indexing

After the sentence sets have been created, a list of keywords for each sentence set is extracted using the Python Natural Language Toolkit (NLTK) (Bird et al., 2009). During indexing stop words[5] are removed. The list of keywords is also filtered to include only keywords that appear in more than one sentence. The list is extended or filtered in accordance with the vocabulary option chosen. There are nine different options to choose from (see section 4.2.2 for details). Some options restrict the vocabulary to nouns and verbs or to nouns only. To extract the word class for the index keywords, the part-of-speech (PoS) tagger from RASP (Briscoe et al., 2006) is used. For other vocabulary options, collocations are extracted from the sentence sets and added to the index. Collocations are groups of words that often appear consecutively within one/all sentence sets. To extract collocations of two and three words, the collocation module of NLTK (Bird et al., 2009) is used. Examples for extracted collocations from the dataset are *Charlie Brown*, *United States*, *Salman Rushdie* and *death sentence*.

TSM creation

Once the indexing is finished, term-by-sentence matrices (TSM) are built according to the space option specified by the user (for more details see section 4.3). The cells of the matrices include ntf-isf weighted frequencies of occurrence of keywords in sentences. The isf value is also used to filter out keywords that appear in almost all sentences of a sentence set.

[5]The stop word list created for the SMART information retrieval system (Salton, 1971b) was used.

SVD

After generating the TSM(s), SVD is applied (section 3.2). In BOSSE^{Clu} SVDLIBC (Rohde, 2008) is used to calculate the singular value decomposition of the TSM(s). The matrices are truncated in accordance with the parameter k specified by the user. The three resulting submatrices $\mathbf{T}_k$, $\mathbf{S}_k$ and $\mathbf{D}_k$ are saved to files alongside all other relevant information.

3.4.2 Clustering

For clustering sentences two steps need to be carried out: clustering space creation and sentence clustering.

Cluster space creation

First the cluster space needs to be calculated. This space is created using the three truncated submatrices from SVD. Depending on what kind of objects will be clustered (terms, sentences, terms and sentences) the appropriate LSA space has to be calculated. For sentence clustering the space is given by $\mathbf{CD_k} = \mathbf{D_k S_k}$.

Sentence clustering

For sentence clustering, BOSSE^{Clu} uses an implementation of a hierarchical agglomerative clustering algorithm (HAC) called hcluster (Eads, 2008a), which is a Python library whose interface and range of functions are comparable to those of MATLAB®'s Statistics Toolbox. All parameters specified in the following section 3.5 can be passed to BOSSE^{Clu}. The only addition to that implementation for my research is that only clusters that contain sentences from different documents are used, thus the clusters created by hcluster are filtered. The clusters that contain sentences from only one document are added to the clustering as singletons. This feature was added since the human annotators were given the rule that each cluster they create must contain sentences from at least two different documents.

The advantage of my own implementation is that I can easily change the source code and can add all cluster options I want to investigate. I retain full control of the indexing, decomposition and clustering process.

3.5 Clustering algorithm

Clustering is a method of unsupervised learning. Clustering is defined as an assignment of objects into clusters such that the clusters are internally as coherent as possible but clearly distinguishable from each other. That is to say the objects within the same cluster should be similar in some sense and objects in one cluster should be different to objects from other clusters.

There are two types of clustering methods – hard and soft. In soft clustering an object can belong to several clusters; a probability of a sentence belonging to a cluster is given. On the other hand in hard clustering a sentence can belong to exactly one cluster. For sentence clustering in MDS, I use a hard clustering algorithm.

Clustering algorithms can also be divided into two groups: (i) hierarchical and (ii) partitional (Manning and Schütze, 1999). Partitional or non-hierarchical algorithms (e.g., k-means clustering) start out with k randomly selected centres of clusters. This k is different to the k in LSA. Here k represents the number of clusters. These randomly selected centres of clusters are called seeds. Each object is then assigned to its nearest seed. When all objects are assigned to a seed, the cluster centres called centroids are computed. A centroid of a cluster is the average of all points in a cluster. Each object is assigned to its nearest centroid and then the centroids are recomputed. These latter steps are repeated until a convergence criterion is met. One disadvantage is that when the algorithm is run on the same data set several times the results may vary, due to the fact that the seeds are randomly selected. Another drawback of this approach is that the number of clusters has to be set a priori. This is not feasible for MDS. There is no way of knowing a priori how many topic clusters there are in a cluster of documents.

Hierarchical clustering algorithms produce trees of clusters also known as cluster hierarchies or dendrograms (see figure 4.1 for an example). Each node represents a cluster consisting of all the objects of its descending nodes (its children). That is, each node (except for the root node) is a subclass of its parent node. The leaves of the tree represent the individual objects to be clustered, here sentences. There are two approaches to hierarchical clustering, agglomerative clustering and divisive clustering. The divisive or top-down algorithm starts with one cluster containing all objects. In each step the cluster that is least cohesive is determined and split. Agglomerative clustering or bottom-up clustering starts with leaves. Each leaf is interpreted as a separate cluster; such clusters containing only one object are called singletons. In each iteration the two clusters with the maximum similarity (or minimum distance) are merged. The algorithm stops when one cluster is left containing all objects.

To determine the similarity (or distance) between clusters it has to be defined (i) what the distance between two clusters is (linkage criteria) and (ii) how the distance is calculated (distance metric). There are three commonly used linkage criteria:

Single link The distance of two clusters is equal to the distance between the two closest (most similar) objects in the clusters.

Complete link The distance of two clusters is equal to the distance between the two furthermost (most dissimilar) objects in the clusters.

Average link The distance of two clusters is equal to the average distance between the objects.

Common distance metrics are:

Cosine metric The similarity of two sentences is given by the cosine of the angle between the two vectors describing the sentences.

Euclidean distance The similarity of two sentences is the length of a line segment connecting two vectors describing the sentences.

Jaccard distance The similarity of two sentences is the number of different entities divided by the number of shared entities.

For sentence clustering I used a Python implementation of the hierarchical agglomerative clustering (HAC) algorithm called hcluster (Eads, 2008a). The Python library hcluster provides functions to generate hierarchical clusterings from distance matrices computed from observation vectors. It also offers visualization of the clusterings with dendrograms and different methods of splitting the cluster tree into flat clusters (Eads, 2008b).

Chapter 4

Parameters in sentence clustering

> Anyone who doesn't take truth seriously in small matters cannot be trusted in large ones either.
>
> ALBERT EINSTEIN

This chapter discusses various parameters that need to be thought of when clustering sentences in general and when using LSA in particular. First the parameters that need to be set for the hierarchical clustering algorithm are examined in section 4.1. In section 4.2 I will discuss the issue of creating an index vocabulary. The idea of different sizes of an LSA space is explained and discussed in section 4.3. The important role of the LSA parameter k and how it might influence results is discussed in section 4.4.

4.1 Clustering algorithm parameter

As described in section 3.5 a hierarchical agglomerative clustering algorithm (HAC) was used for automatic sentence clustering. There are different parameters that can influence the quality of sentence clusters. In section 3.5 I listed several linkage criteria and distance metrics. I will now experimentally determine which of them are appropriate for sentence clustering. The Microsoft Research Paraphrase Corpus (MSRPC) (Dolan et al., 2004; Dolan and Brockett, 2005) was used for this purpose. It was originally designed for research on paraphrase detection. The corpus contains pairs of sentences, which are annotated for equivalence. This corpus was useful because a gold standard for sentence clustering can be easily created from it. Although the corpus was created for research on paraphrasing, it is a good starting point for evaluating the clustering capabilities of LSA, since a pair of paraphrases is a special case of a sentence cluster. It is also a good opportunity to find optimal settings for some clustering parameters.

The MSRPC was created from 11,162 document clusters with a total of 117,095 news articles from thousands of news sources. Different automatic heuristics were applied to find

paraphrases within these clusters. After 49,375 pairs of possible paraphrases were found, a classifier, which uses feature classes such as string similarity features, morphological variants and WordNet lexical mappings, was applied. The resulting data set consists of 20,574 pairs, from which 5,801 were randomly chosen to be evaluated by human judges. 67% of the pairs were rated as semantically equivalent. The human-evaluated segment of 5,801 paragraph pairs was then split into the MSRPC training and the MSRPC test set containing 4,076 and 1,725 sentence pairs respectively.

I derived a gold standard for sentence clustering from the resulting sentence set (hereinafter called MSRPC_GS) by following transitive chains. The assumption is that, if sentence A is a paraphrase of sentence B and sentence B is a paraphrase of sentence C, then the sentences A and C, even if they have not been annotated as paraphrases by the judges, must at least have something in common or have the same topic. Since all 5,801 sentence pairs were rated as similar by the classifier, there is a good chance that even sentence pairs not marked as paraphrases are related. Hence these pairs were also used for the creation of MSRPC_GS. Only clusters containing more than two sentences are included in MSRPC_GS. According to general nomenclature in clustering the clusters in the gold standard are called *classes* as opposed to *clusters* in a clustering automatically generated by a system. For the MSRPC training set this approach resulted in 277 classes containing 889 sentences and for the MSRPC test set in 50 classes with 155 sentences.

The MSRPC training corpus was used to compare the different distance metrics (cosine, euclidean), linkage criteria (single link, complete link and average link). The best parameter combination is the cosine metric combined with the average linkage criterion.

4.1.1 Fine-tuning the clustering algorithm

The only parameter that was not determined by this preliminary experiment was the flattening criterion. In a hierarchical cluster tree, any two objects of the data set are linked together at some point in the algorithm. An example of a hierarchical cluster tree is shown in figure 4.1. The numbers along the horizontal axis represent the indices of the sentences in a sentence set. The numbers along the vertical axis represent the distance between the objects (distance $=$ 1-cos). Beginning at the leaves (here the sentences) the objects in the tree are linked to other objects at different heights ending in one root object. The links between objects are represented by forks, i.e., two vertical lines connected by a horizontal line. The height of a link indicates the distance between the two connected objects. Two objects whose link is low are nearer to each other (and thereby more similar) than two objects with a higher link. The height of a link between two objects is known as the cophenetic distance between two objects: it can range from 0 for the minimum distance between the objects and 1 for the maximum distance between two objects.

For sentence clustering in MDS, partitional clusters (also known as flat clusters) are needed. The gold standard, which is used to evaluate the sentence clusterings, also consists of flat clus-

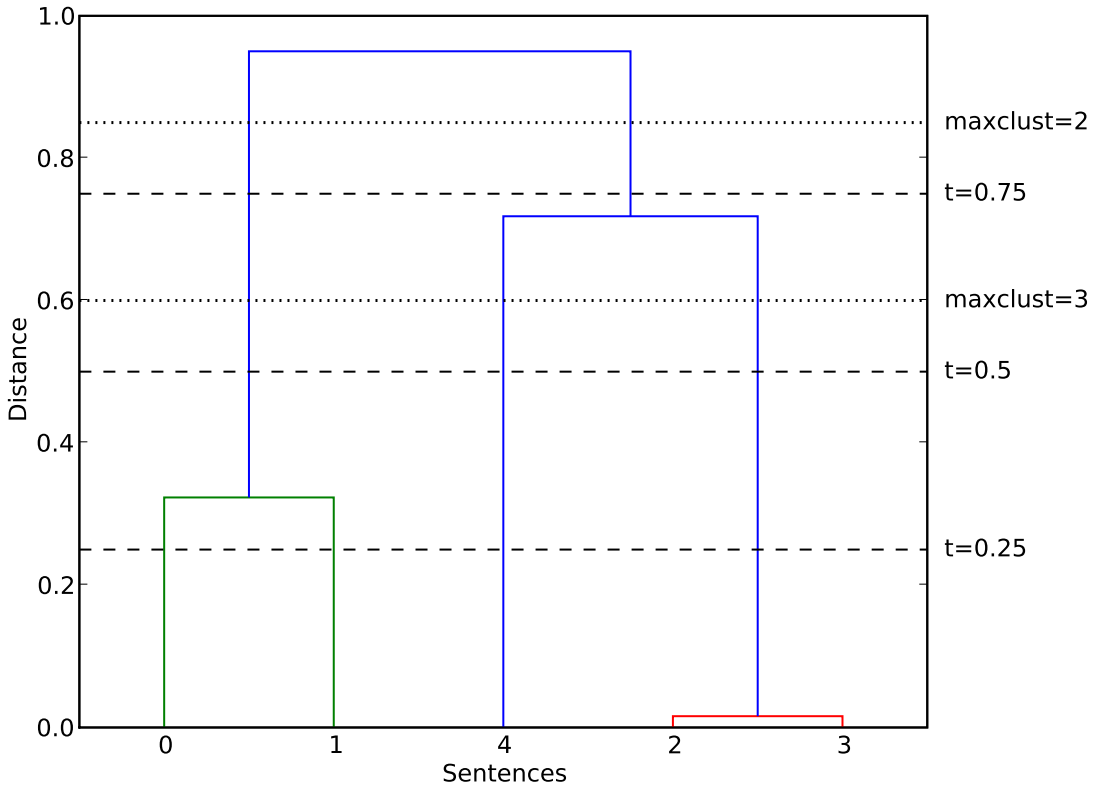


Figure 4.1: *Dendrogram for a sample data set*

ters, where each sentence can belong to exactly one cluster (hard clustering).

In order to separate the cluster tree into flat clusters a cut-off criterion is required, which determines where to cut the tree. This cut-off criterion can be the number of clusters ($maxclust$). If $maxclust = 2$ the tree is partitioned into two clusters. In the dendrogram in figure 4.1 this is visualized by the dotted horizontal line that intersects with two vertical link lines. The dotted line separates the data into two clusters. All objects below the left link line belong to one cluster consisting of objects 0 and 1. All objects below the right line form the second cluster $[4, 2, 3]$. If on the other hand $maxclust = 3$ the tree is cut at the lower dotted line, which results in three clusters $[0, 1][4][2, 3]$. However, since in sentence clustering for MDS the number of clusters is not known a priori, this cut-off criterion is not a feasible way of producing partitional clusters.

Another way to separate the cluster tree into flat clusters is to use a threshold for the cophenetic distance as a cut-off criterion. The cluster tree is then cut at a threshold t represented by dashed lines in figure 4.1. All links with a height $> t$, i.e., above the dashed line, are ignored and only the links below the line (where height $\leq t$) are kept. In figure 4.1, $t = 0.25$ results in the tree being split into four clusters: one cluster consisting of sentence 2 and 3 and three singleton cluster $[0]$, $[1]$ and $[4]$, whereas $t = 0.5$ results in the tree being split into three clusters $[[0, 1][4][2, 3]]$, a division which is identical to the division using $maxclust = 3$. If $t = 0.75$

there are only two clusters $[0, 1]$ and $[4, 2, 3]$. With this criterion the clustering algorithm can be adjusted to create different numbers of flat clusters even if the desired number of clusters is unknown at runtime.

Later I will test four values of t $(0.10, 0.25, 0.50, 0.75)$.

4.2 Vocabulary

The selection of terms for the index vocabulary is an important step in any vector space based application because terms are the entities (i) which can be searched for (IR) and (ii) from which the word usage patterns are built (LSA).

An index vocabulary consists of all the terms (words, n-grams, numbers, acronyms etc.) that are chosen to be a keyword of a sentence. Every term is represented by a weighted frequency vector noted down as a row in the TSM. The weight represents the importance of that term to a given document (or sentence as in the case of sentence clustering). It is often based on the frequency of the term.

4.2.1 Index vocabularies in IR

There are many different ways to create an index vocabulary. Unfortunately several authors do not describe in detail how they created the indexing vocabulary or if term weighting was used. Thus the information provided is not sufficient for reimplementation purposes. IR researchers use many different indexing techniques. I collected results for two standard test collections, MED and Cranfield II. The MED collection (also called MEDLARS 1033) consists of 1033 medical abstracts and 30 queries (Salton et al., 1982; Salton, 1971a). The Cranfield II collection (Cleverdon, 1967) includes 1400 papers on aeronautics and 225 queries. These test collections are widely used to evaluate IR implementations and algorithms. Unfortunately there are no standard indices or indexing procedures, which results in different indices and reduces the comparability of results.

Tables 4.1 and 4.2 give an overview of the different characteristics of indices from IR literature. For the MED corpus I selected three systems that use LSI. Dumais (1990) and Kon-

Article	#terms	Stemmer	Stop list	Weighting	k	P
Dumais (1990)	5831	no	SMART	tf-idf	100	0.67
Dumais (1990)	5831	no	SMART	log E	100	0.72
Zha and Simon (1999)	3681	-	-	-	100	0.66
Kontostathis (2007)	5831	no	SE	log E	75	0.72

Table 4.1: *Differences in indexing methods for MED*

tostathis (2007) seem to use the same index, resulting in similar precision (P) when the log entropy ($\log E$) weighting scheme is used, although they use different number of dimensions (k). Zha and Simon (1999) use considerably fewer terms and they do not specify if a stemmer, stop word removal or weighting were used. For the Cranfield II corpus the situation is similar.

Articles	# terms	Docs	Queries	Stemmer	Stop list	Weighting	k	P
Dumais (1990)	4486	924	100	no	SMART	tf-idf	100	0.40
Kontostathis (2007)	3932	1400	225	no	SE	log E	185	0.45
Hull (1994)	?	1399	219	-	-	-	200	0.45
Jiang and Littman (2000)	3763	1400	225	-	-	-	300	0.41

Table 4.2: *Differences in indexing methods for Cranfield II*

Hull (1994) does not specify how many terms were used to build the TDM or whether a stemmer, stop word removal or term weighting were used. The variance in k is striking, whereas the resulting precision (P) always lies in a range of 0.4 and 0.45.

In the following I will describe two options for extracting an index vocabulary – stop word removal and stemming in more detail. To analyze their influence on the retrieval performance I will test them on the two standard test collections described above using my system BOSSE^{Clu}. I will compare the retrieval performance of different index vocabularies, but all of them include:

- all terms separated by white spaces
- terms that occur in at least two documents
- only year dates (all other numbers are deleted)
- frequencies weighted using $ntf - idf$

Other restrictions are explained where they apply. I use the 3-point-average precision to measure the performance of my system at different numbers for k. For the 3-point-average precision (3-pt-P) the precision of the retrieval system for a given query is averaged at three defined recall levels. Table 4.3 show the results of the experiments.

	MED corpus (1033 documents)				Cranfield II (1398 documents)			
	# terms	k	3-pt-P	VSM 3-pt-P	# terms	k	3-pt-P	VSM 3-pt-P
standard	5810	50	0.70	0.51	4039	250	0.37	0.36
+ stemmer	4244	67	0.71	0.51	2548	200	0.38	0.38
- stop word removal	6141	75	0.63	0.50	4355	400	0.35	0.35

Table 4.3: *Summary of experiments on MED and Cranfield II using* BOSSE^{Clu}

Stop word removal

The idea of a stop word list is to remove all common words such as “he”, “a” or “who”, that do not carry significant information. Gerard Salton and Chris Buckley created a stop word list for their SMART information retrieval system (Salton, 1971b). This list[6] is widely used and includes 571 common English words.

The results show that not removing words listed on the stop word list results in a decrease of performance of 10% for MED and 5.4% for the Cranfield II corpus.

Stemming

Most LSA systems do not use stemming. Dumais (1990) states that the improvement using a stemmer only lies in the range of 1%-5% . In some cases stemmers can even decrease results. On the other hand Frakes (1992) states that there is no evidence that stemming can degrade retrieval performance. He argues that the effect of stemming is dependent on the nature of the vocabulary, therefore a stemmer improves the retrieval performance for some test collections more than for others. For my experiments I used the Porter stemmer (Porter, 1980) and the raw frequencies in the TDM were weighted using the ntf-idf weighting scheme (see section 3.1.1). Only terms that are not listed on a modified SMART list of stop-words were included in the index. All numbers apart from years were deleted. The results of my experiments using the Porter stemmer with Bosse on Cranfield II and MED regarding the effect of stemming can be seen in the first two rows of table 4.3. Using the porter stemmer results in an increase in LSA retrieval performance of 1.4% for the MEDLINE corpus and 2.7% for the Cranfield II collection. However since the increase is marginal and other researchers state that stemming might even decrease retrieval performance I chose not to use stemming for the creation of an index vocabulary.

Conclusion

If an experiment with VSM and LSI is to be repeatable and its results are to be comparable, it is important to specify all steps and tools leading to the index vocabulary and TDM used for the experiments. As a result I chose the standard indexing described above (no stemming, stop word removal, keeping year and removing all other numbers, using the ntf-idf weighting scheme and keeping only these words that occur in more than one document). This works well and offers a good compromise between good average precision and run time.

[6]It is included NLTK and is also available at `ftp://ftp.cs.cornell.edu/pub/smart/english.stop`.

4.2.2 Varying vocabularies for sentence clustering

The selection of index terms, also known as index vocabulary, is an important step for any vector space based application. In LSA only the selected terms can be used to generate word usage patterns and thereby reveal the underlying concepts of the sentence sets and the similarity between the sentences. If the keywords selected do not represent the meaning or *aboutness* of the sentences properly the similarities of the sentences cannot be determined reliably and sentence clustering might produce inadequate clusters. Therefore the selection and processing of index terms is a crucial step and will be examined in more detail.

It is often difficult to find evidence in research papers of how the index vocabulary was obtained, what kind of terms were used (nouns, words, multi-word expressions, n-grams, numbers etc.) and how they were edited. In section 4.2, I gave a short overview of some options in relation to index vocabulary creation. The results from these experiments show that the use of a stemmer results only in marginal improvement of retrieval results, if any at all. Removal of stop words however increases the retrieval performance considerably. As Dumais (1990) reported, weighting has a positive effect on information retrieval. Following these results I will remove common stop words from the index vocabulary and the frequency of a term in a sentence will be weighted using the ntf-isf weighting scheme since the preliminary experiments showed that it outperforms the tf-isf scheme.

In previous research some other options for selecting index terms were introduced. It is claimed that the nouns of a sentence carry most of its meaning and that they are the main characteristics to distinguish between sentences or documents (Baeza-Yates and Ribeiro-Neto, 1999). Thus many IR systems only use nouns as index terms. This trend is also recognizable in summarization. Barzilay and Elhadad (1997) for example only select nouns and noun compounds for candidate terms for lexical chains. Bouras and Tsogkas (2008) reports that boosting weights of nouns results in an increase in summarization performance. The most effective features for finding similarity between short passages of text are simplex noun phrases overlap and noun overlap (Hatzivassiloglou et al., 1999). Other authors do not give information on which terms were selected as keywords. Aliguliyev (2006) only speaks of words/terms occurring in a document. Hachey et al. (2006) used 1000 content-bearing terms.

From the example given above it can be seen that there is no consensus on which terms to use for similarity measurement in summarization or text similarity calculation. Therefore I examined the influence of different index vocabularies on sentence clustering (and therefore sentence similarity measurement) for MDS. I chose eight different strategies of keyword selection:

SV: all tokens separated by white spaces, longer than three characters and not on the stop word list

NUM1: like SV but all numbers are replaced by the string `#num`

NUM2: like SV but all numbers < 1492 and > 3000 are replaced by their numbers of digits

COLL: like SV but collocation (bigrams and trigrams) are added[7]

COLL+NUM1: combination of COLL and NUM1

COLL+NUM2: combination of COLL and NUM2

NV: like SV but all terms that are not nouns or verbs are removed[8]

NV+COLL: like NV but collocations are added[7]

N: like SV but all terms that are not nouns are removed[8]

I tested these options and how they influence the quality of sentence clusterings in an experiment described in section 7.2.

4.3 Size of semantic space

In many fields of NLP a large knowledge base or large corpus is advantageous if one wishes to draw conclusions from data. For example Banko and Brill (2001) presented a study of the effect of data size on machine learning for natural language disambiguation. They showed that various machine learning algorithms can benefit from larger training sets. Foltz et al. (1998) and Barzilay and Lapata (2008) used a larger corpus to get a reliable semantic space in order to automatically judge the coherence of documents. Other applications use a background corpus to broaden the knowledge and the amount of information about words. Zelikovitz and Kogan (2006) did some experiments on using web searches to create background corpora for text classification. By creating a background corpus relevant to the classification domain at hand, one can acquire additional knowledge and improve the accuracy of text classification.

The first attempt to make use of a larger semantic space in MDS was used in the Embra system for DUC 2005 (Hachey et al., 2005). Here a large general semantic space was built from AQUAINT and DUC 2005 data (100+ million words) in order to derive a more robust representation of sentences.

Li et al. (2006) suggest that LSA is more applicable to MDS than to single document summarization as a larger corpus may lead to a more accurate semantic space. However they also point out that a general background corpus might bias term relations. Therefore they propose to create smaller semantic spaces from the sets of documents to be summarized. A similar approached was used in Schütze et al. (1995) for the document routing problem. Wiener et al. (1995) showed that a *local LSI* outperforms *global LSI* for text filtering. Local LSI means that

[7]For extraction of collocations from the data set the collocation finder from NLTK was used (see section 3.4)

[8]For tagging the RASP tagger was used (see section 3.4)

a separate LSI analysis was computed for each category or group of similar categories. They expected that a local representation would be more sensitive to small localized effects, e.g., the correlated use of infrequent terms. They argue that infrequent topics are usually indicated by infrequent terms and that these could be classified as noise by SVD and as a result are "projected out of the LSI representation" (Wiener et al., 1995). Quesada (2007) showed in his experiments that a small data set is indeed more sensitive to isolated groups of objects that are not similar to the rest of the corpus and show only similarity to each other.

To my knowledge the influence of the size of a vector space – in this case an LSA space – on sentence clustering for MDS has not been investigated previously. The question is whether and how cluster quality is influenced by size of the space.

I tested the two hypotheses described above:

- A local space leads to better results since it is more sensitive to local changes.
- A larger space results in higher quality sentence clusters since it provides a more reliable semantic space.

There are three space options that are interesting to test for sentence clustering in MDS:

LOCAL LSA: each sentence set is represented in it own space

EXTENDED LOCAL LSA: all sentence sets from the data set are represented in one space

GLOBAL LSA: not only all sentences from the data set, but also additional external sentences are represented in one space

With LOCAL LSA a separate TSM_{set} is built for each of the six sentence sets (see section 3.3), resulting in six separate clustering spaces for the whole data set. With the EXTENDED LOCAL LSA option one TSM_{all} and hence only one clustering space for all sentences from the six sentence sets is created. Even if only one space is created only the sentences from one set are clustered at a time. This ensures that only sentences from the original sentence set are present in the clusters for that sentence set. The GLOBAL LSA space differs from the previous space options as external sentences, i.e., sentences that are not part of the data set, are added. This results in a larger TSM_{global} containing the index terms and sentences from the data set and from external sources, here from other DUC document sets.

I tested the three space options for sentence clustering using BOSSE^{Clu} in an experiment described in section 7.3.

4.4 Optimal number of dimensions

The most crucial parameter in LSA is the number of dimensions k. As described in section 3.2, k is the number of dimensions that are kept in the three submatrices $\mathbf{T}_k$, $\mathbf{S}_k$ and $\mathbf{D}_k$. If

too many dimensions are kept, the latent semantic structure cannot be revealed because the sentences and words are not projected near enough to each other and too much noise is left. If too few dimensions are kept then words and/or sentences will be superimposed on one another, which means that everything is similar to everything, destroying the latent semantic structure. The choice of the right number of dimensions has always been a problem of LSA. Even the introductory paper Deerwester et al. (1990) noted that the amount of dimensional reduction is critical to LSA. In Dumais (1991) the retrieval performance of LSI on the MED database was evaluated using a range of dimensions. It was reported that the performance increases considerably after 10 or 20 dimension and reaches its climax between 70 and 100 dimensions and then starts to fall slowly. In my preliminary experiments (section 4.2) I obtained similar results. Figure 4.2 shows the development of 3-point-average precision over different values

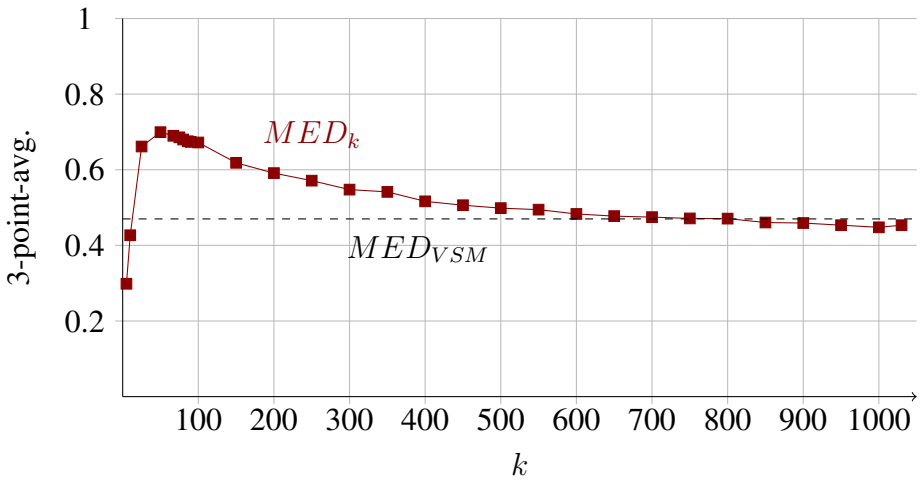


Figure 4.2: *LSA pattern in IR*

of k. This pattern it prototypical for LSI in IR. For very few dimensions the performance is poor, then the performance peaks and falls off slowly. For some IR corpora such as MED there is a substantial range over which LSA outperforms the standard VSM model. Dumais (1991) explains this pattern with the assumption that the performance increases only while added dimensions represent meaningful word usage patterns. The performance then approaches the level of word matching performance. However for other corpora that is not the case. In these cases LSI only outperforms VSM in a small range and the performance of LSI can even lie below that of word matching for some k. Thus it is very important to find the optimal number of dimensions to keep.

Deerwester et al. (1990) states that literature on factor analysis has not provided a proper method to determine the optimal value for k. They choose k empirically instead. This method of choosing the right dimensionality of the search space is only possible when a gold standard for the particular application of LSI is available. Dumais (2007) dismissed this problem and argues that k can be set approximately, as the range of k within which LSA outperforms VSM is rather large (see above). In other papers a standard number of k is used. Dumais (1991) reports that

LSI works well when k is relatively small compared to the number of unique terms. For their further research they use 100 dimensions since they found out that this number works well with the test collections used. Here the assumption is made that the optimal number of dimensions k of the search space is related to the topic variety of the collection and that for homogeneous collections 100 dimensions are sufficient to capture the major word usage patterns.

Landauer and Dumais (2008) specifies that a value of k between 50 and 1000 dimensions is optimal for most language simulations, but that the optimal number of dimensions depends on the domain. For solving the TOEFL test with results similar to those of human learners a 300 dimensional approximation produced the best results (Landauer and Dumais, 1997).
For term comparison Bradford (2008) describes in his study of required dimensionality for large scale LSI applications an *island of stability* in the range of 300 to 500 dimensions. Using a number of dimensions outside this range results in significant distortion in term-term correlations. He specifies that for collections with thousands to tens of thousands of documents, $k = 300$ appeared to be good choice. However for collections containing millions of documents, 400 dimensions produced better results.

For single document summarization Miller (2003) claims that a dimensional reduction of 70-80% works best. He states that $15\% < k < 30\%$ of the original dimensions performs best. For MDS Steinberger and Krišt'an (2007) claim that 10 dimensions are sufficient. Hachey et al. (2005) use a 100 dimensional approximation of their larger semantic space containing 1000 content bearing terms.

On the other hand Quesada (2007) explains that the optimal dimensionality depends on size. He claims that different-sized spaces have completely different properties and that therefore a general method to determine the optimal number of dimensions is not applicable. However there are several techniques for estimating the optimal dimensionality. Skillicorn (2007) described amongst others two methods of determining k automatically: (i) scree plot analysis and (ii) evaluation of the entropy of singular values. In the first method a scree plot, where the singular values are plotted in descending order (Wikipedia, 2011a) is used. This plot sometimes shows a point where the values drop significantly. The first k singular values up to this point are then kept. Another method is Horn's Parallel Analysis (PA) (Horn, 1965). The technique was originally designed to evaluate the components of a principal component analysis (PCA). When used for estimating optimal number of dimensions the eigenvalues that are larger than expected values under term independence (when the columns of $\mathbf{A}$ were orthogonal) are kept. Efron (2002) presents the Amended Parallel Analysis (APA) where standard error is taken into account.

This short overview of optimal number of dimensions in LSA literature shows that the optimal number of remaining dimensions varies between different corpora, domains and applications. That means the optimal dimensionality depends on task, content and space size. In a set of experiments I examined how the quality of sentence clusters is influenced by the number of dimensions in the clustering space $\mathbf{CS}_k$ and if the optimal number of dimensions changes when

other options are incorporated. In the first experiment I checked whether the peaked profile that is typical for LSA in IR is also shown in LSA for sentence clustering in MDS (section 7.4.1). I examined which numbers of dimensions work best for different values of the clustering parameter t (section 7.4.2). In another experiment I tested how the number of optimal dimensions is affected by different sized spaces (section 7.4.3). Another question I tried to answer is whether the size and composition of the indexing vocabulary affects the optimal setting of k (section 7.4.4).

Chapter 5

Evaluation strategy

> True genius resides in the capacity for evaluation of uncertain, hazardous, and conflicting information.
>
> SIR WINSTON CHURCHILL

BOSSE^{Clu} produces clusters of sentences for a given set of related documents. In order to estimate the quality of the clusterings, the sets of clusters have to be evaluated. There are different ways to evaluate clusterings. In general there are two types of evaluation methods for clusterings (Steinbach et al., 2000; Amigó et al., 2009):

Internal cluster evaluation measures intra-cluster similarity, i.e., how close the elements within a cluster are to each other, and inter-cluster similarity, i.e., how close the elements of a cluster are to elements from other clusters.

External cluster evaluation is based on a comparison between the output of a clustering system and an external solution, which is generally a gold standard built by human judges.

Internal cluster evaluation only indicates how well a clustering algorithm performs on the given data representation compared to other algorithms. Internal evaluation can help to fine-tune a cluster algorithm to get the best possible results for the given representation, so that the internal relations are preserved and the clustering solution gives a valid picture of that data representation. Nevertheless this evaluation method does not have the ability to assess the truth of the clustering.

5.1 Gold standard evaluation

By evaluating an automatically generated set of clusters to a gold standard (also called ground truth or set of classes) the correctness and thereby the quality of the clustering solution can be

determined. The system-generated set of clusters is compared to one or several sets of classes and a similarity score is calculated (see section 5.5). The advantage of gold standard evaluation is that it not only rates the quality of the data representation as with internal evaluation, but also measures the quality of the clusters. This is important because a data model that produces clustering with high intra and low inter-cluster similarity can still result in low quality clusterings.

In section 5.2 I present and discuss some existing gold standards for sentence clustering. An overview of inter-annotator agreement on sentence clustering in the literature is given in section 5.3. I decided to create a gold standard for sentence clustering in MDS. Therefore I designed guidelines that help annotators to produce consistent clusterings. A description of the development of the guidelines and the reasons and motivation behind them is given in section 5.4. An important part of the evaluation strategy is also the scores and measures used to determine the similarity between system generated clusterings and gold standard clusterings. An overview and evaluation of several evaluation measures is presented in section 5.5.

5.2 Existing gold standards for sentence clustering

In this section I discuss existing gold standards for sentence clustering, how they differ from my approach and why I chose to create a new gold standard for sentence clustering.

Zha (2002) created a gold standard relying on the section structure of web pages and news articles. In this gold standard the section numbers are used as true cluster labels for sentences. That means that only sentences within the same document and even within the same paragraph can be assigned to the same cluster whereas my approach is to find similar information between documents.

Hatzivassiloglou et al. (1999, 2001) describe the creation of a gold standard for detecting text similarity over short passages. They used 30 articles from the Reuters part of the 1997 TDT pilot corpus and extracted 264 text units (paragraphs) resulting in 10,345 paragraph pairs (10,535 in 2001). Most of the paragraphs contain one sentence. To create a gold standard these paragraph pairs were manually marked for similarity by two reviewers. They were asked to judge if the paragraphs contain common information. They were given the guideline that only paragraphs referring to the same object, performing the same action, or describing something in the same way in both paragraphs are to be considered similar. Here the problem is that only pairs of paragraphs are annotated whereas my work focuses on sentences and the creation of clusters of similar sentences not on pairs of sentences.

For multi-document topic segmentation Sun et al. (2007) used a set of 102 introduction sections of lab reports from a biology course from Pennsylvania State University consisting of 2,264 sentences in total. Each section has two segments: introduction of plant hormones and a description of the content of the lab. For evaluation the sentences were labelled with the

segment number. The difference to my approach is that here the number of topics is predefined. A sentence can belong to either one of the two segments.

A gold standard for Portuguese sentence clustering was build by Seno and Nunes (2008). They used 20 groups of news articles in Portuguese with 1,153 sentences in 71 documents. The first author created the reference clustering. They used the similarity definition from Hatzivassiloglou et al. (1999) (see above). Each sentence of each document was manually classified, but it is not clear if each sentence must belong to a cluster or if there are sentences which do not belong to any cluster. A sentence can only belong to one cluster. If there was more than one possible cluster for a sentence, the sentence was added to the cluster which is most semantically similar.

An approach to event extraction by sentence clustering was described in Naughton et al. (2006, 2008) and Naughton (2007). Naughton et al. (2006) used a collection of 219 news stories describing events related to the war in Iraq. Two volunteers were asked to assign labels to each sentence representing the event(s) in a sentence. A sentence can refer to multiple events and if a sentence does not refer to any event it was labelled with "0". Naughton (2007) used a different corpus – a subset of the Iraq Body Count (IBC) dataset consisting of 342 articles. Ten annotators were asked to identify events in the documents. The events were uniquely identified by integers. The sentences were then assigned to different categories: Category *N* for sentences that describe a new event, *C* for sentences that refer to an event introduced in the preceding sentence, *B* for sentences which refer to events earlier in the document but not in the preceding sentence and *X* for sentences that did not refer to any event. Naughton et al. (2008) used part of the ACE 2005 Multilingual Corpus and part of IBC. In ACE 2005 sentences are – amongst other things – annotated for events like *Die*, *Attack*, *Transport* or *Meet*. Ten annotators were asked to mark all *Die* event instances in the IBC corpus. In these three gold standards the sentences are only labelled for events they describe but not for sentence similarity. The gold standards consist of groups of sentences that contain instances of the same event. The sentences in a group are not necessarily semantically similar since they might describe different aspects of an event.

In conclusion it can be said that there is no gold standard available where the following conditions hold:

- More than two semantically similar sentences are clustered together.
- The sentences in a cluster come from different documents.
- The topics or labels of the clusters are not predefined.
- The sentences are written in English.

The most relevant gold standard for my work is the gold standard for Portuguese sentences clustering (Seno and Nunes, 2008). The problem here is that the gold standard includes only one clustering, but there might not be a single right answer for sentence clustering. Only the first author and not independent judges clustered the sentences.

5.3 Inter-annotator agreement in sentence clustering

In this section I discuss some experiments for calculating inter-annotator agreement in different fields described in the literature.

The gold standard for the experiments in Hatzivassiloglou et al. (1999) consists of 10,345 paragraph pairs that were manually annotated by two reviewers. They marked pairs as either valid paraphrases or invalid. To calculate the inter-annotator agreement in sentence clustering and to validate their definition of similarity, two annotation experiments were performed. Three additional judges were asked to mark a set of 40 randomly chosen paragraph pairs from their gold standard. It is reported that the three judges agreed with the gold standard in 97.6% of the paragraph pairs resulting in $\kappa = 0.5876$. 97% of the sentence pairs from the random sample used in this experiment were marked not similar in the gold standard. In a second experiment a balanced sample consisting of 50 pairs that were marked similar in the gold standard and 50 pairs that were marked not similar were used. Here another two additional judges agreed on the annotations in 91% of the paragraph pairs resulting in $\kappa = 0.82$. However in Hatzivassiloglou et al. (2001) significant disagreement between judges and a large variability in the rate of agreement is reported. For different experiments κ scores between 0.08 and 0.82 are reported. Unfortunately these experiments are not described in detail. The authors mention that the disagreement is significantly lower if the instructions are as specific as their instruction, and that two reviewers who marked the paragraph pairs for similarities could resolve their differences after discussion.

Sun et al. (2007) used introduction sections of lab reports for multi-document topic detection but do not describe how many human judges annotated the data or how much the judges agreed. They only state that it is not hard for humans to identify the boundary between the two segments. Since the annotations were not evaluated it is uncertain if they can be used as a gold standard to compare their system output to it.

To approximate the inter-annotator agreement in the creation of the gold standard for event identification by sentence clustering, in Naughton (2007) two annotators were asked to annotate a disjoint set of 250 documents. The sentences were mapped to one of the four categories described in the previous section. The Fleiss' κscore for this experiment was 0.67 for all categories, 0.69 for *N* (sentence introduces new event), 0.71 for *C* (sentence refers to an event introduced in the previous sentence), 0.52 for *B* (sentence refers to an event introduced before the previous sentence) and 0.72 for *X* (sentence that does not refers to any event). The authors conclude that "*the raters found it difficult to identify sentences that referenced events mentioned earlier in the document*" and that "*the annotations are somewhat inconsistent, but nonetheless are useful for producing tentative conclusions*" (Naughton, 2007, p.4). For the gold standard described in Naughton et al. (2008) the inter-judge agreement was calculated on the basis of another experiment where two annotators annotated 250 documents. The evaluation results in a κ score of 0.67. This inter annotator agreement described the agreement of humans in identifying events in sentences and not in identifying similar sentences.

The results described here lead to two conclusions: (i) humans do not agree entirely on identifying paraphrase pairs or on annotating events in sentences and (ii) that there might be several ideal solutions to event and paraphrase identification. In this aspect the evaluation of paraphrase identification and event extraction is similar to summary evaluation which I will describe very briefly here. There are different ways to evaluate summaries. Spärck Jones and Gallier (1996) define two types of summary evaluation:

Extrinsic summarization evaluation is a task-based evaluation, where the quality of a summary is assessed by determining the effect of the summarization on a given task, e.g., question answering on the basis of the full text vs. question answering on basis of a summary.

Intrinsic summarization evaluation is a system-oriented evaluation, where the quality of the summary is assessed by looking at the quality and the informativeness of the summary.

Often the informativeness of a summary is determined by comparing it to an ideal summary (single summary gold standard) or a set of ideal summaries (compound gold standard). The problem with single summary gold standard is that there is no single best summary, but many good summaries. A system generated summary can be quite different from the ideal summary but can still be a good and acceptable summary. Salton et al. (1997) reported low agreement (only 46% overlap) between human subjects when they were asked to choose the most important 20% of paragraphs from articles. Only 25% overlap in extracts that were selected by four judges was reported by Rath et al. (1961). As van Halteren and Teufel (2003) found, a compound gold standard of 30-40 summaries is needed to counteract the effects of variation in human summaries.

In conclusion, human behaviour in clustering sentences and the rate of agreement between humans have never been evaluated. The results from the literature described here indicate that humans might not agree completely on extracting important information from text documents or on identifying similarity between sentences. However results from summary evaluation have shown that there can be several ideal solutions to a given task and that a compound gold standard can help to counteract the variations in annotations. Therefore I created the first compound gold standard for sentence clustering in MDS. Following the findings of Hatzivassiloglou et al. (2001) that detailed instructions lead to a decrease in disagreement between human judges, I then provide humans annotators with a set of detailed instructions and guidelines.

5.4 Creation of guidelines for human sentence clustering

The results from literature indicate that there is no one ideal sentence clustering solution. Therefore I decided to build a compound gold standard for sentence clustering, where several clusterings created by human annotators are used as ground truth. Humans might not agree com-

pletely when identifying paraphrases or extracting events because different annotators have different views and definitions of similarity. Therefore I tried to reduce disagreement, risk of error and variation in human clustering by giving the annotators detailed instructions and guidelines. These guidelines give, amongst other things, the definition of a cluster, under which conditions sentences should belong to the same cluster and what levels of similarity should be taken into account.

The guidelines for human sentence clustering for MDS evolved gradually and the evolving guidelines were pilot tested by myself and my supervisor. The final set of guidelines and instructions can be found in Appendix A. The starting point for the development of the guidelines was the creation of clusterings for a single DUC document set. Sentences were assigned to clusters, with the task in mind to find groups of sentences that represent the main subtopics in the documents. The annotations were done independently and afterwards the resulting clusterings, the approach and procedure used were compared and discussed. By looking at the differences between the two manual clusterings and reviewing the reasons for the differences, the guidelines were generated and tested on other sentence sets until the differences between the clusterings became noticeably fewer. I now describe the guidelines that evolved from this process and the philosophy behind them in detail.

5.4.1 Characteristics of a cluster

The idea behind sentence clustering is that each cluster represents a subtopic of the document collection to be summarized and that each cluster can be represented by one extracted or generated sentence. Thus the first rule was easily found:

- Clusters should be pure, i.e., each cluster should contain only one topic.

Each cluster must be specific and general enough to be described in one sentence. In addition the annotators were asked to write down a description for each cluster in the form of a sentence. A description or label for a cluster makes it easier to keep an overview of the clusters already created and it ensures that the annotators follow the first rule. It is much easier to assign a sentence to a cluster by comparing it to the label than to each sentence in the cluster. At the end of the clustering process the annotators are asked to review their clusterings. The label helps them to remember the common theme all sentences in that cluster should have.

In MDS it is important to identify redundant information. Redundancy can be a measure of importance. The assumption is that information that is present in many or all of the documents in a DUC cluster is essential for the topic. In the more documents a piece of information is present, the more important it is for the summary. Thus the next rule was established:

- The information in one cluster should come from as many different documents as possible. The more different sources the better.

This also leads to an exclusion criterion, because the reverse conclusion of my definition of importance is that information that is only present in one document is not important and is therefore not included in the summary. This explains the next rule:

- Each cluster **must** include sentences from different documents. A cluster consisting only of sentences from one document is not a valid cluster.

Thus a topic can only be included in the summary if it appears in more than one document. Therefore clusters must contain at least two sentences which come from different documents. Sentences that are not in any cluster that contain at least two sentences are considered irrelevant for the MDS task, which leads to the following rule:

- Each cluster **must** have at least two sentences and should have more than two if possible.

5.4.2 Spectrum of similarity within a cluster

The next set of rules concerns the different types of similarity. Since each cluster will be represented by only one sentence in the summary, the sentences in a cluster should be very similar. Therefore the following guideline was created:

- In an ideal cluster the sentences would be very similar.

Ideally the sentences in a cluster should be paraphrases of each other. Paraphrases are units of text that are semantically equivalent. Barzilay (2003) defines paraphrases as "pairs of units with approximate conceptual equivalence that can be substituted for one another in many contexts." [p. 18]. Consider the following example of a sentence cluster:

- "He also was responsible for helping to form the Big Bang theory of creation."
- "His work gave rise to the Big Bang theory that the universe was created by a tremendous explosion."

The sentences in this cluster are very similar and they can be considered to be paraphrases. This kind of cluster would be perfect for summarization; one of the two sentences could represent the cluster in the summary. However a stringent definition of a sentence cluster that restricts clusters to consist only of paraphrases can lead to different problems:

i) News articles are written by different people with different writing styles, vocabulary and different knowledge. Their sentences might be similar but not always semantically equivalent. They may also include a different amount of information. In restricting the cluster to paraphrase groups only very few clusters will be created.

ii) The clusters found will be very small since all sentences in a cluster must be semantically equivalent to each other.

iii) It is possible that many clusters are created that indeed describe the same subtopic but from different angles. This would result in the over-representation of a topic in the summary and redundancy removal would be impossible.

Therefore sentences that are not paraphrases can be members of the same cluster. In order to guide the human annotators, I defined levels of similarity acceptable. The following ranked list gives a spectrum of similarity, where paraphrases are preferred over other kinds of similarity.

Paraphrases The sentences talk about the same person, same event or the same time. The sentences cover roughly the same amount of information.

Difference in numbers Sentences that are actually paraphrases but differ in numbers.

Partial information overlap A part of a sentence (clause or phrase) is similar to another sentence.

Following these different types of similarity, I defined more rules to guide the human annotators how to handle the different similarities. Often newer articles about an event contain updated information. Consider for example the following two sentences:

- 13.6.1991: *Two people have been reported killed so far.*
- 15.6.1991 *At least four people have died*

When the author of the second sentence wrote his article, which was published two days after the first sentence, new information about the number of casualties was available. Thus the sentences vary in numbers, but they still have the same topic. Sometimes numbers are vague and are not used to give an exact amount but an order of magnitude:

- *Clark Air Base is in Angeles, a city of more than 300,000 people about 50 miles north of Manila.*
- *350,000 residents live in Angeles City, where the air base is located, about 50 miles north of Manila.*

These two sentences communicate the same information that the city next to the air base is a major city with many residents. The exact number of residents is not important. Therefore sentences that only differs from each other in numbers can belong to the same cluster, which is described by the following rule:

- If similar sentences only vary in numbers they can still belong to the same cluster.

There are many sentences in news articles that are not paraphrases but have the same topic. For this kind of sentence another rule was set:

- Not every sentence inside a cluster will be equally similar to all sentences in that cluster. There may be a subset of sentences that is particularly similar to each other. That is okay as long as you think the overall cluster is similar.

Following this rule sentences that are not paraphrases of each other can be joined to form a cluster. The following example shows such a cluster:

- *He also was responsible for helping to form the Big Bang theory of creation.*
- *His work gave rise to the Big Bang theory that the universe was created by a tremendous explosion.*
- *That gave support to the theory that a massive explosion –the Big Bang– created the universe 10 to 20 billion years ago.*

In this example the last sentence is not a paraphrase of the first or second sentence, but it still talks about the same fact. This sample cluster shows that a cluster of sentences less similar to each other than paraphrases can be used for summarization, because they can be summarized by one sentence, e.g., *Hubble's work contributed to the Big Bang theory.*

The last rule concerning similarity of sentences in a cluster, covers partial information overlap. In news articles often more then one piece of information is present in one single sentence. To be able to include these sentences into clusters the following rule was created:

- Generalization is allowed. Sentences in a cluster do not have to be very similar. They still need to be about the same person, fact or event, but they do not have to cover exactly the same information or amount of information.

As long as sentences are about the same person, fact or event, and the resulting cluster can be represented by one sentence the sentences may be members of the same cluster. For example the sentences in the following clusters partially overlap in information and can be represented by the sentence: Charles Schulz died.

- *The coincidence of Charles Schulz's death one day before his final Peanuts appeared in newspapers weighed heavily Sunday on fans of Charlie Brown, Snoopy and Lucy.*
- *Charles Schulz, the creator of Peanuts, the tender and sage comic strip starring Charlie Brown and Snoopy that was read by 355 million people around the world, died in his sleep on Saturday night at his home in Santa Rosa, Calif., just hours before his last cartoon ran in the Sunday newspapers.*
- *He was 77 when he died of cancer at his home in Santa Rosa in February.*
- *He died in his sleep at home Feb. 12.*

5.4.3 Discourse within a cluster

Consecutive sentences always add some new information in comparison to the preceding sentences. Since they still talk about the same event, person or fact, it is very tempting to put them in the same cluster. This following rule is a reminder that clusters of very similar sentences are preferred:

- A sequence of consecutive sentences from one document should not normally be a cluster. There is one exception: if the sentences are very similar they can end up in one cluster but only if they attract at least one sentence from another document.

If it were only allowed to use the information that is present in a sentence, the above cluster about the death of Charles Schulz would not be possible. The first two sentences mention *Charles Schulz* and the latter two only *he*. Humans will infer from the context that *he* refers to *Charles Schulz*. If this kind of inference is not allowed the whole task becomes unnatural, whilst it would make it easier for my system. Thus the following guideline was created:

- Take discourse/context into account. Do not look at that sentence on its own but within context of the whole document. If something important is missing from the previous sentences add it to the sentence.

5.4.4 Standardized procedure for sentence clustering

In general, clustering sentences is not a trivial task, there are several constraints that pull against each other and the human annotators have to find the best compromise. To help the annotators and give them a structure a standardized procedure was proposed:

1. Read all documents. Start clustering from the first sentence in the list. Put every sentence that you think will attract other sentences into an initial cluster. If you feel you will not find any similar sentences to a sentence, put it aside. Continue clustering and build up the clusters while you go through the list of sentences.

2. You can rearrange your clusters at any point.

3. When you are finished with clustering, check that all important information from the documents is covered by your clusters. If you feel that a very important topic is not expressed in your clusters, look for evidence for that information in the text, even in secondary parts of a sentence.

4. Go through your sentences which do not belong to any cluster and check if you can find a suitable cluster.

5. Do a quality check and make sure that you wrote down a sentence for each cluster and that the sentences in a cluster are from more than one document.

6. Rank the clusters by importance.

7. Return a list of clusters in the form:

   rank of cluster – "your sentence": sentence number⟨blank⟩sentence number⟨blank⟩...

These guidelines and description of the clustering task were given to human annotators, who were asked to cluster sets of sentences in order to create a gold standard for sentence clustering.

5.5 Evaluating sentence clusters against a gold standard

There are many different measures available for evaluating a clustering against a gold standard. Each of them has different advantages, disadvantages and constraints. In section 5.5.1, I define and explain requirements for an evaluation measure for sentence clustering. I present the most widely used and promising evaluation measures found in the literature in section 5.5.2. In sections 5.5.3 and 5.5.4 the different measures are tested and evaluated.

5.5.1 Requirements for an ideal evaluation measure

Not all evaluation measures are equally applicable to different clustering tasks. Different measures capture different qualities of a clustering. Here I explain the requirements an evaluation measure for sentence clustering in MDS has to meet. A clustering to be evaluated is called *set of clusters* and a gold standard it is compared to is called *set of classes*. Each requirement will be explained using simple examples which also serve as test cases to test whether the measures meet the requirements[9]. In these examples a class is represented by a set of (coloured) shapes and a cluster by a circle enclosing the members of a cluster.

Homogeneity and completeness

An ideal evaluation measure should reward a set of clusters if the clusters are homogeneous, i.e., if they consist only of sentences from one class (homogeneity). On the other hand it should also reward the set of clusters if all sentences of a class are grouped into one cluster (completeness). In figure 5.1 an sample set of clusters is shown. Class_A consists of three triangles, class_B of four stars and class_C of five circles. All elements of class_A are member of cluster_1, but at the same time cluster_1 also contains one object from another class. In other words this

[9] Some of these test cases were taken from Amigó et al. (2009).

cluster contains a complete class but is not homogeneous. Cluster_2 is homogeneous, i.e., it only contains elements from one class, but not all elements of that class were put into cluster_2. All and only the elements of class_C are grouped into cluster_3, which is with regard to the given classes an ideal cluster. All evaluation measures in consideration will be tested on simple

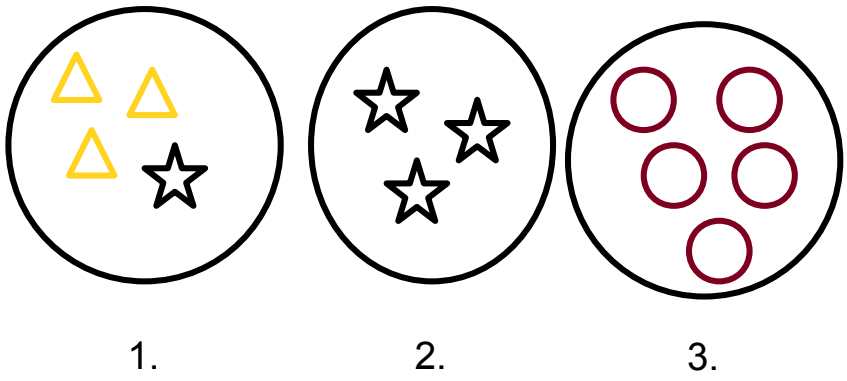


Figure 5.1: *Homogeneity and completeness in a sample set of clusters*

test cases. The test case for homogeneity can be seen in figure 5.2. The first cluster of the left set of clusters (L_1) contains a circle and two squares. In the right set of clusters (L_2) these objects are grouped into two clusters which results in two homogenous clusters. Since L_2 contains now three homogeneous clusters instead of one in L_1, the evaluation score for L_2 should be better than that of L_1. Only if that is the case does the evaluation measure reward homogeneity properly.

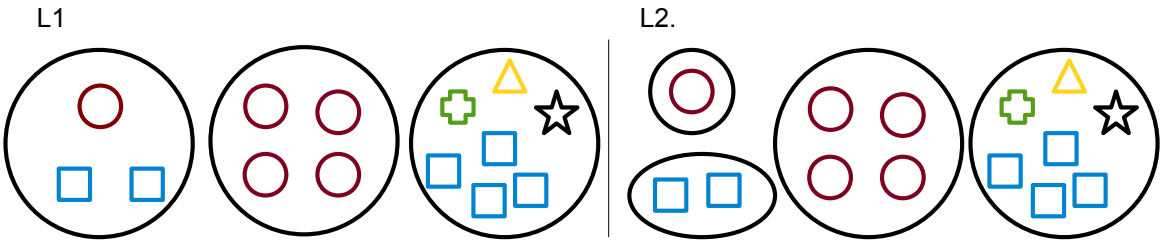


Figure 5.2: *Cluster homogeneity*

Figure 5.3 shows the test case for completeness. Again L_2 should gain better scores than L_1, because the elements of a class (in this case the circles) are distributed over fewer classes in L_2.

Preference of homogeneity over completeness

If sentences that are member of the same class are distributed over several clusters, an evaluation measure should penalize the clustering less than if a complete class was put in one cluster

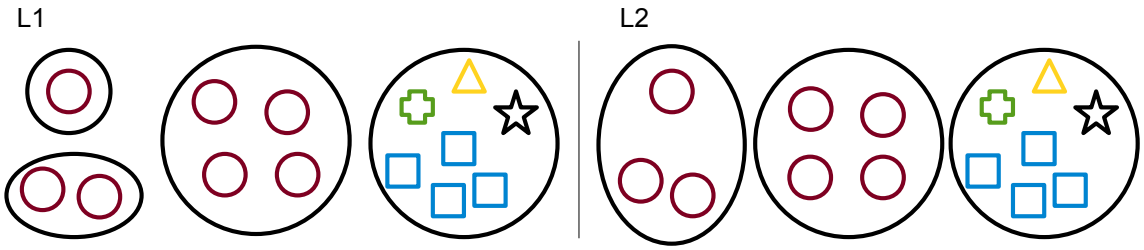


Figure 5.3: *Cluster completeness*

together with a lot of sentences from other classes. For sentence clustering in MDS it is more important that there are only similar sentences (sentences that belong to the same class) in a cluster than to have all of the similar sentences in that cluster. Each cluster will be represented by one sentence in the summary. This sentence is extracted or generated from the sentences present in the cluster. It is easier to choose/generate a representative sentence from a cluster that contains very similar information about the same topic, even if some information is missing, than from a cluster that contains all information about a certain topic but is clouded by irrelevant information. Therefore homogeneity is more important than completeness for sentence clustering in MDS, which also needs to be recognized by an evaluation measure. In the test case for the preference of homogeneity over completeness depicted in figure 5.4 the right set of clusters (L_2) includes two homogeneous clusters but not all circles are members of the same clusters. The other set of clusters (L_1) contains one cluster that includes all circles but in addition this cluster includes objects from other classes. Following the requirement that homogeneity is more important than completeness, the evaluation should result in a better score for L_2 than for L_1.

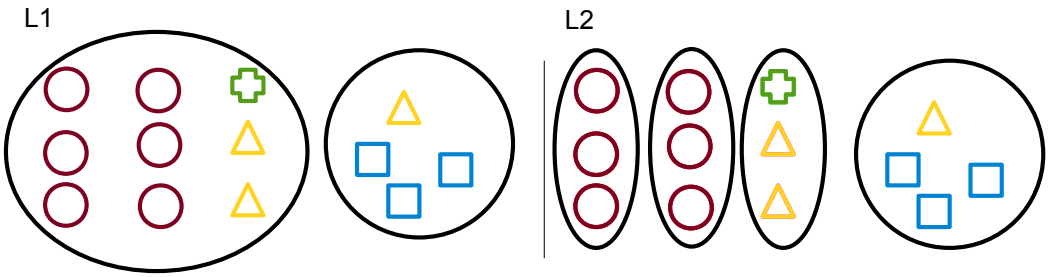


Figure 5.4: *Preference of homogeneity over completeness*

5.5.2 Description of evaluation measures

In this subsection I will present different evaluation measures that are most widely used throughout the literature or look promising to fit the defined requirements. The evaluation measures can be devided into three groups:

1. Entropy based measures
2. Combinatorial measures, i.e., measures based on counting pairs
3. Measures based on mapping clusters to classes

5.5.2.1 Nomenclature

Let S be a set of N sentences s_a so that $S = \{s_a | a = 1, ..., N\}$. A set of clusters $L = \{l_j | j = 1, ..., |L|\}$ is a partition of a data set S into disjoint subsets called clusters, so that $l_j \cap l_n = \emptyset$ and $i \neq n$. $|L|$ is the number of clusters in L. A set of clusters consisting of only one cluster which contains all N sentences of S is called L_{one}. A cluster that contains only one object is called a singleton and a set of clusters that only consists of singletons is called L_{single}.
A set of classes $C = \{c_i | i = 1, ..., |C|\}$ is a partition of a data set S into disjoint subsets called classes, so that $c_i \cap c_n = \emptyset$ and $i \neq n$. $|C|$ is the number of classes in C. C is also called the gold standard of a clustering of data set S because this set contains an ideal solution to the clustering task and other clusterings are compared to it. The number of sentences shared by cluster l_j and class c_i is denoted by n_j^i where n_i is the number of sentences in the class c_i, and n_j is the number of sentences in the cluster l_j.

Entropy based measures

Entropy is a measure of disorder or unpredictability. In clustering a maximum entropy implies that the objects in a set of clusters are maximal disordered in comparison to the objects in a set of classes.

Entropy The entropy of a cluster of objects gives a score of how much information this cluster contains in relation to a set of classes. In other words it measures how the classes are distributed within that cluster. The entropy of one cluster ($H(l_j)$) is calculated by the probability of a sentence $p_{i,j}$ of l_j being a member of a class c_i. Thus given a cluster l_j the entropy of this cluster is given by equation 5.1.

$$\begin{aligned} H(l_j) &= -\sum_{i=1}^{|C|} p_{i,j} \log(p_{i,j}) \\ &= -\frac{1}{\log |C|} \sum_{i=1}^{|C|} \frac{n_j^i}{n_j} \log \frac{n_j^i}{n_j} \end{aligned} \tag{5.1}$$

The entropy of a set of clusters ($H(L, C)$) is given by the sum of the individual cluster entropies weighted by the cluster size in relation to the number of sentences in S (N) as shown in equation 5.2.

$$H(L, C) = \sum_{j=1}^{|L|} \frac{n_j}{N} H(l_j) \tag{5.2}$$

The entropy of a set of clusters ranges from 0 to $\log N$. A minimum entropy of 0 implies that the clustering consists of clusters that only contain sentences from a single class. The maximum entropy of $logN$ is reached if L is maximal disordered in comparison to C, so that the clusters consist of sentences from all classes, e.g., when $L = \{[1, 2, 3]\}$ and $C = \{[1], [2], [3]\}$.
The disadvantage of this measure is that it only measures homogeneity and therefore favours clusterings consisting of many clusters containing few sentences. For example the entropy of L_{single}, e.g., $L = \{[1], [2], [3]\}$ is always 0 regardless of the set of classes it is compared to, since each cluster contains sentences from only one class. However L_{single} is only the ideal clustering in comparison to C_{single}.

$\mathbf{V}_\beta$-measure and $\mathbf{V}_{beta}$ The V-measure (Rosenberg and Hirschberg, 2007) is an external evaluation measure based on conditional entropy:

$$V_\beta(L, C) = \frac{(1 + \beta)hc}{\beta h + c} \tag{5.3}$$

It measures homogeneity (h) and completeness (c) of a clustering solution. By calculating the conditional entropy of the class distribution given the proposed clustering ($H(C|L)$) it can be measured how close the clustering is to complete homogeneity which would result in zero entropy. Because conditional entropy is constrained by the size of the data set and the distribution of the class sizes it is normalized by $H(C)$.

$$\begin{aligned} h =& 1 - \frac{H(C|L)}{H(C)} \\ H(C|L) =& -\sum_{j=1}^{|L|}\sum_{i=1}^{|C|}\frac{n_j^i}{N} log\frac{n_j^i}{n_j} \\ H(C) =& -\sum_{i=1}^{|C|}\frac{n^i}{N} log\frac{n^i}{N} \end{aligned} \tag{5.4}$$

By calculating the conditional entropy of the cluster distribution given the set of classes ($H(L|C)$) it can be measured if all members of a class are grouped into one cluster (completeness). Because conditional entropy is constrained by the size of the data set and the distribution of the cluster sizes it is normalized by $H(L)$.

$$\begin{aligned} c =& 1 - \frac{H(L|C)}{H(L)} \\ H(L) =& -\sum_{j=1}^{|L|}\frac{n^j}{N} log\frac{n^j}{N} \\ H(L|C) =& -\sum_{i=1}^{|C|}\sum_{j=1}^{|L|}\frac{n_j^i}{N} log\frac{n_j^i}{n_i} \end{aligned} \tag{5.5}$$

Like Precision and Recall in IR there is an inverse relationship between completeness and homogeneity: increasing homogeneity often decreases completeness.

The V_β-measure is weighted using β. If $\beta > 1$ completeness is favoured over homogeneity whereas the weight of homogeneity is increased if $\beta < 1$. Since for sentence clustering in MDS homogeneity is favoured over completeness β will be set to 0.5.

Vlachos et al. (2009) proposes V_{beta} where β is set to $\frac{|L|}{|C|}$. This way the shortcoming of the V-measure to favour cluster sets with many more clusters than classes can be avoided. If $|L| > |C|$ the weight of homogeneity is reduced, since clusterings with many clusters can reach high homogeneity quite easily, whereas $|C| > |L|$ decreases the weight of completeness.

V-measure and V_{beta} have a range of $[0, 1]$, where 1 means that the set of clusters is identical to the set of classes. In general, the larger the V_β or V_{beta} score the better the clustering solution.

Normalized mutual information Mutual Information (I) measures the information that C and L share and can be expressed by using entropy and conditional entropy:

$$I = H(C) + H(L) - H(C, L) \tag{5.6}$$

There are different ways to normalize I. Manning et al. (2008) uses

$$NMI = \frac{I}{\frac{H(L)+H(C)}{2}} = \frac{2I}{H(L) + H(C)} \tag{5.7}$$

which represents the average of the two uncertainty coefficients as described in Press et al. (1988). NMI can be generalized to NMI_β:

$$NMI_\beta = \frac{(1 + \beta)I}{\beta H(L) + H(C)} \tag{5.8}$$

Following this generalization it can be shown that $NMI_\beta = V_\beta$ as follows:

from equation 5.4

$$\begin{aligned} h &= 1 - \frac{H(C|L)}{H(C)} && | \times H(C) \\ H(C)h &= H(C) - H(C|L) && |H(C|L) = H(C, L) - H(L) \text{ (Arndt, 2004)} \\ &= H(C) - H(C, L) + H(L) && |\text{see equation 5.6} \\ &= I \end{aligned}$$

from equation 5.5

$$\begin{aligned} c &= 1 - \frac{H(L|C)}{H(L)} && | \times H(L) \\ H(L)c &= H(L) - H(L|C) && |H(L|C) = H(L,C) - H(C) \text{ (Arndt, 2004)} \\ &= H(L) - H(L,C) + H(C) && |H(L,C) = H(C,L) \\ &= I \end{aligned}$$

from equation 5.3

$$\begin{aligned} V_\beta &= \frac{(1+\beta)hc}{\beta h + c} && | \times \frac{H(L)H(C)}{H(L)H(C)} \\ &= \frac{(1+\beta)H(C)hH(L)c}{\beta H(L)H(C)h + H(L)H(C)c} && |H(C)h = I \text{ and } H(L)c = I \\ &= \frac{(1+\beta)I^2}{\beta H(L)I + H(C)I} \\ &= \frac{(1+\beta)I}{\beta H(L) + H(C)} = NMI_\beta && |\text{see equation 5.8} \\ V_1 &= \frac{2I}{H(L) + H(C)} = NMI && \blacksquare \end{aligned}$$

It was shown that $NMI = V_1$ thus NMI has the same properties as the V-measure when $\beta = 1$. V_1 weights homogeneity and completeness equally, whereas in sentence clustering homogeneity is more important.

Variation of information (VI) and normalized variation of information (NVI) The VI-measure (Meila, 2007) also measures completeness and homogeneity using conditional entropy. It measures the distance between two clusterings and thereby the amount of information gained in changing from C to L. This measure is calculated by summing conditional entropies.

$$VI(L,C) = H(C|L) + H(L|C) \tag{5.9}$$

Remember small conditional entropies mean that the clustering is near to complete homogeneity/ completeness, so the smaller VI the better the clustering solution ($VI = 0$ if $L = C$). The maximum score of VI is $\log N$, e.g., when L_{single} is compared to C_{one}.

VI can be normalized to NVI (see equation 5.10), then it can range from 0 when the set of clusters is identical with the set of classes to 1 when L is maximally different from C (Reichart and Rappopor, 2009).

$$NVI(L,C) = \frac{1}{\log \mathrm{N}} VI(L,C) \tag{5.10}$$

Combinatorial measures

These measures compare the two clustering in question by looking at each pair of objects, which can fall into one of four categories:

- TP (true positives) = objects belong to one class and one cluster
- FP (false positives) = objects belong to different classes but to the same cluster
- FN (false negatives) = objects belong to the same class but to different clusters
- TN (true negatives) = objects belong to different classes and to different cluster

Rand index (RI) To calculate the Rand Index (RI) (Rand, 1971) the total number of correctly clustered pairs (TP+TN) is divided by the number of all pairs (TP+FP+TN+FN), thereby the RI gives the percentage of correct decisions.

$$RI = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.11}$$

RI can range from 0 to 1 where 1 corresponds to identical clusterings.

Meila (2007) mentions that in practice RI concentrates in a small interval near 1 (for more detail see subsection 5.5.4). Another shortcoming is that RI gives equal weight to FPs and FNs. In sentence clustering for MDS a sentence that is not assigned to a cluster it should belong to (FN) is not as bad as a sentence that is wrongly assigned to a cluster (FP). An FP sentence can cause a cluster to be less homogeneous, making it more difficult to select/generate a sentence from that cluster.

F-measure The F-measure is a well known metric from IR, which is based on Recall and Precision. The version of the F-score (Hess and Kushmerick, 2003) described here measures the overall Precision and Recall. This way a mapping between a cluster and a class is omitted which may cause problems if $|L|$ is considerably different to $|C|$ or if a cluster could be mapped to more than one class. Precision and Recall here are based on pairs of objects and not on individual objects.

$$\begin{gathered} P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \\ F(L, C) = \frac{2PR}{P + R} \end{gathered} \tag{5.12}$$

The F-measure can range between 0 and 1. The higher the value of the F-measure the better is the clustering solution.

Fleiss' κ $F\kappa$ is a statistical measure of inter-rater agreement (Fleiss, 1971). It takes the agreement occurring by chance into account, therefore it is more robust than a simple percentage of agreement. It can also assess the agreement between any fixed number of annotators.

The Fleiss κ is calculated as shown in equation 5.13. Let Z be the number of pairs of objects, i.e., $Z = \binom{N}{2}$ and z the number of ratings per pair and G the number of categories. In the case of sentence clustering the number of categories is 2 with the categories (i) objects are

assigned to the same cluster and (ii) objects are assigned to different clusters. z_{ij} is the number of annotators who assigned object pair z_i to category j.

$$
\begin{aligned}
F\kappa &= \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \\
\bar{P} &= \frac{1}{Zz(z-1)} \sum_{i=1}^{Z} \sum_{j=1}^{G} z_{ij}^2 - Zz \\
\bar{P}_e &= \frac{1}{Zz} \sum_{j=1}^{G} \sum_{i=1}^{Z} z_{ij}
\end{aligned}
\tag{5.13}
$$

$\bar{P} - \bar{P}_e$ specifies the degree of agreement that was achieved above chance and $(1 - \bar{P}_e)$ describes the degree of agreement that is attainable above chance.

When one set of clusters is compared to one set of classes the number of annotators is 2 and $z = 2$ since both clusterings (L and C) rate each pair of objects. The larger the value of $F\kappa$ the better is the clustering solution L in comparison to C. If $F\kappa = 1$ the clusterings are identical and there is no agreement between the set of clusters and the set of classes if $F\kappa \leq 0$

Measures based on matching

In cluster evaluation the procedure of matching is defined as establishing a one-to-one relation between a cluster and a class. Unfortunately this does not always lead to reliable evaluation results. If a set of clusters is very different from a given set of classes not every cluster can be matched with one class and vice versa. Thereby not every cluster or class is taken into account while evaluating, which leads to an incomplete result. In addition a cluster can be matched with several clusters equally well. The question is how to decide which class to use. In literature this problem is called the problem of matching.

Purity $Purity$ (Zhao and Karypis, 2001) is a widely used evaluation metric that measures the homogeneity of a set of clusters. The $Purity$ measure uses a one-to-one mapping between clusters and a classes. The $Purity$ of a cluster is the fraction of the highest number of common objects between the cluster and any one class and the number of objects in the cluster (see equation 5.14).

$$
Purity(l_j) = \frac{1}{n_j} \max_i (n_j^i) \tag{5.14}
$$

The overall $Purity$ score for a set of clusters is calculated by taking the weighted sum of $Purity(l)$ of the individual cluster as described in equation 5.15.

$$
Purity(L) = \sum_{j=1}^{|L|} \frac{n_j}{N} Purity(l_j) = \frac{1}{N} \sum_{j=1}^{|L|} \max_i \left(n_j^i\right) \tag{5.15}
$$

The values of $Purity$ can range between 0 and 1. The larger the value of $Purity$ the better is the clustering solution.

5.5.3 Evaluation of evaluation measures

I used the examples described in figures 5.2, 5.3 and 5.4 to test whether the evaluation measures described meet the requirements I set earlier (see subsection 5.5.1).

The examples were designed such that the second sets of clusters (L_2) shown in the figures 5.2, 5.3 and 5.4 display the better clustering solution. That means the evaluation measures should give L_2 a better score than L_1 (remember that for *Entropy*, *VI* and *NVI* lower values are better). Table 5.1 shows the results of the tests.

	Homogeneity			Completeness			Preference		
	L_1	L_2		L_1	L_2		L_1	L_2	
$Entropy$	0.44	0.36	✓	0.36	0.36	×	0.55	0.23	✓
$V_{1/NMI}$	0.5	0.58	✓	0.57	0.6	✓	0.51	0.70	✓
$V_{0.5}$	0.48	0.57	✓	0.56	0.58	✓	0.46	0.71	✓
V_{beta}	0.49	0.58	✓	0.57	0.59	✓	0.46	0.70	✓
VI	1.68	1.48	✓	1.52	1.32	✓	1.31	1.13	✓
NVI	0.44	0.39	✓	0.4	0.35	✓	0.35	0.31	✓
RI	0.68	0.7	✓	0.68	0.7	✓	0.68	0.79	✓
F	0.47	0.49	✓	0.47	0.53	✓	0.6	0.56	×
$F\kappa$	0.28	0.32	✓	0.28	0.38	✓	0.38	0.42	✓
$Purity$	0.71	0.79	✓	0.79	0.79	×	0.69	0.85	✓

Table 5.1: *Results of requirement test for evaluation measures*

V_1/NMI, $V_{0.5}$, V_{beta}, VI, NVI, RI and $F\kappa$ fulfil all the requirements. All evaluation measures tested measure homogeneity adequately. $Entropy$ and $Purity$ fail to measure completeness adequately. The F-measure is the only measure that doesn't favour homogeneity over completeness. Since these three measures fail to meet all requirements I will not use them to evaluate sentence clustering for MDS. V_1/NMI meets all requirements set, but gives equal weight to homogeneity and completeness whereas $V_{0.5}$ favours homogeneity over completeness. In the results for the third test case it can be seen that $V_{0.5}$ makes a clearer distinction between L_1 and $L2$ than V_1/NMI. I chose not to use VI. Since VI is measured in bits with an upper bound of $log\ N$, values for different sets are difficult to compare. NVI tackles this problem by normalizing VI by dividing it by $log\ N$. As Meila (2007) pointed out, this is only convenient if the comparison is limited to one data set. Nonetheless I will use NVI instead of VI since normalized values are easier to compare and understand.

5.5.4 Discussion of evaluation measures

In the following discussion and analysis of the behaviour of evaluation measures, the following measures will be considered: $V_{0.5}$, V_{beta}, NVI, RI and $F\kappa$. I used one random set of clusters to analyse the behaviour of the evaluation measures. Variations of that cluster set were created by randomly splitting and merging the clusters. These modified sets were then compared to the original set. This experiment will help to identify what the values reveal about the quality of a set of clusters and how the measures react to changes in the cluster set.

A clustering for the Rushdie sentence set was used for this test[10]. It contains 70 sentences in 15 clusters. This cluster set was modified by splitting and merging the clusters randomly until L_{single} with 70 clusters and L_{one} with one cluster was reached (for details see table 5.2). For

	Number of clusters
L_{single}	70
L_{split3}	61
L_{split2}	48
L_{split1}	30
$C_{original}$	15
L_{merge1}	8
L_{merge2}	4
L_{merge3}	2
L_{one}	1

Table 5.2: *Details of random changes to Judge_A's clustering for the Rushdie sentence set*

the test of how the scores of evaluation measures develop over changes to a clustering $C_{original}$ is compared to the modified clusterings (L_{change}). The results of the test are depicted in figure 5.5.

With each change the resulting clustering becomes more and more dissimilar to $C_{original}$. At the same time homogeneity increases and completeness decreases with increasing number of clusters. When clusters are merged it is the other way round: homogeneity decreases and completeness increases with decreasing number of clusters. Thus the measures should reach their maximum (best) value when $C_{original}$ is compared to itself. The scores of the measure should decrease with each change made to the clustering. Since homogeneity is more important than completeness for sentence clustering, it is expected that the absolute value of the gradient of the first part of the curve (from 1 cluster to 15 clusters) is larger than that of the second part (from 15 clusters to 70 clusters).

[10]For more details on the sentence set see 3.3.

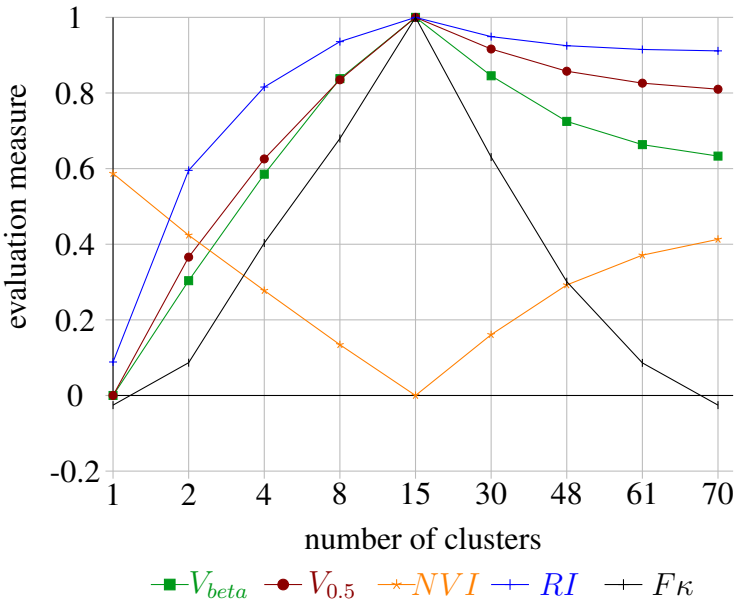


Figure 5.5: *Behaviour of evaluation measures*

All measures apart from the $F\kappa$ show the expected curve. However the curve of the $F\kappa$ is symmetrical, it falls abruptly in both directions. It seems to be very sensitive to changes. Each merging and splitting decreases the value of $F\kappa$ considerably. For L_{single} and L_{one} $F\kappa$ becomes smaller than 0, which implies that these two sets of clusters have nothing in common with $C_{original}$. The other measures however still find some similarity when L_{single} is compared to $C_{original}$.

The RI stays most of the time in an interval between 0.82 and 1. Even for the comparison between $C_{original}$ and L_{single} the RI is 0.91. Only when the number of clusters in L is smaller than 8 do the values of RI fall below 0.9. RI actually uses only a small interval of its range. This behaviour was also described in Meila (2007) who observed that the RI concentrates in a small interval near 1. It implies that a RI score of more than 0.9 does not really tell us anything about the quality of the clustering solution, because even randomly altered clusterings achieve this value.

The graphs for the remaining measures V_{beta}, $V_{0.5}$ and NVI show the expected pattern. All of these measures are more affected by merging than by splitting and use their measuring range appropriately. $V_{0.5}$ favours homogeneity over completeness, but it reacts to changes less than V_{beta}. The V-measure can also be inaccurate if the $|L|$ is considerably different to $|C|$. V_{beta} (Vlachos et al., 2009) tries to overcome this problem and the tendency of the V-measure to favour clusterings with a large number of clusters.

Since V_{beta}, $V_{0.5}$ NMI and NVI fulfil all requirements and passed all tests they will be used to evaluate the sentence clustering produced by BOSSE^{Clu}.

5.5.5 Comparability of clusterings

Following my guidelines, the annotators filtered out all irrelevant sentences that are not related to any other sentence from another document. The number of these irrelevant sentences is different for every sentence set and possibly for every judge or system. The other sentences, hereinafter referred to as content sentences, are the sentences that are part of a cluster. That means that every clustering includes a different number of content sentences. To allow comparison of clusterings, the same number of sentences is required in both clusterings. Here I will discuss and examine three options of equalizing the number of sentences and two options for adding them.

Equalizing the number of sentences in two clusterings

There are three different options for equalizing the number of sentences for two clusterings that are to be compared ($clustering_1$ and $clustering_2$). The different groups of sentences discussed here are visualized in figure 5.6

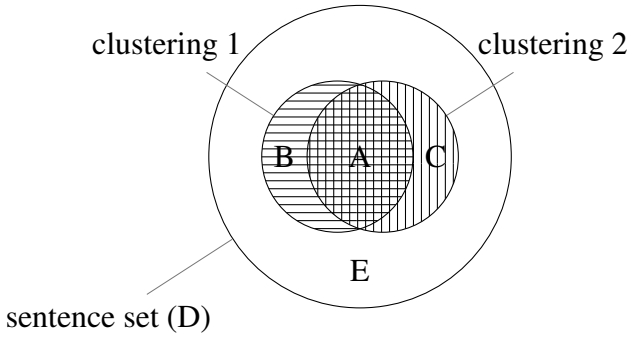
1. **ALL** All sentences from the corresponding sentence set that are not included in a clustering are added to it, so that $L_1 = clustering_1 + C + E$ and $L_2 = clustering_2 + B + E$.

2. **UNION** Only sentences that are included in the first clustering to be compared but missing from the second clustering to be compared are added to the second clustering and vice versa so that $L_1 = clustering_1 + C$ and $L_2 = clustering_2 + B$.

3. **INTERSECTION** Only the sentences that were included in both clusterings to be compared are kept so that $L_1 = clustering_1 - B = A$ and $L_2 = clustering_2 - C = A$.

When these options are applied the clusters in the clusterings remain unchanged. Only when the INTERSECTION option is used sentences might be deleted from clusters. The options determine the number of irrelevant sentences to be added or in case of INTERSECTION to be deleted.

Filtering out irrelevant sentences is an important step in clustering sentences, but assigning the content sentences to clusters is essential. The problem is that these parts cannot be examined independently of each other. If two judges or a set of clusters and a set of classes disagree on one irrelevant sentence then they cannot have the same clusters, because this irrelevant sentence is used as a content sentence in only one of the clusterings. For my research both parts of the clustering process are of great importance but a consensus in the assignment of content sentences is more important. It is important that two judges or the set of clusters in comparison to the set of classes agree on irrelevant sentences, but if they disagree considerably on the assignment of content sentences to clusters, then the clusterings cannot be considered to be similar.



A = sentences used in $clustering_1$ and in $clustering_2$
B = sentences used in $clustering_1$ but not in $clustering_2$
C = sentences used in $clustering_2$ but not in $clustering_1$
E = sentences not used in any of the two clusterings
$clustering_1 = A + B$
$clustering_2 = A + C$
sentence set (D) = $A + B + C + E$

Figure 5.6: *Set diagram of two clusterings for one data set*

Adding sentences to clusterings

After the irrelevant sentences are identified using the different options described above these sentences have to be added or deleted from the clusterings. When irrelevant sentences are added it has to be determined how to add them. In case of INTERSECTION it has to be determined how to treat the singletons that might emerge when all but one sentence are deleted from a cluster. I will examine two different ways to add irrelevant sentences to a clustering, or to deal with emerging singletons:

1. **BUCKET CLUSTER**: All irrelevant sentences are put into one cluster.
2. **SEPARATE CLUSTERS**: Each irrelevant sentence is assigned to a cluster of its own.

Adding each irrelevant sentence as a singleton seems an intuitive way to handle the problem with the irrelevant sentences. However this approach has some disadvantages. The judges will be rewarded disproportionately high for any irrelevant sentence they agree on. Thereby the disagreement on the assignment of the content sentence will be less punished. With every irrelevant sentence the judges agree on the completeness and homogeneity of the whole clustering increases. On the other hand the sentences in a bucket cluster are not all semantically related to each other and the cluster is not homogeneous which is contradictory to my definition of a cluster. Since the irrelevant sentences are combined to only one cluster, the judges will not be rewarded disproportionately high for their agreement. However two bucket clusters from two different sets of clusters will hardly ever be identical and therefore the judges will be punished more for the disagreement on irrelevant sentences.

Experiment

I discuss these options for equalizing the number of sentences in clusterings and illustrate their assets and drawbacks by using three test cases. The sample clusterings $L_{x.x}$ in table 5.3 represent different solutions to the clustering problem for a sentence set D containing 15 sentences $D = \{1, 2, 3, ...15\}$. In each test case two clusterings ($L_{x.1}$ and $L_{x.2}$) are compared to each other. The sentences which were not part of any cluster are listed in $S_{x.1}$ and $S_{x.2}$ respectively.

In the first test case the clusterings are similar. Both clusterings exclude the sentences 12, 13, 14, 15. The clusterings agree on the assignment of the common content sentences and disagree on only 2 irrelevant sentences, namely 10 and 11 of L_1 and 5 and 9 in the second clustering. This test case includes the two most similar clusterings out of the three test cases and should receive the highest similarity scores.

In the second test case the clusterings agree on more irrelevant sentences, which amounts to almost 50% of all sentences. Nonetheless the assignment of the content sentence varies considerably and thus this pair of clusterings should receive lower similarity scores.

In the third test case the clusterings agree on the classification into content sentences and irrelevant sentences. This time the number of irrelevant sentences is smaller: there are only four irrelevant sentences, but the content sentences are clusters completely different. Thus the third test case includes the least similar clusterings and should receive the lowest score.

The fourth test case in table 5.4 shows two clusterings which differ significantly in the numbers of irrelevant sentences: they have only 31% of the irrelevant sentences in common. I use this test case to show the difference between the two attachment options - bucket cluster and separate clusters.

After the irrelevant sentences were added to the clusterings according to the different options, the clusterings were compared to each other using the V_{beta} evaluation measure described in section 5.5.2. The results are shown in tables 5.3 and 5.4.

Results

	$L_{1.1}$={[1,2][3,4,5][6,7,8,9]} $S_{1.1}$={10,11,12,13,14,15}		$L_{2.1}$={[1,2][3,4,5][6,7,8]} $S_{2.1}$={9,10,11,12,13,14,15}		$L_{3.1}$={[1,2,3][4,5,6][7,8,9,10]} $S_{3.1}$={12,13,14,15}	
	$L_{1.2}$={[1,2][3,4,10][6,7,8,11]} $S_{1.2}$={5,9,12,13,14,15}		$L_{2.2}$={[1,3,6][2,4,7][5,8]} $S_{2.2}$={9,10,11,12,13,14,15}		$L_{3.2}$={[1,4,7][2,5,8][3,9,11][6,10]} $S_{3.2}$={12,13,14,15}	
Option	Bucket	Separate	Bucket	Separate	Bucket	separate
ALL	0.52	0.86	0.58	0.76	0.42	0.54
UNION	0.69	0.79	0.42	0.42	0.24	0.24
INTER	1.00	1.00	0.42	0.42	0.24	0.24

Table 5.3: *Comparison of three options to equalize the number of sentences in two clusterings*

ALL This is the easiest option for determining the number of sentences that need to be added to the clusterings. In the end all sentences from the sentence set D are included in both clusterings. The agreement on the irrelevant sentences is taken into account during evaluating. This will influence the similarity score and can lead to imprecise results. This behaviour can be observed in the similarity values for the three test cases. The third test case receives the lowest similarity scores but considering the range of the evaluation measure of $\{0 - 1\}$ they are still too high. The second test set receives higher values than the first case when the bucket cluster of irrelevant sentences is added to the clustering. This is reducible to the large bucket cluster of irrelevant sentences both clusterings have in common. The agreement on the irrelevant sentences in the second test case outweighs the agreement on the content sentences in test case 1. The problem is that the irrelevant sentences and the content sentence are weighted equally. The evaluation measure cannot distinguish between an irrelevant sentence cluster and a content cluster.

UNION When using this option to specify the number of irrelevant sentences, the set of irrelevant sentences to be added is different for every pair of clusterings that is to be compared. The evaluation scores for this option turn out as anticipated. The third test case receives lower values than the second case, which concurrently is awarded with lower scores than the first test set. The values for the bucket and separate options in the second and third test case are identical, because in these examples there are no irrelevant sentences to be added to the clusterings. In the first example the **BUCKET** option receives higher values when applied together with the **UNION** option than with the **ALL** option. This is because when the **UNION** option is used the bucket cluster is smaller, since the common sentences 12, 13, 14 and 15 are removed. Thus the bucket clusters for $clustering_1$ contains only sentences 10 and 11 and the bucket cluster for $clustering_2$ is $[5, 9]$. The sentences from that smaller bucket cluster are found in only two clusters of $L_{1.2}$ whereas with the **ALL** option the bucket cluster for $clustering_1$ is larger ([10, 11, 12, 13, 14, 15] and the sentences are distributed over more clusters in $clustering_2$. Therefore, when the **UNION** option is used, the sentences are distributed over fewer clusters and hence completeness and homogeneity increase and therefore the clusterings receive better similarity values. Here the disagreement on irrelevant sentences is weighted lower than when all missing sentences are added. The agreement on the content sentences becomes more important.

INTERSECTION When this option is used only the sentences that both clusterings share are kept. The first test set receives the best possible score. Here the remaining sentences are clustered identically. However the original clusterings are not identical, so they should not be awarded the best values. With this option to equalize the numbers in the clustering a considerable amount of information is lost. For the next two sets of clusterings the results are the same as for the **UNION** option since no sentences had to be deleted from the clusterings.

In conclusion it can be said that the **UNION** option to equalize the number of sentences in a pair of clusterings represents the best solution to the problem of different numbers of sentences in the clusterings.

	$L_{4.1}$={[1,2] [3,4],[5,6] [7,8],[9,10,11]} $S_{4.1}$={12,13,14,15}	
	$L_{4.2}$={[1,3]} $S_{4.2}$={2,4,5,6,7,8,9,10,11,12,13,14,15}	
Option	Bucket	Separate
ALL	0.15	0.84
UNION	0.34	0.75
INTERSECTION	1.00	0.81

Table 5.4: *Comparison of two options for adding irrelevant sentences*

BUCKET CLUSTER The evaluation scores for the **BUCKET CLUSTER** option are always lower than that for the **SEPARATE** option, when there are irrelevant sentences left. This is because in these cases a clustering to which separate singleton clusters are added includes more clusters than a clusterings to which one bucket cluster is added. Hence the entropy associated with that clustering L $(H(L))$ is larger. On the other hand when the bucket clusters of two clusterings are not identical (as in the first test case) then the completeness and homogeneity of the bucket clusters decrease for every sentence the buckets clusters differ in.

SEPARATE CLUSTERS In the first example this option receives lower V_{beta} scores when applied together with the **UNION** option than with the **ALL** option. This is because when the **UNION** option is used there are fewer separate singleton clusters. Hence the entropy associated with that clustering L $(H(L))$ is smaller and therefore the values of the V-measures decrease. On the other hand with every singleton two clusterings have in common the completeness and homogeneity of the whole clustering increases. This behaviour can be observed in the fourth test case in table 5.4. Here the two clusterings receive considerably better evaluation values when the **SEPARATE** option is applied. When the number of singletons differs considerably between the clusterings, then one of the clusterings (here $L_{4.2}$) will include many more separate singleton clusters. When evaluating the similarity of the two clusterings the clustering with the fewer singletons (here $L_{4.1}$) will receive full completeness score for every separate singleton cluster from $L_{4.2}$. In this case the completeness value is 0.93 out of 1. Consequently the clustering pair receives high V_β-measures although the original clusterings are not very similar. Because of this drawback the **SEPARATE CLUSTER** option is not the right way to attach singletons to a clustering.

In summary the combination of the **UNION** and the **BUCKET** options is the best method to deal with the problem of different number of sentences in clusterings.

5.6 Chapter summary

In this chapter I described the strategy I will use to evaluate my experiments. Part of this strategy is the decision for an external evaluation scheme using a gold standard.

After showing that there is no gold standard available that meets my requirements, I discussed various inter-annotator agreements presented in literature. This overview led to the decision to build a new compound gold standard for sentence clustering with highest possible inter-annotator agreement. Only a high inter-annotator agreement ensures a reliable and effective evaluation of automatic generated sentence clusterings.

In section 5.4 I introduced the guidelines for human sentence clustering I put together. These guidelines assure that the human annotators use the same definition of a cluster and of sentence similarity and follow the same procedure. By providing detailed guidelines the inter-annotator agreement can be maximised.

In the remainder of this chapter I presented and discussed various evaluation measures and showed that only some are suitable for a reliable comparison of clusterings.

Chapter 6

Human generated sentence clusterings

> A man's friendships are one of the best measures of his worth.
>
> CHARLES DARWIN

An experiment was implemented to examine and compare human clustering strategies and to create a gold standard for the automatic clustering experiments. In this experiment, human annotators (section 6.1) were asked to sort sentences into distinct groups, following the guidelines for sentence clustering described in section 5.4. They were given a data set of six sentence sets (section 3.3).

Analysis of the results of these experiments revealed two types of judges with distinct strategies for clustering sentences (section 6.2.1). Other differences in human generated clusterings are listed and discussed in section (Section 6.2.2). The inter-annotator agreement is discussed in Section 6.2.3.

6.1 Human annotators

The data set was annotated by eight annotators – five women and three men, all of whom were unpaid volunteers. They are all second-language speakers of English and hold at least a Master's degree. Five of them have a background in Computational Linguistics, two studied Physics and one judge studied English and History.

The judges were given a task description and a list of guidelines. They worked independently, i.e., they did not confer with each other or me.

Not every judge clustered each sentence set, but each sentence set was manually clustered by at least two judges. Table 6.1 gives a detailed overview of the clusterings each judge created.

Set		Judge								
		A	H	I	B	S	D	O	J	Average
EgyptAir	s	85	75		44			53		64
	c	28	19		15			20		21
	out	106	116		147			138		127
	out %	55	61		77			72		66
	mode	2	2		2			2		2
	max	9	8		8			6		9
Hubble	s				85	93				89
	c				20	25				23
	out				114	106				110
	out %				57	53				55
	mode				2	4				2
	max				8	8				8
Iran	s			69	34					52
	c			19	11					15
	out			116	151					134
	out %			63	82					73
	mode			3	2					2
	max			6	5					6
Rushdie	s	70	74		41	45				58
	c	15	14		10	12				13
	out	33	29		62	58				46
	out %	32	28		60	56				44
	mode	2	4		2	3				2
	max	14	12		12	8				12
Schulz	s	54			38				130	46(74)
	c	16			11				17	14(15)
	out	194			210				128	202(177)
	out %	78			85				52	81(72)
	mode	2			2				7	2
	max	7			7				17	7(10)
Volcano	s	92			57		46			65
	c	30			21		16			22
	out	70			105		116			97
	out %	43			65		72			60
	mode	2			2		2			2
	max	9			5		6			7

Table 6.1: *Details of manual clusterings:* s *= number of sentences in a set,* c *= number of clusters,* out *= number of sentences that are not part of any cluster,* % *= percentage of sentences that are not part of any cluster,* mode *= most frequent number of sentences in a cluster,* max *= maximum number of sentences in one cluster*

6.2 Results

Even before a detailed examination and evaluation of the manually created clusters is carried out, some conclusions can be drawn from the raw details of the clusters shown in table 6.1.

Most of the clusters created are clusters of two sentences: 126 of 320 clusters (124 of 303 clusters without clusterings from Judge_J) have only two members. The percentage of sentences that are not part of any cluster differs greatly between the sentence sets. Only 44% (on av.) of the sentences from the Rushdie sentence set were filtered out, whereas 81% (on av.) where not used for the clusterings of the Schulz sentence set.

Even without a detailed analysis of the collected clusterings, distinct differences between the judges can be identified. There seem to be two groups of judges:

Gatherers Some judges always used more sentences than other judges in their clusterings and thereby filtered out fewer sentences. These judges always used more sentences than average and created often more clusters than average. Judge_A, and Judge_H belong to this group. They used approximately twice as many sentences as Judge_B or Judge_S for their clusterings of the Rushdie sentence set. These judges seem to *gather* all sentences that are similar to each other, even if the similarity is not that obvious.

Hunters Judge_B and Judge_S used fewer sentences for the clusters and thereby filtered out more sentences than the gathering judges. They created fewer clusters than the other annotators. This is clearly visible in the Rushdie sentence set. Judge_B and Judge_S sorted out 60% and 56% respectively, whereas Judge_A filtered out 32% of the sentences and Judge_H 28%. These judges "hunted" for the more obvious similarity relations between sentences.

Judge_D, Judge_I and Judge_O clustered only one sentence set, therefore it is difficult to assign them to one of the two groups of judges, but it seems that Judge_D and Judge_O belong to the hunters and Judge_I to the gatherers.

Judge_J is little bit out of line. He only clustered the Schulz sentence set. He uses 2.4 times more sentences than Judge_A and even 3.4 times more sentences than Judge_B, but only groups them into 17 clusters, which leads to 7.6 ± 2.7 numbers of sentences in a cluster on average, which is much more than any other judge.

While the differences between the judges can be determined by the number of sentences used and number of clusters created, the groups of judges cannot be distinguished by gender or academic background. In both groups men and women and different academic backgrounds can be found.

In the following section I will examine the differences between these two groups in more detail by looking directly at the clusterings the judges produced.

Hunter		Gatherer
119 150 229 246		119 150 229 246
8 57 123 152 155 237 245		8 57 123 152 155 237 245
165 243		165 243
102 126 140		102 126 140
153 179 227 240	⇃	36 38 144 153 227 236
		133 179 240
98 99 168 173		132 173
5 160 161 159 211	⇃	5 160
		159 185 241
12 16 242		4 12 242
95 220		95 147 220
133 177		
11 186		
		1 6 42 156
		24 228 230 238
		43 145 157 164
		56 151
		20 116

Table 6.2: *Comparison of clusterings for the Schulz sentence set created by a gatherer and a hunter*

6.2.1 Hunters and gatherers

In this section I point out and discuss the differences between the two groups of judges, the hunters and the gatherers. For this purpose I compare clusterings from two judges (one gatherer and one hunter) for three sentence sets (Iran, Schulz, Rushdie). For this comparison I aligned the clusters to each other, i.e., I matched each cluster from the hunter with one cluster from the gatherer. It is not always possible to match clusters and this can pose problems for evaluation measures based on matching clusters. Here matching of clusters from different judges was possible, which already indicates that the clusterings are similar. Tables 6.2, 6.3, 6.4 show the selected clusterings. Green numbers indicate that both judges assigned the corresponding sentence to the same cluster. Numbers coloured in orange represent sentences that both judges used in their clustering but assigned them to different clusters. A red number stands for a sentence that was included in a clustering by one judge, but was filtered out by the other.

For the Schulz sentence set the hunter (Judge_B) created 11 clusters of which 9 could be aligned to 11 clusters from the gatherer (Judge_A) (see table 6.2). Two clusters created by the hunter were mapped to two clusters each on the gatherer's side. Besides the sentences in these clusters only one common sentence (coloured in orange) was aligned to different clusters by the two judges. Two clusters created by the hunter could not be mapped to any cluster created by the gatherer and on the gatherer's side 5 clusters consist of sentences the hunter did not include.

Another example is taken from the clusterings for the Iran sentence set (table 6.3). The

Hunter		Gatherer
25 47 66 94		4 10 25 47 66 167
23 48		23 32 48 94 164 179
65 152 160		31 65 152
24 161		24 161
14 15 64 141		14 64 68 141
		15 19 20 63 157
17 58 68 192		17 58 142
69 155		72 155
82 154		82 83 154
81 153		81 153
84 85 158 159		84 85 86 158 159
0 3 1 78 151		1 78 151 168
		6 7 53
		75 76 162
		11 12 169 173 174
		77 143 147 148 150
		9 27 39
		56 74
		18 57 59

Table 6.3: *Comparison of clusterings for the Iran sentence set created by a gatherer and a hunter*

hunters are represented by Judge_B and the gatherers by Judge_I. Most of the sentences used by the hunter are also used by the gatherer, whereas the gatherer generates 6 additional clusters from sentences the hunter filtered out.

For the Rushdie sentence set the clusterings of Judge_S (hunter) and Judge_A (gatherer) are used (see table 6.4). In this example, apart from three sentences, every sentence that was used by the hunter was also used by the gatherer. On the other hand 28 sentences were used by Judge_A (gatherer) which were marked as irrelevant and filtered out by the hunter. These sentences were used by the gatherer to create additional clusters or to add them to common clusters. Excluding the clusters that were split or lumped (see below) the judges assigned only three sentences to different clusters.

Gatherers create larger clusters One distinction between the judges that are considered to be hunters and the judges that are considered to be gatherers is that the gatherers find more sentences for a topic both judges agree on. For example the gatherer includes sentence 147 in cluster 95 220 for the Schulz sentence set. The hunter created the same cluster but excluded sentence 147.

Another example is the cluster consisting of the sentences 84, 85, 86, 158 and 159 from the Iran sentence set (see table 6.3):

Hunter		Gatherer
2 11 21 31 38 66	⇄	2 11 21 30 31 38 48 61 66 79 86 94
22 61 74 79 86 94		
		22 74
63 68 89		63 68 82 83 84
82 88 96		88 96
78 92		78 81 90 92
25 26 33 46 59 64 87 95		19 25 26 27 33 44 46 51 52 54 59 64 95 99
27 75		75 100
3 9 14 57	⇄	3 9 35 55 57 65
35 58 65		
24 43 60	⇄	24 43 60 62 73
48 62 73		
		23 50
32 67		20 32 47 67
		29 42 70 71 72
		30 97
		36 80 101
		16 18 34

Table 6.4: *Comparison of clusterings for the Rushdie sentence set created by a gatherer and a hunter*

84 *At the Vatican, Pope John Paul II sent aid to earthquake victims and a message of condolence to Iranian leaders.*

85 *The Vatican said in a statement that the donation of an unspecified amount was to "provide for the most immediate needs of the population hurt by this huge earthquake."*

86 *John Paul also sent a telegram to the papal nuncio, or diplomatic representative, in Tehran saying the pope "is praying fervently for the wounded and the families of the victims."*

158 *Pope John Paul II was "very saddened" by the earthquake and has donated personal funds to help victims of the disaster, the Vatican said.*

159 *Spokesman Joaquin Navarro did not specify the amount, but in the past the Pope has sent as much as $ 100,000 for disaster relief from his personal funds.*

Only the gatherer includes sentence 86. This sentence clearly is similar to the other four sentences. It can only be guessed why the hunter did not include this sentence. Maybe the hunter just missed that sentence or made a fine- grained distinction between the sentences – the other four sentences all mention that the Pope sent aid.

Gatherers use more sentences to create more specific clusters Another interesting distinction between the gatherers and hunters can be found in the example from the Iran sentence set.

Both judges create a cluster for the topic "Iran is willing to accept help from the U.S.A". Both judges agree on three sentences, i.e., that each of the sentences describe the topic:

14 *Iran said today that it would welcome relief offered by its bitter enemy, the United States, to help victims of the earthquake that has killed as many as 35,000 people, the State Department said in Washington.*

64 *The State Department said Friday that Iran was willing to accept earthquake relief from the American Red Cross and other U.S. humanitarian organizations.*

141 *Iran, at odds with the United States since the 1979 seizure of hostages at the U.S. Embassy in Tehran, has said it would welcome relief from the American Red Cross and other U.S. humanitarian groups.*

Interestingly the hunter includes sentence 15 in this clusters:

15 *The aid would be accepted through the American Red Cross and other U.S. humanitarian organizations, deputy spokesman Richard Boucher said.*

The gatherer on the other hand splits off sentence 15 and, together with other sentences, creates a more specific cluster. The gatherer saw a connection between this sentence and four other sentences which the hunter did not include.

19 *Boucher said Iran told the U.S. government that private donor agencies should contact the Iranian Red Crescent, a humanitarian group that is the conduit for all outside assistance.*

20 *The U.S. government routinely channels humanitarian assistance through the Red Cross and other donor groups.*

63 *Dymally, a member of the House Foreign Affairs Committee, praised President Bush's offer of humanitarian aid to Iran but said Bush must make sure the assistance is channelled through non-government groups.*

157 *The United States on Friday sent $ 300,000 worth of relief supplies through the American Red Cross and is planning more aid, State Department officials said.*

The topic of the first cluster is more general, Iran accepts help from U.S.A. whereas the second cluster is more specific, i.e., the help is channelled through non-government organizations. For the gatherer the similarities between sentence 15 and the four additional sentences seemed great enough to split them off and create a new, more specific cluster. Judge_B however did not include the four additional sentences neither as part of the general cluster nor as a cluster on its own. He did not consider them important for the task.

Another example for this kind of splitting and adding can be found in the Schulz sentence set. The hunter creates a cluster consisting of the sentences 153, 179, 227 and 240. The gatherer splits this cluster into two clusters [179, 240] and [153, 227] and adds to the second cluster four sentences, that the hunter did not include in his clustering.

Hunters create fewer clusters Something else the two groups differ in is that the gatherers consistently create more clusters than the hunters. The gatherers create clusters from sentences the hunters did not consider as important or similar enough to other sentences to be included in their clusterings. An assumption is that these additional clusters cover topics that are not as important for the sentence set as the other clusters. Maybe the connections and similarities between the members of these clusters are not as obvious as in the clusters both judges created. To verify this assumption I will first have a closer look at the clusters the two groups of judges do have in common.

The clusters both groups agree on represent central topics in the document sets. For the Schulz sentence set the common clusters cover the following topics:

- Charles Schulz, the creator of the Peanuts died.
- Charles Schulz was diagnosed with cancer.
- Charles Schulz was born in Minnesota.
- His work reflected the reader's internal world.

For the Iran sentence set, the clusters describe the topics:

- An earthquake of magnitude 7.7 occurred in northern Iran in June 1990.
- Thousands of people died in the earthquake.
- Countries worldwide offer help to Iran.
- Iran accepts help from U.S.A., Israel and South Africa.
- There was a similar devastating earthquake in Armenia in 1988.

The topics in the Rushdie sentence set which both groups of judges found are:

- Salman Rushdie was condemned to death by the Iranian leader Ayatollah Khomeini in 1989.
- Iran distanced itself from the death sentence.
- The death sentence against Salman Rushdie will stay in effect.
- Different groups offer rewards for Rushdies death.
- Rushdie has been living in hiding.
- Salman Rushdie is not allowed to enter India.

- The Rushdie affair impairs the relations between Iran and European countries.

These listings show that the main topics of the sentence sets are covered by both groups of judges.

In the following I have a closer look at some clusters which were only created by gatherers. Here is one example for an additional cluster created by Judge_A for the Rushdie sentence set:

30 *Publication of Salman Rushdie's '"Satanic Verses" in 1988 caused an uproar among Muslims around the world, who contended that it insulted Islam.*

97 *India was the first country in the world to ban "The Satanic Verses" after it provoked angry protests from the Muslim community in 1988.*

The first sentence talks about uproar among Muslims after the publication of "The Satanic Verses" whereas the second sentence is about the ban of the book in India. On first glance these two sentences have different subjects. But both sentences mention uproar among Muslims. However the judge sees a connection between these sentences and values it important enough to create a cluster.

In the next example the sentences are similar but the topics of the sentences are not essential for the understanding of the documents about Charles Schulz:

20 *Jodi Goldfinger of Stone Mountain, Ga., saluted her longtime favorite, Charlie Brown, who "thinks he's a loser but he's not, because he keeps on trying no matter how often the "kite-eating" tree chomps his kite.*

116 *No matter how often his kite crashed, no matter how often his team lost, Charlie Brown never looked back, only up and onward.*

The clusters that were only created by the gatherers cover topics that are not the main topics of the sentence sets. The similarity between the individual sentences is not as clearly visible as in the other clusters.

Spitting vs. lumping A good example for splitting and lamping can be found in the example clusterings for the Rushdie sentence set. In three cases two clusters created by the hunter make up one single cluster in the clustering of the gatherer. The following example shows five sentences (24, 43, 60, 62, 73) which were assigned to one cluster by the gatherer but to two clusters (24, 43, 60 and 62, 73) by the hunter:

24 *Although the Iranian government has said in the past that it would not actively encourage anyone to kill Rushdie, it has never explicitly revoked the death sentence, nor cancelled the bounty.*

43 *While stopping short of revoking a death sentence against Rushdie, Iran says it won't adopt any measures that threaten his life, or anyone connected to his book – The Satanic Verses.*

60 *He did not, however, say that the government was rescinding the edict.*

62 *In Tehran, however, two influential clerics – Grand Ayatollah Mohammad Fazel Lankarani and Grand Ayatollah Nouri Hamedani – were quoted as saying the edict, or fatwa, must be enforced and no one can reverse it.*

73 *Last week, more than half the members of Iran's hard-line parliament signed a letter saying the death sentence stands.*

The hunter distinguishes between the government that did not revoke the death sentence (24, 43, 60) and the hard-liner and clerics that say the death sentence stands (60, 73). The gatherer assigns all five sentences to one cluster.
This phenomenon is known as lumping and splitting behaviour. A lumper is a judge that chooses to emphasize on the similarities rather than on the differences. He generalizes and sees larger units. On the other hand a splitter creates new clusters to classify sentences that differ in key ways. A splitter emphasizes on the differences. He creates smaller but highly differentiated units. I discuss this distinctive feature in detail in section 6.2.2. In this example the gatherer creates a more general cluster – he is a lumper – and the hunter creates two distinct clusters – he is a splitter. In the example from the Schulz sentence set (table 6.2) it is the other way round. Two clusters created by the hunter are split into two clusters each on the gatherers side. Splitting and lumping is a feature that can be found in human sentence clustering but this particular feature can not be associated with only one of the two groups of judges.

In conclusion it can be said that there are two distinct type of judges – hunters and gatherers. A gatherer uses more sentences for his clustering than a hunter. Almost all sentences a hunter chooses to include in his clustering are also used by the gatherer. The additional sentences a gatherer includes in his clustering are used to create:

- Larger clusters for topics both groups of judges have in common
- More specific clusters for topics both groups of judges agree on
- Additional clusters for more insignificant topics

Thus a gatherer produces more clusters than a hunter. Both groups of judges agree on the main topics for a given sentence set and create clusters to represent them. The gatherers seem to have a broader concept of similarity. Since the gatherers created additional clusters it can be assumed that is it harder for gatherers than for hunters to agree among each other. In addition to sentence clusters which represent key topics of a sentence set the gatherers create clusters for less important topics. Humans can reach a consensus about the main topic of a document collection reasonable well but with lower level of importance the agreement seems to diminish. Hence the probability of different clusterings increase with every additional cluster.

Often clusters created by one judge are split in two clusters by another judge. This behaviour cannot be assigned to one of the two groups because in both groups judges can be found that tend to lump or split. There are even judges that do not consistently split or lump (see section 6.2.2 for details).

6.2.2 Semantic differences in human generated clusterings

Each judge clustered the sentence sets differently. No two judges came up with the same separation into clusters or the same amount of irrelevant sentences. I distinguished between two groups of judges, hunters and gatherers, and discussed characteristic behaviour, but there were other types of differences which are not distinctive for one of the groups. While analysing the differences between the judges I found three main categories.

Lumpers and splitters This phenomenon was already introduced in section 6.2.1. One judge creates a cluster that from his point of view is homogeneous:

1. *Since then, the Rushdie issue has turned into a big controversial problem that hinders the relations between Iran and European countries.*
2. *The Rushdie affair has been the main hurdle in Iran's efforts to improve ties with the European Union.*
3. *In a statement issued here, the EU said the Iranian decision opens the way for closer cooperation between Europe and the Tehran government.*
4. *"These assurances should make possible a much more constructive relationship between the United Kingdom, and I believe the European Union, with Iran, and the opening of a new chapter in our relations," Cook said after the meeting.*

Another judge however puts these sentences into two separate cluster (1,2) and (3,4).The first judge (a lumper) chose a more general approach and created one cluster about the relationship between Iran and the EU, whereas the other judge (splitter) distinguishes between the improvement of the relationship and the reason for the problems in the relationship. As discussed in section 6.2.1 this behaviour cannot be directly associated with one of the two groups of judges. Even within clusterings created by one judge splitting and lumping can be found. For example Judge_B showed characteristic lumping behaviour in his clustering for the Schulz data set. He created more general clusters, whereas other judges who created clusterings for this data set created more fine grained clusters. On the other hand in his clustering for EgyptAir or Rushdie Judge_B created more specific clusters whereas other judges created general clusters.

Emphasis Two judges can emphasise on different parts of a sentence. One judge for example assigned the sentence:

All 217 people aboard the Boeing 767-300 died when it plunged into the Atlantic off the Massachusetts coast on Oct. 31, about 30 minutes out of New York's Kennedy Airport on a night flight to Cairo.

to a cluster of sentences about the number of casualties in that plane crash. Another judge emphasized on the course of events and put it into a different cluster.

Inference Humans use different level of interference. One judge clustered the sentence

Schulz, who hated to travel, said he would have been happy living his whole life in Minneapolis.

together with other sentences which said that Schulz is from Minnesota although this sentence does not clearly state this. This judge interfered from *he would have been happy living his whole life in Minneapolis* that he actually is from Minnesota.

6.2.3 Inter-annotator agreement

After the clusterings created by the human annotators were modified in order to equalize the number of sentences as described in section 5.5.5 the evaluation measures discussed in section 5.5 were used to calculate the inter-annotator agreement. Table 6.5 shows the results of the evaluation. The F-measure and the Fleiss-κ measure are only given for reasons of comparability.

The average scores for each evaluation measure over each set was calculated. In the lower part of table 6.5 the average score of the comparisons between all annotators over all sets, the average scores of the comparisons between the hunters (coloured in red) and between the gatherers (coloured in green) are shown. For each sentence set 100 random clusterings were created and compared to the annotator's clusterings totalling in 2600 comparisons. The average of these comparisons is used as a baseline and shown at the bottom of the table. The lowest average similarity values of all sets receives the Schulz sentence set. Within this set the comparisons with the clustering of Judge_J produce considerably lower values than the comparison between Judge_A and Judge_B. As already discussed, Judge_J used considerably more sentences. As a result, I do not include the clustering of Judge_J in the gold standard. The inter-annotator agreement always exceeds the baseline.

Set	Annotators		Evaluation measures			
			V_β	NVI	F	$F\kappa^*$
EgyptAir	A	B	0.61	0.22	0.11	0.31
	A	H	0.70	0.21	0.21	0.40
	A	O	0.63	0.24	0.10	0.29
	B	H	0.64	0.19	0.14	0.21
	B	O	0.68	0.16	0.13	0.28
	H	O	0.65	0.22	0.12	0.23
	Average		0.65	0.21	0.14	0.30**
Hubble	B	S	0.79	0.15	0.32	0.54
Iran	B	I	0.67	0.15	0.15	0.31
Rushdie	A	B	0.62	0.28	0.26	0.41
	A	H	0.75	0.24	0.51	0.52
	A	S	0.67	0.26	0.27	0.37
	B	H	0.64	0.27	0.27	0.41
	B	S	0.75	0.16	0.39	0.49
	H	S	0.65	0.29	0.26	0.38
	Average		0.68	0.25	0.33	0.43**
Schulz	A	B	0.79	0.07	0.29	0.52
	A	J	0.60	0.22	0.15	0.16
	B	J	0.53	0.22	0.15	0.12
	Average		0.64	0.17	0.20	0.18**
	Average(without J)		0.79	0.07	0.29	0.52
Volcano	A	B	0.66	0.27	0.11	0.31
	A	D	0.60	0.27	0.10	0.23
	B	D	0.69	0.18	0.14	0.34
	Average		0.65	0.24	0.12	0.29**
Average	all		0.67	0.21	0.21	0.34
	all (without J)		0.68	0.21	0.22	0.40
	hunter		0.73	0.16	0.25	0.41
	gatherer		0.73	0.23	0.36	0.46
base	all		0.28	0.65	0.06	-0.01
	hunter		0.24	0.64	0.07	-0.01
	gatherer		0.32	0.66	0.06	-0.01

Table 6.5: *Inter-annotator agreement between the human annotators: the hunters are coloured in red and the gatherers in green.*

* For the calculation of $F\kappa$ all singletons were used.

** Here the Fleiss-κ for all judges were calculated.

The overall average V_β is 0.68 and the NVI is 0.21. The agreement within the two groups of judges is considerably higher, V_β=0.73 (0.7275 for the hunters, 0.725 for the gatherers). The hunters receive a lower NVI than average whereas the gatherers receive a higher than average NVI. These results indicate that the agreement between the hunters is slightly better than between the gatherers. The comparison with the baseline and the whole gold standard receives V_β of 0.28 and an NVI of 0.65. When compared to the hunter subset of the gold standard the baseline clusterings receive a value for V_β of 0.24 and when compared to the

gatherers clusterings a V_β of 0.32.

6.3 Conclusion

The clusterings created by the human judges can be used as a gold standard for sentence clustering in multi-document summarization. The agreement between the judges is high and always exceeds the baseline.

This gold standard can be used in different ways. The clusterings of all judges or just the clusterings created by the hunters or the gatherers can be used to compare the output of a sentence clustering system to. The complete gold standard (CGS) represents the whole range of human sentence clustering. The subsets – hunter gold standard (HGS) and the gatherer gold standard (GGS) – represent two kinds of human sentence clustering. One system cannot be of both kinds. The best strategy is to choose the gold standard subset which best fits the purpose of the system. If the goal is to create small and precise clusters of the main topics within a document cluster, the system's output should be compared to the HGS. If the goal is to find more sentences for a topic or to find all redundant but not necessarily important information, the clusterings produced by a system should be evaluated against the GGS.

The particular average inter-annotator agreement (J) can be used as an upper bound for evaluation of the performance of a sentence cluster system. A lower bound can be defined by the baseline (B). The score for the sentence cluster system (S) can then be mapped to these bounds, so that it receives a score of 1 if the result is equal to the average inter-annotator agreement and 0 if the result of the algorithm is equal to the baseline. It would receive negative values if the result is lower than the baseline. The linear function to normalize the system performance is $D = (S - B)/(J - B)$ (Radev et al., 2004). That means the normalized V_β for CGS, HGS and GGS are:

$$N_{CGS}V_\beta = (S - 0.28)/0.38 \tag{6.1}$$

$$N_{HGS}V_\beta = (S - 0.24)/0.49 \tag{6.2}$$

$$N_{GGS}V_\beta = (S - 0.32)/0.41 \tag{6.3}$$

Chapter 7

Experiments for optimizing parameters

> No amount of experimentation can ever prove me right; a single experiment can prove me wrong.
>
> ALBERT EINSTEIN

Several parameters need to be considered when clustering sentences for MDS and especially when using LSA. These parameters, including vocabulary, size of semantic space or the number of dimensions k, have not been evaluated in connection with sentence clustering. In this chapter I describe the experiments to optimize these parameters for LSA and evaluate the results using the gold standard described in chapter 6 and the evaluation measures described in chapter 5.

7.1 Fine-tuning the clustering algorithm

The first experiment focuses on fine-tuning the clustering algorithm. As described in section 4.1, several parameters are required to perform agglomerative hierarchical sentence clustering. These parameters include information about the linkage criteria which defines the distance between clusters and about the distance metric, which defines how the distance is calculated. These two parameters were optimized using the MSRPC as described in section 4.1. I use a combination of the average linkage criterion and the cosine distance metric.

One parameter that was not determined using the MSRPC was the threshold for the cophenetic distance t, by which the cluster tree is cut in order to determine partitional clusters. This parameter is optimized during the first experiment using part of the data set extracted from DUC document clusters as described in section 3.3. From the data set a training set was extracted consisting of 2 of the 6 sentence sets for which human clusterings are available. The two sentence sets which are closest to the average number of sentences in a set were chosen. On average a sentence set has 181 sentences, the *Iran* sentence set consists of 162 and the *EgyptAir* sentence

set of 191 sentences, they were chosen for the training set hereinafter called the *Iran_EgyptAir subset*. The system and clustering algorithm used are described in section 3.

For each of the two sentence sets from the Iran_EgyptAir subset a separate latent semantic space is built including only the terms and sentences present in that sentence set. A co-occurrence matrix TSM with i rows representing the terms and j columns representing the sentences is created. Each cell $c_{i,j}$ includes the weighted frequency of occurrence of term i in sentence j (see section 3.1.1). SVD is used on this matrix and the dimensions of the submatrices are reduced to k dimensions. Using the reduced submatrices $\mathbf{S}_k$ and $\mathbf{D}_k$ clustering spaces $\mathbf{CS}_k$ with $k \in \{10, 25, 50, 75, 100, 125, 150, 175\}$ were obtained. On basis of each of the eight $\mathbf{CS}_k$ for a sentence set four different automatic clusterings are built with $t \in \{0.10, 0.25, 0.50, 0.75\}$. In total 32 $(8 * 4)$ clusterings for each sentences set are created. Each clustering is evaluated and the best clustering L in dependence of k is determined resulting in one clustering for each value of t and each sentence set.

7.1.1 Statistical significance

The Cochran Q-test (Siegel, 2001) was used to verify that different settings for t result in different clusterings. The data was processed to obtain a pairwise representation of the clusterings. This results in three matrices (one for each gold standard), where each row i represents a sentence pair and each column j a value of t. Each cell $c_{i,j}$ includes a binary value, where 1 indicates that the two sentences are members of the same cluster in the clustering created using the designated value of t, 0 if otherwise. For each gold standard subset, different clusterings were chosen for evaluation, thus one matrix was built for each gold standard. On the basis of these matrices the Cochran Q-test was performed. The null hypothesis H_0 reads that the probability that two sentences are members of the same cluster is equal for all t, i.e., $p(t_{0.10}) = p(t_{0.25}) = p(t_{0.50}) = p(t_{0.75})$. The alternative hypothesis H_1 states that the probabilities are different. The results show that this null hypotheses can be rejected in favour of H_1. For all three gold standard subsets the probability that H_0 applies is $p < 0.001$. That is to say the different settings of t have an effect on the creation of the clusterings, i.e., the clusterings created with the different settings are significantly different.

7.1.2 Results

Tables 7.1, 7.2 and 7.3 show the results of the evaluation of the Iran_EgyptAir subset. When the automatic generated clusterings for the Iran_EgyptAir subset are compared to the gatherer subset of the gold standard (GGS, see table 7.1), the values for V_{beta} lie in an interval of 0.54 to 0.57, the highest value is achieved when $t = 0.50$. For the $V_{0.5}$ measure the best results are achieved when $t = 0.75$ whereas $t = 0.50$ is again the best setting when the results are evaluated with the NMI measure. The NVI shows a slightly different picture; here the best results are

GGS				
t	0.10	0.25	**0.50**	0.75
k	10	25	75	175
V_{beta}	0.54	0.56	0.57	0.54
NV_{beta}	0.55	0.59	0.61	0.54
$V_{0.5}$	0.54	0.56	0.58	0.59
NMI	0.54	0.56	0.57	0.56
NVI	0.33	0.32	0.33	0.38
F	0.1	0.1	0.08	0.06
$F\kappa$	0.04	0.09	0.08	0.07

Table 7.1: *Evaluation of the Iran_EgyptAir subset against GGS for different values of* t

achieved when $t = 0.25$. The Friedman test for the GGS shows that, with t as treatment and the evaluation measures (V_{beta}, NV_{beta}, $V_{0.5}$,NMI) and NVI) as blocks, the values are significantly different. That is to say there is a difference in the quality of the clusterings for different values of t ($X^2_{GGS} = 8.28$ with $0.05 > p > 0.02$). Since three of the five evaluation measures under consideration rise to their best values when $t = 0.50$ and drop considerably again when $t = 0.75$, it is concluded that $t = 0.50$ works best for the GGS.

HGS				
t	0.10	**0.25**	0.50	0.75
k	25	50	175	175
V_{beta}	0.62	0.63	0.61	0.51
NV_{beta}	0.78	0.80	0.76	0.56
$V_{0.5}$	0.61	0.63	0.60	0.60
NMI	0.63	0.63	0.63	0.56
NVI	0.21	0.23	0.20	0.38
F	0.11	0.11	0.11	0.07
$F\kappa$	0.17	0.18	0.16	0.17

Table 7.2: *Evaluation of the Iran_EgyptAir subset against HGS for different values of* t

For the HGS the results are clearer and the best setting is easier to specify. For four of the five evaluation measures (V_{beta}, NV_{beta}, $V_{0.5}$ and NMI) the best results are achieved when $t = 0.25$. Only the NVI measure reaches its best value when $t = 0.50$. Once more the Friedman test was used to establish if the clusterings created using different values for t are significantly different. Again the different settings for t are considered to be the treatments and the evaluation measures (V_{beta}, NV_{beta}, $V_{0.5}$,MNI and NVI) are the blocks. With $X^2_{HGS} = 9.72$ with $0.05 > p > 0.02$ it can be said that the results are different. Thus I conclude that $t = 0.25$ is the best setting when the automatically created clusterings are compared to the HGS.

If all human clusterings together are used to form the gold standard (CGS) the values of

CGS				
t	0.10	0.25	0.50	0.75
k	10	50	100	100
V_{beta}	0.56	0.57	0.55	0.48
NV_{beta}	0.72	0.74	0.69	0.51
$V_{0.5}$	0.61	0.57	0.57	0.57
NMI	0.58	0.58	0.57	0.52
NVI	0.33	0.25	0.29	0.43
F	0.11	0.11	0.09	0.07
$F\kappa$	0.15	0.16	0.15	0.17

Table 7.3: *Evaluation of the Iran_EgyptAir subset against CGS for different values of* t

V_{beta} range from 0.48 to 0.57 for $t = 0.25$. The results for NV_{beta} and NVI also indicate that the best setting for t when the clusterings are compared to CGS is $t = 0.25$. On the other hand the highest values of $V_{0.5}$ and NMI are reached when $t = 0.25$. Again the Friedman test showed that the different setting of t results in significantly different results ($X^2_{CGS} = 8.04$ with $0.05 > p > 0.02$).

For all three gold standard subsets the value of V_{beta} rises to a maximum value with increasing t. This maximum is reached at $t = 0.25$ for HGS, $t = 0.50$ for the GGS and $t = 0.25$ for the CGS. After the maximum is reached the values drop to 0.54 for the GGS, to 0.51 for the HGS and to 0.53 for the CGS. A similar behaviour can be observed with the other evaluation measures. In the following I focus on the HGS and GGS. I omit the CGS because it is a combination of the two subsets. In the experiment $t = 0.25$ provides the best result for the CGS, which is the same setting as for the HGS. This might be because the hunter subset is larger than the gatherer subset.

7.1.3 Discussion

When the GGS is used as ground truth for the evaluation, the best results are achieved when $t = 0.50$. Whereas for the comparison with the HGS the clusterings created with the threshold $t = 0.25$ obtain the highest evaluation scores. How is the parameter t linked to the human behaviour of hunting and gathering? What influence does the parameter t have on the flattening of a hierarchical cluster tree and how is it connected to the similarity with the clusterings created by humans? As described in section 6.2.1, the gatherers created more clusters and included more sentences in their clusters. It also seems that the sentences in a cluster created by a gatherer are not as similar as in a cluster created by a hunter. Figure 7.1 shows two dendrogramss for the EgyptAir dataset. In both dendrogramss the cluster trees are identical. As described in section 4.1, the height of the links between the objects (clusters or single sentences) represent the distance between them, known as cophenetic distance. By using the threshold t as a cut-off criterion the tree is cut at a horizontal line where $distance = t$ and all links above this line , i.e.,

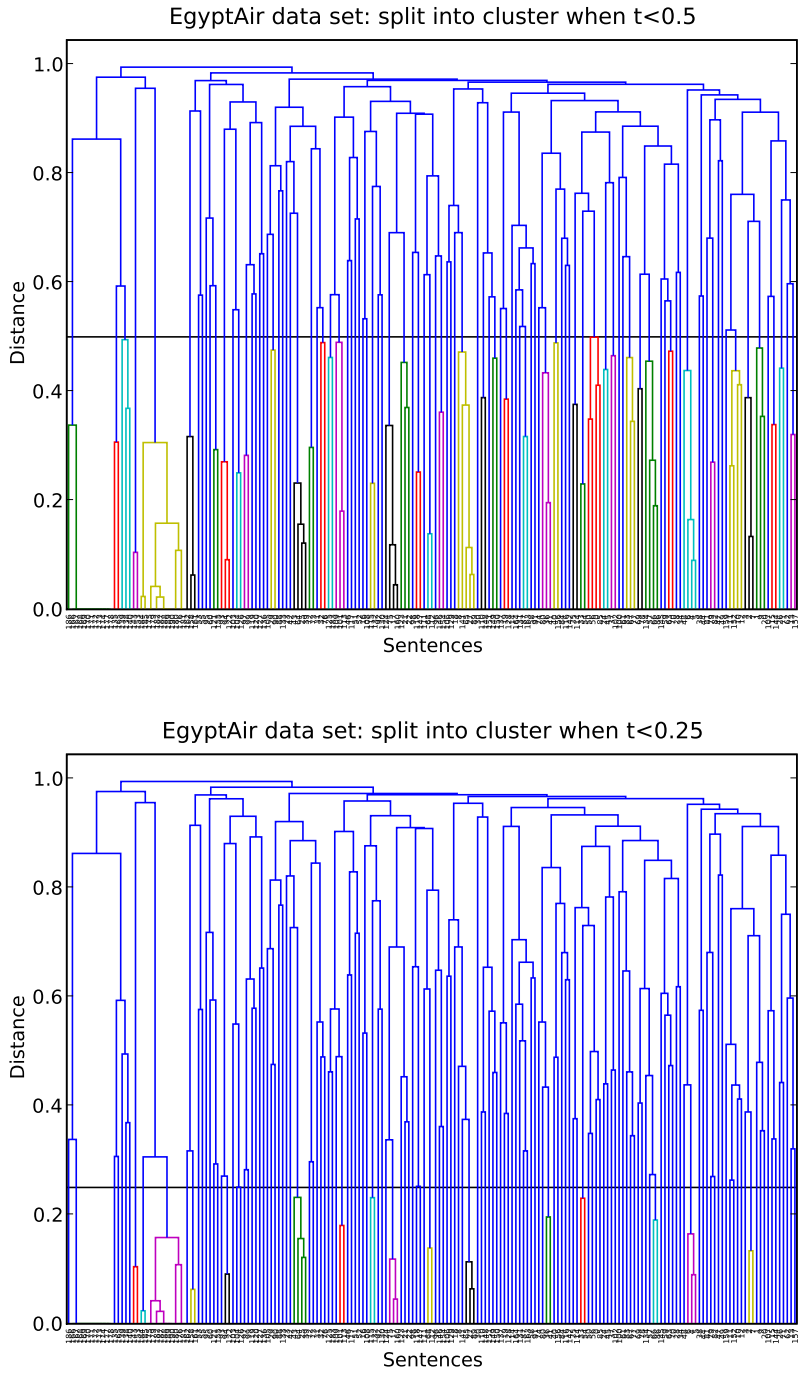


Figure 7.1: *Dendrograms for clusterings of EgyptAir dataset with* $t = 0.5$ *and* $t = 0.25$

with a height $> t$, are ignored. The upper dendrogram shows the cluster tree of the EgyptAir dataset which is cut at $t = 0.5$ resulting in 46 clusters and 53 singletons. The lower dendrogram shows the cluster tree cut at $t = 0.25$. Here the separation results in only 16 clusters and a lot more singletons. Table 7.4 shows the number of clusters[11] and singletons for the EgyptAir

[11]These numbers include all clusters created by the algorithm. Later these clusters are filtered. Only clusters that contain sentences from different documents are allowed.

sentence set for different t. A lower t here results in fewer clusters and more singletons than a higher value for t. This is equivalent to the behaviour of humans considered to be hunters and gatherers. Hunters create fewer clusters and use fewer sentences than gatherers.

t	clusters	singletons
0.10	7	176
0.25	16	151
0.50	46	53
0.75	48	7

Table 7.4: *Number of clusters and singletons in relation to t for the EgyptAir sentence set with k=75*

Analysis of human behaviour in sentence clustering in section 6.2.1 suggested that gatherers tend to build clusters from sentences that are not equally similar to each other. By comparing clusters created by a gatherer with cluster created by a hunter it was apparent that the two groups agreed on the general topic of a cluster, but that the gatherer included additional sentences, whose connection to the topic of the cluster was not immediately visible. The assumption was that this might result in lower intra cluster similarity. The problem was that within the human generated cluster it was not possible to calculate intra- and inter-cluster similarity since the human did not determine or rate the degree of membership of a sentence to a cluster. However it is possible to calculate these internal evaluation measures for the automatic generated clusterings. Table 7.5 shows the intra- and inter-cluster similarity for the two clusterings for the EgyptAir

t	intra	inter
0.1	0.93	0.02
0.25	0.85	0.04
0.5	0.65	0.05
0.75	0.49	0.06

Table 7.5: *Intra- and inter-cluster similarity for clusterings of the EgyptAir sentence set with $t = 0.5$ and $t = 0.25$ and k=75*

sentence set created with different t. The intra-cluster similarity decreases with increasing t whereas the inter-cluster similarity, i.e., the similarity between clusters grows when the value of t increases. These result are consistent with the assumptions made above.

In conclusion it can be said that specific human behaviour with regard to sentence clustering for MDS can be emulated to a certain degree by fine-tuning the cluster algorithm. The cut off threshold t can be used to adjust the clustering algorithm to produce clusterings that exhibit typical features of a hunter or a gatherer.

Having said that, it is striking that the comparison with the HGS almost always receives higher evaluation values than the comparison with the GGS. As can be seen in tables 7.1 and 7.2 only for $t = 0.75$ is the V_{beta} for the HGS smaller than for the GGS. In section 6.2.1 I made the assumption that it might be harder for gatherers to agree on clusters. In addition to sentence clusters which represent key topics of a sentence set the gatherers create clusters for less important topics. Humans can reach a consensus about the main topic of a document collection reasonably well (Barzilay and Elhadad, 1997; Marcu, 1997) but with lower level of importance the agreement seems to diminish. Hence the probability that different clusterings are created increases with every additional cluster. This fact could lead to continuously lower evaluation values. To confirm this hypothesis I calculated the normalized V_{beta} as described in section 6.3. The normalized V_{beta} puts the result into perspective with regard to the upper bound and lower bound of the evaluation scale. In principle the V_{beta} can range between 0 and 1 but the interjudge agreement (J) acts like an upper bound for the performance of the system (Radev et al., 2000). I would assume that with the normalized V_{beta} the score for the two gold standard subset are similar. Unfortunately the results did not confirm this hypothesis. In section 6.2.3 both groups of human annotators receive a similar average inter annotator agreement (hunter: 0.7275, gatherer 0.725) and therefore, even with the NV_{beta}, the comparison with the HGS receives higher values than the CGS. This might be due to the fact that this kind of clustering algorithm favours the creation of clusterings that are more similar to the clusterings of hunters. On the other hand it might be due to the selection of sentence sets. The sentence sets were chosen so that one generic summary can be created. The requirement was that a set describes a single person or event. This selection might already favour hunter-like clusterings. The gatherer subset of the gold standard might be more useful to summarization system which generate topic focused summaries. But these are only the results for the Iran_EgyptAir subset. If this finding holds true for the whole data set remains to be seen.

Nonetheless in consequence it can be said that the clustering algorithm can be tuned to act more like a gatherer or more like a hunter by changing the value of t. Following this experiment the threshold value t will be set to 0.5 to create clusterings that are compared to the GGS and to 0.25 to create clusterings that are compared to the HGS.

7.2 The influence of different index vocabulary

In this section I describe experiments concerning the different options for creating an index vocabulary. As explained in section 4.2 the selection of an index vocabulary is a vital step for any application using vector spaces. Basic criteria for creating an index vocabulary like excluding stop words, stemming and weighting have been evaluated using IR test corpora as described in section 4.2.1.

The objective of this experiment described here is to test whether different index vocabularies listed in section 4.2.2 have an influence on the quality of sentence clusterings.

Here eight different approaches to index term selection were tested. The first vocabulary (called *Standard Vocabulary* or SV in the following) includes all tokens separated by white spaces, which are not on the SMART stop word list. Another vocabulary called NUM1 includes all terms from the *Standard Vocabulary* but all numbers are replaced by the character string `#num`. The NUM2 also focuses on processing numbers. It uses the same tokens as in the *Standard Vocabulary*, but here all numbers smaller than 1492 and larger than 3000 are replaced by their number of digits. For example '29987' becomes '00000' and '137' becomes '000'. The numbers between 1492 and 3000 are considered to be year dates and are kept. The next vocabulary used called COLL consists of the tokens from the *Standard Vocabulary* and in addition of multi-word expressions extracted from the data set using the NLTK collocation module. The vocabularies COLL+NUM1 and COLL+NUM2 use the same terms as the COLL vocabulary but the numbers are processed as in the NUM1 and NUM2 vocabulary respectively. The last group of vocabularies tested are NV, NV+COLL and N. For the NV vocabulary the *Standard Vocabulary* was filtered for nouns and verbs only. Accordingly the NV+COLL vocabulary includes all tokens from NV and the collocations extracted with the NLTK collocation finder. The N vocabulary only consists of nouns from the SV. The index vocabularies created with the different index selection methods described above resulted in different term-by-sentences matrices for the complete data set consisting of all six sentence sets, whose dimensions are shown in table 7.6. In these experiments one TSM for each vocabulary option was built.

	SV	Num_1	Num_2	$Coll$	$Coll + Num_1$	$Coll + Num_2$	N	NV	$NV + Coll$
Terms	1710	1620	1652	1753	1663	1695	696	1072	1115
Sentences	1088	1088	1088	1088	1088	1088	1088	1088	1088

Table 7.6: *Size of different index vocabularies*

The largest TSM was created for the COLL vocabulary, the smallest for the N vocabulary. After these matrices were created and SVD was performed, the clusterings were created on the basis of reduced clustering spaces (see section 3.2). As described in section 7.1, for each sentence set (here only the sets from the training set) and each value of k ($k < rank(TSM)$ and $k \in [10, 25, 50, 75, 100, 125, 150, 175, 200, 225, 300, 400, 500, 750, 1000]$) two clusterings are created , one with the clustering algorithm's parameter t set to 0.50, which is then compared to the GGS, and the other one with $t = 0.25$, which is compared to the HGS. This resulted in 15 for each index vocabulary and each gold standard. For each of the two gold standards used and each vocabulary under consideration the best clustering L in dependence of k was determined.

7.2.1 Statistical significance

The Cochran Q-test (Siegel, 2001) was used to verify that the use of different index vocabularies results in different clusterings. The data was processed to obtain a pairwise representation of

the clusterings. This results in two matrices (one for each gold standard), where each row i represents a sentence pair and each column j a vocabulary. Each cell $c_{i,j}$ includes a binary value, where 1 indicates that the two sentences are members of the same cluster and 0 otherwise. Since for each gold standard different clusterings were chosen for evaluation, one matrix was built for each gold standard. On the basis of these matrices the Cochran Q-test was performed. The null hypothesis H_0 reads that the probability that two sentences are members of the same cluster is the same for all eight index vocabularies. The alternative hypothesis H_1 states that the probabilities are different. The results show that this null hypothesis can be rejected in favour of H_1. For both gold standards the probability that H_0 applies is $p < 0.001$. That is to say the different index vocabularies have an effect on the creation of the clusterings, i.e., the clusterings created using different index vocabularies are indeed different.

7.2.2 Results

	GGS								
	SV	Num_1	Num_2	$Coll$	$Coll + Num_1$	$Coll + Num_2$	N	NV	$NV + Coll$
k	200	250	275	275	275	300	500	400	550
V_{beta}	0.58	0.57	0.59	0.57	0.56	0.58	0.50	0.55	0.54
NV_{beta}	0.64	0.61	0.66	0.61	0.60	0.65	0.43	0.57	0.53
$V_{0.5}$	0.59	0.58	0.59	0.58	0.58	0.58	0.49	0.55	0.52
NMI	0.58	0.57	0.59	0.57	0.57	0.59	0.50	0.55	0.54
NVI	0.32	0.33	0.30	0.33	0.33	0.30	0.35	0.33	0.31
F	0.08	0.09	0.10	0.08	0.08	0.09	0.46	0.21	0.20
$F\kappa$	0.10	0.08	0.09	0.07	0.07	0.08	-0.05	0.03	0.02

Table 7.7: *Evaluation of the Iran_EgyptAir subset against GGS for different index vocabularies*

Table 7.7 shows the results of the evaluation of sentence clustering using different index vocabularies against the GGS. For three of the five evaluation measures under consideration the NUM2 vocabulary obtains the best scores. The highest $V_{0.5}$ score was received by the SV and the best NVI score by the COLL+NUM2 vocabulary.

The scores were used for a significance test with the Friedman signed rank test. The null hypothesis H_0 implies that the clusterings produced on basis of different indexing vocabularies are of the same quality whereas the alternative hypothesis H_1 states that the different index vocabulary contribute to the creation of different quality clusterings. Remember the Cochran Q-test has only proven that the clusterings are different, but not whether one clustering is better than the other. On basis on the Friedman test H_0 could be rejected only with $0.1 > p > 0.05$, which lies above the rejection threshold $\alpha = 0.05$.

With a series of Wilcoxon signed rank tests I analysed which vocabulary performs best in comparison to the other. These tests revealed that the NUM2 performs significantly better (+)

than all other vocabularies. The NV vocabulary performed significantly worse ($-$) than six other vocabularies and the N vocabulary lead to the lowest quality clusterings. From these significance tests results an order of precedence (see table 7.8) can be created for the index vocabulary with regard to the evaluation against the GGS.

Vocabulary	+	-
NUM2	8	0
COLL+NUM2	7	1
SV	6	2
COLL	5	3
NUM1	4	4
COLL+NUM1	3	5
NV	2	6
NV+coll	1	7
N	0	8

Table 7.8: *Order of precedence of index vocabularies for GGS*

	HGS								
	SV	Num_1	Num_2	$Coll$	$Coll + Num_1$	$Coll + Num_2$	N	NV	$NV + coll$
k	100	100	100	125	100	100	250	125	175
V_{beta}	0.66	0.65	0.66	0.67	0.65	0.65	0.61	0.62	0.62
NV_{beta}	0.86	0.83	0.86	0.88	0.84	0.84	0.75	0.78	0.78
$V_{0.5}$	0.66	0.65	0.66	0.67	0.66	0.66	0.60	0.62	0.61
NMI	0.66	0.65	0.66	0.67	0.65	0.65	0.62	0.63	0.63
NVI	0.21	0.21	0.21	0.20	0.22	0.22	0.23	0.23	0.21
F	0.13	0.12	0.13	0.14	0.13	0.13	0.33	0.21	0.13
$F\kappa$	0.21	0.19	0.20	0.21	0.19	0.20	0.05	0.08	0.12

Table 7.9: *Evaluation of the Iran_EgyptAir subset against HGS for different index vocabularies*

Table 7.9 shows the results of the evaluation of the usage of different indexing vocabularies with regard to the HGS. The values for the HGS were tested for significance with the Friedman signed rank test. The null hypothesis H_0 implies that the clusterings produced on basis of different index vocabularies are of the same quality whereas the alternative hypothesis H_1 states that the different index vocabulary contribute to the creation of different quality clusterings. On basis on the Friedman test H_0 could be rejected with $p < 0.01$. So the clusterings created with different index vocabularies are significantly different in quality.

A sequence of Wilcoxon signed rank tests shows that the COLL vocabulary performs significantly better than the other seven vocabularies. The vocabularies can be ranked according to these tests (see table 7.10). Again the N vocabulary performs significantly worse than the other vocabularies.

Vocabulary	+	-
COLL	8	0
NUM2	7	1
SV	6	2
COLL+NUM2&COLL+NUM1	4	3
NUM1	3	5
NV & *NV+coll*	1	6
N	0	8

Table 7.10: *Order of precedence of index vocabularies for HGS*

Both rankings show that the index vocabularies based on nouns and verbs (N, NV and NV+COLL) always perform worse than the other (full) index vocabularies. The vocabulary where all numbers are replaced with `#num` (NUM1) obtains better scores than the noun/verb vocabularies, but worse than the others. For the GGS the vocabulary versions where the collocations are added receive lower scores than the same version without the collocations. Here the vocabularies where the numbers are replaced with the number of digits (NUM2) perform best. For the HGS the COLL vocabulary works best, followed by the NUM2 and the SV vocabulary. Overall the differences in the scores are fairly small. That means that the differences in quality are significant but not very high.

7.2.3 Discussion

Reducing the index terms to nouns or noun and verbs does not seem to have a positive effect on sentence clustering. On the contrary, vocabularies that include only nouns or nouns and verbs perform worse than the other vocabularies. For IR it was claimed that nouns bear the most semantic meaning and are the main characteristics to distinguish documents (Baeza-Yates and Ribeiro-Neto, 1999). Thus some information retrieval applications use only nouns as index terms. However for automatic sentence clustering for MDS it is not sufficient to represent a sentence by nouns and verbs, as the results have shown.

Furthermore it seems to be important to keep the numbers. The vocabularies NUM1 and COLL+NUM1 where all numbers are replaced with `#num` obtain lower scores then the vocabularies where the year dates are kept and the remaining numbers are replaced with their numbers of digits (NUM2). For both gold standard subsets tested the NUM2 vocabulary results in better quality clusters than the standard vocabulary. Thus numbers are important to sentence clustering. However it seems that the order of magnitude of a number is more important than the exact number. In newspaper articles numbers can vary. For example an article that was published shortly after a disaster has different numbers of casualties than an article that was published later when most of the casualties were reported. That was the reason why the human annotators were allowed to cluster sentences that vary in numbers. So the system should have the same possibilities. This can be achieved by representing numbers by their number of digits.

Therefore the *num2* vocabulary works best for gold standards were variation in numbers are allowed.

When compared to the GGS the version of the vocabularies where collocations are included always receives lower scores than the vocabularies without these additional entries. Similarly the vocabularies where the numbers are replaced by the number of digits outperform their complement with the original numbers.

The situation is different when the clusterings are compared to the HGS. Here COLL outperforms its complement COLL+NUM2. The COLL vocabulary results in the term index with the most entries (1753). Maybe the hunters do not concentrate on single words in a sentence to find similarity but on the semantics of the words. Whereas the gatherer sometimes cluster together sentences that just share the same words and therefore a vocabulary with fewer index terms is sufficient (1652).

In conclusion it can be said that the NUM2 works well for both gold standards. However the differences in the qualities of the clustering produced with the full vocabularies are marginal, so that each of them could be used. For the later comparison with the standard VSM I use the NUM2, COLL and SV vocabularies.

7.3 The influence of different sized spaces

As explained in section 4.3 the size of the latent semantic space in which sentence similarities are calculated might have an effect on sentence clustering for MDS. In the first experiment described in section 7.1 each sentence set was represented within its own latent semantic space (called LOCAL LSA). In other words for each set a separate Term-Sentence Matrix (TSM) was created on which SVD was performed and different reduced clustering spaces were created using a range of dimensions where $k < rank(TSM)$ and $k \in [10, 25, 50, 75, 100, 125, 150, 175, 200, 225]$. For each set the clusterings were created within these clustering spaces.

I will examine two more possible setups, the EXTENDED LOCAL LSA and the GLOBAL LSA(described in section 4.3). When the EXTENDED LOCAL LSA is used, one TSM_{ALL} is created for all the sentence sets used. SVD is performed on this single TSM_{ALL} and different rank k approximations are created. Here $k < rank(TSM)$ and $k \in [10, 25, 50, 75, 100, 125, 150, 175, 200, 225, 300, 400, 500, 750, 1000]$. The GLOBAL LSA is similar but in addition to the six sentence sets, more sentences from other DUC documents are added to the TSM before SVD is applied and the clustering spaces are created with $k < rank(TSM_{big})$ and $k \in [10, 25, 50, 75, 100, 125, 150, 175, 200, 225, 300, 400, 500, 750, 1000, 2500, 5000]$.

The idea of testing different space options is based on the assumption that a larger semantic space might contribute to a more robust representation of sentences (Hachey et al., 2006) and a reliable semantic space (Barzilay and Lapata, 2008). Banko and Brill (2001) showed that

increasing corpus size has a positive effect on machine learning for natural language disambiguation. On the other hand (Li et al., 2006) and (Wiener et al., 1995) suggest using LOCAL LSA, expecting a representation that will be more sensitive to small, localized effects (see section 4.3).

Using the *SV* the spaces created with the different space size options resulted in different term document matrices whose dimensions are shown in table 7.11.

Option	Set	Terms	Sentences
LOCAL	EGYPTAIR	326	191
	HUBBLE	342	199
	IRAN	321	185
	RUSHDIE	207	103
	SCHULZ	332	248
	HUBBLE	290	162
EXTENDED LOCAL	all six sentence sets	1710	1088
GLOBAL	all six sentence sets + 5000 additional sentences	7421	6088

Table 7.11: *Size of different latent semantic spaces*

After these matrices were created and SVDs were performed, the clusterings were obtained from the reduced spaces (see section 3.2). As described in the previous section, for each sentences set (here only for the sets from Iran_EgyptAir subset) and each value of k two clusterings are created, one with the clustering algorithm's parameter t set to 0.50, which is then compared to the GGS, and the other one with $t = 0.25$, which is compared to the HGS. For each space option and each gold standard ten or more clusterings (depending on the number of k-values tested for the specific space option) were created. For each of the two gold standard subsets used and each space option under consideration the best clustering L in dependence on k was determined.

7.3.1 Statistical significance

The Cochran Q-test (Siegel, 2001) was used again to verify that the use of different space options result in different clusterings. The data was processed to obtain a pairwise representation of the clusterings. This results in two matrices (one for each gold standard), where each row i represents a sentence pair and each column j a space option. Each cell $c_{i,j}$ includes a binary value, where 1 indicates that the two sentences are members of the same cluster in the clustering created with designated space and 0 if otherwise. Since for each gold standard different clusterings were chosen for evaluation, one matrix was built for each gold standard. On the basis of these matrices the Cochran Q-test was performed. The null hypothesis H_0 reads that the probability that two sentences are members of the same cluster and is the same for all three space options. The alternative hypothesis H_1 states that the probabilities are different.

The results show that the null hypotheses can be rejected in favour of H_1. For both gold standards the probability that H_0 applies is $p < 0.001$. That is to say the different space options have an impact of the creation of the clusterings, i.e., the clusterings created in the different spaces are significantly different.

7.3.2 Results

GGS ($t = 0.50$)				HGS ($t = 0.25$)			
	Local	Extended local	global		Local	Extended local	Global
k	75	200	750	k	50	100	300
V_β	0.57	0.58	0.56	V_β	0.63	0.66	0.63
NV_β	0.61	0.64	0.58	$NV\beta$	0.80	0.86	0.79
$V_{0.5}$	0.58	0.59	0.55	$V_{0.5}$	0.63	0.66	0.61
NMI	0.57	0.58	0.56	NMI	0.63	0.66	0.63
NVI	0.33	0.32	0.29	NVI	0.23	0.21	0.20
F	0.08	0.08	0.11	F	0.11	0.13	0.10
$F\kappa$	0.08	0.10	0.06	$F\kappa$	0.1816	0.21	0.16

Table 7.12: *Evaluation of clusterings of Iran_EgyptAir subset created in different sized latent semantic spaces against GGS and HGS*

Table 7.12 shows the results of the evaluation of sentence clustering using different sized spaces. The Friedman test on these values – where the options are considered to be the treatments and the measures the blocks – shows that the treatments produce different quality clusterings.

The EXTENDED LOCAL LSA receives the highest values from four of the evaluation measures namely V_β, NV_β, $V_{0.5}$ and NMI. The NVI measures gives a different result. Here the GLOBAL LSA performs best. The Wilcoxon signed rank test (Siegel, 2001), considering the values of all five evaluation measures from both gold standards, showed that the EXTENDED LOCAL LSA performs significantly better then the *local LSA* with $p < 0.005$. It also showed that the EXTENDED LOCAL LSA performs significantly better then the GLOBAL LSA with $0.025 > p > 0.01$. The values also suggest that the LOCAL LSA outperforms the GLOBAL LSA. However the Wilcoxon signed rank test cannot verify this assumption (p cannot be determined, it is only known that $p > 0.025$). Even the weaker Sign test (Siegel, 2001) can only confirm this hypothesis with $p = 0.055$, which is only just higher than usual level of significance $\alpha = 0.05$.

7.3.3 Discussion

In this experiment the two gold standards respond similarly. In the previous experiments the gold standards behaved differently with regard to the parameter t. The parameter t has a direct

impact on the separation of a hierarchical clustering tree into distinct clusters. In this experiment different sized spaces were tested. In contrast to the parameter t, the different spaces have an impact on the creation of the cluster tree.

In some experiments a larger more general space led to an improvement in the quality of the results (see Banko and Brill, 2001; Foltz et al., 1998; Barzilay and Lapata, 2008). On the other hand in order to emphasize smaller, more localized effects a smaller (local) space seems to perform better (see Li et al., 2006; Wiener et al., 1995). In the case of sentence clustering for MDS the local spaces perform best. Especially the EXTENDED LOCAL space outperforms the other two spaces tested. This EXTENDED LOCAL space is a compromise between the two extremes of local space and global space. The extended local space combines the strategy of the global space with that of the local space by using only the sentence set that is to be clustered, but building one space from them instead of six separated spaces. Thereby the semantic representation becomes more robust as some background information, which in contrast to the global space was not selected at random, was added.

However this may not always be the case. The corpus used in this experiment consists entirely of newspaper articles, which is a special text genre. All articles are factual, non-fiction texts, which are targeted at a general mass audience. In addition the topics of the articles are limited to three categories (i) single natural disaster, (ii) single event and (iii) biography. As a result the sets are quite similar with regard to their structure and topic category. Thus the articles are not in contradiction to each other, which means that for example there are only a very few words that are used ambiguously, e.g., the term *plane* always refers to an aeroplane in these articles and not to a plain or extension. Furthermore two sentence sets talk about the Iran: (i) the *Rushdie* set is about the death sentence proclaimed by Iran on Rushdie (ii) the *Iran* set talks about an earthquake in Iran. The sets *Iran* and *Volcano* both talk about natural disaster and the sets *EgyptAir* and *Hubble* both have vocabulary about flying and planes in common. That implies that the word usage patterns are similar. Thus the fact that the EXTENDED LOCAL LSA space performs better than the LOCAL LSA space might be due to these similarities. Nonetheless the results show that the local spaces perform better than the global LSA spaces.

The global space represents an extension of the EXTENDED LOCAL space with random selected background information. This added background information seems to bias the word usage patterns in the sentences. This is undesirable in sentence clustering. Here small differences or similarities are important to be able to cluster sentences of similar meaning in one group. Therefore a global LSA space is not applicable to sentence clustering for MDS.

For other application it might be advisable to create bigger semantic spaces or create a huge background corpus. However for sentence clustering using LSA a localized corpus works best. In sentence clustering small differences in word usage and term relations are important in order to group sentences into different topic groups therefore a homogenous space works better. This localized space can consist only of the sentences to be clustered or in additional of sentences from documents with similar topics.

7.4 The optimal number of dimensions

As described in section 4.4 the number of dimensions k of the clustering space is essential to the performance of LSA. If too many dimensions are kept, the latent semantic structure cannot be revealed since the documents and words are not projected near enough to each other and too much noise is left. If k is too small then too many words and/or sentences will be superimposed on each other, destroying the latent semantic structure. In this section I describe several experiments that focus on the optimal number of dimensions in connection with other parameters for sentence clustering.

7.4.1 LSA pattern

The first analysis was carried out to find out whether the quality of the resulting clusterings changes for different numbers of dimensions k. When quality is plotted against k for LSA in IR the graph shows low performance for very few dimensions, then the quality improves considerably until it peaks and falls off slowly.

For this experiment I compared the output of the clustering algorithm for the Iran_EgyptAir subset with the two subsets of the gold standard GGS and HGS using the V_{beta} evaluation measure. The standard vocabulary was used as indexing vocabulary and the EXTENDED LOCAL LSA space containing all six sentence sets was built. In this experiment different k dimensional clustering spaces were built where $k \in [10, 25, 50, 75, 100, 125, 150, 175, 200, 225, 300, 400, 500, 750, 1000]$.

Figure 7.2 shows the development of the quality of the clustering solutions measured in V_{beta} for different numbers of dimensions k of the clustering space.

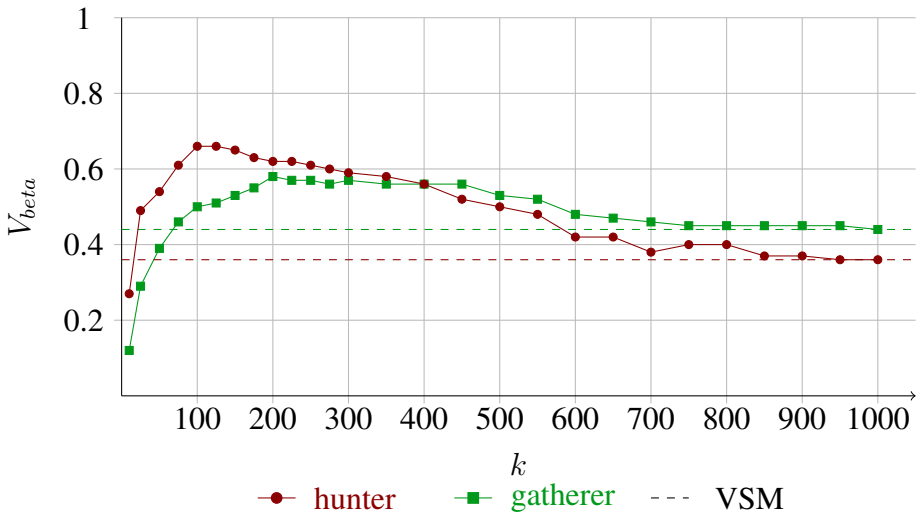


Figure 7.2: *LSA pattern in sentence clustering*

The graph shows the same characteristics as a plot for LSA in IR. If only very few dimensions are used the quality of the clusterings is very low and lies below the quality of sentence clustering using the standard VSM. The performance peaks at 100 dimensions when the automatic generated clusterings are compared to the hunters and at 200 when compared to the gatherers. After this maximum is reached both curves drop off slowly and approach the level of quality attained by the VSM.

For this training data set and this application the quality of the clusterings created using LSA is higher than that using the VSM for most of the number of dimensions tested.

7.4.2 The optimal number of dimensions: dependence on *t*

The second experiment regarding the optimal number of dimensions is intended to find out whether the optimal number of dimensions depends on the cophenetic distance used to cut the clustering tree in partitional clusters.

For this analysis I used the results from the fine-tuning experiment. The results described in section 4.1 were obtained comparing the automatic created clusterings for the Iran_EgyptAir subset with the two subsets of the gold standard GGS and HGS using the evaluation measures described in section 5.5. The standard vocabulary was used as indexing vocabulary and local spaces were built. In this experiment different k dimensional clustering spaces were built where $k \in [10, 25, 50, 75, 100, 125, 150, 175]$. For each t the number of dimensions k was chosen which lead to the best quality clustering in comparison to the HGS and GGS.

Figure 7.3 displays the results for this experiment. The optimal number of dimensions k_{best} is plotted against the distance parameter t.

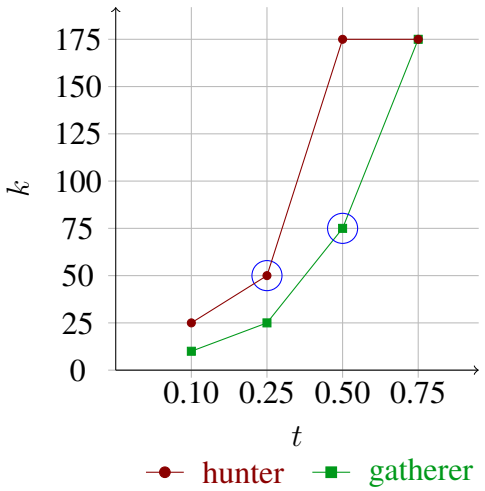


Figure 7.3: *Optimal number of dimensions in relation to the threshold* t

It is striking that the optimal number of dimensions k_{best} grows with increasing t. The results are similar for the EXTENDED LOCAL LSA space.

The comparison of the optimal number of dimensions k_{best} for the HGS with k_{best} for GGS shows that for the same t less dimensions are needed for the GGS. At the same time the optimal t for the GGS is higher than for the HGS. This means that in the end more dimensions are needed for the GGS as for the HGS.

In conclusion it can be said that the number of dimensions k depends on threshold t of the cluster algorithm. The higher the threshold the more dimensions are needed to obtain an optimal result.

7.4.3 The optimal number of dimensions: dependence on the LSA space

In this section I show that the optimal number of dimensions varies with the number of sentences in a clustering space. In section 7.3 I described that the quality of the automatically generated clusterings is contingent on the size of the clustering space. Here I analyse how the optimal number of dimensions k_{best} varies over the different spaces.

I used the results from the experiment described in section 7.3. In this experiment the automatically generated clusterings for the Iran_EgyptAir subset were compared with the two subsets of the gold standard GGS and HGS using the evaluation measures described in section 5.5. The standard vocabulary was used as indexing vocabulary. Clusterings were determined for different values of k with $k < rank(TSM)$ and $k = 10, 25, 50, 75, 100, 125,150, 175, 200, 225, 300, 400, 500, 750, 1000$. For each space size and each gold standard subset the number of dimensions producing the best quality clusterings were selected.

In figure 7.4 the optimal number of dimensions k_{best} is plotted against the number of sentences.

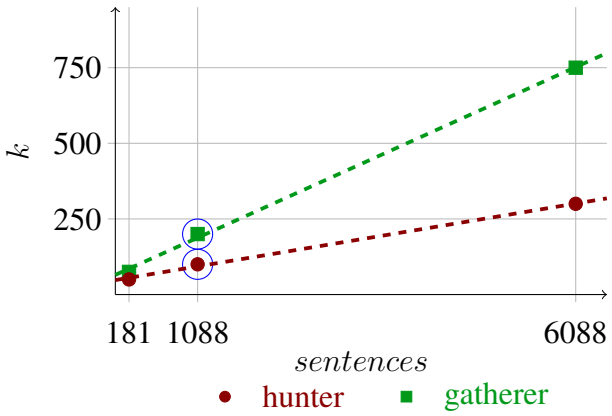


Figure 7.4: *Optimal number of dimensions in relation to number of sentences*

The marks represent the optimal number of dimensions for a given number of sentences. The dashed lines represent linear regressions of the optimal number of dimensions over number

of sentences. The function for the linear regressions are:

$$f_{HGS}(x) = 0.04x + 48.13 \text{ with } R^2 = 0.9978 \tag{7.1}$$

$$f_{GGS}(x) = 0.11x + 64.97 \text{ with } R^2 = 0.9990 \tag{7.2}$$

The R^2 coefficient of determination is a statistical measure that measure how well the regression line fits the data points. R^2 can range from 0 to 1 where 1 represents a perfect fit of the model. In this case the R^2 values show that the regression lines almost perfectly fit the data.
This means that for both gold standard subsets the optimal number of dimensions is almost linearly proportional to the number of sentences. The only difference between the two gold standard subsets is that the slope and intercept for the HGS (red circles) are smaller than for the GGS (green squares). Thus the optimal number of dimensions for comparison with the HGS is always lower than for the GGS.

For the LOCAL space 50 respectively 75 dimensions produce the best results. In the LOCAL space each sentence set is represented in its own space with 181 sentences on average. These separate sentence sets are considerable smaller than the corpora normally used in IR. The results for IR described in section 4.4 are based on corpora containing thousands to millions of documents. These small local spaces are homogeneous. They consist of sentences from newspaper articles about one certain topic. Therefore a relatively small number of dimensions is sufficient to produce top quality results. The best results for the EXTENDED LOCAL LSA space (1088 sentences) are produced when a clustering space with 100 dimensions is used when the clusterings created are compared to the HGS and with 200 dimensions for the GGS. However when a GLOBAL space with thousands of documents (here 6088 documents) is used, the number of dimensions increase considerably. Here a space with 300 dimensions works best for the comparison with HGS and for that with GGS a space with 750 dimensions.

This linear model of relation between k and number of sentences is only applicable if the increase in subtopics is linearly proportional to the increase in sentences. This holds true in this case, which is due to the structure of the collections. The additional sentences are not part of the documents the original sentences sets were created from. So with every additional sentence the number of subtopics and word usage patterns increase and hence the optimal number of dimensions grows. If the sentences added had the same topics and used the same or similar word usage patterns the increase in the required dimensions is expected to be much lower.

These results are consistent with the observation that the broader the topics or conceptual content of a test collection the larger the optimal number of dimensions (Dumais, 1991). Bradford (2008) suggest for term comparison that the increase in dimensions is not more than logarithmic. However he tested collections which range from several thousand to several millions documents. He refers to an "island of stability" in the range of 300 to 500 dimensions for corpora with 1, 2 and 5 million documents. Other papers suggest using fewer dimensions. With regard to corpora containing ≈ 1000 documents the use of 100 dimensions was suggested (Dumais, 1991). Later when test collection containing thousands or tens of thousands of documents

were used, Landauer and Dumais (2008) proposed to use 350 +/- 50 dimensions, which seemed to work best for most corpora in IR.

It is noticeable that for the comparison with the hunters, clusterings created from a space with fewer dimensions produce best results, whereas for the comparison with the GGS more dimensions are needed. This observation is consistent with the presumption that less dimensions work better for broader comparisons and more dimensions for more specific comparisons (Wikipedia, 2011a). As described in section 6.2, hunters create coarser clusters, only using the sentences that at first glance fit into a specific cluster. Gatherers however look at a sentence in more detail. They find more fine-grained clusters. Therefore they use more sentences and create more clusters. For this more specific comparison more dimensions work better.

Thus for the standard vocabulary and the EXTENDED LOCAL LSA space 100 dimensions for the HGS and 200 dimensions for the GGS are the optimal number of dimensions for the clustering space.

7.4.4 The optimal number of dimensions: dependence on vocabulary

In this section I examine whether the optimal number of dimensions k_{best} depends on the size and/or on the content of the indexing vocabulary.

For this analysis the results from section 7.2 were used. In that experiment eight different approaches to index term selection were tested. The SV includes all tokens separated by white spaces, which are not listed on the SMART stop word list and occur in more than one sentence. NUM1 includes all terms from the SV but all numbers are replaced by the character string `#num`. NUM2 uses the same tokens as in the SV but here all number smaller than 1492 and larger than 3000 are replaced by their number of digits. The vocabulary COLL consists of the tokens from the SV and in addition of multi-word expression extracted from the data set. The vocabularies *Coll+Num1* and *Coll+Num2* use the same terms as the COLL vocabulary but the numbers are processed as in the NUM1 and NUM2 vocabulary respectively. For the *NV* vocabulary the SV was filtered for nouns and verbs only. Accordingly the *NV+Coll* vocabulary includes all tokens from *NV* and the collocations extracted. The vocabulary *N* contains only nouns. With each indexing vocabulary different EXTENDED LOCAL LSA spaces were built with $k = 10, 25, 50, 75, 100, 125, 150, 175, 200, 225, 300, 400, 500, 750, 1000$. The clusterings automatically created from these spaces were compared against the two subsets of the gold standard, HGS and GGS. For each vocabulary the k dimensional space that produces the best clusterings was selected. Figure 7.5 shows the optimal number of dimensions in relation to the indexing vocabularies.

In the previous section it was shown that the optimal number of dimensions k_{best} for the *SV* is 100 for the HGS and 200 dimensions for the GGS. The correlation between the optimal number of dimensions and the number of sentences was linear. This scatter plot of k_{best} over

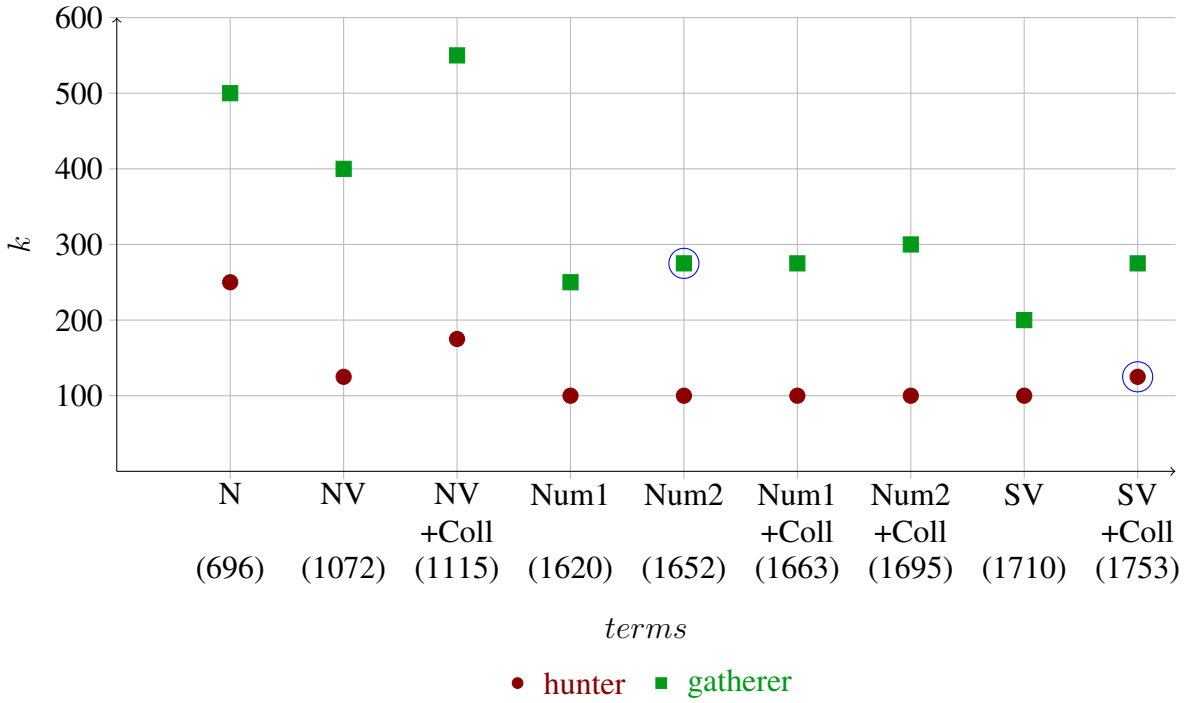


Figure 7.5: *Optimal number of dimensions in relation to index vocabularies*

number of terms and vocabularies, however, does not show a linear correlation. Furthermore the optimal number of dimensions seems to fall while the number of terms increases.

This plot shows that the development of k_{best} over different vocabularies is similar for both GS subsets. To obtain top results with vocabularies restricted to nouns or nouns and verbs (*N*, *NV* and *NV+Coll*) more dimensions are needed in the clustering space than when full vocabularies are used. Vocabularies including collocations always need more dimensions than their complement without collocations. Unfortunately with the available results this behaviour cannot always be confirmed for the HGS. However this might be because the interval in which k is tested is too large. But it is not feasible to test all possible values for k. Anyway this would not be possible in a real world application.

Earlier it was shown that the number of dimensions increases with the number of sentences in a corpus. Normally with the number of sentences the number of terms increases as well (until a certain degree). For example the standard indexing vocabularies for the local spaces have 303 terms on average, the *SV* for the larger local space includes 1710 terms and the global space 7421 terms. In this analysis of a possible correlation between dimensions and indexing vocabularies things are different. Here the number of dimensions with which the best results are achieved fall with increasing number of terms. This leads to the conclusion that the optimal number of dimensions is not only dependent on the size of the TSM but also on the nature of the indexing vocabularies. Furthermore, the fewer word classes are used the more dimensions are needed: for the *NV* vocabulary 400 (resp. 125) dimensions works best and for the vocabulary including only nouns 500 (resp. 250) dimensions are needed. It seems the less (semantic)

information is gathered from the sentences, the more dimensions are needed.

7.4.5 Discussion

The optimal number of dimensions depends not only on the task and the corpus, as Quesada (2007) argues, but is also influenced by other factors.

The fine-tuning of the clustering algorithm has an influence on k_{best}. The larger the threshold values t the more dimensions are needed to achieve top results. Another factor that affects the optimal number of dimensions is the clustering space. The more sentences adding new information to the corpus are included in a space, the larger is the optimal number of dimensions. On the other hand the optimal dimensionality also depends on the content of the indexing vocabulary. If only certain lexical categories like noun or verb are present in a vocabulary, more dimensions are required in order to represent the sentences adequately. Last but not least the gold standard used plays a part in choosing the right number of dimensions for the clustering space. When the hunter subset of the gold standard for sentence clustering is used a lower rank approximation of the original TSM is sufficient (100-275 in the larger local space). This contrasts to the gatherer subset of the gold standard (GGS) where more dimensions are needed to produce top results (200-550 dimensions for the larger local space, depending on the vocabulary). This is because the GGS requires a higher value for the threshold value t and as explained before a larger t requires more dimensions. It can be concluded from the results that is is not reasonable simply to take the values from IR. The standard value used for IR tends to be 300 dimensions, which would only be optimal for some space-vocabulary-GS settings.

The conclusions drawn from the results disprove the theories that is is possible to deduce the optimal numbers of dimensions from the singular values or from the TSM. For a given space size the co-occurrence matrices (TSM) and singular values are the same for both gold standard subsets. Still the results show that different number of dimensions are needed for different vocabularies, which are based on the same number of sentences but which differ only slightly in the number of terms. Another argument against the idea of calculating the optimal number of dimensions from the TSM is the difference in k_{best} for the two gold standard subsets.

For the remaining experiments I use the values for k in dependence on the gold standard and vocabularies shown in table 7.13.

	HGS ($t = 0.25$)			GGS ($t = 0.5$)		
vocab	SV	NUM2	COLL	SV	NUM2	COLL
k	100	100	125	200	275	275

Table 7.13: *Optimal number of dimensions for the* LARGER LOCAL LSA *clustering space*

Chapter 8

Final Experiments – comparison of LSA and VSM

42

The Hitchhiker's Guide to the Galaxy
DOUGLAS ADAMS

The experiments described in chapter 7 have shown which parameters and settings influence the performance of LSA for sentence clustering and which setups produce the best quality sentence clusterings to be used in MDS. In the next experiment I analyse how well LSA performs in comparison with the simple word matching approach of VSM.

In the VSM (section 3.1) the measurement of similarity of two sentences is based on world overlap since each dimension of a document vector corresponds to a term in the corpus. The value of that vector cell represents the weight of the term the cell corresponds to in that sentence. The number of dimensions of a document vector is determined by the number of terms in the indexing vocabulary. In contrast to the VSM the dimensions or elements of a sentence vector in the reduced LSA space corresponds to a word usage pattern (mathematically to a singular value). The value in a vector element corresponds therefore to the weight of that word usage pattern in the sentence.

I will illustrate this using the sample dataset from table 3.2. The sentences in the sample data set are titles of technical reports: d1-d5 are about human computer interaction and d6-d9 about graphics. The sentence vectors created with the two different models – LSA and VSM – are shown in tables 8.1 and 8.2.

For LSA, SVD was applied to TSM shown in table 3.3 and the three resulting sub-matrices were reduced to three dimensions ($k = 3$). For sentence to sentence comparison the sub-matrix $\mathbf{D}_k$ was scaled by $\mathbf{S}_k$ which results in the matrix shown in table 8.1. Whereas the sentence similarity estimation in VSM is based on the overlap in index terms, in LSA the calculations are based on the three dimensions or in other words on concepts or word usage patterns. Thus

	dim1	dim2	dim3
d1	0.38	-0.30	-0.16
d2	1.37	-1.83	-0.62
d3	1.41	-0.31	-0.21
d4	4.13	0.86	0.32
d5	0.30	-1.35	-0.65
d6	0.00	-0.16	0.52
d7	0.01	-0.41	1.13
d8	0.01	-0.60	1.56
d9	0.08	-0.80	1.13

Table 8.1: *Sentence vectors for the sample data set in LSA space (k=3)*

	computer	human	interface	response	survey	system	time	user	eps	trees	graph	minors
d1	1	1	1	0	0	0	0	0	0	0	0	0
d2	1	0	0	1	1	1	1	1	0	0	0	0
d3	0	0	1	0	0	1	0	1	1	0	0	0
d4	0	1	0	0	0	2	0	0	1	0	0	0
d5	0	0	0	1	0	0	1	1	0	0	0	0
d6	0	0	0	0	0	0	0	0	0	1	0	0
d7	0	0	0	0	0	0	0	0	0	1	1	0
d8	0	0	0	0	0	0	0	0	0	1	1	1
d9	0	0	0	0	1	0	0	0	0	0	1	1

Table 8.2: *Sentence vectors for the sample data set in traditional vector space*

for example the two sentences d8 "*Graph minors IV: Widths of trees and well-quasi-ordering*" and d9 "*Graph minors: A survey*" have a cosine similarity of 0.97 as opposed to 0.67 in the VSM. In vector space the sentences d6 "*The generation of random, binary, unordered trees*" and d9 receive a cosine similarity of 0 since they do not share any keywords. In LSA the cosine score for the two sentences is 0.95. This value is due to the fact that sentence d6 contains only one keyword *trees*. This term co-occurs twice with *graph*, so the sentences d6 and d9 are link through this second order co-occurrence.

To demonstrate the influence of the similarity calculation of sentences on sentence clustering, I use HAC to cluster the sentences from the sample dataset automatically. The results are shown in the dendrograms in figure 8.1.

The left figure shows the cluster tree for LSA, the right figure for VSM. It can be seen that in both vector spaces two distinct groups of sentences emerge [d6, d7, d8, d9] and [d1, d2, d3, d4, d5]. This distinction into two groups is correct as the sentences d1-d5 are about human-computer interaction and d6-d9 about graphics. However the links in the VSM cluster tree are higher than the links in the LSA cluster tree. That means that the distance between the objects is greater in the traditional vector space whereas in LSA the objects are closer to each other. If the tree were cut at $t = 0.5$ in order to get flat clusters, the clusterings would be CL_{LSA}=[d1, d2, d3, d4, d5][d6, d7, d8, d8] with an intra-cluster similarity of 0.82 and inter-cluster similarity of 0.01. For the VSM the clustering would be CL_{VSM}=[d2, d5][d6, d7, d8, d9][d1, d3, d4] with an

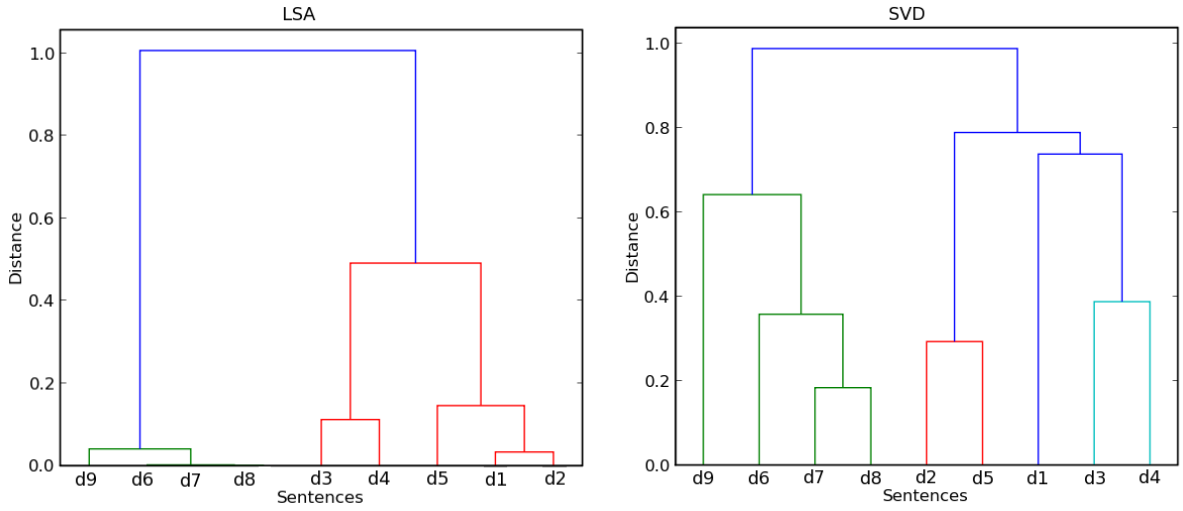


Figure 8.1: *Dendrogram for a sample data set*

intra cluster similarity of 0.54 and inter cluster similarity of 0.08. This small example shows that there is a difference between clusterings created on the basis of a traditional vector space (VSM) and on the basis of an LSA space. In this small data set the word usage pattern were very small and the scope was limited. Some MDS systems described in section 2.1, for example Goldstein et al. (2000); Lin and Hovy (2002), use the traditional vector space to calculate similarities between passages or sentences. The experiments described in this chapter show whether LSA performs better in estimating the similarity between sentences or passages than the VSM.

8.1 Comparison of LSA and VSM

In this experiment I compare the sentence clustering performance of two models – LSA and VSM. The models use different criteria for estimating the similarity of two sentences. In VSM the similarity of two sentences is based on simple world overlap, whereas in LSA high-order co-occurrences and word usage patterns provide the foundation for the similarity calculation. This is due to the different clustering spaces. In the VSM the complete space described by the original TSM is used. In LSA the original TSM was split into three submatrices using SVD (section 3.2). Then the matrices were reduced to k dimensions and the clustering space $\mathbf{CD_k}$ was created using the submatrices $\mathbf{D_k}$ and $\mathbf{S_k}$.

The performance of LSA for sentence clustering for MDS depends on some parameters. For this experiment the cophenetic distance t for cutting the clustering tree is set to 0.25 for the HGS and to 0.5 for the GGS (section 7.1). An extended local space is created containing all sentence sets (section 7.3) and the indexing vocabularies *SV*, *Num2* and *Coll* are used (section 7.2). The numbers of dimensions used for the LSA clustering space are listed in table 7.13. For the VSM the same settings were used apart from k because in VSM the complete space is used.

For each sentence set 12 clusterings were created – six to compare to the HGS and six for comparison with the GGS. For each indexing vocabulary tested two clustering were created – one on the basis of LSA and one on basis of the VSM. First the clusterings created using the two models described above were evaluated against the gold standard described in section 6.2 using the evaluation measures from section 5.5.

8.2 Results

The results of the evaluation can be seen in tables 8.3 and 8.4.

Table 8.3 shows the V_{beta} scores for each sentence set. These results show that for almost all sentence sets and all vocabularies the quality of the clusterings created on the basis of the LSA is higher than the quality of the clustering created using the VSM. The sole exception is the Schulz data set. Regardless of the indexing vocabulary and the gold standard subset, here the VSM always outperforms LSA.

	HGS						GGS					
Vocabulary	*SV*		*Num2*		*Coll*		*SV*		*Num2*		*Coll*	
Model	LSA	VSM	LSA	VSM	LSA	VSM	LSA	VSM	LSA	VSM	LSA	VSM
k	100		100		125		200		275		275	
EgyptAir	0.61	0.34	**0.63**	0.34	0.62	0.39	0.58	0.39	**0.59**	0.36	0.57	0.41
Hubble	0.70	0.37	0.69	0.36	**0.71**	0.37						
Iran	0.60	0.48	0.60	0.54	**0.60**	0.48	0.57	0.56	**0.61**	0.55	0.60	0.57
Rushdie	0.57	0.46	0.58	0.46	**0.60**	0.46	**0.64**	0.36	0.61	0.38	0.62	0.41
Schulz	0.39	0.58	0.40	0.58	0.43	**0.58**	0.39	0.61	0.39	0.61	0.39	**0.62**
Volcano	0.64	0.52	**0.66**	0.52	0.66	0.52	0.64	0.41	**0.68**	0.41	0.66	0.44

Table 8.3: *V_{beta} scores for each sentence set*

The *Num2* vocabulary seems to produce the best clusterings for LSA when compared to the GGS and the *Coll* vocabulary when compared to HGS. For the VSM the *Coll* vocabulary seems to work best for both gold standard subsets. Leaving the Schulz dataset out of consideration, LSA produces considerably better clustering than the VSM.

Table 8.4 shows the average V_{beta} scores for the sentence sets. The first row shows the scores for the training set (Iran_EgyptAir subset), the following two rows show the average V_{beta} scores for all sentence sets from the test set, i.e., excluding the training sets, with and without the Schulz dataset. The lower part of the table shows the scores for the whole data set including the training set, again with and without the Schulz set.

There is a noticeable difference in the scores between the training set and the test set, especially for the HGS. However when compared to the average V_{beta} scores for the whole data

	HGS						GGS					
Vocabulary	*SV*		*Num2*		*Coll*		*SV*		*Num2*		*Coll*	
k	100	VSM	100	VSM	125	VSM	200	VSM	275	VSM	275	VSM
training set	0.66	0.36	0.66	0.35	0.67	0.38	0.58	0.39	0.59	0.36	0.57	0.41
test set	0.55	0.51	0.56	0.53	0.57	0.51	0.56	0.48	0.57	0.49	0.57	0.51
- w/o Schulz	0.60	0.49	0.62	0.51	0.62	0.49	0.62	0.44	0.64	0.45	0.63	0.47
all	0.59	0.46	0.59	0.47	0.60	0.47	0.56	0.47	0.58	0.46	0.57	0.49
-w/o Schulz	0.63	0.44	0.63	0.45	0.64	0.44	0.61	0.43	0.62	0.43	0.61	0.46

Table 8.4: *Average V_{beta} scores for different combinations of the data set*

set excluding the Schulz sentence set the differences are considerably lower, meaning that fine-tuning of parameters and options on the basis of the training set works well. Nevertheless it is not impossible that there might be a better combination of parameters and options. However in a real world application a gold standard for the whole data set at the time of testing is rarely available.

The results show that LSA produces the best results in comparison to the HGS when the *Coll* vocabulary is used, whereas the VSM performs best with the *Num2* vocabulary. When the clusterings are compared to the GGS it is the other way round.

	HGS		GGS	
Model	LSA_{125}	VSM	LSA_{275}	VSM
Vocabulary	Coll	Num2	Num2	Coll
V_{beta}	0.64	0.45	0.62	0.46
NV_{beta}	0.81	0.42	0.74	0.34
$V_{0.5}$	0.66	0.49	0.63	0.47
NMI	0.65	0.56	0.63	0.52
NVI	0.22	0.16	0.31	0.27
F	0.13	0.13	0.13	0.11
$F\kappa$	0.17	0.04	0.17	0.10

Table 8.5: *Detailed results for LSA and VSM*

Table 8.5 shows the scores from all evaluation measures chosen in section 5.5. For this comparison the vocabulary that works best for both models was chosen. The results show that when the clusterings are evaluated using different measures, LSA receives for all measures except the NVI better scores than the VSM.

Figure 8.2 shows the quality of the automatically generated sentence clusterings in comparison with the three different gold standard subsets HGS, GGS and CGS calculated using the V_{beta} measure for different numbers of k. The values were obtained using the EXTENDED LOCAL

LSA space and the SV vocabulary. For evaluation the average V_{beta} for the five sentence sets EgyptAir, Hubble, Iran, Rushdie and Volcano was taken. As explained above the Schulz data set was excluded.

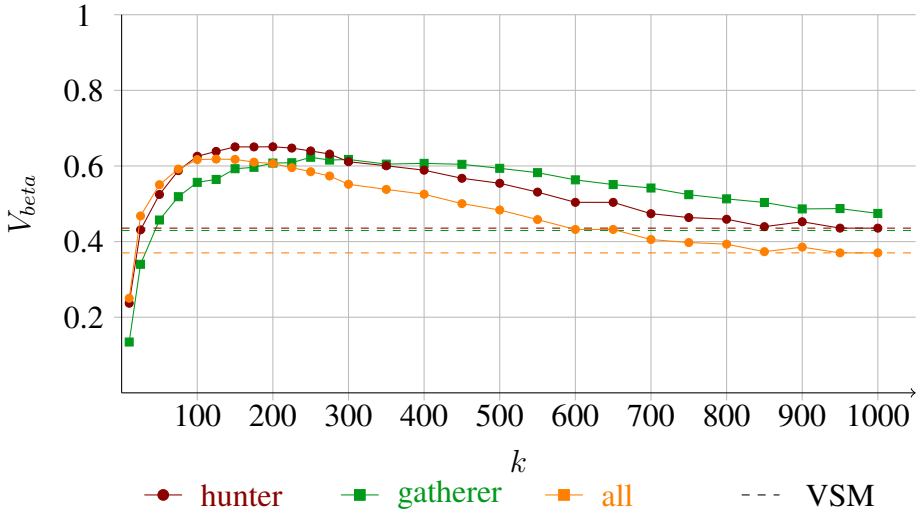


Figure 8.2: *V_{beta} scores for clusterings created using LSA and VSM with different k*

These results show that the range of k in which LSA outperforms VSM is quite large. So even if the values for k can not be determined using a training set the probability that k is chosen so that LSA produces better results than VSM is high.

8.3 Discussion

Differences between clusterings In this section I evaluate how the clusterings created on the basis of an LSA space differs from clusterings created on the basis of the VSM space. For this analysis I chose sentence clusters from the clusterings created for the Rushdie data set. For the comparison to the HGS the clustering created by Judge_S was chosen to represent the HGS. The clustering spaces for LSA and VSM were created using COLL vocabulary since for the Rushdie sentence set the best results for both models were achieved using this vocabulary. The LSA space was reduced to $k = 125$ dimensions. I chose the cluster with the topic "Rushdie spent a decade in hiding following a death sentence issued by Ayatollah Khomeini in 1989" from the clustering created by Judge_S. Table 8.6 gives an overview of the clusters and sentences discussed. The full sentence cluster and the equivalences created automatically on basis of the LSA space and the standard space are shown in Table 8.8. The original cluster from the gold standard clustering contains the sentences 22, 61, 74, 79, 86 and 94. With the VSM only a cluster containing the sentences 61 and 94 was created. The other four sentences are not part of any cluster in the VSM clustering. In the LSA clustering there is a cluster containing three of the six reference sentences namely 61, 86 and 94. Sentences 61 and 94 are almost identical

LSA	VSM	Hunter
11 21 61 66 86 94 31 38 43 79	61 94	22 61 74 79 86 94 2 11 21 31 38 66

Table 8.6: *Comparison of clusters created by LSA, VSM and by a hunter*

sentences differing in only a few words. The LSA cluster and the reference cluster have one more sentence in common, namely 86. Two of the remaining sentences from the reference cluster (22 and 74) are not part of any cluster in the LSA clustering. Sentence 79 was clustered together with sentence 39 ("While stopping short of revoking a death sentence against Rushdie, Iran says it won't adopt any measures that threaten his life, or anyone connected to his book – The Satanic Verses.") in the LSA clustering.

The LSA cluster contains the sentences 11, 21, 61, 66, 86 and 94. From reading the sentences they are seem to be very similar. But the human judge discriminated between the two topics in these sentences. That is why the LSA cluster and the reference cluster have only three sentences in common. The other three sentences from the LSA cluster were put in a new cluster by Judge_S with the topic headline "Ayatollah Khomeini issued a death sentence on Rushdie in 1989". For the human annotator there is a distinction between the conviction and the result of it, namely that Rushdie has been living in hiding since then. Nevertheless in the LSA clustering these sentences were at least used and put into one cluster in which the sentences are similar. In the VSM clustering the sentences don't even occur. On basis of the VSM only very few clusters were created: three cluster including 6 sentences in total. The LSA clustering consists of 18 clusters with 50 sentences in total. The clustering created by Judge_S includes 45 sentences in 10 clusters.

The gatherers did not make the distinction between the two groups of sentences discussed before. Here the clustering created by Judge_A was chosen to be compared to the clustering created on the basis of 200-dimensional LSA space including the standard vocabulary and a full VSM space including the *Coll* vocabulary. Table 8.7 gives an overview of the sentences and clusters in consideration.

LSA	VSM	Gatherer
11 21 31 61 66 74 79 86 94 2 38	11 21 61 66 86 94	2 11 21 31 38 48 61 66 79 86 94

Table 8.7: *Comparison of a cluster created by LSA, VSM, and a gatherer*

LSA	VSM	Hunter
61 - Rushdie has spent nearly a decade in hiding since the late Ayatollah Ruhollah Khomeini called for his death on Feb. 14, 1989, claiming his book, "The Satanic Verses" blasphemed Islam.	**61** - Rushdie has spent nearly a decade in hiding since the late Ayatollah Ruhollah Khomeini called for his death on Feb. 14, 1989, claiming his book, "The Satanic Verses" blasphemed Islam.	**61** - Rushdie has spent nearly a decade in hiding since the late Ayatollah Ruhollah Khomeini called for his death on Feb. 14, 1989, claiming his book, "The Satanic Verses" blasphemed Islam.
94 - Rushdie has spent nearly a decade in hiding since Iran's late Ayatollah Ruhollah Khomeini called for his death in 1989, claiming "The Satanic Verses" blasphemed Islam.	**94** - Rushdie has spent nearly a decade in hiding since Iran's late Ayatollah Ruhollah Khomeini called for his death in 1989, claiming "The Satanic Verses" blasphemed Islam.	**94** - Rushdie has spent nearly a decade in hiding since Iran's late Ayatollah Ruhollah Khomeini called for his death in 1989, claiming "The Satanic Verses" blasphemed Islam.
86 - Rushdie has been protected by the government since 1989, when the Iranian religious leader Ayatollah Ruhollah Khomeini decreed that the author of "The Satanic Verses" should be killed.		**86** - Rushdie has been protected by the government since 1989, when the Iranian religious leader Ayatollah Ruhollah Khomeini decreed that the author of " The Satanic Verses" should be killed.
11 - Rushdie was condemned to death in 1989 by Iran's late Spiritual Leader Ayatollah Ruhollah Khomenei for his novel "The Satanic Verses," which was regarded as an insult to the Islamic Prophet Muhammad.		**22** - Since then, he has lived in safe houses around London, guarded around the clock by British agents.
21 - After he wrote the novel "The Satanic Verses" Rushdie was accused of blasphemy against Islam and condemned to death in a religious decree by Ayatollah Ruhollah Khomeini in 1989.		**74** - Rushdie had spent nearly a decade in hiding since Khomeini issued the edict.
66 - Ayatollah Ruhollah Khomeini issued the fatwa, or religious edict, on Rushdie in 1989, claiming his book "The Satanic Verses" blasphemed Islam.		**79** - Booker prize winner Rushdie had been living a furtive life since 1989 under police protection in Britain following a death sentence issued by the late Iranian leader Ayatollah Khomeini for blaspheming Islam in his book the Satanic Verses.

Table 8.8: *Comparison of sentence clusters created by LSA, VSM, and a hunter*

Here the LSA cluster and the human created cluster have 8 sentences in common. In contrast the VSM cluster and the reference cluster share only six sentences. The VSM clustering contains seven clusters with 18 sentences. 73 sentences are grouped into 24 clusters in the LSA clustering. Judge_A used 70 sentences in 15 clusters.

The examples from both gold standard subsets show that the clusters created on basis of the LSA space bear much more resemblance to the clusters created by human judges than the VSM clusters. Moreover the overall characteristics (number of sentences, number of clusters) of the clusterings are more similar between the LSA clusterings and the human created clusterings as between the VSM clusterings and the human clusterings.

Schulz data set The LSA clusterings for the Schulz data set received the lowest scores. The clusterings for this set created on basis of the traditional space received higher scores. The question remains, why this set is different.

The Schulz sentence set is the largest sentence set in the corpus. It contains 248 sentences which come from only 5 documents. The other sets have 168 sentences on average. The Schulz sentence set is the only set extracted from a DUC document set from 2003. In contrast to DUC 2002, in 2003 the tasks were not limited to general single and multi document summarization. In 2003 new tasks were introduced. One of the tasks is to produce a short summary for a cluster given a viewpoint. The viewpoints for the Schulz document set are:

- Peanuts comic strip defined Charles M. Schulz's life
- Chronology of "Peanuts" creator Charles Schulz's career
- The life, death, and accomplishments of Charles Schultz, creator of the popular comic strip "Peanuts"
- Spread of Peanuts comic strip to world-wide audience.

Thus the document was designed to capture all of these different viewpoint and topics. The other documents set from which the sentence sets were drawn, were designed with a general summary in mind. The Schulz data set is not as homogenous as the other sets.

Since there are more topics within the Schulz sentence set than there are in the other sentence sets, other parameters may have to be used. To examine this, I ran another experiment using different number of dimensions where $k = 10$, 25, 50, 75, 100, 125, 150, 175, 200, 225, 300, 400, 500, 750, 1000. It turned out that for the Schulz data set many more dimensions were needed. Table 8.9 shows V_{beta} scores for the three selected vocabularies. In contrast to the earlier experiment here the optimal number of dimensions for the Schulz sentence set was used.

In the previous experiment k was set to 100 or 125 for comparison to the HGS and to 200 or 275 for comparison to the GGS. Here k ranges from 600 to 700 when compared to the HGS and

	HGS						GGS					
Vocabulary	*SV*		*Num2*		*Coll*		*SV*		*Num2*		*Coll*	
Model	LSA	VSM	LSA	VSM	LSA	VSM	LSA	VSM	LSA	VSM	LSA	VSM
k	650		600		700		1000		1000		1000	
Schulz	0.67	0.58	0.65	0.58	0.66	0.58	0.59	0.61	0.61	0.61	0.59	0.62

Table 8.9: *V_{beta} scores for the Schulz sentence set with optimal number of dimensions for selected vocabulary*

for the GGS comparison the optimal number of dimensions is $k = 1000$. However even when the optimal number of dimensions is used the VSM clusterings receives higher scores when compared to the gatherer subset. In comparison to the hunters the LSA clusterings outperforms the VSM clustering when the optimal number of dimensions is used.

8.4 Chapter summary

After this set of experiments it can be said that clusterings created on basis of a reduced LSA clustering space are more similar to clustering created by humans than clusterings created on basis of a standard VSM.

Various factors have an impact on the quality of the clusterings created on the basis of an LSA space. One very important factor is the optimal number of dimensions. This is still the most crucial point in sentence clustering with LSA for MDS. When the clusterings created should include clusters for the most important themes in a document collection, fewer dimensions are needed. If clusterings are to be created which are more similar to clusterings created by gatherers, that is to say more detailed clusters should be created, more dimensions are needed.

The optimal number of dimensions also depends on the size of the clustering space, the homogeneity of the data set and the clustering algorithm. I also showed that even when the optimal number of dimensions cannot be estimated using a training set, the probability of choosing a value for k where LSA does not outperform VSM is quite low, since the range of k for which LSA produces better results than VSM is quite big as can be seen in figure 8.2.

Another important factor in this setup is the threshold parameter t which specifies where to cut the clustering tree into separate clusters. For clustering resembling hunter clusterings the tree can be cut at a lower level, whereas when gatherer-style clusterings should be created the threshold has to be raised.

The size of the clustering space is another factor that has a great impact on the quality of a clusterings. For sentence clustering for MDS a LOCAL or better an EXTENDED LOCAL space is preferable to a GLOBAL space.

The selection of an indexing vocabulary has a minor effect of the quality of the clusterings. Indexing vocabularies including only nouns or nouns and verbs does lead to lower quality clus-

terings. However the differences in full indexing vocabularies (those not limited to certain word classes) are marginal. A full indexing vocabulary including collocations seems to work best for a hunter strategy and an indexing vocabulary where numbers are replaced by their numbers of digits are best used for clustering resembling gatherer clusterings.

Another conclusion that can be drawn from the experiments is that in general it is possible to train the sentence clustering system on a smaller training set. However this is only possible if the data corpus is homogeneous and the document sets were all designed for the same task. Then even the optimal number of dimensions k can be predefined without great loss of performance. Of course it is possible that for each single sentence set better settings can be found. However, given that gold standards for sentence clustering in MDS are rare, a system can be trained on a small set of data.

Chapter 9

Conclusion

Das also war des Pudels Kern![12]

Faust
JOHANN WOLFGANG GOETHE

In this thesis, the applicability of Latent Semantic Analysis (LSA) to sentence clustering for MDS was investigated. The assumption was made that LSA could provide a text generation system with better quality sentence clusters than a more shallow approach like VSM. In contrast to VSM, which measures the similarity of sentences by word overlap, LSA takes word usage patterns into account.

In chapter 4, I introduced and discussed essential parameters that might influence the quality of sentence clusterings in general and when using LSA in particular. The parameters include options for creating partitional clusters from hierarchical cluster trees, type of index vocabularies, size of the semantic space, and the number of dimensions used in LSA.

To evaluate sentence clusterings directly, which has never been done before in MDS, and to assess the influence of the described parameters on the quality of sentence clusterings, a clearly laid out evaluation strategy was developed (chapter 5). This strategy includes an external evaluation scheme using a compound gold standard. Different evaluation measures for comparing automatically generated clusterings against the gold standard were discussed and evaluated in section 5.5, with the conclusion that metrics based on entropy that measure both homogeneity and completeness of a clustering solution are most suitable for sentence clustering in MDS.

The first major contribution of this thesis is the creation of the first compound gold standard for sentence clustering. Several human annotators were asked to group sentences from a data set, extracted from DUC document clusters, into clusters of similar sentences. They were requested to follow the guidelines provided to them. These guidelines were especially created for human sentence clustering in order to reduce the variation in human clusterings (section

[12] "So that was the quintessence of the cur!"

5.4). While analysing the clusterings created by human judges two distinct types were identified: hunters and gatherers, who are clearly distinguishable by the size and the structure of their clusterings.

The second major contribution is the analysis of the most important parameters for latent semantic sentence clustering. Using the gold standard the influence of the parameters introduced in chapter 4 on the quality of the sentence clusterings was evaluated (chapter 7). These experiments showed that the threshold for the cophenetic distance to extract the partitional clusters from a dendrogram varies for the two gold standard subsets. For the HGS a lower threshold is most suitable whereas a higher t is required to obtain larger clusters that are more similar to clusterings produced by gatherers. Different indexing vocabularies on the other hand hardly influence the sentence clustering quality. The size of the semantic space has a larger impact on the quality of sentence clusters. The EXTENDED LOCAL LSA provides the best results. During analysis of the optimal number of dimensions for LSA in sentence clustering it was verified that the quality of the clusterings vary for different k and the LSA pattern is similar to that of LSA in IR. Furthermore it was established that the optimal number of dimensions depends on different parameters, e.g., the size of the semantic space and the threshold for the cophenetic distance. The most striking result is that the optimal number for k depends on the gold standard subset the clusterings are compared to. If the system is to create smaller clusterings incorporating only the most salient information from a document set, that is to say if the clusterings are more similar to clusterings created by hunters, fewer dimensions are needed than when the system-generated clusterings are compared with clusterings created by gatherers.

In the final experiment (chapter 8), it was shown that when the same parameter settings are used LSA outperforms VSM significantly in clustering sentences for MDS. I also showed that the range of k in which LSA outperforms VSM is large, so that the probability to choose a value for k where LSA does not outperform VSM is very low.

In future work I would like to extend my approach to co-cluster words, phrases and sentences. Different sized units can help text-to-text generation systems to produce better and grammatical sentences for an abstract. The challenge is to extract the most content bearing words and phrases from the sentences for clustering. For word selection different strategies need to be tested and evaluated. One strategy would be to select only nouns and verbs, another to restrict the words for clustering to include only subjects, objects and predicates. Another problem that needs to be addressed is the extraction of phrases from sentences. Here the definition of a phrase will be essential. Other considerations include the usefulness of a type of phrase to a text-to-text generation system and the applicability of the clustering algorithm to co-clustering. The different approaches and algorithms to be used need to be evaluated. Therefore the gold standard needs to be extended to cover words and phrases. Another interesting continuation of this work would be to extract the most important information from the clusters to create extracts or abstracts. A definition of most important information and a strategy to extract it, has to be defined. One approach is to select the text units that are nearest to the center of a cluster.

Bibliography

Aliguliyev, R. (2006). A Novel Partitioning-Based Clustering Method and Generic Document Summarization. In *WI-IATW '06: Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, pages 626—629, Washington, DC, USA.

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.

Arndt, C. (2004). *Information Measures: Information and its description in Science and Engineering*. Springer.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley, Reading, MA, USA, 1st edition.

Banko, M. and Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 26–33, Morristown, NJ, USA.

Barzilay, R. (2003). *Information fusion for multidocument summarization: Paraphrasing and generation*. PhD thesis, DigitalCommons@Columbia.

Barzilay, R. and Elhadad, M. (1997). Using Lexical Chains for Text Summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

Barzilay, R. and Lapata, M. (2008). Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Barzilay, R. and McKeown, K. R. (2005). Sentence Fusion for Multidocument News Summariation. *Computational Linguistics*, 31(3):297–327.

Bing, Q., Ting, L., Yu, Z., and Sheng, L. (2005). Research on Multi-Document Summarization Based on Latent Semantic Indexing. *Journal of Harbin Institute of Technology*, 12(1):91–94.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. O'Reilly, Sebastopol, CA, USA.

Bouras, C. and Tsogkas, V. (2008). Improving Text Summarization Using Noun Retrieval Techniques. In Lovrek, I., Howlett, R., and Jain, L., editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 5178 of *Lecture Notes in Computer Science*, pages 593–600. Springer.

Bradford, R. B. (2008). An empirical study of required dimensionality for large-scale latent semantic indexing applications. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 153–162, New York, NY, USA.

Briscoe, T., Carroll, J., and Watson, R. (2006). The Second Release of the RASP System. In *COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, Australien.

Chalker, S. and Weiner, E. S. C. (1994). *The Oxford dictionary of English grammar*. Clarendon Press, Oxford, UK, 1. edition.

Cleverdon, C. W. (1967). The Cranfield tests on index language devices. In *ASLIB proceedings*, volume 19(6), pages 173–192.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *Proceedings of COLING 2004*, pages 350–356, Geneva, Switzerland.

Dolan, W. B. and Brockett, C. (2005). Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of The 3rd International Workshop on Paraphrasing (IWP2005)*, Jeju Island, South Korea.

Dubin, D. (2004). The most influential paper Gerard Salton never wrote. *Library Trends*, 52(4):748–764.

DUC (2007). Document Understanding Conference. `http://duc.nist.gov/` Online; accessed 2011-06-25.

Dumais, S. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23(2):229–236.

Dumais, S. T. (1990). Enhancing Performance in Latent Semantic Indexing. Technical Report TM-ARH-017527, Bellcore (now Telcordia Technologies), Morristown, TN, USA.

Dumais, S. T. (2004). Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1):188–230.

Dumais, S. T. (2007). LSA and Information Retrieval: Getting Back to Basics. In *Handbook of Latent Semantic Analysis*, chapter 16, pages 293–321. Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Eads, D. (2008a). hcluster: Hierarchical Clustering for Scipy. "`http://scipy-cluster.googlecode.com/` Online; accessed 2011-03-28".

Eads, D. (2008b). hcluster: Hierarchical Clustering for SciPy API Documentation. "`http://www.cs.ucsc.edu/~eads/cluster.html`Online; accessed 2011-03-28".

Efron, M. (2002). Amended parallel analysis for optimal dimensionality reduction in latent semantic indexing. Technical Report No. TR-2002-03, University of North Carolina, Chapel Hill, NC, USA.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA, USA.

Fleiss, J. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Foltz, P. W., Kintsch, W., and Landauer, T. K. (1998). The Measurement of Textual Coherence with Latent Semantic Analysis. *Discourse Processes*, 25(2&3):285–307.

Frakes, W. B. (1992). Stemming Algorithms. In Frakes, W. B. and Baeza-Yates, R., editors, *Information Retrieval: Data Structures & Algorithms*, chapter 8, pages 131–160. Prentice Hall, NJ, USA.

Geiß, J. (2006). Latent Semantic Indexing and Information Retrieval - a Quest with BosSE. Master's thesis, Ruprecht-Karls-Universität, Germany.

Goldstein, J., Mittal, V., Carbonell, J., and Kantrowitz, M. (2000). Multi-document summarization by sentence extraction. In *Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*, pages 40–48.

Hachey, B., Murray, G., and Reitter, D. (2005). The Embra system at DUC 2005: Query-oriented multi-document summarization with a very large latent semantic space. In *Proceedings of the Document Understanding Conference (DUC) 2005*.

Hachey, B., Murray, G., and Reitter, D. (2006). Dimensionality Reduction Aids Term Co-Occurrence Based Multi-Document Summarization. In *Proceedings of the COLING-ACL Workshop Task-Focused Summarization and Question Answering 2006*, SumQA '06, pages 1–7, Sydney, Australia.

Haenelt, K. (2009). Information Retrieval Modelle - Vektormodell. `http://kontext.fraunhofer.de/haenelt/kurs/folien/Haenelt_IR_Modelle_Vektor.pdf` Online; accessed 2011-03-02.

Harman, D. and Liberman, M. (1993). *TIPSTER Complete*. Linguistic Data Consortium, Philadelphia. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93T3A Online; accessed 2011-07-02.

Hatzivassiloglou, V., Klavans, J. L., Holcombe, M. L., Barzilay, R., Kan, M.-Y., and McKeown, K. R. (1999). Detecting text similarity over short passages: Exploring Linguistic Feature Combinations via Machine Learning. In *Proceedings of the 1999 Joint SIGDAT Conference on empirical Methods in Natural Language Processing and very large corpora*, pages 203–212, College Park, MD, USA.

Hatzivassiloglou, V., Klavans, J. L., Holcombe, M. L., Barzilay, R., Kan, M.-Y., and McKeown, K. R. (2001). SIMFINDER: A Flexible Clustering Tool for Summarization. In *NAACL Workshop on Automatic Summarization*, pages 41–49, Pittsburgh, PA, USA.

Hess, A. and Kushmerick, N. (2003). Automatically Attaching Semantic Metadata to Web Services. In *Proceedings of the 2nd International Semantic Web Conference (ISWC 2003)*, Florida, USA.

Horn, J. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185.

Hull, D. (1994). Improving Text Retrieval for the Routing Problem using Latent Semantic Indexing. In *Proceedings of the 17th Annual International Conference on Research and Development in Information Retrieval*, pages 282–291, London, UK.

Jiang, F. and Littman, M. l. (2000). Approximate dimension equalization in vector-based information retrieval. In *Proceedings of the Seventeenth International Conference on Machine Learning*, Stanford, CA, USA.

Kontostathis, A. (2007). Essential Dimensions of Latent Semantic Indexing (LSI). *Hawaii International Conference on System Sciences*.

Kontostathis, A. and Pottenger, W. M. (2006). A framework for understanding Latent Semantic Indexing (LSI) performance. *Information Processing and Management*, 42(1):56–73.

Landauer, T. and Dumais, S. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review;*, 104(2):211–240.

Landauer, T. K. and Dumais, S. T. (2008). Latent semantic analysis. *Scholarpedia*, 3(11):4356.

Lang, C. B. and Pucker, N. (1998). *Mathematische Methoden in der Physik*. Spektrum Akademischer Verlag, Heidelberg, Germany.

Li, W., Li, B., and Wu, M. (2006). Query focus guided sentence selection strategy for duc 2006. In *Proceedings of the Document Understanding Workshop at the HLT/NAACL Annual Meeting*, Brooklyn, New York.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Proceedings of ACL workshop on Text Summarization Branches Out*, Barcelona, Spain.

Lin, C.-Y. and Hovy, E. (2002). From Single to Multi-document Summarization: A Prototype System and its Evaluation. In *Proceedings of the ACL conferenc*, pages 457–464, Philadelphia, PA, USA.

Linguistic Data Consortium (2002). The AQUAINT Corpus of English News Text. `http://www.ldc.upenn.edu/Catalog/docs/LDC2002T31/` Online; accessed 2011-07-02.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Marcu, D. (1997). From Discourse Structures to Text Summaries. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 82–88, Madrid, Spain.

Marcu, D. and Gerber, L. (2001). An Inquiry into the Nature of Multidocument Abstracts, Extracts, and Their Evaluation. In *Proceedings of the NAACL-2001 Workshop on Automatic Summarization*, Pittsburgh, PA.

McKeown, K. R., Klavans, J. L., Hatzivassiloglou, V., Barzilay, R., and Eskin, E. (1999). Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of AAAI-99*, pages 453–460, Orlando, FL, USA.

Meila, M. (2007). Comparing clusterings–an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

Miller, T. (2003). Generating coherent extracts of single documents using latent semantic analysis. Master's thesis, University of Toronto, Toronto, CA.

Naughton, M. (2007). Exploiting Structure for Event Discovery Using the MDI Algorithm. In *Proceedings of the ACL 2007 Student Research Workshop*, pages 31–36, Prague, Czech Republic.

Naughton, M., Kushmerick, N., and Carthy, J. (2006). Clustering sentences for discovering events in news articles. *Lecture Notes in Computer Science*.

Naughton, M., Stokes, N., and Carthy, J. (2008). Investigating Statistical Techniques for Sentence-Level Event Classification. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK.

Nenkova, A. and Passonneau, R. (2004). Evaluating content selection in summarization: The Pyramid method. In *Proceedings of HLT/NAACL 2004*, Boston, MA USA.

Persson, P.-O. (2007). Script to MIT 18.335: Introduction to Numerical Methods (Fall 2007) Lecture 3. `http://persson.berkeley.edu/18.335/lec3.pdf` Online; accessed 2011-02-21.

Peters, P. (2004). *The Cambridge guide to English usage*. Cambridge University Press, Cambridge, UK.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988). *Numerical Recipies in C: The art of Scientific Programming*. Cambridge University Press, Cambridge, UK.

Quesada, J. (2007). Creating your own LSA space. In Landauer, T. K., McNamara, D. S., Dennis, S., and Kintsch, W., editors, *Handbook of Latent Semantic Analysis*, chapter 1, pages 71–85. Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Radev, D. R., Jing, H., and Budzikowska, M. (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *ANLP/NAACL Workshop on Summarization*, pages 21–29, Morristown, NJ, USA.

Radev, D. R., Jing, H., Styś, M., and Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing and Management*, 40:919–938.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *American Statistical Association Journal*, 66(336):846–850.

Rath, G., Resnick, A., and Savage, R. (1961). The formation of abstracts by the selection of sentences. In *American Documentation*, pages 57–64, Morristown, NJ, USA. Association for Computational Linguistics.

Reichart, R. and Rappopor, A. (2009). The NVI Clustering Evaluation Measure. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, Boulder, CO, USA.

Rohde, D. (2008). SVDLIBC Doug Rohde's SVD C Library version 1.34. `http://tedlab.mit.edu/~dr/SVDLIBC/` Online; accessed 2011-03-29.

Rosenberg, A. and Hirschberg, J. (2007). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Proceedings of the 2007 Joint Conference on Empirical*

Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 410–420.

Salton, G. (1971a). A New Comparison Between Conventional Indexing (MEDLARS) and Automatic Text Processing (SMART). Technical Report TR71-115, Cornell University, Computer Science Department.

Salton, G., editor (1971b). *The SMART Retrieval System Experiments in Automatic Document Processing*. Prentice Hall, Englewood Cliffs, NJ, USA.

Salton, G. (1979). Mathematics and information retrieval. *Journal of Documentation*, 35(1):1–29.

Salton, G., Buckley, C., and Yu, C. T. (1982). An Evaluation of Term Dependence Models in Information Retrieval. In *SIGIR '82: Proceedings of the 5th annual ACM conference on Research and development in information retrieval*, pages 151–173, New York, NY, USA. Springer.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Salton, G., Singhal, A., Mitra, M., and Buckley, C. (1997). Automatic text structuring and summarization. *Information Processing and Management*, 33(2):193–207.

Schütze, H., Hull, D. A., and Pedersen, J. O. (1995). A Comparison of Classifiers and Document Representations for the Routing Problem. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, series = SIGIR '95, Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 229–237.

Seno, E. and Nunes, M. (2008). Some experiments on clustering similar sentences of texts in portuguese. *Proceedings of the 8th international conference on Computational Processing of the Portuguese Language*.

Siegel, S. (2001). *Nichtparametrische statistische Methoden*. Verlag Dietmar Klotz, Eschborn bei Frankfurt a.M., Germany, 5. edition.

Skillicorn, D. (2007). *Understanding complex datasets: data mining with matrix decompositions*. Chapman & Hall/CRC data mining and knowledge discovery series. Chapman & Hall/CRC Press.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Spärck Jones, K. and Gallier, J. R. (1996). *Evaluating Natural Language Processing Systems, An Analysis and Review*, volume 1083 of *Lecture Notes in Computer Science*. Springer.

Steinbach, M., Karypis, G., and Kumar, V. (2000). A Comparison of Document Clustering Techniques. In Grobelnik, M., Mladenic, D., and Milic-Frayling, N., editors, *KDD-2000 Workshop on Text Mining, August 20*, pages 109–111, Boston, MA.

Steinberger, J. and Krišt'an, M. (2007). LSA-Based Multi-Document Summarization. In *8th International PhD Workshop on Systems and Control, a Young Generation Viewpoint*, pages 87–91, Balatonfured, Hungary.

Strang, G. (2003). *Lineare Algebra*. Springer, Germany.

Sun, B., Mitra, P., Giles, C. L., Yen, J., and Zha, H. (2007). Topic segmentation with shared topic detection and alignment of multiple documents. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 199–206, New York, NY, USA.

Thadani, K. and McKeown, K. R. (2008). A framework for identifying textual redundancy. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 873–880, Stroudsburg, PA, USA.

TREC (2011). Text REtrieval Conference. `http://trec.nist.gov/` Online; accessed 2011-07-02.

van Halteren, H. and Teufel, S. (2003). Examining the consensus between human summaries: initial experiments with factoid analysis. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop*, pages 57–64, Morristown, NJ, USA. Association for Computational Linguistics.

Vlachos, A., Korhonen, A., and Ghahramani, Z. (2009). Unsupervised and Constrained Dirichlet Process Mixture Models for Verb Clustering. In *Proceedings of the EACL workshop on GEometrical Models of Natural Language Semantics*.

Wiener, E. D., Pedersen, J. O., and Weigend, A. S. (1995). A neural network approach to topic spotting. In *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, Las Vegas, US.

Wikipedia (2011a). Latent semantic indexing – wikipedia, the free encyclopedia. "`http://en.wikipedia.org/w/index.php?title=Latent_semantic_indexing&oldid=408562332` Online; accessed 2011-01-18".

Wikipedia (2011b). News style – wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=News_style&oldid=414160896` Online; accessed 2011-03-02.

Yu, C., Cuadrado, J., Ceglowski, M., and Payne, J. S. (2002). Patterns in Unstructured Data – Discovery, Aggregation, and Visualization. "`http://www.knowledgesearch.org/lsi/`, Online, accessed 2011-03-29".

Zelikovitz, S. and Kogan, M. (2006). Using web searches on important words to create background sets for lsi classification. In *FLAIRS Conference*, pages 598–603.

Zha, H. (2002). Generic Summarization and Keyphrase Extraction using Mutual Reinforcement Principle and Sentence Clustering. In *Proceedings of the 25th Annual ACM SIGIR Conference*, pages 113–120, Tampere, Finland.

Zha, H. and Simon, H. D. (1999). On Updating Problems in Latent Semantic Indexing. *SIAM Journal on Scientific Computing*, 21(2):782–791.

Zhao, Y. and Karypis, G. (2001). Criterion functions for document clustering: Experiments and analysis. Technical report, Department of Computer Science, University of Minnesota, MN, USA.

Appendix A

Guidelines for sentence clustering

Task T1 Form a set of clusters. A cluster corresponds to one set of sentences that are similar or closely related to each other. Not every sentence will belong to a cluster.

T2 Create your own description for each of clusters in form of a sentence and write it down.

T3 Rank the clusters by importance.

Material You will get a list of sentences ordered by date. Each sentence has a unique ID consisting of the document number x and a sentence number y followed by the date the document was published, the news-agency and the actual sentence:

`17.149 1996-03-07 XIE We have to defend the Islamic culture, " he said.`

The document number, the date and agency is purely for your information, because sometimes it is helpful to know which sentences belong to the same document and in which order the documents were published.

Procedure

P1 Read all documents. Start clustering from the first sentence in the list. Put every sentence that you think will attract other sentences into an initial cluster. If you feel, that you will not find any similar sentences to a sentence, put it immediately aside. Continue clustering and build up the clusters while you go through the list of sentences.

P2 You can rearrange your clusters at any point.

P3 When you are finished with clustering, check that all important information from the documents is covered by your clusters. If you feel that a very important topic is not expressed in your clusters, look for evidence for that information in the text, even in secondary parts of a sentence.

P4 Go through your sentences which do not belong to any cluster and check if you can find a suitable cluster.

P5 Do a quality check and make sure that you wrote down a sentence for each cluster and that the sentences in a cluster are from more than one document.

P6 Rank the clusters by importance.

P7 Return a list of clusters in the form:
rank of cluster – ”your sentence”: sentence number⟨blank⟩sentence number⟨blank⟩...

General Principal There will be several demands that pull against each other; choose what to you seems the best compromise. There is no ideal clustering. Do your best.

Similarity There is a whole spectrum of similarities including the following:

a) paraphrases (same person, same event, same time but different wording; roughly same amount of information in the sentences)

b) sentences that are actually paraphrases but differ in numbers (see rule 7)

c) part of a sentence (clause or phrase) is similar to another sentence (partial information overlap)

d) similarity between a sentence and a pair of sentences

In general prefer a)

Most important rules They should always be kept in mind.

R1 Clusters should be pure, i.e. each cluster should contain only one topic.

R2 In an ideal cluster the sentences (or at least one part of each sentence) would be very similar (almost paraphrases).

R3 The information in one cluster should come from as many different documents as possible. The more different sources the better.

More specific rules

R4 Each cluster **must** have at least two sentences and should have more than two if possible.

R5 Each cluster **must** include sentences from different documents. A cluster consisting only of sentences from one document is not a valid cluster.

R6 A sequence of consecutive sentences from one document *should* not normally be a cluster. There is one exception: if the sentences are very similar they can end up in one cluster (but only if they attract at least one sentence from another document).

R7 If similar sentences only vary in numbers they can still belong to the same cluster:

a) Vagueness in numbers

- `Clark Air Base is in Angeles, a city of more than 300,000 people about 50 miles north of Manila.`
- `350,000 residents live in Angeles City, where the air base is located, about 50 miles north of Manila.`

b) If a sentence provides new or updated numerical information and only differs from another sentence in numbers, these sentences can still belong to the same cluster.

- `Two people have been reported killed so far.`
- `At least four people have died, 24 have been hurt and four have been listed as missing since Pinatubo began erupting Sunday.`

R8 Not every sentence inside a cluster will be equally similar to all sentences in that cluster.There may be a subset of sentences that is particularly similar to each other. That is o.k. as long as you think the overall cluster is similar.

R9 Do not use too much inference.
Only because $A \Rightarrow B$ (B follows from A) does not mean that they should be within the same cluster.

R10 If a sentence consists of more than one part and the parts would belong to different clusters, put the sentence in the more important cluster, particularly if this cluster does not yet include a sentence from the document the sentence in question belongs to.

R11 Generalisation is allowed. Sentences in a cluster do not have to be very similar. They still need to be about the same person, fact or event, but they do not have to cover exactly the same information or amount of information.

R12 Take discourse/context into account. Do not look at that sentence on its own but within context of the whole document. If something important is missing from the previous sentences add it to the sentence.

- `Charles Schulz, the creator of ``Peanuts,'' died in his sleep on Saturday night at his home in Santa Rosa, Calif.`
- `He was 77.` ⇒ Charles Schulz was 77 when he died.

R13 If two sentences cover exactly the same information as one other sentence, only put them into a cluster if the information is very important for the summary.

- `"No hypothesis for the cause of this accident has been accepted, and the activities that I have outlined indicate that there is much that still needs to be done before a determination of cause can be reached. "`
- `"No hypothesis for the cause of this accident has been accepted," Hall said Friday in a statement.`
- `"There is much that still needs to be done before a determination of cause can be reached."`

Index