University of West Bohemia

Faculty of Applied Sciences

# Doctoral Thesis

in partial fulfillment of the requirements

for the degree of

*Doctor of Philosophy*

in specialization

*Computer Science and Engineering*

**Ing. Josef Steinberger**

# Text Summarization within the LSA Framework

*Supervisor:* Ing. Doc. Karel Ježek, CSc.

*Date of state doctoral exam:* June 30, 2005

*Date of thesis consignation:* January 26, 2007

Pilsen, 2007

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

# Disertační práce

k získání akademického titulu

*doktor*

v oboru

*Informatika a výpočetní technika*

**Ing. Josef Steinberger**

# Text Summarization within the LSA Framework

*Školitel:* Ing. Doc. Karel Ježek, CSc.

*Datum státní závěrečné zkoušky:* 30. června 2005

*Datum odevzdání práce:* 26. ledna 2007

V Plzni, 2007

# Prohlášení

Předkládám tímto k posouzení a obhajobě disertační práci zpracovanou na závěr doktorského studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji tímto, ze tuto práci jsem vypracoval samostatně, s použitím odborné literatury a dostupných pramenů uvedených v seznamu, jenž je součástí této práce.

V Plzni dne 26. ledna 2007 Ing. Josef Steinberger

# Abstract

This thesis deals with the development of a new text summarization method that uses the latent semantic analysis (LSA). The language-independent analysis is able to capture interrelationships among terms, so that we can obtain a representation of document topics. This feature is exploited by the proposed summarization approach. The method originally combines both lexical and anaphoric information. Moreover, anaphora resolution is employed in correcting false references in the summary. Then, I describe a new sentence compression algorithm that takes advantage from the LSA properties. Next, I created a method which evaluates the similarity of main topics of an original text and its summary, motivated by the ability of LSA to extract topics of a text. Using summaries in multilingual searching system MUSE led to better user orientation in the retrieved texts and to faster searching when summaries were indexed instead of full texts.

# Abstrakt

Disertační práce se zabývá vývojem nové metody sumarizace textů, která využívá latentní sémantickou analýzu (LSA). Tato analýza, která je nezávislá na jazyku textu, je schopna zachytit vzájemné vztahy mezi termy. Získáme tak reprezentaci témat dokumentu. Textové jednotky (např. věty), které obsahují nejvýznamnější témata, jsou potom extrahovány. Sumarizační metoda byla dále originálně vylepšena přidáním anaforických informací, které jsou také zužitkovány při opravě chybných anafor v extraktu. Dále popisuji nový algoritmus komprese souvětí, který je založen také na LSA. Fakt, že díky LSA obdržíme témata textu motivoval k vytvoření metody hodnocení kvality extraktů, která měří podobnost hlavních témat extraktu a původního textu. Praktické užití extraktů ve vícejazyčném vyhledávacím systému MUSE ukázalo lepší orientaci uživatele ve vyhledaných textech a rychlejší vyhledávání, když byly extrakty indexovány místo plných textů.

# Contents

# List of Figures

# List of Tables

# Acknowledgement

First of all, I would like to thank doc. Karel Ježek, my thesis supervisor, for his leadership during my PhD studies.

Next, special thanks go to my wife and the rest of the family who supported me both physically and mentally.

Then, I have to thank my colleagues from the University of Essex, Massimo Poesio and Mijail A. Kabadjov, who collaborated with me and gave me so many great advices.

Furthermore, I thank Michal Toman for executing the MUSE experiments.

Last, I would like to thank my colleagues from Text-mining group for a friendly working atmosphere.

# Chapter 1

# Introduction

The main objective of this Ph.D. thesis is the developement of a new text summarization method that would take advantage of latent semantic analysis strengths. The proposed method combines both lexical and anaphoric information.

In this chapter I will describe the motivation of my research and the aims of the dissertation work. In the end the structure of the work will be introduced.

## 1.1 Motivation

Research and development in automatic text summarization has been growing in importance with the rapid growth of on-line information services. The aim of automatic text summarization is to take a source text and present the most important content in a condensed form in a manner sensitive to the needs of the user and the task.

Summarization is a hard problem of natural language processing (NLP) because, to do it properly, one has to really understand the point of a text. This requires semantic analysis, discourse processing, and inferential interpretation (grouping of the content using world knowledge). The last step,

especially, is complex, because systems without a great deal of world knowledge simply cannot do it. Therefore, attempts of performing true abstraction, creating abstracts as summaries, have not been very successful so far. Fortunately, an approximation called extraction is more feasible today. To create an extract, a system needs simply to identify the most important/central topic(s) of the text, and return them to the reader. Although the summary is not necessarily coherent, the reader can form an opinion of the content of the original. Most automated summarization systems today produce extracts only.

Latent semantic analysis (LSA - [29]) is a technique for extracting the 'hidden' dimensions of the semantic representation of terms, sentences, or documents, on the basis of their contextual use. It has been extensively used for various NLP applications including information retrieval [7] or text segmentation [12]. The fact that LSA can identify the most important topics induce a possibility to use it for text summarization as well. However, the number of topics that will be included in the summary is a crucial decision. We should not omit an important topic but on the other hand unimportant ones should be ignored in summary generation. I simply follow the approach to extract the most important sentences about the most important topics. And because the analysis is language-independent the summarization system can be used in multilingual environment.

Discourse structure is an essential indicator for sentence quality judgements [9]. Methods that rely only on lexical information to identify the main topics of a text, such as the word-based LSA, can only capture part of the information about which entities are frequently repeated in the text. An anaphora resolution system can identify repeatedly mentioned entities even when different forms of mention are used. This motivates to improve the basic lexical LSA topic determination.

The recent direction of the summarization field is going beyond sentence extraction. Sentence compression is a task that could move automatic sum-

marization closer to what humans produce. A long sentence is likely to be favored because it has better chance to contain more topic words. However, usually it tends to contain some clauses that are unimportant for the summary. Therefore, its simplification would make the summary more concise. Another recent focus in summarization is to apply coreference resolution and to use the information to improve the summary readability.

The evaluation of the summary quality is a very important task. The fact that we can obtain topics of the text by the LSA motivates to create the summary evaluation method whose idea would be that the summary should retain the main topics of the full text.

## 1.2 Aims of the Dissertation Work

The latent semantic analysis is able to capture interrelationships among terms, so that terms and sentences can be clustered on a semantic basis rather than on the basis of words only. Thus, we can obtain a representation of document topics. The main goal of my work is to find a way how to use this powerful feature for summarization.

The particular aims:

1. Not all terms in the text contain the same important information. I will present an analysis of different weighting schemes for LSA-based summarization.

2. Further, we have to deal with the level of dimensionality reduction. I will find a way how to reduce the number of dimensions on the basis of the required size of the summary. If we take too few dimensions/topics in the summary, we may lose topics which are important from a summarization point of view, but if we take too many, we end up including less important ones.

3. The basic lexical summarization can be enhanced by the knowledge of anaphors. This information can be used to determine the text topics more accurately. The task will be to find out if the integration of an automatic anaphora resolution system yields an improvement despite the system's incorrect interpretations. I will show that not all ways of the integration are equally good. Anaphora resolution would not be employed only in improving sentence selection, however, it can also correct false references in the summary. To find an automatic method that would correct the summary references is another goal of the thesis.

4. Then, I will go beyond the sentence extraction and I will design a simple sentence compression algorithm that would remove unimportant clauses from the full sentence.

5. I will analyse a possilibity of summary quality evaluation through LSA.

6. The automatic summaries will be used in multilingual searching system (MUSE). I will discuss searching in summaries that would boost the retrieval response time. In addition, the summaries will be presented for better and faster user orientation.

## 1.3 Structure of the Work

The rest of this thesis is organized as follows:

The next chapter covers related work in text summarization. I discuss there the aspects that affect the process of summarization, approaches to sentence extraction, approaches that go beyond sentence extraction and at the end standard evaluation measures.

The third chapter describes the model of latent semantic analysis and the new sentence extraction method based on it. In the fourth part the summarizer is originally extended by anaphoric knowledge. Moreover, I propose a new approach that is able to correct anaphoric references in the summary

(summary reference checker). The next chapter enriches the summarizer by the novel sentence compression algorithm.

Chapter 6 discusses possitive and negative assets of the new evaluation method based on LSA. The practical usage of summarization is showed in the seventh chapter. I describe there main modules of a prototype multilingual searching and summarization system. After all, I conclude all and I outline my future research.

# Chapter 2

# Related Work

Summarization has traditionally been decomposed into three phases [1, 56]:

- analyzing the input text to obtain text representation,
- transforming it into summary representation, and
- synthesizing an appropriate output form to generate the summary text.

Effective summarizing requires an explicit and detailed analysis of context factors. I will follow [56], who distinguished three main aspects that affect the process of text summarization: input, purpose and output.

**Input aspects:** The way a summary can be obtained is crucially determined by the features of the text to be summarized. Here are some aspects of input relevant to the task of text summarization:

- *Document Structure*: Besides textual content heterogenous documental information can be found in a source document (e.g. labels that mark headers, chapters, sections, lists, tables, etc.). If it is well systematized and exploited, this information can be used to analyze the document.
- *Domain*: Domain-sensitive systems are only capable of obtaining summaries of texts that belong to a pre-determined domain, with varying

degrees of portability. The restriction to a certain domain is usually compensated by the fact that specialized systems can apply knowledge intensive techniques which are only feasible in controlled domains.

- *Scale*: Different summarizing strategies have to be adopted to handle different text lengths. In the case of news articles, sentences or even clauses are usually considered the minimal meaning units, whereas for longer documents like reports or books, paragraphs seem a more adequate unit of meaning.

- *Unit*: The input to the summarization process can be a single document or multiple documents.

- *Language*: Systems can be language independent, exploiting characteristics of documents that hold cross-lingualistically or else their architecture can be determined by the features of a concrete language.

**Purpose Aspects**: Summarization systems can perform general summarization or else they can be embedded in larger systems, as an intermediate step for other NLP task (e.g., Information Retrieval, Document Classification, etc.). Task-driven summarization presents the advantage that systems can be evaluated with respect to the improvement they introduce in the final task they are applied to.

- *Audience*: In case a user profile is accessible summaries can be adapted to the needs of specific users (e.g., the user's prior knowledge on a determined subject). Background summaries assume that the reader's prior knowledge is poor, and so intensive information is supplied, while just-the news are those kinds of summaries conveying only the newest information on an already known subject.

- *Usage*: Summaries can be sensitive to determined uses, retrieving the source text, previewing the text, refreshing the memory of an already read text, etc.

**Output Aspects**:

- *Content*: A summary may try to represent all relevant features of a source text or it may focus on some specific ones, which can be determined by queries, subjects, etc. Generic summaries are text-driven, while user-focused (or query-driven) ones rely on a specification of the user's information need, like a question or keywords.

- *Style*: A summary can be informative - if it covers the topics of in the source text, indicative - if it provides a brief survey of the topics addressed in the original, aggregative - if it supplies information non present in the source text that completes some of its information or elicits some hidden information or critical - if it provides additional valorization of the summarized text.

- *Production Process*: The resulting summary can be an extract - if it is composed by literal fragments of text, or an abstract - if it is generated.

- *Surrogate*: Summaries can stand in place of the source as a surrogate, or they can be linked to the source, or even be presented in the context of the source (e.g., by highlighting source text).

- *Length*: The targeted length of the summary crucially affects the informativeness of the final result. This length can be determined by a compression rate - a ratio of the summary length with respect to the length of the original text.

As for input factors I am concerned with single-document summarization. I do not use any structure or domain information because the aim is to create a domain-independent summarizer. The scale and language factors are influenced by available resources. The method is not dependent on a specific scale or language but the texts used for evaluation are mostly news articles in English. However, I experimented with scientific articles and Czech language as well. The main goal is to do general summarization that means I do not

assume any prior knowledge about the target users and the summaries should be used for previewing the text. The exception is the usage of summaries in the multilingual searching system where summaries whould be sensitive to retrieving the source text. As for output factors the focus is on text-driven informative and extractive summaries with various lenght that would be linked to the source.

## 2.1 Sentence Extraction

A lot of text summarization approaches can be found in literature. Most of them are based on sentence extraction. In this shallow approach, statistical heuristics are used to identify the most salient sentences of a text. It is a low-cost approach compared to more knowledge-intensive deeper approaches which require additional knowledge bases such as ontologies or linguistic knowledge. I classify and discuss here some approaches to sentence exraction.

### 2.1.1 Surface Level Approaches

The oldest approaches use surface level indicators to decide what parts of a text are important. The first sentence extraction algorithm was developed in 1958 [32]. It used term frequencies to measure sentence relevance. The idea was that when writing about a given topic, a writer will repeat certain words as the text is developed. Thus, term relevance is considered proportional to its in-document frequency. The term frequencies are later used to score and select sentences for the summary. Other good indicators of sentence relevance are the position of a sentence within the document [4], the presence of title words or certain *cue-words* (i.e., words like "important" or "relevant"). In [15] it was demonstrated that the combination of the presence of cue-words, title words and the position of a sentence produces the most similar extracts to abstracts written by a human.

### 2.1.2 Corpus-based Approaches

It is likely that documents in a certain field share common terms in that field that do not carry salient information. Their relevance should be reduced. [54] showed that the relevance of a term in the document is inversely proportional to the number of documents in the corpus containing the term. The normalized formula for term relevance is given by $tf_i * idf_i$, where $tf_i$ is the frequency of term $i$ in the document and $idf_i$ is the inverted document frequency. Sentence scores can then be computed in a number of ways. For instance, they can be measured by the sum of term scores in the sentence.

In [18] an alternative to measuring term relevance was proposed. The authors presented *concept relevance* which can be determined using WordNet. The occurrence of the concept "bicycle" is counted when the word "bicycle" is found as well as when, for instance, "bike", "pedal", or "brake" are found.

In [28] a Bayesian classifier that computes the probability that a sentence in a source document should be included in a summary was implemented. In order to train the classifier the authors used a corpus of 188 pairs of full documents/summaries from scientific fields. They used, for example, the following features: sentence length, phrase structure, in-paragraph position, word frequency, uppercase words. The probability that a sentence should be selected is computed by the Bayesian formula.

### 2.1.3 Cohesion-based Approaches

Extractive methods can fail to capture the relations between concepts in a text. Anaphoric expressions[1] that refer back to events and entities in the text need their antecedents in order to be understood. The summary can become difficult to understand if a sentence that contains an anaphoric link is extracted without the previous context. Text cohesion comprises relations

[1]Anaphoric expression is a word or phrase which refers back to some previously expressed word or phrase or meaning (typically, pronouns such as herself, himself, he, she).

between expressions which determine the text connectivity. Cohesive properties of the text have been explored by different summarization approaches.

In [2] a method called *Lexical chains* was introduced. It uses the WordNet database for determining cohesive relations (i.e., repetition, synonymy, antonymy, hypernymy, and holonymy) between terms. The chains are then composed by related terms. Their scores are determined on the basis of the number and type of relations in the chain. Sentences where the strongest chains are highly concentrated are selected for the summary. A similar method where sentences are scored according to the objects they mention was presented in [9]. The objects are identified by a *co-reference resolution system*. Co-reference resolution is the process of determining whether two expressions in natural language refer to the same entity in the world. Sentences where the frequently mentioned objects occur go to the summary.

### 2.1.4 Rhetoric-based Approaches

*Rhetorical Structure Theory* (RST) is a theory about text organization. It consists of a number of rhetorical relations that tie together text units. The relations connect together a *nucleus* - central to the writer's goal, and a *satellite* - less central material. Finally, a tree-like representation is composed. Then the text units have to be extracted for the summary. In [43] sentences are penalized according to their rhetorical role in the tree. A weight of 1 is given to satellite units and a weight of 0 is given to nuclei units. The final score of a sentence is given by the sum of weights from the root of the tree to the sentence. In [34], each parent node identifies its nuclear children as salient. The children are promoted to the parent level. The process is recursive down the tree. The score of a unit is given by the level it obtained after promotion.

### 2.1.5 Graph-based Approaches

Graph-Based algorithms, such as HITS [24] or Google's PageRank [8] have been successfully used in citation analysis, social networks, and in the analysis of the link-structure of the Web. In graph-based ranking algorithms, the importance of a vertex within the graph is recursively computed from the entire graph. In [36] the graph-based model was applied to natural language processing, resulting in TextRank. Further, the graph-based ranking algorithm was applied to summarization [37]. A graph is constructed by adding a vertex for each sentence in the text, and edges between vertices are established using sentence inter-connections. These connections are defined using a similarity relation, where similarity is measured as a function of content overlap. The overlap of two sentences can be determined simply as the number of common tokens between lexical representations of two sentences. After the ranking algorithm is run on the graph, sentences are sorted in the reverse order of their score, and the top ranked sentences are included in the summary.

## 2.2 Beyond Sentence Extraction

There is a big gap between the summaries produced by current automatic summarizers and the abstracts written by human professionals. One reason is that systems cannot always correctly identify the important topics of an article. Another factor is that most summarizers rely on extracting key sentences or paragraphs. However, if the extracted sentences are disconnected in the original article and they are strung together in the summary, the result can be incoherent and sometimes even misleading. Lately, some non-sentence-extractive summarization methods have started to develop. Instead of reproducing full sentences from the text, these methods either compress the sentences [21, 25, 57, 60], or re-generate new sentences from scratch [35]. In [22] a *Cut-and-paste strategy* was proposed. The authors have iden-

tified six editing operations in human abstracting: (i) sentence reduction; (ii) sentence combination; (iii) syntactic transformation; (iv) lexical paraphrasing; (v) generalization and specification; and (vi) reordering. Summaries produced this way resemble the human summarization process more than extraction does. However, if large quantities of text need to be summarized, sentence extraction is a more efficient method, and it is robust towards all kinds of input, even slightly ungrammatical ones.

## 2.3 Evaluation Measures

I follow the taxonomy of summary evaluation measures in [50] (see figure 2.1). *Text quality* is often assessed by human annotators. They assign a value from a predefined scale to each summary. The main approach for summary quality determination is the intrinsic *content evaluation* which is often done by comparison with an ideal summary (written by a human). For sentence extracts, it is often measured by *co-selection*. It finds out how many ideal sentences the automatic summary contains. *Content-based measures* compare the actual words in a sentence, rather than the entire sentence. Their advantage is that they can compare both human and automatic extracts with human abstracts that contain newly written sentences. Another significant group are *task-based methods*. They measure the performance of using the summaries for a certain task.

### 2.3.1 Text Quality Measures

There are several aspects of text (linguistic) quality:

- **grammaticality** - the text should not contain non-textual items (i.e., markers) or punctuation errors or incorrect words.
- **non-redundancy** - the text should not contain redundant information.

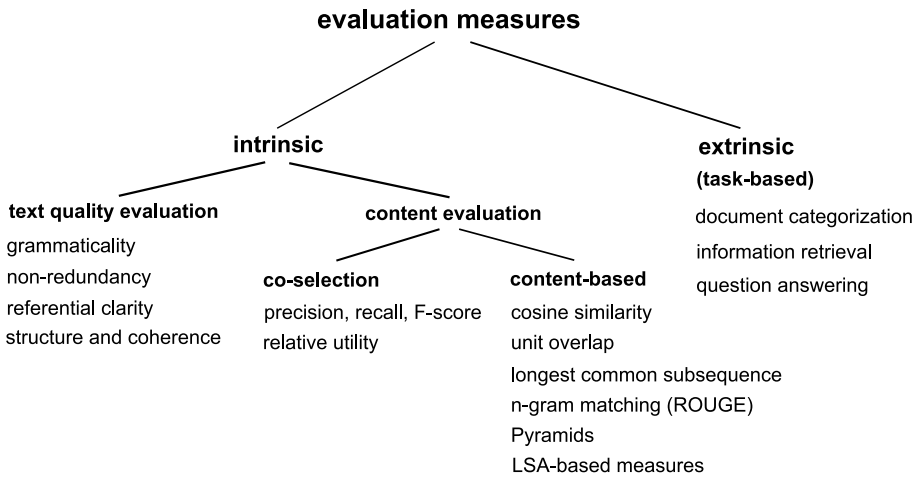


Figure 2.1: The taxonomy of summary evaluation measures.

- **reference clarity** - the nouns and pronouns should be clearly referred to in the summary. For example, the pronoun *he* has to mean somebody in the context of the summary.
- **coherence and structure** - the summary should have good structure and the sentences should be coherent.

This cannot be done automatically. The annotators mostly assign marks (i.e., from A - very good - to E - very poor - at DUC 2005) to each summary.

### 2.3.2 Co-Selection Measures

#### Precision, Recall and F-score

The main evaluation metrics of co-selection are precision, recall and F-score. *Precision* (P) is the number of sentences occurring in both system and ideal summaries divided by the number of sentences in the system summary. *Recall* (R) is the number of sentences occurring in both system and ideal summaries divided by the number of sentences in the ideal summary. F-score is a com-

posite measure that combines precision and recall. The basic way how to compute the F-score is to count a harmonic average of precision and recall:

$$F = \frac{2 * P * R}{P + R}. \tag{2.1}$$

Below is a more complex formula for measuring the F-score:

$$F = \frac{(\beta^2 + 1) * P * R}{\beta^2 * P + R}, \tag{2.2}$$

where $\beta$ is a weighting factor that favours precision when $\beta > 1$ and favours recall when $\beta < 1$.

**Relative Utility**

The main problem with P&R is that human judges often disagree on what the top $n\%$ most important sentences are in a document. Using P&R creates the possibility that two equally good extracts are judged very differently. Suppose that a manual summary contains sentences [1 2] from a document. Suppose also that two systems, A and B, produce summaries consisting of sentences [1 2] and [1 3], respectively. Using P&R, system A will be ranked much higher than system B. It is quite possible that sentences 2 and 3 are equally important, in which case the two systems should get the same score.

To address the problem with precision and recall, the *relative utility* (RU) measure was introduced [49]. With RU, the model summary represents all sentences of the input document with confidence values for their inclusion in the summary. For example, a document with five sentences [1 2 3 4 5] is represented as [1/5 2/4 3/4 4/1 5/2]. The second number in each pair indicates the degree to which the given sentence should be part of the summary according to a human judge. This number is called the *utility* of the sentence. It depends on the input document, the summary length, and the judge. In the example, the system that selects sentences [1 2] will not get a higher score than a system that chooses sentences [1 3] because both

summaries [1 2] and [1 3] carry the same number of utility points (5+4). Given that no other combination of two sentences carries a higher utility, both systems [1 2] and [1 3] produce optimal extracts. To compute relative utility, a number of judges, ($N \geq 1$) are asked to assign utility scores to all $n$ sentences in a document. The top $e$ sentences according to utility score[2] are then called a sentence extract of size $e$. We can then define the following system performance measure:

$$RU = \frac{\sum_{j=1}^{n} \delta_j \sum_{i=1}^{N} u_{ij}}{\sum_{j=1}^{n} \epsilon_j \sum_{i=1}^{N} u_{ij}}, \tag{2.3}$$

where $u_{ij}$ is a utility score of sentence $j$ from annotator $i$, $\epsilon_j$ is 1 for the top $e$ sentences according to the sum of utility scores from all judges, otherwise its value is 0, and $\delta_j$ is equal to 1 for the top $e$ sentences extracted by the system, otherwise its value is 0. For details, see [49].

### 2.3.3 Content-based Measures

Co-selection measures can count as a match only exactly the same sentences. This ignores the fact that two sentences can contain the same information even if they are written differently. Furthermore, summaries written by two different annotators do not in general share identical sentences. In the following example, it is obvious that both headlines, $H_1$ and $H_2$, carry the same meaning and they should somehow count as a match.

$H_1$: "The visit of the president of the Czech Republic to Slovakia"

$H_2$: "The Czech president visited Slovakia"

Whereas co-selection measures cannot do this, content-based similarity measures can.

[2] In the case of ties, an arbitrary but consistent mechanism is used to decide which sentences should be included in the summary.

### Cosine Similarity

A basic content-based similarity measure is Cosine Similarity [54]:

$$cos(X, Y) = \frac{\sum_i x_i \cdot y_i}{\sqrt{\sum_i (x_i)^2} \cdot \sqrt{\sum_i (y_i)^2}}, \tag{2.4}$$

where $X$ and $Y$ are representations of a system summary and its reference document based on the vector space model.

### Unit Overlap

Another similarity measure is Unit Overlap [53]:

$$overlap(X, Y) = \frac{\|X \cap Y\|}{\|X\| + \|Y\| - \|X \cap Y\|}, \tag{2.5}$$

where $X$ and $Y$ are representations based on sets of words or lemmas. $\|X\|$ is the size of set $X$.

### Longest Common Subsequence

The third content-based measure is called Longest Common Subsequence (LCS) [50]:

$$lcs(X, Y) = \frac{length(X) + length(Y) - edit_{di}(X, Y)}{2}, \tag{2.6}$$

where $X$ and $Y$ are representations based on sequences of words or lemmas, lcs(X,Y) is the length of the longest common subsequence between $X$ and $Y$, $length(X)$ is the length of the string $X$, and $edit_{di}(X, Y)$ is the edit distance of $X$ and $Y$ [50].

### N-gram Co-occurrence Statistics - ROUGE

In the last editions of DUC[3] conferences, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) was used as an automatic evaluation method. The ROUGE family of measures, which are based on the similarity of n-grams[4], was firstly introduced in 2003 [30].

Suppose a number of annotators created reference summaries - reference summary set (RSS). The ROUGE-$n$ score of a candidate summary is computed as follows:

$$\text{ROUGE-}n = \frac{\sum_{C \in RSS} \sum_{gram_n \in C} Count_{match}(gram_n)}{\sum_{C \in RSS} \sum_{gram_n \in C} Count(gram_n)}, \tag{2.7}$$

where $Count_{match}(gram_n)$ is the maximum number of $n$-grams co-occurring in a candidate summary and a reference summary and $Count(gram_n)$ is the number of $n$-grams in the reference summary. Notice that the average $n$-gram ROUGE score, ROUGE-$n$, is a recall metric. There are other ROUGE scores, such as ROUGE-L, a longest common subsequence measure (see the previous section), or ROUGE-SU4, a bigram measure that enables at most 4 unigrams inside of bigram components to be skipped [31].

### Pyramids

*The Pyramid method* is a novel semi-automatic evaluation method [42]. Its basic idea is to identify summarization content units (SCUs) that are used for comparison of information in summaries. SCUs emerge from annotation of a corpus of summaries and are not bigger than a clause. The annotation starts with identifying similar sentences and then proceeds with finer grained inspection that can lead to identifying more tightly related subparts. SCUs

[3]The National Institute of Standards and Technology (NIST) initiated the Document Understanding Conference (DUC) series to evaluate automatic text summarization. Its goal is to further the progress in summarization and enable researchers to participate in large-scale experiments.

[4]An n-gram is a subsequence of $n$ words from a given text.

that appear in more manual summaries will get greater weights, so a pyramid will be formed after SCU annotation of manual summaries. At the top of the pyramid there are SCUs that appear in most of the summaries and thus they have the greatest weight. The lower in the pyramid the SCU appears, the lower its weight is because it is contained in fewer summaries. The SCUs in peer summary are then compared against an existing pyramid to evaluate how much information is agreed between the peer summary and manual summary. However, this promising method still requires some annotation work.

### 2.3.4 Task-based Measures

Task-based evaluation methods do not analyze sentences in the summary. They try to measure the prospect of using summaries for a certain task. Various approaches to task-based summarization evaluation can be found in literature. I mention the three most important tasks - document categorization, information retrieval and question answering.

#### Document Categorization

The quality of automatic summaries can be measured by their suitability for surrogating full documents for *categorization*. Here the evaluation seeks to determine whether the generic summary is effective in capturing whatever information in the document is needed to correctly categorize the document. A corpus of documents together with the topics they belong to is needed for this task. Results obtained by categorizing summaries are usually compared to ones obtained by categorizing full documents (an upper bound) or random sentence extracts (lower bound). Categorization can be performed either manually [33] or by a machine classifier [19]. If we use an automatic categorization we must keep in mind that the classifier demonstrates some inherent errors. It is therefore necessary to differentiate between the error

generated by a classifier and one caused by a summarizer. It is often done only by comparing the system performance with the upper and lower bounds.

In SUMMAC evaluation [33], apart from other tasks, 16 participating summarization systems were compared by a manual categorization task. Given a document, which could be a generic summary or a full text source (the subject was not told which), the human subject chose a single category (from five categories, each of which had an associated topic description) to which the document is relevant, or else chose "none of the above".

Precision and recall of categorization are the main evaluation metrics. *Precision* in this context is the number of correct topics assigned to a document divided by the total number of topics assigned to the document. *Recall* is the number of correct topics assigned to a document divided by the total number of topics that should be assigned to the document. The measures go against each other and therefore a composite measure - the F-score - can be used (see the section 2.3.2).

### Information Retrieval

*Information Retrieval* (IR) is another task appropriate for the task-based evaluation of a summary quality. *Relevance correlation* [50] is an IR-based measure for assessing the relative decrease in retrieval performance when moving from full documents to summaries. If a summary captures the main points of a document, then an IR machine indexed on a set of such summaries (instead of a set of the full documents) should produce (almost) as good a result. Moreover, the difference between how well the summaries do and how well the full documents do should serve as a possible measure for the quality of summaries.

Suppose that given query $Q$ and a corpus of documents $D$, a search engine ranks all documents in $D$ according to their relevance to query $Q$. If instead of corpus $D$, the corresponding summaries of all documents are substituted

for the full documents and the resulting corpus of summaries $S$ is ranked by the same retrieval engine for relevance to the query, a different ranking will be obtained. If the summaries are good surrogates for the full documents, then it can be expected that the ranking will be similar. There exist several methods for measuring the similarity of rankings. One such method is Kendall's tau and another is Spearman's rank correlation [55]. However, since search engines produce relevance scores in addition to rankings, we can use a stronger similarity test, linear correlation.

Relevance correlation (RC) is defined as the linear correlation of the relevance scores assigned by the same IR algorithm in different data sets (for details see [50]).

### Question Answering

An extrinsic evaluation of the impact of summarization in a task of *question answering* was carried out in [39]. The authors picked four Graduate Management Admission Test (GMAT) reading comprehension exercises. The exercises were multiple-choice, with a single answer to be selected from answers shown alongside each question. The authors measured how many of the questions the subjects answered correctly under different conditions. Firstly, they were shown the original passages, then an automatically generated summary, furthermore a human abstract created by a professional abstractor instructed to create informative abstracts, and finally, the subjects had to pick the correct answer just from seeing the questions without seeing anything else. The results of answering in the different conditions were then compared.

# Chapter 3

# Applying LSA to Summarization

LSA [29] is a technique for extracting the 'hidden' dimensions of the semantic representation of terms, sentences, or documents, on the basis of their use. It has been extensively used in educational applications such as essay ranking [29], as well as in NLP applications including information retrieval [7] and text segmentation [12].

More recently, a method for using LSA for summarization has been proposed in [16]. This purely lexical approach is the starting point for my own work. The heart of Gong and Liu's method is a document representation developed in two steps. The first step is the creation of a term by sentences matrix $A = [A_1, A_2, \ldots, A_n]$, where each column $A_i$ represents the weighted term-frequency vector of sentence $i$ in the document under consideration[1]. The vector $A_i = [a_{1i}, a_{2i}, \ldots, a_{ni}]^T$ is defined as:

$$a_{ij} = L(t_{ij}) \cdot G(t_{ij}), \tag{3.1}$$

[1] A sentence is usually used to express context in summarization. However, for instance, a context can be represented by a paragraph for longer documents.

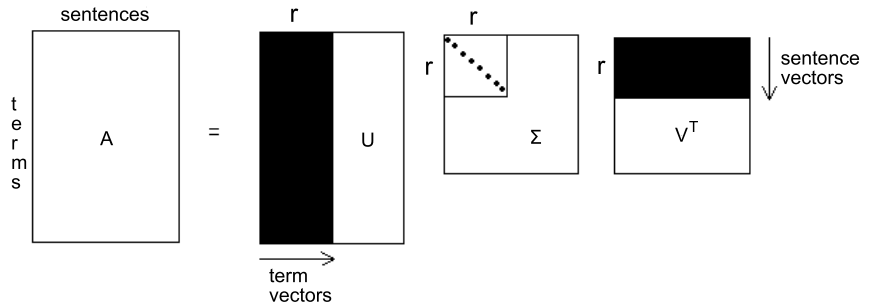


Figure 3.1: Singular Value Decomposition.

where $t_{ij}$ denotes the frequency with which term $j$ occurs in sentence $i$, $L(t_{ij})$ is the local weight for term $j$ in sentence $i$, and $G(t_{ij})$ is the global weight for term $j$ in the whole document. There are many possible weighting schemes. A detailed analysis of finding the best weighting system for summarization can be found in the section 3.2.3.

If there are $m$ terms and $n$ sentences in the document, then we will obtain an $m \times n$ matrix $A$ for the document. The next step is to apply Singular Value Decomposition (SVD) to matrix $A$. The SVD of an $m \times n$ matrix $A$ is defined as:

$$A = U\Sigma V^T \tag{3.2}$$

where $U = [u_{ij}]$ is an $m \times n$ column-orthonormal matrix whose columns are called *left singular vectors*. $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is an $n \times n$ diagonal matrix, whose diagonal elements are non-negative *singular values* sorted in descending order. $V = [v_{ij}]$ is an $n \times n$ orthonormal matrix, whose columns are called *right singular vectors*.

From a mathematical point of view, SVD derives a mapping between the $m$-dimensional space specified by the weighted term-frequency vectors and the $r$-dimensional singular vector space.

From an NLP perspective, what SVD does is to derive the *latent semantic*

*structure* of the document represented by matrix $A$: i.e. a breakdown of the original document into $r$ linearly-independent base vectors which express the main 'topics' of the document.

SVD can capture interrelationships among terms, so that terms and sentences can be clustered on a 'semantic' basis rather than on the basis of words only. Furthermore, as demonstrated in [7], if a word combination pattern is salient and recurring in document, this pattern will be captured and represented by one of the singular vectors. The magnitude of the corresponding singular value indicates the importance degree of this pattern within the document. Any sentences containing this word combination pattern will be projected along this singular vector, and the sentence that best represents this pattern will have the largest index value with this vector. Assuming that each particular word combination pattern describes a certain topic in the document, each *triplet* (left singular vector, singular value, and right singular vector) can be viewed as representing such a topic [13], the magnitude of its singular value representing the degree of importance of this topic.

## 3.1 Sentence Selection Based on LSA

### 3.1.1 Gong and Liu's Approach

The summarization method proposed in [16] uses the representation of a document thus obtained to choose the sentences to go in the summary on the basis of the relative importance of the 'topics' they mention, described by the matrix $V^T$. The summarization algorithm simply chooses for each 'topic' the most important sentence for that topic: i.e., the $k^{th}$ sentence chosen is the one with the largest index value in the $k^{th}$ right singular vector in matrix $V^T$.

The main drawback of Gong and Liu's method is that when $l$ sentences are extracted the top $l$ topics are treated as equally important. As a result, a

summary may include sentences about 'topics' which are not particularly important.

### 3.1.2 My Approach - Length Strategy

In [13] it was proved that the statistical significance of each LSA dimension (i.e., topic) is approximately the square of its singular value. I exploited this result by changing the selection criterion to include in the summary the sentences whose vectorial representation in the matrix $B = \Sigma^2 \cdot V^T$ has the greatest 'length', instead of the sentences containing the highest index value for each 'topic'. Intuitively, the idea is to choose the sentences with greatest combined weight across all topics, possibly including more than one sentence about an important topic, rather than always choosing one sentence for each topic as done by Gong and Liu. More formally: after computing the SVD of a term by sentences matrix, we compute marix $B$:

$$B = \begin{pmatrix} v_{1,1}\sigma_1^2 & v_{1,2}\sigma_1^2 & \dots & v_{1,n}\sigma_1^2 \\ v_{2,1}\sigma_2^2 & v_{2,2}\sigma_2^2 & \dots & v_{2,n}\sigma_2^2 \\ \dots & \dots & \dots & \dots \\ v_{r,1}\sigma_r^2 & v_{r,2}\sigma_r^2 & \dots & v_{r,n}\sigma_r^2 \end{pmatrix}. \tag{3.3}$$

Then, we measure the length $s_k$ of each sentence vector in $B$:

$$s_k = \sqrt{\sum_{i=1}^{r} b_{i,k}^2}, \tag{3.4}$$

where $s_k$ is the length of the vector of $k$'th sentence in the modified latent vector space, and its significance score for summarization too. We then include in the summary the sentences with the highest values in vector $s$. I demonstrate that this modification results in a significant improvement over Gong and Liu's method (see section 3.2.4).

### Dimensionality reduction

The length strategy still requires a method for deciding how many LSA dimensions/topics to include in the latent space and therefore in the summary. If we take too few, we may lose topics which are important from a summarization point of view. But if we take too many, we end up including less important topics.

When we perform SVD on an $m$ x $n$ matrix, we can view the new dimensions as some sort of pseudo sentences: linear combinations of the original terms (left singular vectors), sorted according to their significance within the document. From a summarization point of view, the number of extracted sentences is dependent on the summary ratio. We know what percentage of the full text the summary should be: part of the input to the summarizer is that a $p\%$ summary is needed. (The length is usually measured in the number of words, but there are other possibilities.) If the pseudo sentences were real sentences that a reader could interpret, we could simply extract the top $r$ pseudo sentences, where $r = p/100 * n$. However, because the linear combinations of terms are not really readable sentences, we use the above sentence selection algorithm to extract the actual sentences that 'overlap the most' in terms of vector length with top $r$ pseudo sentences. In addition, the algorithm takes into account the significance of each dimension by multiplying the matrix $V^T$ by $\Sigma^2$.

The summarizer can thus automatically determine the number of significant dimensions dependent on the summarization ratio. The larger the summary (measured in the percentage of the full text), the more topics are considered important in the process of summary creation. And because we know the contribution of each topic from the square of its singular value we can measure how much information is considered important by the dimensionality reduction approach for each full text percentage. Figure 3.2 shows the logarithmic dependency between summary ratio and sum of relative significances

of $r$ most important dimensions[2]: for instance, a 10% summary contains the sentences that best cover 40% of document information, whereas a 30% summary will contain the sentences that most closely include 70% of document information.

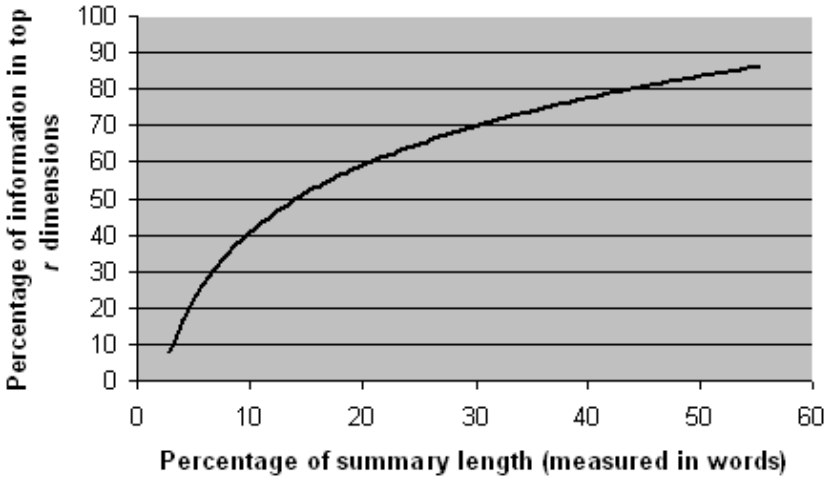


Figure 3.2: The dependency of the sum of significances of $r$ most important dimensions on the summary length. DUC-2002 data were used to create the curve.

### 3.1.3 Other LSA Approaches

In [41] an LSA-based summarization of meeting recordings was presented. The authors followed the Gong and Liu's approach, but rather than extracting the best sentence for each topic, $n$ best sentences were extracted, with $n$ determined by the corresponding singular values from matrix $\Sigma$. The number

[2]Suppose, for example, we have singular values [10, 7, 5, ...], that their significances (squares of singular values) are [100, 49, 25, ...], and that the total significance is 500 (sum of all singular value squares). Then the relative significances are [20%, 9.8%, 5%, ...]: i.e., the first dimension captures 20% of the information in the original document. Thus, when the latent space contains 30 dimensions in total and summary ratio is 10% THEN $r$ is set to 3. The sum of the relative significances of the three most important dimensions is 34.8%.

of sentences in the summary that will come from the first topic is determined by the percentage that the largest singular value represents out of the sum of all singular values, and so on for each topic. Thus, dimensionality reduction is no longer tied to summary length and more than one sentence per topic can be chosen.

Another summarization method that uses LSA was proposed in [65]. It is a mixture of graph-based and LSA-based approaches. After performing SVD on the word-by-sentence matrix and reducing the dimensionality of the latent space, they reconstruct the corresponding matrix $A' = U'\Sigma'V'^T$.[3] Each column of $A'$ denotes the semantic sentence representation. These sentence representations are then used, instead of a keyword-based frequency vector, for the creation of a text relationship map to represent the structure of a document. A ranking algorithm is then applied to the resulting map (see section 2.1.5).

## 3.2 ROUGE Evaluation over DUC 2002 data

In order to assess the quality of the proposed summarization method I used the DUC 2002 corpus and the ROUGE measure, which would make it easier to contrast the results with those published in the literature.

### 3.2.1 The DUC 2002 Corpus

In 2002 DUC (see section 2.3.3) included a single-document summarization task, in which 13 systems participated [69]. 2002 is the last version of DUC that included single-document summarization evaluation of informative summaries. Later DUC editions (2003 and 2004) contained a single-document summarization task as well, however only very short summaries (75 Bytes) were analyzed. However, I am not focused on producing headline-length

[3] $U'$, resp. $\Sigma'$, $V'^T$, $A'$, denotes matrix $U$, resp. $\Sigma$, $V^T$, $A$, reduced to $r$ dimensions.

summaries. The test corpus used for the task contains 567 documents from different sources; 10 assessors were used to provide for each document two 100-word human summaries. In addition to the results of the 13 participating systems, the DUC organizers also distributed baseline summaries (the first 100 words of a document). The coverage of all the summaries was assessed by humans.

### 3.2.2 The ROUGE Evaluation Metric

In DUC-2002, the SEE (Summary Evaluation Environment - [70]) was used, but in later editions of the initiative the ROUGE measure was introduced [30], which is now standard. I used ROUGE to compare my system with those that participated in DUC. A detailed description of ROUGE computation can be found in the section 2.3.3.

ROUGE is actually a family of metrics; however, different ROUGE scores correlate in different ways with human assessments. As shown in Table 3.1, there is a strong correlation between humans and ROUGE-1 (and ROUGE-L) when we include all summarizers including human ones.

| Score | Correlation |
|---|---|
| **ROUGE-1** | 0.92465 |
| **ROUGE-2** | 0.80044 |
| **ROUGE-SU4** | 0.78412 |
| **ROUGE-L** | 0.92269 |

Table 3.1: Correlations between ROUGE scores and human assessments (all summarizers including human ones are included).

On the other hand, when we take only system summarizers - Table 3.2, ROUGE-2 shows the highest correlation.

For that reason, I do not compare the systems only from the angle of a specific

| Score | Correlation |
|---|---|
| **ROUGE-1** | 0.90317 |
| **ROUGE-2** | 0.96119 |
| **ROUGE-SU4** | 0.93897 |
| **ROUGE-L** | 0.91143 |

Table 3.2: Correlations between ROUGE scores and human assessments (only system extractive summarizers are included).

ROUGE metric but I use results of all main ROUGE metrics to determine significance results.[4]

### 3.2.3 Finding the Best Weighting System

I studied the influence of different weighting schemes on the summarization performance. As shown by equation 3.1, given a term $j$ and sentence $i$, its weighting scheme is defined by two parts: local weighting $L(t_{ij})$ and global weighting $G(t_{ij})$. Local weighting $L(t_{ij})$ has the following four possible alternatives [14]:

- Frequency weight (FQ in short): $L(t_{ij}) = tf_{ij}$, where $tf_{ij}$ is the number of times term $j$ occurs in sentence $i$.
- Binary weight (BI): $L(t_{ij}) = 1$, if term $j$ appears at least once in sentence $i$; $L(t_{ij}) = 0$, otherwise.
- Augmented weight (AU): $L(t_{ij}) = 0.5 + 0.5 * (tf_{ij}/tfmax_i)$, where $tfmax_i$ is the frequency of the most frequently occurring term in the sentence.
- Logarithm weight (LO): $L(t_{ij}) = \log(1 + tf_{ij})$.

[4] I should point out however that in DUC 2005, ROUGE-2 and ROUGE-SU4 were used.

Global weighting $G(t_{ij})$ has the following four possible alternatives:

- No weight (NW): $G(t_{ij}) = 1$ for any term $j$.
- Inverse document frequency (IDF): $G(t_{ij}) = \log(N/n_j) + 1$, where $N$ is the total number of sentences in the document, and $n_j$ is the number of sentences that contain term $j$.
- GFIDF (GF): $G(t_{ij}) = \frac{gf_j}{sf_j}$, where the sentence frequency $sf_j$ is the number of sentences in which term $j$ occurs, and the global frequency $gf_j$ is the total number of times that term $j$ occurs in the whole document.
- Entropy frequency (EN): $G(t_{ij}) = 1 - \sum_i \frac{p_{ij} \log(p_{ij})}{log(nsent)}$, where $p_{ij} = tf_{ij}/gf_j$ and $nsent$ is the number of sentences in the document.

All combinations of these local and global weights for the new LSA-based summarization method are compared in the figures 3.3 (ROUGE-1) and 3.4 (ROUGE-2). We can observe that the best performing local weight was the binary weight and the best performing global weight was the entropy weight. Thus, for the local weight it is not important how many times a term occurs in a sentence. And the global weight of the term increases with the ratio of the frequency of the term in the sentence and frequency of the term in the document. The combination of the binary local weight and the entropy global weight is used throughout the rest of the thesis.

### 3.2.4 Comparison with DUC Participating Systems

I show in Table 3.3 the ROUGE scores[5] of two LSA summarizers - GLLSA (Gong and Liu's approach) and LELSA (Length strategy, my approach); and

[5]All system summaries were truncated to 100 words as traditionally done in DUC. ROUGE version and settings: `ROUGEeval-1.4.2.pl -c 95 -m -n 2 -l 100 -s -2 4 -a duc.xml`.

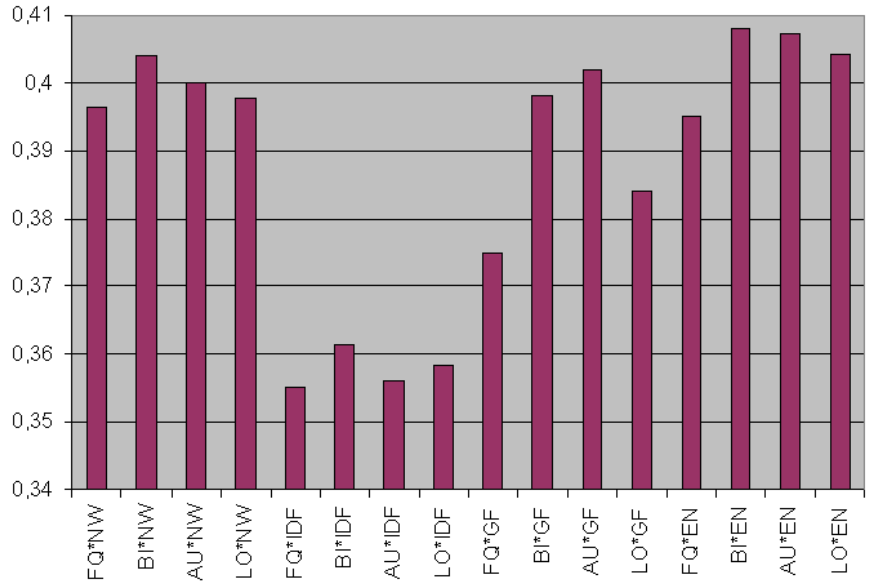


Figure 3.3: The comparison of different weighting systems for LSA - ROUGE-1. The meaning of the letters is as follows: Local weight * Global weight.

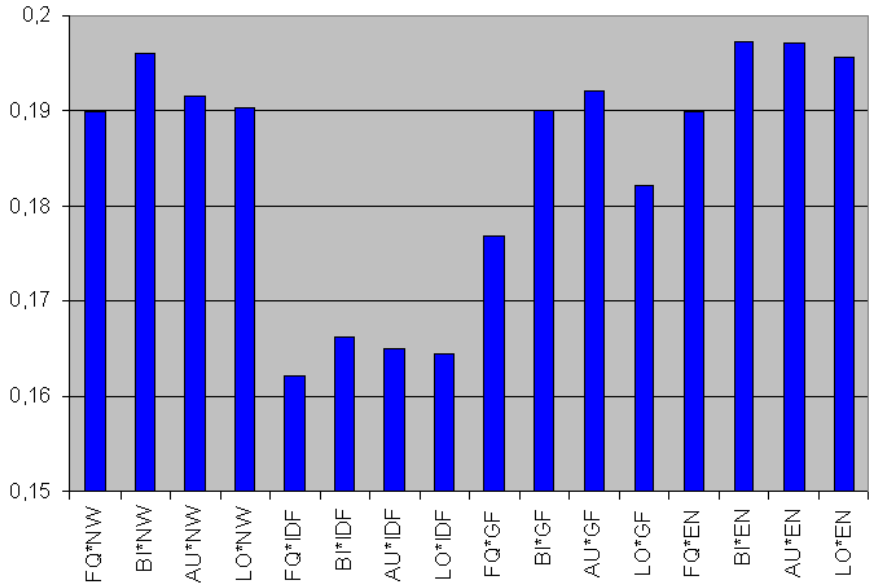


Figure 3.4: The comparison of different weighting systems for LSA - ROUGE-2. The meaning of the letters is as follows: Local weight * Global weight.

of the 13 systems which participated in DUC-2002[6]. I also list a baseline and a random summarizer (the lowest baseline): 17 summarizers in total.

Table 3.4 shows a multiple comparison of ROUGE scores between systems. Systems not sharing a common letter are significantly different (at the 95% confidence level).

Example: The average performance of System 1 is 0.44 and its 95% confidence interval (CI) is (0.39 - 0.49). Similarly, the average of System 2 is 0.40 and CI is (0.35 - 0.45). The last System 3: average is 0.32, CI is (0.27 - 0.37). We can say that System 1 is significantly better than System 3, but other differences are not statistically significant. To show the significances we assign to System 1 letters $A$ and $B$, to System 2 $B$ and $C$, and to System 3 $C$. Systems 1 and 2 share $B$ and Systems 2 and 3 share $C$, and thus, they are not significantly different. Systems 1 and 3 do not share the same letter and so we can say that system 1 is significantly better than System 3 with 95% confidence.

The first result highlighted by these tables is that the LELSA summarizer is state of the art. Its performance is significantly worse only than that the best system in DUC 2002, system 28, in ROUGE-1, ROUGE-2 and ROUGE-SU4, and significantly better than that of 9 in ROUGE-1, 7 in ROUGE-2, 7 in ROUGE-SU4 and 10 in ROUGE-L of the systems that participated in that competition. The second result is that the LELSA system significantly outperforms Gong and Liu's LSA approach (GLLSA).

## 3.3 SVD and Complexity

The crucial complexity part of the summarization method is the singular value decomposition. I use the SVDPACKC [6] which is a package for the computation of SVD written in C language. This software package implements

[6]The two systems with the poorest performance produce only headlines, which are much shorter than 100 words. This may be the reason for their poor results.

| System | ROUGE-1 | ROUGE-2 | ROUGE-SU4 | ROUGE-L |
|---|---|---|---|---|
| **28** | 0.42776 | 0.21769 | 0.17315 | 0.38645 |
| **21** | 0,41488 | 0.21038 | 0.16546 | 0.37543 |
| **DUC baseline** | 0.41132 | 0.21075 | 0.16604 | 0.37535 |
| **19** | 0.40823 | 0.20878 | 0.16377 | 0.37351 |
| **LeLSA** | 0.40805 | 0.19722 | 0.15728 | 0.37878 |
| **27** | 0.40522 | 0.20220 | 0.16000 | 0.36913 |
| **29** | 0.39925 | 0.20057 | 0.15761 | 0.36165 |
| **31** | 0.39457 | 0.19049 | 0.15085 | 0.35935 |
| **15** | 0.38884 | 0.18578 | 0.15002 | 0.35366 |
| **23** | 0.38079 | 0.19587 | 0.15445 | 0.34427 |
| **GLLSA** | 0.38068 | 0.17440 | 0.13674 | 0.35118 |
| **16** | 0.37147 | 0.17237 | 0.13774 | 0.33224 |
| **18** | 0.36816 | 0.17872 | 0.14048 | 0.33100 |
| **25** | 0.34297 | 0.15256 | 0.11797 | 0.31056 |
| **Random** | 0.29963 | 0.11095 | 0.09004 | 0.27951 |
| **17** | 0.13528 | 0.05690 | 0.04253 | 0.12193 |
| **30** | 0.07452 | 0.03745 | 0.02104 | 0.06985 |

Table 3.3: Systems' comparison - ROUGE scores.

Lanczos and subspace iteration-based methods for determining several of the largest singular triplets for large sparse matrices.

The computational complexity is $O(3rz)$, where $z$ is the number of non-zero elements in the term by sentences matrix and $r$ is the number of dimensions returned. The maximum matrix size one can compute is usually limited by the memory requirement, which is $(10+r+q)N+(4+q)q$, where $N = m+n$, $m$ is the number of terms, $n$ is the number of sentences, and $q = min(N, 600)$, plus space for the term by sentences matrix [29].

The time complexity shows that it is directly proportional to the number of returned dimensions. Figure 3.5 shows the difference between the full SVD

| System | Significance Groups | | | |
|---|---|---|---|---|
| | **ROUGE-1** | **ROUGE-2** | **ROUGE-SU4** | **ROUGE-L** |
| **28** | A | A | A | AB |
| **LeLSA+AR** | AB | AB | AB | A |
| **21** | ABC | AB | AB | ABCD |
| **DUC baseline** | ABCD | AB | AB | ABCD |
| **19** | BCD | AB | ABC | BCD |
| **LeLSA** | BCD | BC | BC | ABC |
| **27** | BCD | ABC | ABC | BCDE |
| **29** | CDE | ABC | BC | CDEF |
| **31** | DE | CD | CDE | DEF |
| **15** | EF | CDE | CDE | EF |
| **23** | EFG | BC | BCD | FG |
| **GLLSA** | FG | DE | E | EF |
| **16** | FG | E | E | G |
| **18** | G | DE | DE | G |
| **25** | H | F | F | H |
| **Random** | I | G | G | I |
| **17** | J | H | H | J |
| **30** | K | I | I | K |

Table 3.4: Systems' comparison - 95% significance groups for ROUGE scores.

time[7] and the reduced SVD time[8]. There are always two values over each other. The purple value corresponds to the reduced time and the blue value over it corresponds to the full time. We can observe that for documents with up to 50 sentences the speed-up is not that substantial[9]. However, when a matrix with more contexts is decomposed the computation of the reduced SVD is considerably faster[10].

The memory complexity is dependent on the number of returned dimensions as well. Although, the memory consumption is not that noticeable for the

[7]The time needed for the computation of all dimensions.

[8]The time of the computation of the dimensions used by the summarization method.

[9]Partly because the time-consuming loading data procedure is the same in both runs.

[10]Testing machine: AMD Opteron 1.6GHz, 512MB RAM

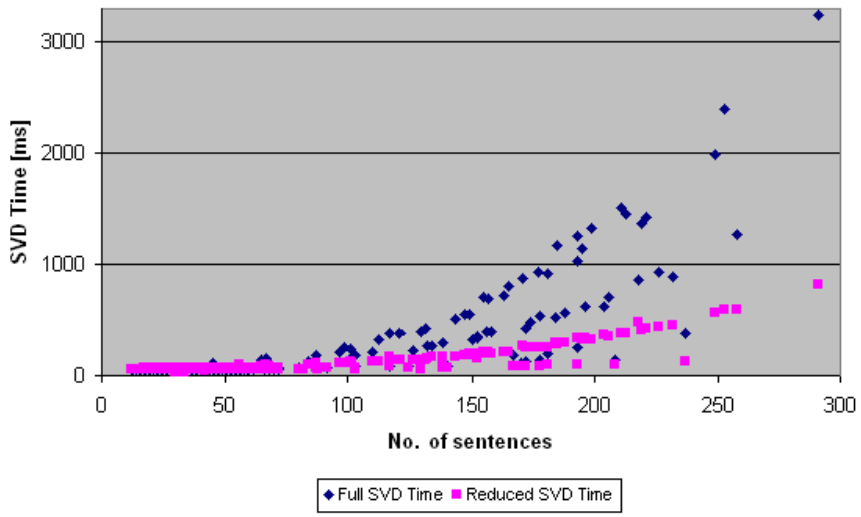


Figure 3.5: Dependency of the full and reduced SVD time on the length of a document (measured in sentences).

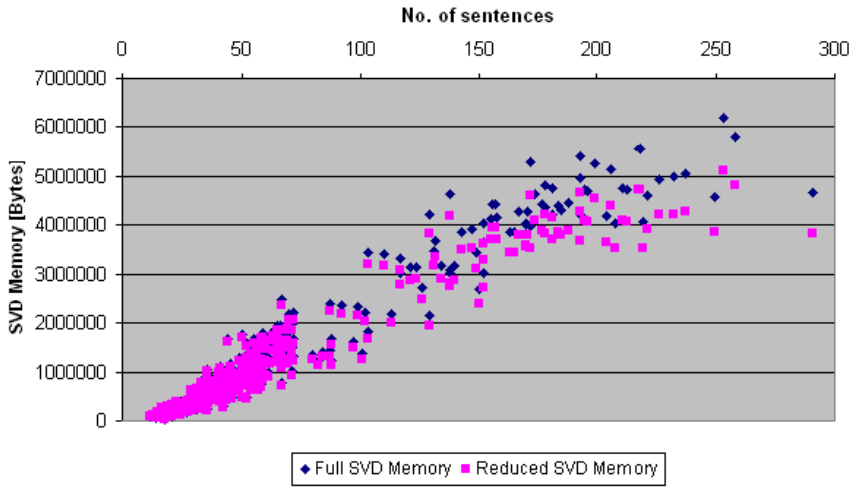


Figure 3.6: Dependency of the full and reduced SVD memory on the length of a document (measured in sentences).

matrix sizes that are used in summarization (see figure 3.6).

To conclude all, the time and memory complexity does not present a problem for summarization matrix sizes. Time is in the order of milliseconds and memory is in the order of kilobytes for newspaper articles. Reducing the number of returned dimension can lead to savings in time and memory especially for longer documents.

# Chapter 4

# Improving LSA-based Summarization with Anaphora Resolution

During my stay at the University of Essex, UK, I cooperated with the NLP group (especially with Massimo Poesio and Mijail A. Kabadjov) that works on anaphora resolution. This topic is closely related to summarization. We were trying to find a way how to use anaphoric information to improve the proposed LSA-based summarization method. They are responsible for the anaphora resolution part of the experiments and I am responsible for the summarization part. The primary aim was to improve the summarizer's performance. Additionally, it gave an opportunity of task-based evaluation of their anaphora resolver GUITAR.

Information about anaphoric relations could be beneficial for applications such as summarization, that involve extracting discourse models (possibly very simplified) from text. In this work we investigated exploiting automatically extracted information about the anaphoric relations in a text for two different aspects of the summarization task. First of all, we used anaphoric information to enrich the latent semantic representation of a document [29],

from which a summary is then extracted. Secondly, we used anaphoric information to check that the anaphoric expressions contained in the summary thus extracted still have the same interpretation that they had in the original text.

Lexical approaches to summarization use word similarity and other lexical relations to identify central terms [2]; we would include previous approaches based on LSA, such as in section 3.1. Coreference- or anaphora-based approaches[1] [3, 9, 5, 61] identify these terms by running a coreference- or anaphoric resolver over the text. we are not aware, however, of any attempt to use both lexical and anaphoric information to identify the main terms. In addition, we also developed a new algorithm for checking the use of anaphoric expressions in the summary (SRC).

## 4.1 Using Anaphora Resolution to find the Most Important Terms

Boguraev and Kennedy [9] use the following news article to illustrate why being able to recognize anaphoric chains may help in identifying the main topics of a document.

> PRIEST IS CHARGED WITH POPE ATTACK
>
> *A Spanish priest* was charged here today with attempting to murder the Pope. *Juan Fernandez Krohn*, aged 32, was arrested after a man armed with a bayonet approached the Pope while he was saying prayers at Fatima on Wednesday night. According to the police, *Fernandez* told the investigators today that *he* trained for the past six months for the assault.

[1] I use the term 'anaphora resolution' to refer to the task of identifying successive mentions of the same discourse entity, as opposed to the task of 'coreference resolution' which involves collecting all information about that entity, including information expressed by appositions.

> ... If found guilty, *the Spaniard* faces a prison sentence of 15-20 years.

As they point out, the title of the article is an excellent summary of the content: an entity (*the priest*) did something to another entity (*the pope*). Intuitively, this is because understanding that *Fernandez* and *the pope* are the central characters is crucial to provide a summary of texts like these.[2] Among the clues that help us to identify such 'main characters,' the fact that an entity is repeatedly mentioned is clearly important.

Methods that only rely on lexical information to identify the main topics of a text, such as the word-based methods discussed in the previous chapter, can only capture part of the information about which entities are frequently repeated in the text. As the above example shows, stylistic conventions forbid verbatim repetition, hence the six mentions of *Fernandez* in the text above contain only one lexical repetition, '*Fernandez*'. The main problem are pronouns, that tend to share the least lexical similarity with the form used to express the antecedent (and anyway are usually removed by stop-word lists, therefore do not get included in the SVD matrix). The form of definite descriptions (*the Spaniard*) doesn't always overlap with that of their antecedent, either, especially when the antecedent was expressed with a proper name. The form of mention which more often overlaps to a degree with previous mentions is proper nouns, and even then at least some way of dealing with acronyms is necessary (cfr. *European Union / E.U.*). On the other hand, it is well-known from the psychological literature that proper names often are used to indicate the main entities in a text. What anaphora resolution can do for us is to identify which discourse entities are repeatedly mentioned, especially when different forms of mention are used. We can then use the anaphoric chains identified by the anaphoric resolvers as additional terms in the initial matrix $A$ in equation 3.2.

[2] In many non-educational texts only a 'entity-centered' structure can be clearly identified, as opposed to a 'relation-centered' structure of the type hypothesized in Rhetorical Structures Theory and which serves as the basis for discourse structure-based summarization methods [27, 45].

### 4.1.1 General Tool for Anaphora Resolution (GUITAR)

The anaphora resolution system used for experiments, GUITAR [46, 23, 59], was implemented by University of Essex. It is a publically available tool designed to be modular and usable as an off-the-shelf component of a NLP pipeline. The system can resolve pronouns, definite descriptions and proper nouns.

#### Preprocessing

The anaphora resolution proper part of GUITAR is designed to take XML input, in a special format called MAS-XML, and produce an output in the same format, but which additionally contains anaphoric annotation. The system can therefore work with a variety of preprocessing methods, ranging from a simple part-of-speech tagger to a chunker to a full parser, provided that appropriate conversion routines into MAS-XML are implemented. The version used for these experiments uses Charniak's parser [11].

#### Anaphora Resolution Algorithms

The earlier version, GUITAR 2.1, included an implementation of the MARS pronoun resolution algorithm [38] to resolve personal and possessive pronouns. This system resolves definite descriptions using a partial implementation of the algorithm proposed in [64], augmented with a statistical discourse new classifier. The latest version, GUITAR 3.2, includes also an implementation of the shallow algorithm for resolving coreference with proper names proposed in [10].

##### Personal Pronouns

GUITAR includes an implementation of the MARS pronoun resolution algorithm [38]. MARS is a robust approach to pronoun resolution which only requires input text to be part-of-speech tagged and noun phrases to be iden-

tified. Mitkov's algorithm operates on the basis of antecedent-tracking preferences (referred to hereafter as "antecedent indicators"). The algorithm identifies the noun phrases which occur in the two sentences preceding the pronoun, checks their gender and number agreement with the anaphor, and then applies genre-specific antecedent indicators to the remaining candidates [38]. The noun phrase with the highest aggregate score is proposed as antecedent.

### Definite Descriptions

GUITAR also includes a partial implementation of the algorithm for resolving definite descriptions proposed in [64]. This algorithm attempts to classify each definite description as either direct anaphora, discourse-new, or bridging description. The first class includes definite descriptions whose head is identical to that of their antecedent, as in *a house ... the house*. Discourse-new descriptions are definite descriptions that refer to objects not already mentioned in the text and not related to any such object. [3]

Bridging descriptions are all definite descriptions whose resolution depends on knowledge of relations between objects, such as definite descriptions that refer to an object related to an entity already introduced in the discourse by a relation other than identity, as in *the flat ... the living room*. The Vieira / Poesio algorithm also attempts to identify the antecedents of anaphoric descriptions and the anchors of bridging ones. GUITAR incorporates an algorithm for resolving direct anaphora derived quite directly from Vieira / Poesio, as well as statistical methods for detecting discourse new descriptions [48].

### Proper Nouns

As the above example shows, proper nouns such as *Juan Fernandez Krohn*

[3]Some of these definite descriptions refer to objects whose existence is widely known, such as discourse-initial references to the pope; other to objects that can be assumed to be unique, even if unfamiliar, such as *the first woman to climb all Scottish Munros*.

(and quasi-proper nouns such as *the Pope*) are generally used in at least one mention of the main entities of a text. However, the version of GUITAR used in our previous work, 2.1, could not even identify coreferential proper names. Therefore a new version of GUITAR, 3.2, includes also an implementation of a shallow algorithm for resolving proper names proposed in [10].

The proper name resolution algorithm consists of a number of rules, which fall into two categories: rules that apply to all types of named entities (exact match, equivalent, possessives, spurious); and rules that apply only to organisations and persons (word token match, first token match, acronyms, last token match, prepositional phrases, abbreviations, multi-word name matching).

The algorithm uses knowledge of the type of entities being resolved, hence needs a Named Entity Recognizer (NER - [10]). It is worth noting that in our experiments we did not have access to a NER, meaning the current performance of the Proper Name resolution module can be considerably enhanced by incorporating a NER.

### Evaluation

GUITAR has been evaluated over a variety of corpora. I report here the results with a corpus in which noun phrases have been identified by hand the GNOME corpus [47], consisting of a variety of texts from different domains. The results of version 3.2 of the system with each type of anaphoric expression, are summarized in table 4.1. "Anaphor" is the type of anaphor, "Target #" is the number of observed anaphors, P, R, F are precision, recall and F-score. We can observe that the performance of the resolver reaches .7 level in all $P$, $R$ and $F$. We expect the performance of the anaphoric resolver on the documents used in experiments in this work to be similar to this one[4].

[4]Reference annotations were not available for these texts.

| **Anaphor** | **Target #** | **P** | **R** | **F** |
|---|---|---|---|---|
| DD | 195 | 70.4 | 63.6 | 66.8 |
| PersPro | 307 | 78.1 | 77.8 | 78 |
| PossPro | 202 | 79.1 | 73.3 | 76.1 |
| PN | 132 | 49 | 72 | 58.3 |
| TOTAL | 836 | 70.2 | 72.5 | 71.3 |

Table 4.1: Evaluation of GUITAR 3.2. on GNOME corpus.

### 4.1.2 Combining Lexical and Anaphoric Information

'Purely lexical' LSA methods discussed in the previous chapter determines the main 'topics' of a document on the basis of the simplest possible notion of term, simple words, as usual in LSA. In this section we will see, however, that anaphoric information can be easily integrated in an mixed lexical / anaphoric LSA representation by generalizing the notion of 'term' used in SVD matrices to include *discourse entities* as well, and counting a discourse entity $d$ as occurring in sentence $s$ whenever the anaphoric resolver identifies a noun phrase occurring in $s$ as a mention of $d$.

The simplest way of using anaphoric information with LSA is the SUBSTITUTION METHOD: keep using only words as terms, and use anaphora resolution as a pre-processing stage of the SVD input matrix creation. I.e., after identifying the anaphoric chains, replace all anaphoric nominal expressions with the first element of their anaphoric chain. In the example at the beginning of this chapter, for example, all occurrences of elements of the anaphoric chain beginning with *A Spanish priest* would be substituted by *A Spanish priest*. The resulting text would be as follows:

> PRIEST IS CHARGED WITH POPE ATTACK
>
> *A Spanish priest* was charged here today with attempting to murder the Pope. *A Spanish priest*, aged 32, was arrested after a man armed with

> a bayonet approached the Pope while he was saying prayers at Fatima on Wednesday night. According to the police, *a Spanish priest* told the investigators today that *a Spanish priest* trained for the past six months for the assault. ... If found guilty, *a Spanish priest* faces a prison sentence of 15-20 years.

This text could then be used to build an LSA representation as discussed in the previous chapter. I will show shortly, however, that this simple approach does not lead to improved results.

A better approach, it turns out, is what we call the ADDITION METHOD: generalize the notion of 'term,' treating anaphoric chains as another type of 'term' that may or may not occur in a sentence. The idea is illustrated in figure 4.1, where the input matrix $A$ contains two types of 'terms': terms in the lexical sense (i.e., words) and terms in the sense of discourse entities, represented by anaphoric chains. The representation of a sentence then specifies not only if that sentence contains a certain word, but also if it contains a mention of a discourse entity. With this representation, the chain 'terms' may tie together sentences that contain the same anaphoric chain even if they do not contain the same word. The resulting matrix would then be used as input to SVD as before.

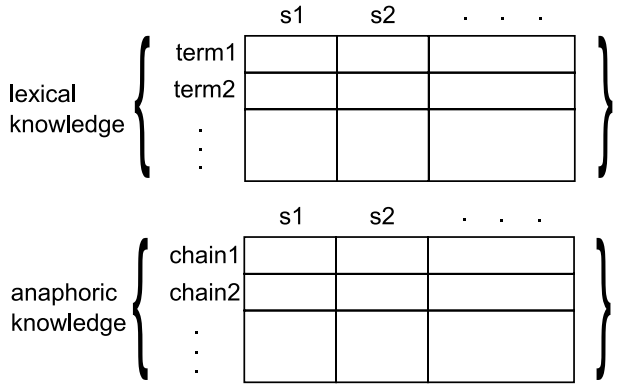


Figure 4.1: Using discourse entities as terms.

### 4.1.3 First Experiments: the CAST Corpus

#### The CAST Corpus

In this pilot evaluation, we used the corpus of manually produced summaries created by the CAST project [44]. The CAST corpus contains news articles taken from the Reuters Corpus and a few popular science texts from the British National Corpus. Summaries are specified by providing information about the importance of sentences [17]: sentences are marked as **essential** or **important** for the summary. The corpus also contains annotations for linked sentences, which are not significant enough to be marked as important/essential, but which have to be considered as they contain information essential for the understanding of the content of other sentences marked as essential/important.

Four annotators were used for the annotation, three graduate students and one postgraduate. Three of the annotators were native English speakers, and the fourth had advanced knowledge of English. Unfortunately, not all of the documents were annotated by all of the annotators. To maximize the reliability of the summaries used for evaluation, we chose the documents annotated by the greatest number of the annotators; in total, the evaluation corpus contained 37 documents.

For acquiring manual summaries at specified lengths and getting the sentence scores (for relative utility evaluation) we assigned a score 3 to the sentences marked as essential, a score 2 to important sentences and a score 1 to linked sentences.

#### Evaluation Measures

In this early study, we addressed the problem by using a combination of evaluation measures. The first of these was relative utility (section 2.3.2), then I present a standard F-measure values (section 2.3.2) and at last cosine

similarity (section 2.3.3) was computed. For details see section 2.3. The most complex measure, relative utility, was used to draw conclusions.

### How Much May Anaphora Resolution Help? An Upper Bound

In order to determine whether anaphoric information might help, and which method of adding anaphoric knowledge to the LSA summarizer is better, we annotated by hand all the anaphoric relations in the 37 documents in the CAST corpus using the annotation tool MMAX [40]. Results for the 15%, resp. 30%, summarization ratio using a variety of summarization evaluation measures are presented in table 4.2, resp. 4.3.

| **Evaluation Method** | **Lexical LSA** | **Manual Subst.** | **Manual Addition** |
|---|---|---|---|
| **Relative Utility** | 0.595 | 0.573 | **0.662** |
| **F-score** | 0.420 | 0.410 | **0.489** |
| **Cosine Sim.** | 0.774 | 0.806 | **0.823** |

Table 4.2: Improvement over word-based LSA with manually annotated anaphoric information - summarization ratio: 15%.

| **Evaluation Method** | **Lexical LSA** | **Manual Subst.** | **Manual Addition** |
|---|---|---|---|
| **Relative Utility** | 0.645 | 0.662 | **0.688** |
| **F-score** | 0.557 | 0.549 | **0.583** |
| **Cosine Sim.** | 0.863 | 0.878 | **0.886** |

Table 4.3: Improvement over word-based LSA with manually annotated anaphoric information - summarization ratio: 30%.

These tables clearly shows that even with perfect knowledge of anaphoric links, the substitution method would lead to worse results than lexical LSA.

On the other hand, the addition method could potentially lead to significant improvements.

### Results with GUITAR 2.1

To use GUITAR, the texts were parsed using Charniak's parser [11]. The output of the parser was then converted into the MAS-XML format expected by GUITAR by one of the preprocessors that come with the system. (This step includes heuristic methods for guessing agreement features.) Finally, GUITAR was ran to add anaphoric information to the files. The resulting files were then processed by the summarizer.

The results obtained by the summarizer using GUITAR's output are presented in tables 4.4 and 4.5 (relative utility, F-score, and cosine similarity).

| **Evaluation Method** | **Lexical LSA** | **GuiTAR Substitution** | **GuiTAR Addition** |
|---|---|---|---|
| **Relative Utility** | 0.595 | 0.530 | **0.640** |
| **F-score** | 0.420 | 0.347 | **0.441** |
| **Cosine Similarity** | 0.774 | 0.804 | **0.805** |

Table 4.4: Improvement over word-based LSA with GUITAR annotations - summarization ratio: 15%.

| **Evaluation Method** | **Lexical LSA** | **GuiTAR Substitution** | **GuiTAR Addition** |
|---|---|---|---|
| **Relative Utility** | 0.645 | 0.626 | **0.678** |
| **F-score** | 0.557 | 0.524 | **0.573** |
| **Cosine Similarity** | 0.863 | 0.873 | **0.879** |

Table 4.5: Improvement over word-based LSA with GUITAR annotations - summarization ratio: 30%.

Tables 4.4-4.5 clearly show that using GUITAR and the addition method leads to significant improvements over our baseline LSA summarizer. The improvement in relative utility measure was significant (95% confidence by the t-test). On the other hand, the substitution method did not lead to significant improvements, as was to be expected given that no improvement was obtained with 'perfect' anaphora resolution (see previous section).

### ROUGE Evaluation of Pilot Study

I also evaluated the results using the ROUGE measure (see section 2.3.3) - tables 4.6 and 4.7[5] - obtaining improvements with the addition method, but the differences were not statistically significant.

| System | ROUGE-1 | ROUGE-2 | ROUGE-SU4 | ROUGE-L |
|---|---|---|---|---|
| **Manual Add.** | 0.64257 | 0.57896 | 0.55134 | 0.56609 |
| **GuiTAR Add.** | 0.62297 | 0.55353 | 0.52693 | 0.54783 |
| **Lexical LSA** | 0.60359 | 0.53140 | 0.50115 | 0.53516 |
| **GuiTAR Sub.** | 0.59273 | 0.50666 | 0.47908 | 0.52006 |
| **Manual Sub.** | 0.53144 | 0.40629 | 0.37347 | 0.46431 |

Table 4.6: ROUGE scores for the pilot study - summarization ratio: 15%.

### Pilot Study Conclusion

In conclusion, this pilot study showed that (i) we could expect performance improvements over purely lexical LSA summarization using anaphoric information, (ii) that significant improvements at least by the Relative Utility score could be achieved even if this anaphoric information was automatically

[5]The values are larger than it is usual in standard DUC comparison of summaries and abstracts because summaries and extracts were compared here. Truncation of summaries to exact leghth could not be performed because the summary length was derived proportionally from source text length. ROUGE version and settings: `ROUGEeval-1.4.2.pl -c 95 -m -n 2 -s -2 4 -a cast.xml`.

| System | ROUGE-1 | ROUGE-2 | ROUGE-SU4 | ROUGE-L |
|---|---|---|---|---|
| **Manual Add.** | 0.64257 | 0.57896 | 0.55134 | 0.56609 |
| **GuiTAR Add.** | 0.62297 | 0.55353 | 0.52693 | 0.54783 |
| **Lexical LSA** | 0.60359 | 0.53140 | 0.50115 | 0.53516 |
| **GuiTAR Sub.** | 0.59273 | 0.50666 | 0.47908 | 0.52006 |
| **Manual Sub.** | 0.53144 | 0.40629 | 0.37347 | 0.46431 |

Table 4.7: ROUGE scores for the pilot study - summarization ratio: 30%.

extracted, and (iii) that, however, these results were only achievable using the Addition method.

What this earlier work did not show was how well our results compared with the state of the art, as measured by evaluation over a standard reference corpus such as DUC 2002, and using the by now standard ROUGE measure. Furthermore, these results were obtained using a version of the anaphoric resolver that did not attempt to identify coreference links realized using proper names, even though proper names tend to be used to realize more important terms. Given our analysis of the effect of improvements to the anaphoric resolver [23], we expected an improved version to lead to better results. The subsequent experiments were designed to address these questions.

### 4.1.4 Experiments with the DUC 2002 Corpus

The version of our system used for this second experiment differs from the version discussed in the previous experiment. A new version of the anaphoric resolver, GUITAR 3.2, was developed. As discussed above, it also resolves proper names. It was expected that this new version could lead to significant improvements also by the ROUGE measure, as well as being more usable. In section 3.2.4 I discussed the comparison of the purely lexical summarizer with other summarizers. Now we can update the figures by adding an improved summarizer with anaphora resolution (LELSA+AR).

| System | ROUGE-1 | ROUGE-2 | ROUGE-SU4 | ROUGE-L |
|---|---|---|---|---|
| **28** | 0.42776 | 0.21769 | 0.17315 | 0.38645 |
| **LeLSA+AR** | 0.42280 | 0.20741 | 0.16612 | 0.39276 |
| **21** | 0,41488 | 0.21038 | 0.16546 | 0.37543 |
| **DUC baseline** | 0.41132 | 0.21075 | 0.16604 | 0.37535 |
| **19** | 0.40823 | 0.20878 | 0.16377 | 0.37351 |
| **LeLSA** | 0.40805 | 0.19722 | 0.15728 | 0.37878 |
| **27** | 0.40522 | 0.20220 | 0.16000 | 0.36913 |
| **29** | 0.39925 | 0.20057 | 0.15761 | 0.36165 |
| **31** | 0.39457 | 0.19049 | 0.15085 | 0.35935 |
| **15** | 0.38884 | 0.18578 | 0.15002 | 0.35366 |
| **23** | 0.38079 | 0.19587 | 0.15445 | 0.34427 |
| **GLLSA** | 0.38068 | 0.17440 | 0.13674 | 0.35118 |
| **16** | 0.37147 | 0.17237 | 0.13774 | 0.33224 |
| **18** | 0.36816 | 0.17872 | 0.14048 | 0.33100 |
| **25** | 0.34297 | 0.15256 | 0.11797 | 0.31056 |
| **Random** | 0.29963 | 0.11095 | 0.09004 | 0.27951 |
| **17** | 0.13528 | 0.05690 | 0.04253 | 0.12193 |
| **30** | 0.07452 | 0.03745 | 0.02104 | 0.06985 |

Table 4.8: Updated systems' comparison - ROUGE scores.

Systems are compared in table 4.8 by the ROUGE scores[6] and DUC 2002 data. Table 4.9 shows a multiple comparison of ROUGE scores between systems. Systems not sharing a common letter are significantly different at the 95% confidence level (see section 3.2.4 for explanation).

In the section 3.2.4 I concluded that the LELSA summarizer is state of the art: its performance is significantly worse only than that the best system in DUC 2002, system 28, in ROUGE-1, ROUGE-2 and ROUGE-SU4, and significantly better than that of 9 in ROUGE-1, 7 in ROUGE-2, 7 in ROUGE-SU4 and

[6]All system summaries were truncated to 100 words as traditionally done in DUC. ROUGE version and settings: `ROUGEeval-1.4.2.pl -c 95 -m -n 2 -l 100 -s -2 4 -a duc.xml`.

| System | Significance Groups | | | |
|---|---|---|---|---|
| | **ROUGE-1** | **ROUGE-2** | **ROUGE-SU4** | **ROUGE-L** |
| **28** | A | A | A | AB |
| **LeLSA+AR** | AB | AB | AB | A |
| **21** | ABC | AB | AB | ABCD |
| **DUC baseline** | ABCD | AB | AB | ABCD |
| **19** | BCD | AB | ABC | BCD |
| **LeLSA** | BCD | BC | BC | ABC |
| **27** | BCD | ABC | ABC | BCDE |
| **29** | CDE | ABC | BC | CDEF |
| **31** | DE | CD | CDE | DEF |
| **15** | EF | CDE | CDE | EF |
| **23** | EFG | BC | BCD | FG |
| **GLLSA** | FG | DE | E | EF |
| **16** | FG | E | E | G |
| **18** | G | DE | DE | G |
| **25** | H | F | F | H |
| **Random** | I | G | G | I |
| **17** | J | H | H | J |
| **30** | K | I | I | K |

Table 4.9: Updated systems' comparison - 95% significance groups for ROUGE scores.

10 in ROUGE-L of the systems that participated in that competition and it outperforms significantly Gong and Liu's LSA approach (GLLSA). However, the LELSA+AR summarizer is even better: it is significantly better than 11 systems in ROUGE-1, 9 in ROUGE-2, 9 in ROUGE-SU4 and 13 in ROUGE-L, it is significantly better than the baseline in ROUGE-L at the 90% confidence level, and it is not significantly worse than any of the systems.

### 4.1.5 An Example: a Summary Before and After Anaphora Resolution

Examples (4.1.5) and (4.1.5) illustrate the difference between a summary created by the pure LSA summarizer and the corresponding one created by the summarizer enhanced by anaphora resolution (addition method).

JURORS DEADLOCKED ON 13 CHARGES
(summary before anaphora resolution)

Jurors who have reached verdicts on 52 counts in the McMartin preschool molestation case said Wednesday they are deadlocked on the remaining 13 charges in the nation's longest, costliest criminal trial. Superior Court Judge William Pounders received a note from the jurors as they ended their day's deliberation and called a hearing for Thursday to discuss the deadlock and possibly opening the 52 sealed verdicts. In an interview Wednesday evening, Pounders said he would deal with the deadlocked counts first , either declaring a mistrial on those counts and reading the sealed verdicts, or sending the jury back to resume deliberations on the undecided counts.

JURORS DEADLOCKED ON 13 CHARGES
(summary after anaphora resolution)

*Jurors* who have reached verdicts on 52 counts in the McMartin preschool molestation case said Wednesday *they* are deadlocked on the remaining 13 charges in the nation's longest, costliest criminal trial. Superior Court Judge William Pounders received a note from *the jurors* as *they* ended *their* day's deliberation and called a hearing for Thursday to discuss the deadlock and possibly opening the 52 sealed verdicts. *The jurors* are deciding whether *Raymond Buckey, 31,* and *his mother, Peggy McMartin Buckey, 63,* are guilty or innocent of charges *they* molested children at *their family-owned McMartin Pre-School in Manhattan Beach.*

From the examples it can be seen that the first two sentences selected by the summarizers are the same, whereas the third one is different. When using anaphora resolution, sentence selection was affected by strong anaphoric chains refering to salient entities (e.g., *the jurors*, *Raymond Buckey*, *Peggy McMartin Buckey*). The presence of the dominant entity, *the jurors*, in all three sentences served as 'glue' and kept the three sentences together throughout the process influencing the outcome of that summarizer. ROUGE scores for this particular document were significantly better when anaphora resolution was used.

## 4.2 A Summary Reference Checker

Anaphoric expressions can only be understood with respect to a context. This means that summarization by sentence extraction can wreak havoc with their interpretation: there is no guarantee that they will have an interpretation in the context obtained by extracting sentences to form a summary, or that this interpretation will be the same as in the original text. Consider the following example.

> PRIME MINISTER CONDEMNS IRA FOR MUSIC SCHOOL EXPLOSION
>
> (**S1**) [Prime Minister Margaret Thatcher]$_1$ said Monday [[the Irish Republican Army]$_2$ members who blew up [the Royal Marines School of Music]$_3$ and killed [10 bandsmen]$_4$ last week]$_5$ are monsters who will be found and punished.
>
> (**S2**) "[The young men whom we lost]$_4$ were murdered by [common murderers who must be found and brought to justice and put behind bars for a very long time]$_5$," [she]$_1$ said following a tour of [[the school's]$_3$ wrecked barracks]$_6$ in Deal, southeast England.
>
> ...
>
> (**S3**) [Gerry Adams, president of [Sinn Fein, the legal political arm of

> [the IRA]$_2$ ]$_8$ ]$_7$ issued a statement disputing [Mrs. Thatcher's]$_1$ remarks, saying "[she]$_1$ knows in [her]$_1$ heart of hearts the real nature of the conflict, its cause and the remedy".
>
> ...
>
> (**S4**) "[We]$_8$ want an end to all violent deaths arising out of the present relationship between our two countries," [Adams]$_7$ said.
>
> ...
>
> (**S5**) [The IRA]$_2$ claimed responsibility for the explosion, and police said they are looking for [three men with Irish accents who rented a house overlooking [the barracks]$_6$]$_5$.

If sentence S2 were to be extracted to be part of the summary, but not S1, the pronoun *she* would not be understandable as it would not have a matching antecedent anymore. The reference to *the school* would also be uninterpretable. The same would happen if S5 were extracted without also extracting S2; in this case, the problem would be that the antecedent for *the barracks* is missing.

Examples such as the one just shown suggested another use for anaphora resolution in summarization – correcting the references in the summary. Our idea was to replace anaphoric expressions with a full noun phrase in the cases where otherwise the anaphoric expression could be misinterpreted. We discuss this method in detail next.

### 4.2.1 The Reference Correction Algorithm

The proposed correction algorithm works as follows.

1. Run anaphora resolution over the source text, and create anaphoric chains.
2. Identify the sentences to be extracted using a summarization algorithm such as the one discussed in the previous sections.

3. For every anaphoric chain, replace the first occurrence of the chain in the summary with its first occurrence in the source text. After this step, all chains occuring in both source and summary start with the same lexical form.

   For example, in the text in (4.2), if sentence S4 is included in the summary, but S3 isn't, the first occurrence of chain 7 in the summary, *Adams*, would be substituted by *Gerry Adams, president of Sinn Fein, the legal political arm of the IRA*.

4. Run the anaphoric resolver over the summary.

5. For all nominal expressions in the summary: if the expression is part of a chain in the source text and it is not resolved in the summary (the resolver was not able to find an antecedent), or if it is part of a different chain in the summary, then replace the anaphoric expression with the full expression from the source text.

This method can be used in combination with the summarization system discussed in earlier sections, or with other systems; and becomes even more important when doing sentence compression, because intrasentential antecedents can be lost as well. However, automatic anaphora resolution can introduce some new errors. We discuss our evaluation of the algorithm next.

### 4.2.2 Evaluation of Reference Correction

To measure the recall of the reference checker algorithm we would need anaphoric annotations, that were not available for DUC data. We measured its precision manually as follows. To measure the precision of the step where the first occurrences of a chain in the summary were replaced by the first mention of that chain in the source text, we took a sample of 155 document, obtaining the results shown in table 4.10.

| **Statictic** | Overall | Per Document |
|---|---|---|
| Chains in full text | 2906 | 18.8 |
| Chains in summary | 1086 (37.4% of full text chains) | 7.0 |
| First chain occurrence was in the summary | 714 (65.7% of the chains in summaries) | 4.6 |
| First element of chain had same lexical form | 101 (9.3% of the chains in summaries) | 0.7 |
| First chain occurrence replaced | 271 (25% of the chains in summaries) | 1.7 |
| Correctly replaced | 186 (Precision: 68.6%) | 1.2 |

Table 4.10: Evaluation of SRC step 3, the first chain occurrence replacement.

We can observe that full texts contained on average 19 anaphoric chains, a summaries about 7. In 66% of the summary chains the sentence where the first chain occurrence appeared was selected into the summary, and in 9% there was no need to replace the expression because it already had the same form as the first element of the chain. So overall the first chain occurence was replaced in 25% of the cases; the precision was 68.6%. This suggest that the success in this task correlates with anaphora resolver's quality.

After performing anaphora resolution on the summary and computing its anaphoric chains, the anaphors without an antecedent are replaced. We analyzed a sample of 86 documents to measure the precision by hand. Overall, 145 correct replacements were made in this step and 65 incorrect, for a precision of 69%. Table 4.11 analyzes the performance on this task in more detail.

The first row of the table lists the cases in which an expression was placed in a chain in the summary, but not in the source text. In these cases, our algorithm does not replace anything.

| **Observed state** | **Correct** | **Incorrect** |
|---|---|---|
| Full text: expression in no chain<br>Summary: expression in a chain | 16<br>(66.7%) | 8<br>(33.3%) |
| Full text: expression in a chain<br>Summary: expression in no chain | 32 (replaced +)<br>(69.6%) | 14 (replaced -)<br>(30.4%) |
| Expression in the same chain<br>in the full text and its summary | 336<br>(83%) | 69<br>(17%) |
| Expression in a different chain<br>in the full text and its summary<br>(correctly resolved in full text ) | 81 (replaced +)<br>(71.7%)<br>(correctly resolved<br>in summary) | 32 (replaced +)<br>(28.3%)<br>(incorrectly resolved<br>in summary) |
| Expression in a different chain<br>in the full text and its summary<br>(incorrectly resolved in full text) | 39 (replaced -)<br>(76.5%)<br>(in summary<br>correctly resolved) | 12 (replaced -)<br>(23.5%)<br>(in summary<br>incorrectly resolved) |
| **Replacements overall** | **145**<br>**(69%)** | **65**<br>**(31%)** |

Table 4.11: Evaluation of SRC step 5, checking the comprehensibility of anaphors in the summary. (Replaced + means that the expressions were correctly replaced; replaced - that the replacement was incorrect).

Our algorithm however does replace an expression when it finds that there is no chain assigned to the expression in the summary, but there is one in the source text; such cases are listed in the second row. We found that this replacement was correct in 32 cases; in 14 cases the algorithm replaced an incorrect expression. The third row lists summarizes the most common case, in which the expression was inserted into the same chain in the source text and in the summary. (I.e., the first element of the chain in the summary is also the first element of the chain in the source text.) When this happens, in 83% of cases GUITARs' interpretation is correct; no replacement is necessary.

Finally, there are two subcases in which different chains are found in the source text and in the summary (in this case the algorithm performs a replacement). The fourth row lists the case in which the original chain is correct; the last, cases in which the chain in the source text is incorrect. In the first column of this row are the cases in which the anaphor was correcly resolved in the summary but it was substituted by an incorrect expression because of a bad full text resolution; the second column shows the cases in which the anaphor was incorrectly resolved in both the full text and the summary, however, replacement was performed because the expression was placed in different chains.

### 4.2.3 An Example: a Summary Before and After Reference Checking

Examples (4.2.3) and (4.2.3) illustrate the difference between a summary before and after reference checking. A reader of the following summary may not know who *the 71-year-old Walton* or *Sively* are, and what *store* it is the text is talking about. In addition, the pronoun *he* in the last sentence is ambiguous between *Walton* and *Sively*. On the other hand, *the singer* in the last sentence can be easily resolved. This is because the chains *Walton*, *Sively* and *the store* do not start in the summary with the expression used for the first mention in the source text. These problems are fixed by step 3 of the SRC. The ambiguous pronoun *he* in the last sentence of the summary is resolved to *Sively* in the summary and *Walton* in the source text[7]. Because the anaphor occurs in a different chains in the summary and in the full text, it has to be substituted by the head of the first chain occurrence noun phrase, *Walton*. *The singer* in the last sentence is resolved identically in the summary and in the full text: the chains are the same, so there is no need for replacement.

[7]The previous sentence in the source is: " *Walton* continued talking with customers during the concert."

WAL-MART FOUNDER PITCHES IN AT CHECK-OUT COUNTER
(summary before reference checking)

***The 71-year-old Walton***, considered to be one of the world's richest people, grabbed a note pad Tuesday evening and started hand-writing merchandise prices for customers so their bills could be tallied on calculators quickly. *Walton*, known for his down-home style, made a surprise visit to ***the store*** that later Tuesday staged a concert by *country singer Jana Jea* in *its* parking lot. *Walton* often attends promotional events for *his* Arkansas-based chain, and ***Sively*** said *he* had suspected the boss might make an appearance. ***He*** also joined *the singer* on stage to sing a duet and led customers in the Wal-Mart cheer.

WAL-MART FOUNDER PITCHES IN AT CHECK-OUT COUNTER
(summary after reference checking)

***Wal-Mart founder Sam Walton***, considered to be one of the world's richest people, grabbed a note pad Tuesday evening and started hand-writing merchandise prices for customers so their bills could be tallied on calculators quickly. *Walton*, known for his down-home style, made a surprise visit to ***his store in this Florida Panhandle city*** that later Tuesday staged a concert by *country singer Jana Jea* in *its* parking lot. *Walton* often attends promotional events for *his* Arkansas-based chain, and ***store manager Paul Sively*** said *he* had suspected the boss might make an appearance. ***Walton*** also joined *the singer* on stage to sing a duet and led customers in the Wal-Mart cheer.

# Chapter 5

# Sentence Compression based on LSA

Sentence compression can be linked to summarization at the sentence level. The task has an immediate impact on several applications ranging from summarization to caption generation [26]. Previous data-driven approaches [26, 51] relied on parallel corpora to determine what is important in a sentence. The models learned correspondences between long sentences and their shorter counterparts, typically employing a rich feature space induced from parse trees. In [57] an algorithm based on Rhetorical Structure Theory (see section 2.1.4) is proposed. They use the fact that nuclei are more likely to be retained when summarizing than satellites. The output can be then obtained simply by removing satellites. The task is challenging since the compressed sentences should retain essential information and convey it grammatically.

I present a simple sentence compression algorithm that works on a clause level. The aim is to simplify sentences that have many clauses by removing unimportant ones. The first stage is to obtain compression candidates (e.g., sentences to which an original sentence can be compressed). Further, the algorithm tries to select the best candidate. An ideal candidate preserves the LSA score of the full sentence and has a reasonable compression ratio.

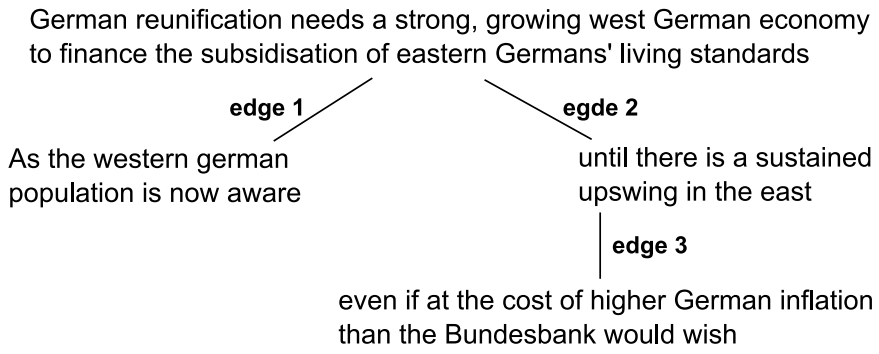


Figure 5.1: Tree structure of an example sentence.

## 5.1 Identification of Compression Candidates

If we want to compress a sentence we firstly need to identify a set of possible compression candidates (CC). They should preserve the grammaticality and the meaning of the full sentence. For this task we need a parser that can derive a sentence tree structure. I used Charniak parser [11], which is able to mark clauses and catch their dependencies. Let me describe the identification method on the following sentence[1]:

> As the western german population is now aware, German reunification needs a strong, growing west German economy to finance the subsidisation of eastern Germans' living standards until there is a sustained upswing in the east even if at the cost of higher German inflation than the Bundesbank would wish.

The Charniak parser will capture the tree structure at figure 5.1. We can see three edges there. If we cut the tree in an edge we get a compressed sentence where all subordinate clauses of the edge are removed. And more, we can cut the tree more than once - in combination of edges. However, we can not delete dependent edges. If we apply these rules on the example sentence above we obtain the following six compression candidates:

[1]It is a sentence from DUC 2002 corpus, cluster 69, document FT923-6509.

**CC1:** German reunification needs a strong, growing west German economy to finance the subsidisation of eastern Germans' living standards. (**edge 1** and **edge 2** were removed)

**CC2:** German reunification needs a strong, growing west German economy to finance the subsidisation of eastern Germans' living standards until there is a sustained upswing in the east. (**edge 1** and **edge 3**)

**CC3:** German reunification needs a strong, growing west German economy to finance the subsidisation of eastern Germans' living standards until there is a sustained upswing in the east even if at the cost of higher German inflation than the Bundesbank would wish. (**edge 1**)

**CC4:** As the western german population is now aware, German reunification needs a strong, growing west German economy to finance the subsidisation of eastern Germans' living standards. (**edge 2**)

**CC5:** As the western german population is now aware, German reunification needs a strong, growing west German economy to finance the subsidisation of eastern Germans' living standards until there is a sustained upswing in the east. (**edge 3**)

The last candidate **CC6** is the full sentence. Other combinations of edges could not be applied due to the rules above.

## 5.2 Finding the Best Candidate

The summarization method proposed in 3.1 favours long sentences. It is obvious because a long sentence is more likely to contain significant terms or topics. There is a correlation between sentence length and its summarization value, however, not that strong (Pearson correlation coeficient is approximately 0.6). Regardless, the medium correlation encourages us to normalize

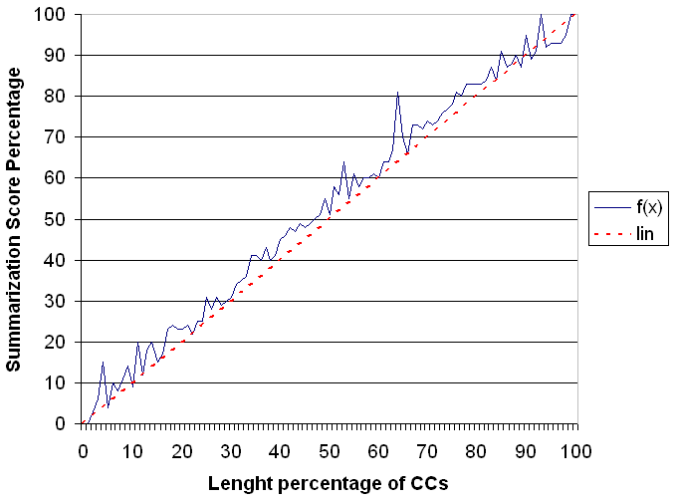


Figure 5.2: Dependency of the candidate's summarization score on its length. Both the length and the summarization score are measured proportionally to the full sentence.

the summarization value by the sentence length to avoid the negative correlation effect. I created all compression candidates of the sentences in DUC 2002 corpus. This gave enough data to show the linear dependency of the summarization score on the sentence length (figure 5.2). Notice that when we cut a sentence to 50% a random compression candidate contains approximately 50% of full sentence information, measured by LSA score.

Now we can update the summarization score formula 3.4:

$$s_k = \frac{\sqrt{\sum_{i=1}^{r} b_{i,k}^2}}{nwords}, \tag{5.1}$$

where *nwords* is a number of words in the sentence $k$. This will measure an average summarization score for a word in a sentence. The difficulty is now that a long significant sentence with many unimportant clauses will be assigned by a low score. However, with compression algorithm that would be able to delete the unimportant clauses we can suppress this negative effect. After obtaining compression candidates we assign each a summariza-

tion score. Firstly we need to create an input vector for the candidate. The process is the same as when producing the input matrix $A$ (see chapter 3). The vector has to be transformed to the latent space[2]:

$$v_q = \Sigma U^T q, \tag{5.2}$$

where $q$ is the CC's weighted term-frequency vector and $v_q$ is the vector transformed to the latent space.

Now, we can compute the summarization score (5.1) for each candidate. We substitute the vector $v_k$ by $v_q$ in equation 5.1. The candidate with the highest summarization score within the sentence candidates is considered to be the best candidate of the sentence. In the example above **CC1** received the highest score. Among the best candidates we select the ones with the highest summarization score for the summary. A full sentence can be selected as well because it is allways among the candidates.

After selecting the best candidate we can find out what percentage of loss in summarization score we get when we compress a sentence. For this task we obtained the best compression candidates of all sentences in our corpus. Figure 5.3 shows a dependency of the best candidate's summarization score on the sentence length.

The difference between this graph and the previous one is that here we take into account only the best candidate, however, in the previous we took all of them. We can observe that this dependency is defined by the square root. It means that when we shorten a sentence to 50% it will contain on average approximately 71% of full sentence information, measured by the LSA score.

[2]The process is the same as a query is transformed to the latent space in information retrieval.

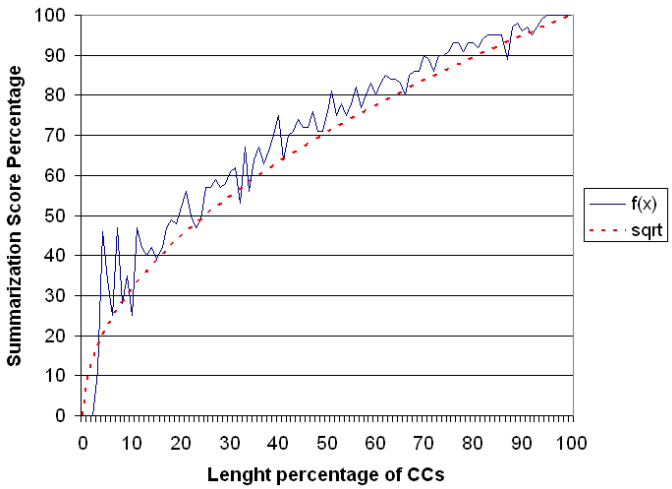


Figure 5.3: Dependency of the best candidate's summarization score on its length. Both the length and the summarization score are measured proportionally to the full sentence.

## 5.3 Experimental Results

Requirements of sentence compression algorithms are to preserve the main content of the sentence and produce a grammatically correct sentence. I made the following experiment. I used the DUC 2002 corpus for the evaluation. I randomly selected 20 documents, each from a different cluster. The documents were parsed by the Charniak parser and summaries with compressed sentences were created. I needed to compare these compressions to those made by humans. For each summary sentence one compression by an annotator was written. And more, I generated a baseline candidate where bottom-most leaves of the clause tree (see figure 5.1) are cut. When we return to the example above **CC5** represents the baseline compression. Thus, for each summary sentence three different compressions were produced - by a human, the LSA score and the baseline.

Three annotators assigned a correctness score from 1 to 5 to each candidate. It received score 1 when unimportant information was removed, score 2 when

not so important information was missing, score 3 when some important information was missing, score 4 when the sense of the sentence was changed, and score 5 where the sentence had no sense. If they found a grammatical problem they marked the candidate as grammatically incorrect. See the example sentences in the following annotation scale:

- 1 = loss of unimportant information

  Example: *A huge explosion in a barracks staff room Friday morning leveled the three-story building as bandsmen took a coffee break between practice sessions on the school's parade ground.*

  COMPRESSED TO

  *A huge explosion in a barracks staff room Friday morning leveled the three-story building.*

  The fact that "*bandsmen took a coffee break*" is obviously unimportant for inclusion into the summary.

- 2 = loss of slightly important information

  Example: *Forecasters said the hurricane had been gaining strength as it passed over the ocean after it dumped 5 to 10 inches of rain on the Dominican Republic and Haiti, which share the island of Hispaniola.*

  COMPRESSED TO

  *The hurricane had been gaining strength as it passed over the ocean.*

  If we compress the sentence this way we are loosing the information that the hurricane dumped inches of rain on the Dominican Republic and Haiti, but it does not seem to be that important for the summary. However, it is always a bit subjective point of view. Two is still a positive mark.

- 3 = loss of important information

Example: *The Royal Marines Music School is one of 30 military establishments in Britain that use private security firms, Defense Ministry figures show.*

COMPRESSED TO

*The Royal Marines Music School is one of 30 military establishments in Britain, Defense Ministry figures show.*

This is already a negative mark. If we throw out the clause "*that use security firms*", we lost an important information and more serious is that the compressed sentence is a bit misleading.

- 4 = the sense changed

  Example: *If Mrs. Thatcher resigns while in office, her successor elected by the party automatically becomes prime minister with the approval of Queen Elisabeth II, but convention dictates that he or she seek a mandate in a general election as soon as possible.*

  COMPRESSED TO

  *If Mrs. Thatcher resigns automatically becomes prime minister with the approval of Queen Elisabeth II.*

  This is a bit funny example. We can see that it can happen that the sense of the original sentence can totally change.

- 5 = no sense

  Example: *Earlier this year, Mrs. Thatcher overtook Liberal Lord Asquith's 1908-1916 tenure as prime minister of the 20th century.*

  COMPRESSED TO

  *Earlier this year Mrs. Thatcher overtook.*

  These are the most problematic compressions. In most of them an object is missing. We know only that Mrs. Thatcher overtook but we do not know what.

| **Compression approach** | **Baseline** | **LSA score** | **Human** |
|---|---|---|---|
| **% of compressed sentences** | 66.07 | 51.79 | 43.75 |
| **% of average compression rate** | 58.41 | 63.64 | 64.60 |
| **correctness score (95% conf. interval)** | 2.883 (2.577 - 3.189) | 2.193 (1.929 - 2.456) | 1.486 (1.363 - 1.610) |
| **% of grammatically incorrect sentences** | 5.86 | 7.66 | 0 |

Table 5.1: A comparison of compression approaches.

Table 5.1 compares the compression approaches. Totally, 74 sentences from 20 documents were analyzed.

A critical problem in doing the compressions was that sometimes the main object of the sentence was removed. These compressions were assigned by the score 5. We can observe that the compression method based on the LSA score performed significantly better than the simple baseline and significantly worse than humans. It compressed 51.79% of the total number of sentences (the rest of the sentences was left in their full form). When a sentence was compressed it was shortened on average to 63.64% of its original length (measured by the number of words). The correctness was a bit lower than 2. It can imply that on average some information was removed but it was not so important. Notice that even some human compressions were marked 2 - so the annotators advised to remove some information that is not so important for the full sentence sense. A few gramatical problems arose, however, 7% does not represent a serious problem. There were high correlations between annotator assessments - 76%, 81% and 85%.

In table 5.2 I present the evaluation by ROUGE scores. For this evaluation

| **Summarizer setting** | **ROUGE-1** | **ROUGE-2** | **ROUGE-SU4** | **ROUGE-L** |
|---|---|---|---|---|
| **F3.4woc** | 0.40805 | 0.19722 | 0.15728 | 0.37878 |
| **F5.1woc** | 0.40271 | 0.19402 | 0.15414 | 0.37621 |
| **F5.1wc** | 0.41211 | 0.19975 | 0.15757 | 0.38598 |

Table 5.2: The ROUGE evaluation of LSA-based sentence compression. F3.4woc - formula (3.4) without compression, F5.1woc - formula (5.1) without compression, F5.1wc - formula (5.1) with compression.

I used the whole DUC 2002 corpus - totally 567 documents. I compared 3 settings of the system. The first does not apply sentence compression and uses the formula (3.4) to assign a score to a sentence. The only difference of the second system setting is that it takes into account the sentence length adjustment - formula (5.1) but it still extracts full sentences. And finally, the last system use also formula (5.1) but it apply sentence compression.

Here we can observe that when we added the normalization by sentence length to the formula for assigning the sentence quality, ROUGE scores came down. It is caused by decreasing the score of significant sentences with many unimportant subordinate clauses. However, when we include sentence compression that is able to identify and remove these clauses the summary quality goes up. Long significant sentences are included but shortened and we can add more sentences to the summary because it is usually limited by the number of words.

# Chapter 6

# LSA-based Summary Evaluation

The LSA's ability to capture the most important topics is used by two summarization evaluation metrics I proposed. The idea is that a summary should contain the most important topic(s) of the reference document (e.g., full text or abstract). The method evaluates a summary quality via content similarity between a reference document and the summary like other content-based evaluation measures do. The matrix $U$ of the SVD breakdown (see section 3) represents the degree of term importance in salient topics. The method measures the similarity between the matrix $U$ derived from the SVD performed on the reference document and the matrix $U$ derived from the SVD performed on the summary. To appraise this similarity I have proposed two measures.

## 6.1 Main Topic Similarity

The first measure compares first left singular vectors of the SVD performed on the reference document and the SVD performed on the summary. These vectors correspond to the most important word pattern in the reference text and in the summary. I call it the *main topic*. The cosine of the angle between

the first left singular vectors is measured. The vectors are normalized, thus we can use the following formula:

$$cos\varphi = \sum_{i=1}^{n} ur_i \cdot us_i, \tag{6.1}$$

where $ur$ is the first left singular vector of the reference text SVD, $us$ is the first left singular vector of the summary SVD[1] and $n$ is the number of unique terms in the reference text.

## 6.2 Term Significance Similarity

The second LSA measure compares a summary with the reference document from an angle of $r$ most salient topics. The idea behind it is that there should be the same important topics/terms in both documents. The first step is to perform the SVD on both the reference document and summary matrices. Then we need to reduce the dimensionality of the documents' SVDs to leave only the important topics there.

### Dimensionality Reduction

If we perform SVD on $m$ x $n$ matrix we can look at the new dimensions as descriptions of document's topics or some sort of pseudo sentences. They are linear combinations of original terms. The first dimension corresponds to the most important pseudo sentence[2]. From the summarization point of view, the summary contains $r$ sentences, where $r$ is dependent on the summary length. Thus, the approach of setting the level of dimensionality reduction $r$ is the following:

- We know what percentage of the reference document the summary is - $p$%. The length is measured in the number of words. Thus,

[1] Values which correspond to particular terms are sorted by the reference text terms and instead of missing terms there are zeroes.

[2] It is the first left singular vector.

$p = min(sw/fw * 100, 100)$, where $sw$ is the number of words in the summary and $fw$ is the number of words in the reference text.[3]

- We reduce the latent space to $r$ dimensions, where $r = min(sd, p/100 * rd$, $sd$ is the total number of dimensions in the summary SVD and $rd$ is the total number of dimensions in the reference text SVD. In our case, the total number of dimensions is the same as the number of sentences.

The evaluator can thus automatically determine the number of significant dimensions dependent on the summary/reference document length ratio.

Example: The summary contains 10% of full text words, 4 sentences and the full text contains 30 sentences. Thus, SVD creates a space of 30 dimensions for the full text and we choose the 3 most important dimensions ($r$ is set to 3).

However, $p$% dimensions contain more than $p$% information. For instance, when evaluating a 10% summary against its source text, the 10% most important dimensions used for evaluation deal with 40% of source document information, or when evaluating 30% summary, the top 30% dimensions deal with 70% of source document information (see section 3.1.2).

### Term significances

After obtaining the reduced matrices we compute the significance of each term in the document latent space. Firstly, the components of matrix $U$ are multiplied by the square of its corresponding singular value that contains the topic significance as discussed above. The multiplication favours the values

[3]When the reference document is represented by an abstract, the *min* function arranges that even if the summary is longer then the reference document, $p$ is 100%, (e.g., we take all topics of the abstract).

that correspond to the most important topics. The result is labeled $C$:

$$C = \begin{pmatrix} u_{1,1}\sigma_1^2 & u_{1,2}\sigma_2^2 & \dots & u_{1,r}\sigma_r^2 \\ u_{2,1}\sigma_1^2 & u_{2,2}\sigma_2^2 & \dots & u_{2,r}\sigma_r^2 \\ \dots & \dots & \dots & \dots \\ u_{m,1}\sigma_1^2 & u_{m,2}\sigma_2^2 & \dots & u_{m,r}\sigma_r^2 \end{pmatrix}. \tag{6.2}$$

Then, we take matrix $C$ and measure the length of each row vector:

$$|c_i| = \sqrt{c_{i,1}^2 + c_{i,2}^2 + \ldots + c_{i,r}^2} \tag{6.3}$$

This corresponds to the importance of each term within the $r$ most salient topics. From these lengths, we compute the resultant term vector $rtv$:

$$rtv = \begin{bmatrix} |c_1| \\ |c_2| \\ \dots \\ |c_n| \end{bmatrix} \tag{6.4}$$

Vector $rtv$ is further normalized. The process is performed for both reference and summary documents. Thus, we get one resultant vector for the reference document and one for the summary. Finally, the cosine of the angle between the resultant vectors, which corresponds to the similarity of the compared documents, is measured.

## 6.3 Correlations on DUC Data

To assess the usefulness of the LSA-based evaluation measures, I used the DUC-2002 corpus. This gave me the opportunity to compare the quality of the systems participating in DUC from an angle of several evaluation measures. Furthermore, I could compare the system rankings provided by the LSA measures against human rankings.

In 2002 the family of ROUGE measures had not yet been introduced. However, now, I could perform ROUGE evaluation. This gave me another interesting comparison of standard evaluation measures with the LSA-based ones.

I included in the computation ROUGE-1, ROUGE-2, ROUGE-SU4, ROUGE-L, Cosine similarity, top $n$ keywords and our two measures - Main topic similarity and Term significance similarity. The systems were sorted from each measure's point of view. Then, I computed the Pearson correlation between these rankings and human ones.

### 6.3.1 Term Weighting Schemes for SVD

Firstly, I needed to find a suitable term weighting scheme for the LSA-based evaluation measures. I analysed the same schemes as for the LSA-based summarization method (see section 3.2.3).

All combinations of these local and global weights for the LSA-based evaluation methods are compared in figures 6.1 (reference document is an abstract) and 6.2 (reference document is the full text).

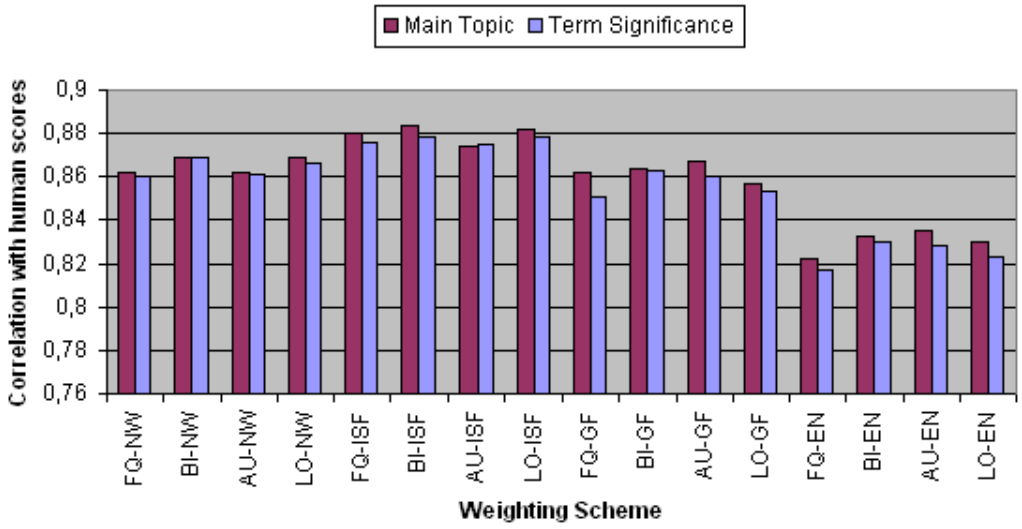


Figure 6.1: The influence of different weighting schemes on the evaluation performance measured by the correlation with human scores. The meaning of the letters is as follows: [Local weight]-[Global weight]. Reference document is abstract.

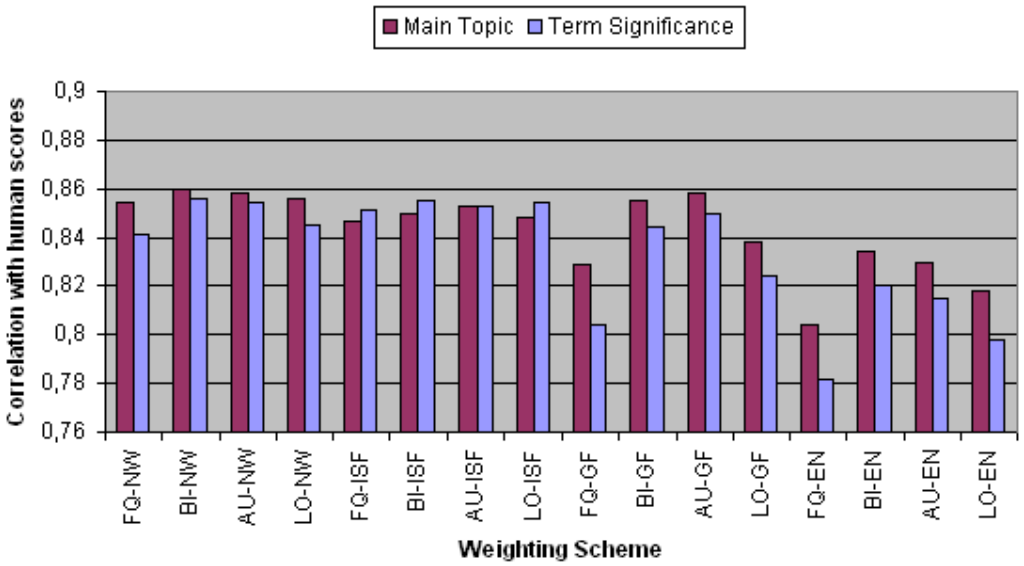


Figure 6.2: The influence of different weighting schemes on the evaluation performance measured by the correlation with human scores. The meaning of the letters is as follows: [Local weight]-[Global weight]. Reference document is full text.

We can observe that the best performing weighting scheme when comparing summaries with abstracts was binary local weight and inverse sentence frequency global weight. When comparing summaries with full texts, a simple Boolean local weight and no global weight performed the best. However, not all of the differences are statistically significant. The best performing weightings are used for the comparison of evaluators in tables 6.1 and 6.2.

### 6.3.2 Baseline Evaluators

I included in the evaluation two baseline evaluators. The first one - cosine similarity - was described in section 2.3.3. The second baseline evaluator compares the set of keywords of a systems summary and that of its reference document. The most frequent lemmas of words in the document which do not occur in stop-word list were labeled as keywords. The top $n$ keywords

were compared in the experiments - see figure 6.3. The best performing value of $n$ for the 100-word summaries was 30. This setting is used in tables 6.1 and 6.2.

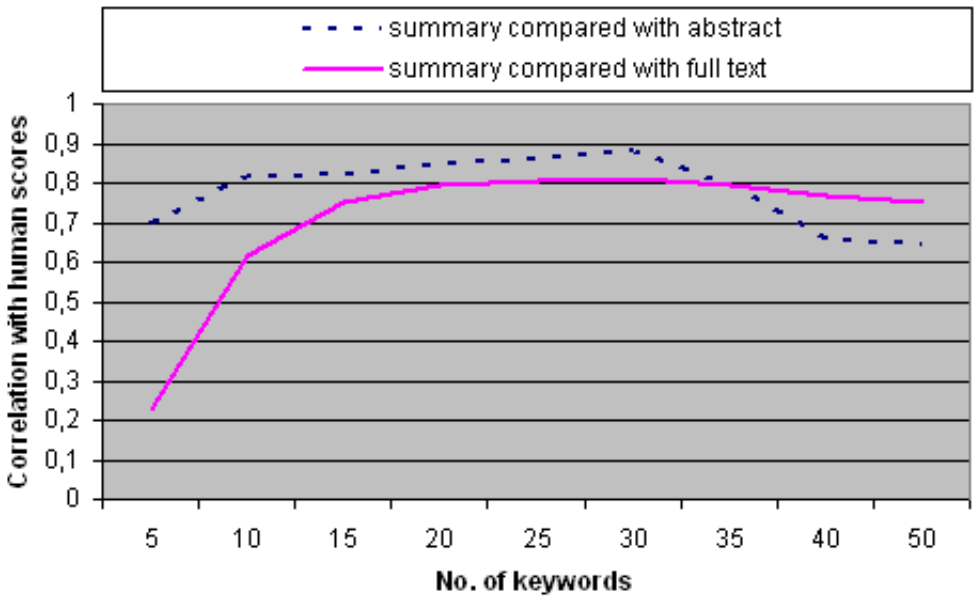


Figure 6.3: The dependency of the performance of the keyword evaluator on the number of keywords.

### 6.3.3 Summary and Abstract Similarity

In this experiment we measured the similarity of summaries with human abstracts from the angle of the studied evaluators. The correlation results can be found in table 6.1.

We can observe that when comparing summaries with abstracts, ROUGE measures demonstrate the best performance. The measures showing the best correlation were ROUGE-2 and ROUGE-SU4, which is in accord with the latest DUC observations. For the LSA measures we obtained worse correlations. The first reason is that abstractors usually put in the abstract some words

| Score | Correllation |
|---|---|
| ROUGE-2 | **0.96119** |
| ROUGE-SU4 | 0.93897 |
| ROUGE-L | 0.91143 |
| ROUGE-1 | 0.90317 |
| LSA - Main Topic Similarity | 0.88206 |
| Keywords | 0.88187 |
| LSA - Term Significance Similarity | 0.87869 |
| Cosine similarity | 0.87619 |

Table 6.1: Correlation between evaluation measures and human assessments - reference document is an abstract.

not contained in the original text and this can make the main topics of the abstract and an extractive summary different. Another reason is that the abstracts were sometimes not long enough to find the main topics and therefore to use all terms in evaluation, as ROUGE does, results in better performance. The differences between LSA measures and baselines were not statistically significant at 95% confidence.

### 6.3.4 Summary and Full Text Similarity

In the second experiment I took the full text as a reference document. I compared Cosine similarity, top $n$ keywords, and LSA-based measures with human rankings. ROUGE is not designed for comparison with full texts. I report the results in table 6.2. These results showed that the simple Cosine similarity did not correlate well with human rankings. Here we can see the positive influence of dimensionality reduction. It is better to take only the main terms/topics for evaluation instead of all, as Cosine similarity does. Keyword evaluator holds a solid correlation level. However, the LSA measures

| **Score** | **Correllation** |
|---|---|
| LSA - Main Topic Similarity | **0.85988** |
| LSA - Term Significance Similarity | 0.85573 |
| Keywords | 0.80970 |
| Cosine similarity | 0.27117 |

Table 6.2: Correlation between evaluation measures and human assessments - reference document is a full text.

correlate even significantly better. The difference between the LSA measures is not statistically significant at 95% confidence and, therefore, it is sufficient to use the simpler Main topic similarity. The results suggest that the LSA-based similarity is appropriate for the evaluation of extractive summarization where abstracts are not available.

# Chapter 7

# Using Summaries in Multilingual Searching

Multilingual aspects are increasing in importance in text processing systems. We proposed possible solutions to new problems arising from these aspects [63]. A multilingual system will be useful in digital libraries, as well as the web environment.

The contribution deals with methods of multilingual searching enriched by the summarization of retrieved texts. This is helpful for a better and faster user navigation in retrieved results. I also present our system, MUSE (Multilingual Search and Extraction). The core of our multilingual searching approach is the EuroWordNet thesaurus (EWN) [66]. An implementation of the summarization algorithm described in this thesis provides the extraction task.

The searching part and evaluation of MUSE was designed and performed by Michal Toman and I am responsible for summarization.

MUSE consists of several relatively self contained modules, namely language recognition, lemmatization, word sense disambiguation, indexing, searching and user query expansion, and finally, summarization. Methods based on the

frequency of specific characters and words are used for language recognition. All terms are lemmatized and converted into the internal EWN format - Inter Lingual Index (ILI). The lemmatization module executes mapping of document words to their basic forms, which are generated by the ISPELL utility package [67].[1] Documents are then indexed by the indexing module. The main search engine is based on the modified vector retrieval model with the TF-IDF scoring algorithm (see section 7.6). It uses an SQL database as an underlying level to store indexed text documents, EWN relations and lemmatization dictionaries for each language. Queries are entered in one of the languages (currently Czech and English). However, it should be noted that the principles remain the same for an arbitrary number of languages. Optionally, the query can be expanded to obtain a broader set of results. EWN relations between synsets (sets of synonymous words) are used for query expansion. Hypernym, holonym, or other related synsets can enhance the query. The expansion setting is determined by user's needs.

The amount of information retrieved by the search engine can be reduced to enable the user to handle this information more effectively. MUSE uses the summarizer for presenting summaries of retrieved documents. Moreover, we study the possibility of speeding up document retrieval by searching in summaries, instead of in full texts.

## 7.1 MUSE architecture

To verify our solution, we created a prototype system. It demonstrates possibilities, advantages, and disadvantages of the approach. MUSE was designed as a modular system, and it consists of relatively independent parts. The overall description is shown in figure 7.1. The system contains five logical

[1] The module complexity depends on the specific language. We experimented with both morphologically simple (English) and complicated (Czech) languages. The Czech language requires a morphological analysis [68].

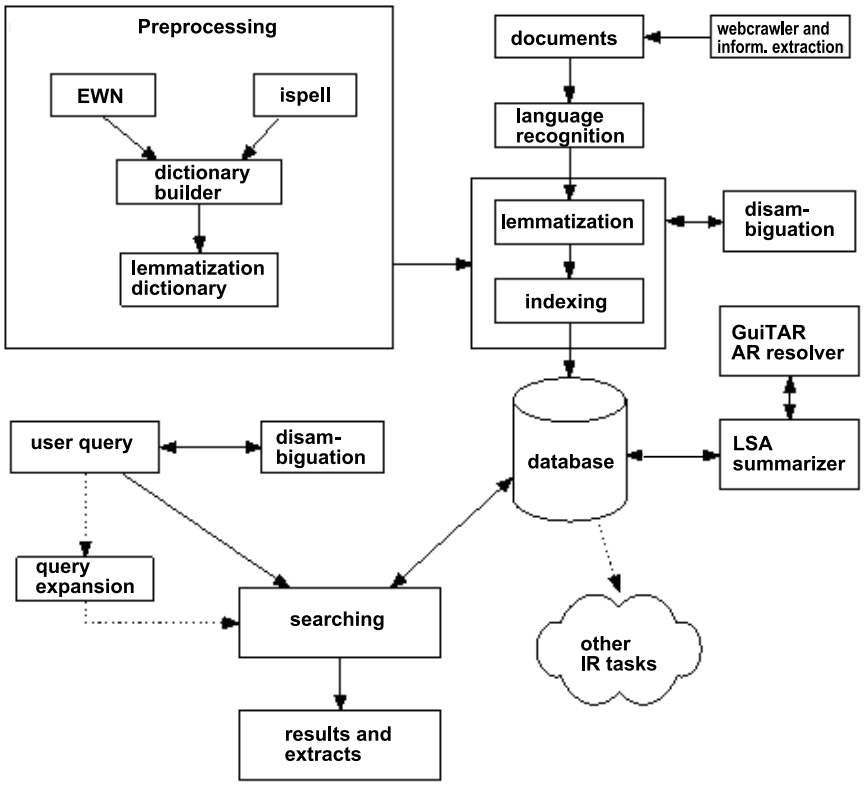


Figure 7.1: MUSE architecture.

parts: preprocessing, lemmatization, indexing, a summarizer, and searching. It is necessary to acquire a high quality lemmatization dictionary for indexing and successive processing. This task is covered by the preprocessing module. It processes the word forms derived from ISPELL, and creates a lemmatization dictionary for each language. A morphological analyzer, which improves lemmatization precision, is applied to the Czech language. Basic word forms are mapped on EWN synsets, and the resulting dictionary is used in the indexing module for document transformation into the language independent form. The summarization module can be considered a breakthrough part of the system. It transforms full documents into shorter ones with a minimal information loss. It is very important for an easier user's navigation in a larger number of documents. The summarization method was described in

chapters 3, 4, and 5. The main part of MUSE is the searching module enriched by query expansion. Terms can be expanded in different ways (e.g. hypernyms, hyponyms).

## 7.2 Language Recognition

The multilingual processing implies a need for a language recognition module. Its task is not only to distinguish the language but to recognize the text coding as well. There are many language recognition approaches. We used two of them.

The first one results from a different letter frequency in languages. Except for language determination, letters are also used for text coding recognition. For each language and document, a binary vector is created where ones are at the position of characteristic letters (e.g. letters with diacritics). The document vectors are compared with the language vectors by the well-known Hamming distance measure (i.e. the number of disagreements between two vectors).

The second method is based on a stop-word list. The list includes words not carrying any particular information. They are highly specific for each language. English stop-words are for example: a, an, the, of, from, at, is, etc. Finally, the module chooses the correct lemmatization dictionary, according to the recognized language.

The comparison of both methods was discussed in [20].

## 7.3 Lemmatization

Lemmatization transforms words into their basic forms. Dictionary lemmatization was used because of its simplicity and generality. The lemmatization dictionary was created by the extraction of word forms from the Ispell program (see [67]). Thanks to ISPELL, we were able to generate all existing word

forms from stems stored in the ISPELL dictionary. We considered the stem a basic form of the word. This works perfectly in the case of English, but some problems appear in Czech. In general, languages with a rich flex are more difficult to process in general. We used a Czech morphological analyzer [68] to overcome this problem. In the case of English, lemmatization is relatively simple. It is possible to apply an algorithmic method - Porter's algorithm.

## 7.4 Word Sense Disambiguation

Word sense disambiguation (WSD; [62]) is a necessary module in most of the natural language processing (NLP) systems. It allows distinguishing of the meaning of a text or a message. Polysemous words may occur in any language. Ambiguity causes many problems, which may result in the retrieval of irrelevant documents. Disambiguation is a relatively self-contained task, which has to be carried out within the indexing. It has to distinguish between words which have identical basic forms but different meanings. The decision about the right meaning requires the knowledge of the word's context.

We implemented a disambiguation method based on the Bayesian classifier. Each meaning of the word was represented by a class in the classification task. The total number of meanings for each ambiguous word was obtained from the EWN thesaurus. Our analysis discovered that nearly 20% of English words are ambiguous. This shows the importance of disambiguation in all NLP tasks. In the course of our implementation, some heuristic modifications were tested with the aim to refine the disambiguation accuracy, as discussed in [20].

## 7.5 Indexing

We introduced a bit of an unusual approach to indexing. For language independent processing, we designed a technique which transforms all the mul-

tilingual texts into an easily processed form. The EWN thesaurus was used for this task (see [66]). It is a multilingual database of words and relations for most European languages. It contains sets of synonyms - synsets - and relations between them. A unique index is assigned to each synset; it interconnects the languages through an inter-lingual-index in such a way, that the same synset in one language has the same index in another one. Thus, cross-language searching can easily be performed. We can, for example, enter a query in English, and the system can retrieve Czech documents as a result, and vice versa.

With EWN, completely language independent processing and storage can be carried out, and moreover, synonyms are identically indexed.

## 7.6 Searching

Our system deals with the representation, storage, and presentation of multilingual information sources. Documents are transformed into the internal language independent form. This is done in the lemmatization and indexing phase. Each document can be described by a set of indexes, representing its main topics. Such indexes can be determined in a fully automatic way. A weight is assigned to each word. It implies its expected semantic significance within the whole document. This framework is proposed to accomplish partial matching based on the similarity degree of a document and a query. Moreover, term weighting and scoring according to user queries enables the sorting of retrieved documents according to their relevance.

We use a slightly modified TF-IDF (Term Frequency - Inverse Document Frequency) principle for the term scoring algorithm. The weight of the term $t_i$ in the document $d_j$ denoted $w_{ij}$ is the product $w_{ij} = tf_{ij} \cdot idf_i$, where $tf_{ij}$ is the term frequency of $t_i$ in $d_j$ and $idf_i$ is the inverted document frequency of $t_i$ in the corpus D.

A resultant candidate set is computed for each term in the user query. The set

is scored by the relevance measured with regard to the term. If more terms are used in the query, candidate sets' intersection or union is performed according to the logical operation in the user query (AND or OR). In the case of intersection, document weights are adjusted by simple summation of candidate values.

From the user's point of view, the searching process is intuitive. The user query is interpreted as a set of terms describing the desired result set. Query terms are lemmatized and indexed into an internal form, and the query can be expanded with the use of EWN. This step is optional. Each word from the query should be disambiguated to prevent a retrieval of irrelevant documents. Afterwards, the searching is performed, and the results are displayed. For each document, a full text and its summary are available. All operations are performed upon a relational database. It contains summarized data, the lemmatization dictionary, and the EWN thesaurus.

## 7.7 Query Expansion

It is not simple to create a query which fully covers the topic of our interest. We introduced a query expansion module that provides a simple, yet powerful, tool for changing the queries automatically. The expansion can be done in different ways. Synsets' interconnections were obtained from the EWN thesaurus for this purpose. We used 10 different relationships. They are presented together with their weights and types in table 7.1. The weights are used in the TF-IDF scoring algorithm. They were subjectively designed according to the relationship between the query term and its expansion.

A query expansion can significantly improve the system recall. It will retrieve more documents, which are still relevant to the query (see Results section). The user is able to restrict the expansion level to any combination of similar, subordinate and superordinate words. The expanding terms have a lower weight than those entered directly by the user.

| Relationship | Relationship weight | Relation type |
|---|---|---|
| **similar to** | 8 | Similar |
| **be in state** | 6 | Similar |
| **also see** | 8 | Similar |
| **derived** | 3 | Similar |
| **hypernym** | 2 | Superordinates |
| **Holo portion** | 3 | Superordinates |
| **Holo part** | 3 | Superordinates |
| **Holo member** | 3 | Superordinates |
| **Particle** | 3 | Subordinates |
| **Subevent** | 2 | Subordinates |

Table 7.1: Expansion relationships

## 7.8 Summarization

The LSA-based summarizer presented in this work is responsible for the summarization task in MUSE. The main accent was put on the multilingualism of the summarizer. LSA is a totally language independent process. The only difference in processing different languages is the stop-word list and lemmatization. In anaphora resolution, the situation is different. So far, we have enriched our summarization method with anaphoric knowledge only for texts written in English. Now, we plan to create an anaphora resolver for the Czech language in which we intend to implement similar resolution algorithms as the ones in GUITAR.

## 7.9 Experiments with MUSE

We created a testing corpus which includes Czech and English texts, in particular - press articles from ČTK and Reuters news agencies. The corpus consists of a total number of 82000 Czech and 25000 English articles. They were chosen from 5 classes - weather, sport, politics, agriculture, and health. A 100-word extract was created for each document.

Retrieval results are presented in table 7.2. We show a total number of retrieved documents and the number of documents that are relevant in the top 30. The average precision was 94.3%. The second table (7.3) contains similar figures in which query expansion was applied. When we include similar, subordinate and superordinate relations in the query, a larger set of documents is obtained. On the contrary the precision went a bit down. However, a high precision is hold anyway.

| **Query** | **Total number of retrieved documents** | **Relevant documents in top 30** | **Precision** |
|---|---|---|---|
| **formula & one & champion** | 88 | 27 | 90% |
| **terorismus & útok** | 265 | 29 | 96.7% |
| **white & house & president** | 2393 | 29 | 96.7% |
| **povodeň & škody** | 126 | 29 | 96.7% |
| **cigarettes & health** | 366 | 25 | 83.3% |
| **rozpočet & schodek** | 2102 | 30 | 100% |
| **plane & crash** | 221 | 29 | 96.7% |
| **Average** | 790 | 28.3 | 94.3% |

Table 7.2: Relevance of documents retrieved by MUSE (without query expansion).

| Query | Total number of retrieved documents | Relevant documents in top 30 | Precision |
|---|---|---|---|
| **formula & one & champion** | 465 | 26 | 86.7% |
| **terorismus & útok** | 300 | 29 | 96.7% |
| **white & house & president** | 6116 | 23 | 76.7% |
| **povodeň & škody** | 126 | 29 | 96.7% |
| **cigarettes & health** | 393 | 25 | 83.3% |
| **rozpočet & schodek** | 2174 | 30 | 100% |
| **plane & cash** | 2306 | 29 | 96.7% |
| **Average** | 1697 | 27.3 | 91.0% |

Table 7.3: Relevance of documents retrieved by MUSE (with all query expansions - similar, subordinate, superordinate relations).

We compared the retrieval performance of the GOOGLE approach, the widely accepted search method, and that of our MUSE system. Our approach and the state-of-the-art GOOGLE search engine are compared in tables 7.4 and 7.5. In the former table we show the MUSE performance when query expansion was disabled and in the latter when all possible query expansion levels were used. We measured the intersection between MUSE and GOOGLE in the first 10 and 30 retrieved documents on the same query. In the first 10 documents there was an intersection of 70% and in the first 30 documents 57.5% were common. Enabling the query expansion led to a small decrease in precision.

We also tested the influence of summarization on the quality of the retrieved results. To verify the influence, we performed the same queries on both the full text and summarized corpus. Searching in summaries improves the response times of the system significantly (table 7.8), without any remarkable loss in precision (table 7.7). The number of relevant documents in the top

| Query | Intersec. in top 10 (percentage) | Intersec. in top 30 (percentage) |
|---|---|---|
| **formula & one** | 9 (90%) | 25 (83.3%) |
| **national & park** | 3 (30%) | 9 (30%) |
| **religion & war** | 7 (70%) | 20 (66.7%) |
| **water & plant** | 7 (70%) | 11 (36.7%) |
| **hockey & championship** | 7 (70%) | 20 (66.7%) |
| **traffic & jam** | 6 (60%) | 18 (60%) |
| **heart & surgery** | 7 (70%) | 16 (53.3%) |
| **weather & weekend** | 10 (100%) | 19 (63.3%) |
| **Average** | 7.0 (70%) | 17.3 (57.5%) |

Table 7.4: Intersection with GOOGLE (query expansion disabled).

30 retrieved results is basically the same (table 7.7). The intersection of the documents retrieved by searching in both corpuses is 37.5%.

In conclusion, the results show approximately 70% similarity with the GOOGLE approach in the top 30 retrieved documents. However, MUSE has several advantages in comparison with GOOGLE. Firstly, it respects a multilingual environment. If we enter a query in English, GOOGLE is not able to find any relevant documents written in another language. On the contrary, MUSE will retrieve both English and Czech documents. Secondly, synonyms are considered equal in the searching process. Moreover, we provide query expansion, and finally, a part of the system is an automatic summarizer. Searching in summaries is reasonably precise and six times faster.

There is a problem related to the actual EWN structure - a missing word's equivalents in non-English languages. This can cause some difficulties in cross-language searching. As EWN is gradually being completed, this problem will disappear.

| **Query** | **Intersec. in top 10 (percentage)** | **Intersec. in top 30 (percentage)** |
|---|---|---|
| **formula & one** | 9 (90%) | 24 (80%) |
| **national & park** | 3 (30%) | 9 (30%) |
| **religion & war** | 7 (70%) | 20 (66.7%) |
| **water & plant** | 4 (40%) | 6 (20%) |
| **hockey & championship** | 7 (70%) | 20 (66.7%) |
| **traffic & jam** | 6 (60%) | 16 (53.3%) |
| **heart & surgery** | 7 (70%) | 17 (56.7%) |
| **weather & weekend** | 10 (100%) | 16 (53.3%) |
| **Average** | 6.6 (66%) | 16.0 (53.3%) |

Table 7.5: Intersection with GOOGLE (query expansion enabled).

| **Query** | **Summary and fulltext intersection in the first 30 retrieved documents** |
|---|---|
| **formula & one** | 21 (70%) |
| **national & park** | 10 (33.3%) |
| **religion & war** | 4 (13.3%) |
| **water & plant** | 7 (23.3%) |
| **hockey & championship** | 16 (53.3%) |
| **traffic & jam** | 11 (36.7%) |
| **heart & surgery** | 16 (53.3%) |
| **weather & weekend** | 5 (16.6%) |
| **Average** | 11.3 (37.5%) |

Table 7.6: Intersection between searching in full texts and summaries.

| **Query** | **Summary relevance in the first 30 retrieved documents** |
|---|---|
| **formula & one** | 26 (86.6%) |
| **national & park** | 20 (66.7%) |
| **religion & war** | 26 (86.6%) |
| **water & plant** | 14 (46.7%) |
| **hockey & championship** | 29 (96.6%) |
| **traffic & jam** | 23 (76.7%) |
| **heart & surgery** | 30 (100%) |
| **weather & weekend** | 28 (93.3%) |
| **Average** | 24.5 (81.7%) |

Table 7.7: Relevance of documents retrieved by searching in summaries.

| **Query** | **Searching time in full texts [ms]** | **Searching time in full texts [ms]** |
|---|---|---|
| **formula & one** | 6359 | 984 |
| **national & park** | 8797 | 1312 |
| **religion & war** | 6172 | 922 |
| **water & plant** | 8734 | 1015 |
| **hockey & championship** | 1938 | 547 |
| **traffic & jam** | 3656 | 688 |
| **heart & surgery** | 5656 | 1031 |
| **weather & weekend** | 4125 | 703 |
| **Average** | 5680 | 900 |
| **Speed-up** | **6.3x** | |

Table 7.8: The comparison of searching times.

# Chapter 8

# Conclusion

This chapter summarizes the current state of work. At the end I outline future research directions.

## 8.1 Current State of Work

I presented a summarization method that is based on latent semantic analysis. The analysis can capture the main topics of the processed document. The method takes advantage of this property. The document is firstly converted into the SVD input matrix format. The matrix is then decomposed into three final matrices. They contain information about topics of the document. Moreover, we can find there to what extent each sentence contains each topic. To decide how many topics are considered important I propose an automatic dimensionality reduction algorithm. The longer is the summary, compared to the original text, the more topics it contains. In addition, only a few SVD dimensions need to be computed and this makes the time needed for summary creation shorter. I presented an analysis how much information is contained in the top $p$% dimensions. For instance, a summary than contains 10% of source text words deals with 40% of document information, or 30% summary deals with 70% of document information. In the final stage, sen-

tences are sorted according to how they contain the important topics. The evaluation revealed the appropriate weighting scheme for the creation of the input SVD matrix. Furthermore, the comparison with other summarization systems shows that this novel method is comparable with the best systems in the DUC evaluation corpus. However, the advantage of the proposed LSA-based method is that its core is completly language independent.

The basic lexical summarization was enhanced by the knowledge of anaphors. I proposed the addition method that is able to determine document topics more accurately then the simple substitution method does. Moreover, I invented a summary reference checking method that can check and correct the false anaphoric references in the summary. The improved system was evaluated as significantly better than the basic lexical-based one.

Then I proposed a sentence compression algorithm. Its aim is to mark unimportant clauses in long sentences. These clauses can be then removed. This makes the summary more concise and shorter. The evaluation showed a quality gap between human compressions and system ones. However, the performance is significantly over the baseline.

Furthermore, I proposed an LSA-based evaluation method that measures how much are the most important topics of the reference document contained in the summary. Experiments showed that the best usage of this method was to measure the similarity between source text and its summary when abstracts were not available.

At last, the practical usage of the summaries in the experimental searching system MUSE was shown. The response times and searching quality of full texts' and summaries' data sets were compared. The precision of searching in summaries was at the same level as that of searching in original texts. The retrieval speed was boosted by more than six times when full documents we substituted by summaries.

## 8.2 Future Work

My future work will be concerned mainly with multi-document summarization. As in the last DUC volumes, the summary is not created from the only one source text but from a cluster of documents. The single-document LSA approach can be easily extended to process multiple documents by including all sentences in a cluster of documents in the SVD input matrix. The latent space would be then reduced to $r$ dimensions according to the dimensionality reduction approach as done currently (see section 3.1.2). The sentence selection approach can be used as well; however, care has to be taken to avoid including very similar sentences from different documents. Therefore, before including a sentence in the summary we have to check if there are any sentences whose similarity with the observed one is above a given threshold. (The easiest way of measuring the similarity between two sentences is to measure the cosine of the angle between them in the term space.)

Cross-document coreference, on the other hand, is a fairly different task from within-document coreference, as even in the case of entities introduced using proper names one cannot always assume that the same object is intended, let alone in the case of entities introduced using definite descriptions. We are currently working on this problem.

The aim is to produce a system that would be able to compete in future DUC competition.

The core of the summarization method presented here is language independent. I tested the system with English texts from DUC corpus. So far, there is no corpus of Czech documents annotated for summarization. Therefore, I will work on the creation of a multilingual corpus annotated for multi-document summarization.

With multilingual processing anaphora resolution becomes more complicated. The creation of anaphora resolution system for Czech language can be another research direction.

# Bibliography


[1] Laura Alonso, Irene Castellón, Salvador Climent, Maria Fuentes, Lluís Padró, and Horacio Rodríguez: **Approaches to Text Summarization: Questions and Answers**. In *Ibero-American Journal of Artificial Intelligence, No. 20*, pp. 34–52, Spain, 2003.

[2] Regina Barzilay, Michael Elhadad: **Using Lexical Chains for Text Summarization**. In *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pp. 10–17, Madrid, Spain, 1997.

[3] Breck Baldwin, Thomas S. Morton: **Dynamic coreference-based summarization**. In *Proceedings of EMNLP*. Granada, Spain, 1998.

[4] Phyllis B. Baxendale: **Man-made Index for Technical Literature - an experiment**. In *IBM Journal of Research Development, Vol. 2, No. 4*, pp. 354–361, 1958.

[5] Sabine Bergler, René Witte, Michelle Khalife, Zhuoyan Li, Frank Rudzicz: **Using knowledge-poor coreference resolution for text summarization**. In *Proceedings of DUC*. Edmonton, Canada, 2003.

[6] Michael W. Berry: **SVDPACKC (Version 1.0) User's Guide**, *University of Tennessee Tech. Report CS-93-194*, 1993 (Revised October 1996). See also http://www.netlib.org/svdpack/index.html.

[7] Michael W. Berry, Susan T. Dumais, Gavin W. O'Brien: **Using linear algebra for intelligent IR**. In *SIAM Review, 37(4)*, 1995.

[8] Sergey Brin and Lawrence Page: **The anatomy of a large-scale hypertextual Web search engine**. In *Computer Networks and ISDN Systems*, 30, pp. 1–7, 1998.

[9] Branimir Boguraev and Christopher Kennedy: **Salience-based content characterization of text documents**. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, pp. 2–9, Madrid, Spain, 1997.

[10] Kalina Bontcheva, Marin Dimitrov, Diana Maynard, Valentin Tablan, Hamish Cunningham: **Shallow methods for named entity coreference resolution**. In *Chaĭnes de références et résolveurs d'anaphores, workshop TALN 2002*. Nancy, France, 2002.

[11] Eugene Charniak: **A maximum-entropy-inspired parser**. In *Proceedings of NAACL*. Philadelphia, US, 2000.

[12] Freddy Y. Y. Choi, Peter Wiemer-Hastings, Johanna Moore: **Latent semantic analysis for text segmentation**. In *Proceedings of EMNLP*, pp. 109–117, Pittsburgh, USA, 2001.

[13] Chris H. Q. Ding: **A probabilistic model for latent semantic indexing**. In *Journal of the American Society for Information Science and Technology, 56(6)*, pp. 597–608, 2005.

[14] Susan T. Dumais: **Improving the retrieval of information from external sources**. In *Behavior Research Methods, Instruments & Computers, 23(2)*, pp. 229–236, 1991.

[15] H. P. Edmundson: **New methods in automatic extracting**. In *Journal of the ACM, Vol. 16, No. 2*, pp. 264–285, 1969.

[16] Yihong Gong, Xin Liu: **Generic text summarization using relevance measure and latent semantic analysis**. In *Proceedings of ACM SIGIR*. New Orleans, USA, 2002.

[17] Laura Hasler, Constantin Orasan, Ruslan Mitkov. **Building better corpora for summarization**. In *Proceedings of Corpus Linguistics*. Lancaster, United Kingdom, 2003.

[18] Eduard Hovy, Chin-Yew Lin: **Automated Text Summarization in SUMMARIST**. In *Advances in Automatic Text Summarization* edited by Inderjeet Mani and Mark T. Maybury, pp. 81–94, MIT Press, Cambridge MA, USA, 1999.

[19] Jiří Hynek, Karel Ježek: **Practical Approach to Automatic Text Summarization**. In *Proceedings of the ELPUB '03 conference*, pp. 378–388, Guimaraes, Portugal, 2003.

[20] Karel Ježek and Michal Toman: **Documents Categorization in Multilingual Environment**. In *Proceedings of ELPUB*, Leuven, Belgium, 2005.

[21] Hongyan Jing: **Sentence Reduction for Automatic Text Summarization**. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pp. 310–315, Seattle, USA, 2000.

[22] Hongyan Jing, Kathleen McKeown: **Cut and Paste Based Text Summarization**. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 178–185, Seattle, USA, 2000.

[23] Mijail A. Kabadjov, Massimo Poesio, and Josef Steinberger: **Task-Based Evaluation of Anaphora Resolution: The Case of Summarization**. In *RANLP Workshop "Crossing Barriers in Text Summarization Research"*. Borovets, Bulgaria, 2005.

[24] Jon M. Kleinberg: **Authoritative sources in a hyper-linked environment**. In *Journal of the ACM*, 46(5), pp. 604–632, 1999.

[25] Kevin Knight, Daniel Marcu: **Statistics-Based Summarization — Step One: Sentence Compression**. In *Proceeding of The 17th National Conference of the American Association for Artificial Intelligence*, pp. 703–710, Austin, USA, 2000.

[26] Kevin Knight, Daniel Marcu: **Summarization beyond sentence extraction: A probabilistic approach to sentence compression.** In *Artificial Intelligence, 139(1)*, pp. 91–107, 2003.

[27] Alistair Knott, Jon Oberlander, Michael O'Donnell, Chris Mellish: **Beyond elaboration: The interaction of relations and focus in coherent text**. In *T. Sanders, J. Schilperoord, and W. SpoorenText (eds), Text representation: linguistic and psycholinguistic aspects*. John Benjamins, 2001.

[28] Julian M. Kupiec, Jan Pedersen, and Francine Chen: **A Trainable Document Summarizer**. In *Research and Development in Information Retrieval*, pp. 68–73, 1995.

[29] Thomas K. Landauer and Susan T. Dumais: **A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge**. *Psychological Review, 104*, pp. 211–240, 1997.

[30] Chin-Yew Lin, Eduard Hovy: **Automatic evaluation of summaries using n-gram co-occurrence statistics**. In *Proceedings of HLT-NAACL*, Edmonton, Canada, 2003.

[31] Chin-Yew Lin: **Rouge: a package for automatic evaluation of summaries**. In *Proceedings of the Workshop on Text Summarization Branches Out*, Barcelona, Spain, 2004.

[32] Hans P. Luhn: **The Automatic Creation of Literature Abstracts**. In *IBM Journal of Research Development, Vol. 2, No. 2*, pp. 159–165, 1958.

[33] Inderjeet Mani, David House, Gary Klein, Lynette Hirschman, Therese Firmin, Beth Sundheim: **The TIPSTER Summac Text Summarization Evaluation**. In *Proceedings of the 9th Meeting of the European Chapter of the Association for Computational Linguistics*, pp. 77–85, 1999.

[34] Daniel Marcu: **From Discourse Structures to Text Summaries**. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pp. 82–88, Madrid, Spain, 1997.

[35] Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin: **Towards Multidocument Summarization by Reformulation: Progress and Prospects**. In *Proceedings of AAAI-99*, pp. 453–460, Orlando, USA, 1999.

[36] R. Mihalcea and P. Tarau: **Text-rank - bringing order into texts**. In *Proceeding of the Comference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004.

[37] R. Mihalcea and P. Tarau: **An Algorithm for Language Independent Single and Multiple Document Summarization**. In *Proceedings of the International Joint Conference on Natural Language Processing*, Korea, 2005.

[38] Ruslan Mitkov: **Robust pronoun resolution with limited knowledge**. In *Proceedings of COLING*. Montreal, Canada, 1998.

[39] Andrew Morris, George Kasper, Dennis Adams: **The Effects and Limitations of Automatic Text Condensing on Reading Comprehension Performance**. In *Information Systems Research, 3(1)*, pp. 17–35, 1992.

[40] Christoph Mueller, Michael Strube: **MMAX: A tool for the annotation of multi-modal corpora**. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. Seattle, US, 2003.

[41] Gabriel Murray, Steve Renals, Jean Carletta: **Extractive Summarization of Meeting Recordings**. In *Proceedings of Interspeech*. Lisboa, Portugal, 2005.

[42] Ani Nenkova, Rebecca Passonneau: **Evaluating Content Selection in Summarization: The Pyramid Method**. In *Document Understanding Conference* Vancouver, Canada, 2005.

[43] Kenji Ono, Kazuo Sumita, Seiji Miike: **Abstract generation based on rhetorical structure extraction**. In *Proceedings of the 15th International Conference on Computational Linguistics*, volume 1, pp. 344–384, Kyoto, Japan, 1994.

[44] Constantin Orasan, Ruslan Mitkov, Laura Hasler: **CAST: a computer-aided summarisation tool**. In *Proceedings of EACL*, Budapest, Hungary, 2003.

[45] Massimo Poesio, Rosemary Stevenson, Barbara Di Eugenio, Janet Hitzeman: **Centering: A parametric theory and its instantiations**. *Computational Linguistics, 30(3)*, 2004.

[46] Massimo Poesio, Mijail A. Kabadjov: **A general-purpose, off-the-shelf anaphora resolution module: implementation and preliminary evaluation**. In *Proceedings of LREC*. Lisbon, Portugal, 2004.

[47] Massimo Poesio: **Discourse Annotation and Semantic Annotation in the GNOME Corpus**. In *Proceedings of the ACL Workshop on Discourse Annotation*. Barcelona, Spain, 2004.

[48] Massimo Poesio, Mijail A. Kabadjov, Renata Vieira, Rodrigo Goulart, Olga Uryupina: **Does discourse-new detection help definite description resolution?** *Proceedings of the Sixth IWCS*, Tilburg, 2005.

[49] Dragomir Radev, Hongyan Jing, Malgorzata Budzikowska: **Centroid-based summarization of multiple documents**. In *ANLP/NAACL Workshop on Automatic Summarization*, pp. 21–29, Seattle, USA, 2000.

[50] Dragomir R. Radev, Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Hong Qi, Arda Celebi, Danyu Liu, Elliott Drabek: **Evaluation Challenges in Large-scale Document Summarization**. In *Proceeding of the 41st meeting of the Association for Computational Linguistics*, pp. 375–382, Sapporo, Japan, 2003.

[51] Stefan Riezler, Tracy H. King, Richard Crouch, Annie Zaenen: **Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar.** In *Proceedings of HLT/NAACL*, Edmonton, Canada, 2003.

[52] Horacio Saggion: **Génération automatique de résumés par analyse sélective**. In *Ph.D. thesis, University of Montreal*, Canada, 2000.

[53] Horacio Saggion, Dragomir Radev, Simone Teufel, Wai Lam, Stephanie M. Strassel: **Developing infrastructure for the evaluation of single and multi-document summarization systems in a cross-lingual environment**. In *Proceedings of LREC*, Las Palmas, Spain, 2002.

[54] Gerard Salton: **Automatic text processing**. Addison-Wesley Publishing Company, 1988.

[55] Sidney Siegel, N. John Castellan Jr.: **Nonparametric Statistics for the Behavioral Sciences**. Berkeley, CA: McGraw-Hill, 2nd edn., 1988.

[56] Karen Sparck-Jones: **Automatic summarising: factors and directions**. In *Advances in Automatic Text Summarization* edited by Inderjeet Mani and Mark T. Maybury, pp. 1–12, MIT Press, Cambridge MA, USA, 1999.

[57] Caroline Sporleder, Mirella Lapata: **Discourse chunking and its application to sentence compression**. In *Proceedings of HLT/EMNLP*, pp. 257–264, Vancouver, Canada, 2005.

[58] Josef Steinberger, Karel Ježek: **Text Summarization and Singular Value Decomposition**. In *Proceedings of ADVIS'04*, Springer Verlag, 2004.

[59] Josef Steinberger, Mijail A. Kabadjov, Massimo Poesio, Olivia E. Sanchez-Graillet: **Improving LSA-based Summarization with Anaphora Resolution**. In *Proceedings of HLT/EMNLP*, pp. 1-–8, The Association for Computational Linguistics, Vancouver, Canada, 2005.

[60] Josef Steinberger, Karel Ježek: **Sentence Compression for the LSA-based Summarizer**. In *Proceedings of the 7th International Conference on Information Systems Implementation and Modelling*, pp. 141–148, Přerov, Czech Republic, 2006.

[61] Roland Stuckardt: **Coreference-based summarization and question answering: a case for high precision anaphor resolution**. In *International Symposium on Reference Resolution*. Venice, Italy, 2003.

[62] Michal Toman, Karel Ježek: **Modifikace bayesovského disambiguátoru**. In *Znalosti 2005*, VŠB-Technická univerzita Ostrava, 2005.

[63] Michal Toman, Josef Steinberger, Karel Ježek: **Searching and Summarizing in Multilingual Environment**. In *Proceedings of the 10th International Conference on Electronic Publishing*, pp. 257–265 FOI-Commerce, Bansko, Bulgaria, 2006.

[64] Renata Vieira and Massimo Poesio: **An empirically-based system for processing definite descriptions**. *Computational Linguistics, 26(4)*, 2000.

[65] Jen-Yuan Yeh, Hao-Ren Ke, Wei-Pang Yang, I-Heng Meng: **Text summarization using a trainable summarizer and latent semantic analysis**. In *Special issue of Information Processing and Management on An Asian digital libraries perspective*, 41(1), pp. 75–95, 2005.

[66] http://www.illc.uva.nl/EuroWordNet/

[67] http://fmg-www.cs.ucla.edu/fmg-members/geoff/ispell.html

[68] http://quest.ms.mff.cuni.cz/pdt/Morphology_and_Tagging/Morphology/index.html

[69] http://www-nlpir.nist.gov/projects/duc/data.html

[70] http://haydn.isi.edu/SEE/

# Author's Publications

The following papers were published in conference proceedings:

1. Josef Steinberger and Karel Ježek: **Using Latent Semantic Analysis in Text Summarization and Summary Evaluation**. In *Proceedings of the 5th International Conference on Information Systems Implementation and Modelling*, pp. 93–100, MARQ Ostrava, April 2004, ISBN 80-85988-99-2.

2. Josef Steinberger and Karel Ježek: **Text Summarization and Singular Value Decomposition**. In *Proceedings the 3rd International Conference on Advances in Information Systems, Lecture Notes in Computer Science 2457*, pp. 245–254, Springer-Verlag, October 2004, ISBN 3-540-23478-0, ISSN 0302-9743.

3. Josef Steinberger and Karel Ježek: **Hodnocení kvality sumarizátorů textů**. In *Proceedings of ZNALOSTI 2005*, pp. 96-–107, Stará lesná, Slovakia, 2005, ISBN 80-248-0755-6.

4. Mijail A. Kabadjov, Massimo Poesio and Josef Steinberger: **Task-Based Evaluation of Anaphora Resolution: The Case of Summarization**. In *Proceedings of Recent Advances in Natural Language Processing Workshop "Crossing Barriers in Text Summarization Research"*, pp. 18–25, Incoma Ltd., Shoumen, Bulgaria, September 2005, ISBN 954-90906-8-X.

5. Josef Steinberger, Mijail A. Kabadjov and Massimo Poesio: **Improving LSA-based Summarization with Anaphora Resolution**. In *Proceedings of Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing*, pp. 1–8, The Association for Computational Linguistics, Vancouver, Canada, October 2005, ISBN 1-932432-55-8.

6. Josef Steinberger and Karel Ježek: **Sentence Compression for the LSA-based Summarizer**. In *Proceedings of the 7th International Conference on Information Systems Implementation and Modelling*, pp. 141–148, MARQ Ostrava, Přerov, Czech Republic, April 2006, ISBN 80-86840-19-0.

7. Michal Toman and Josef Steinberger, Karel Ježek: **Searching and Summarizing in Multilingual Environment**. In *Proceedings of the 10th International Conference on Electronic Publishing*, pp. 257–265, FOI-Commerce, Bansko, Bulgaria, June 2006, ISBN 954-16-0049-9.

8. Richard F. E. Sutcluffe, Josef Steinberger, Udo Kruschwitz, Mijail A. Kabadjov, Massimo Poesio: **Identifying Novel Information using Latent Semantic Analysis in the WiQA Task at CLEF 2006** In *Proceedings of the CLEF 2006 Workshop*, Alicante, Spain, 2006.

All technical reports from the following list are available on-line at `http://-www.kiv.zcu.cz/publications/techreports.php`.

1. Josef Steinberger: **Text Summarization via Latent Semantic Analysis and Anaphora Resolution**, *Technical Report DCSE/TR-2005-01*, Pilsen, Czech Republic, April 2005.

The following paper was accepted by a journal:

1. Josef Steinberger, Massimo Poesio, Mijail A. Kabadjov and Karel Ježek: **Two Uses of Anaphora Resolution in Summarization**. In *Spe-*

*cial Issue of Information Processing & Management on Summarization*, Elsevier Ltd., 2007.

The following paper is submitted to a journal:

1. Josef Steinberger and Karel Ježek: **Evaluation Measures for Text Summarization**. In *Special Issue of Computing and Informatics on Knowlegde Discovery*, Slovak Academic Press Ltd., 2007.