Computing and Informatics, Vol. 28, 2009, 251–275

EVALUATION MEASURES FOR TEXT SUMMARIZATION

Josef Steinberger, Karel Ježek

Department of Computer Science and Engineering
University of West Bohemia in Pilsen
Univerzitní 8
306 14 Plzeň, Czech Republic
e-mail: {jstein, jezek_ka}@kiv.zcu.cz

Revised manuscript received 20 March 2007

Abstract. We explain the ideas of automatic text summarization approaches and the taxonomy of summary evaluation methods. Moreover, we propose a new evaluation measure for assessing the quality of a summary. The core of the measure is covered by Latent Semantic Analysis (LSA) which can capture the main topics of a document. The summarization systems are ranked according to the similarity of the main topics of their summaries and their reference documents. Results show a high correlation between human rankings and the LSA-based evaluation measure. The measure is designed to compare a summary with its full text. It can compare a summary with a human written abstract as well; however, in this case using a standard ROUGE measure gives more precise results. Nevertheless, if abstracts are not available for a given corpus, using the LSA-based measure is an appropriate choice.

Keywords: Text summarization, automatic extract, summary evaluation, latent semantic analysis, singular value decomposition

1 INTRODUCTION

Automatic text summarization is a process that takes a source text and presents the most important content in a condensed form in a manner sensitive to the user or task needs. The importance of having a text summarization system has been growing with the rapid expansion of information available on-line. The production

of summaries is directly associated with the processes of text understanding and production. Firstly, the source text is read and its content is recognized. Afterwards, the central ideas are compiled in a concise summary.

Summarization is a tough problem because the system has to understand the point of a text. This requires semantic analysis and grouping of the content using world knowledge. However, the system cannot do it without a great deal of world knowledge. Therefore, attempts at performing true abstraction have not been very successful so far. Fortunately, an approximation called extraction is more feasible today. The system simply needs to identify the most important passages of the text to produce an extract. The problem is that the summary is mostly not coherent. Nevertheless, the reader can form an opinion of the original content. Thus at present, most automated systems produce extracts only. Several theories ranging from text linguistics to artificial intelligence have been proposed.

The evaluation of a summary quality is a very ambitious task. Serious questions remain concerning the appropriate methods and types of evaluation. There are a variety of possible bases for the comparison of summarization systems performance. We can compare a system summary to the source text, to a human-generated summary or to another system summary. Summarization evaluation methods can be broadly classified into two categories [37]. In *extrinsic* evaluation, the summary quality is judged on the basis of how helpful summaries are for a given task, and in *intrinsic* evaluation, it is directly based on analysis of the summary. The latter can involve a comparison with the source document, measuring how many main ideas of the source document are covered by the summary or a content comparison with an *abstract* written by a human. The problem of matching the system summary against an "ideal summary" is that the ideal summary is hard to establish. The human summary may be supplied by the author of the article, by a judge asked to construct an abstract, or by a judge asked to extract sentences. There can be a large number of abstracts that can summarize a given document. The intrinsic evaluations can then be broadly divided into *content evaluation* and *text quality evaluation*. Whereas content evaluations measure the ability to identify the key topics, text quality evaluations judge the readability, grammar and coherence of automatic summaries.

Latent semantic analysis (LSA) [19] is a technique for extracting the hidden dimensions of the semantic representation of terms, sentences, or documents, on the basis of their contextual use. We have developed a summarization method that is based on LSA [39]. The idea is to identify the most important topics from the source text and then to choose the sentences with the greatest combined weights across the topics. Afterwards, we enriched the document representation by anaphoric relations [40]. It was found that the addition of anaphoric knowledge leads to improved performance of the summarizer. Later, we went beyond sentence extraction and proposed a simple sentence compression algorithm for our summarizer [41]. Summaries are used in our MUSE (Multilingual Searching and Extraction) system [42]. They enable better and faster user orientation in retrieved results. Nowadays, we investigate additional techniques for producing personalized summaries (i.e., favouring

sentences that either include words from the user query or match the user profile [17]). The fact that LSA can identify the most important topics induces the possibility of using it for summary content evaluation. We present here a summary evaluation method whose idea is that the summary should retain the main topics of the source text.

The rest of the paper is organized as follows: Section 2 covers related work in text summarization. Then the taxonomy of summary evaluation measures is presented (Section 3). Afterwards, we describe the LSA principles and we pay close attention to related work in LSA-based summarization (Section 4). In Section 5 we propose our LSA-based evaluation method. The experimental part (Section 6) covers a comparison of 13 summarization systems that participated in DUC 2002[1] from the point of view of several evaluation measures: two baselines, the standard ROUGE measure (see Section 3.3.4) and our proposed LSA measures. Firstly, the similarity of system summaries and abstracts and then the similarity of system summaries and full texts were studied. The correlation between system rankings produced by the evaluation measures and a manual ranking provided by DUC organizers was measured.

2 TEXT SUMMARIZATION

The earliest work in automatic text summarization dates back to the 1950s. In the last ten years a lot of new approaches have appeared as a result of the information overload on the Web. Recently, several LSA-based approaches have been developed. They are described in separate Section 4.

2.1 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 [22]. 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 [2], the presence of title words or certain *cue-words* (i.e., words like "important" or "relevant"). In [9] it was demonstrated that the combination of the presence of cue-words, title words and the position of a sentence produce the most similar extracts to abstracts written by a human.

[1] 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.

2.2 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. [35] 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 $tf_i \cdot idf_i$, 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 [11] 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 [18] 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.

2.3 Cohesion-Based Approaches

Extractive methods can fail to capture the relations between concepts in a text. Anaphoric expressions[2] 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 between expressions which determine the text connectivity. Cohesive properties of the text have been explored by different summarization approaches.

In [1] 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 [5]. 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] 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).

2.4 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 [31] 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 [24], 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.

2.5 Graph-Based Approaches

Graph-Based algorithms, such as HITS [15] or Google's PageRank [6] 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 [26] the graph-based model was applied to natural language processing, resulting in TextRank. Further, the graph-based ranking algorithm was applied to summarization [27]. 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.

2.6 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 [13, 16, 38, 41], or re-generate new sentences from scratch [25]. In [14] a *Cut-and-paste strategy* was proposed. The authors have identified six editing operations in human abstracting:

1. sentence reduction,
2. sentence combination,
3. syntactic transformation,
4. lexical paraphrasing,
5. generalization and specification, and
6. 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.

3 EVALUATION MEASURES

The taxonomy of summary evaluation measures can be found in Figure 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. 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.

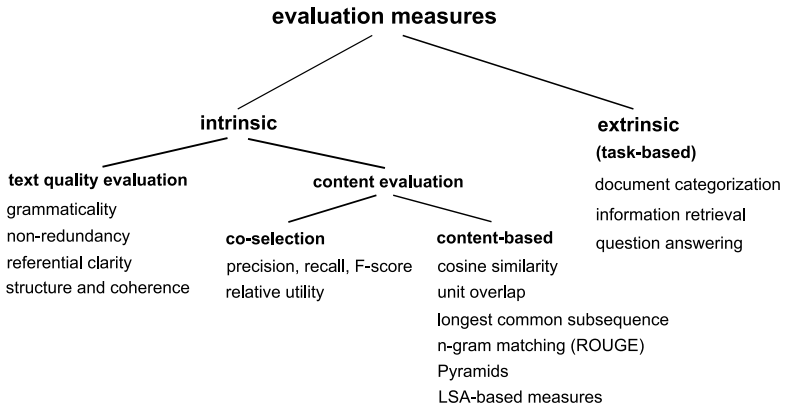


Fig. 1. The taxonomy of summary evaluation measures

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

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.

3.2 Co-Selection Measures

3.2.1 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 composite 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:

$$F = \frac{2 \cdot P \cdot R}{P + R}. \tag{1}$$

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

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

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

3.2.2 Relative Utility

The main problem with P & R is that human judges often disagree on what the top $p\,\%$ 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 [32]. 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[3] are then called a sentence extract of size e. We can then define the following system performance metric:

$$RU = \frac{\sum_{j=1}^{n} \delta_j \sum_{i=1}^{N} u_{ij}}{\sum_{j=1}^{n} \epsilon_j \sum_{i=1}^{N} u_{ij}}, \tag{3}$$

where u_{ij} is a utility score of sentence j from annotator i, ϵ_j is 1 for the top e sentences according to the sum of utility scores from all judges, otherwise its value is 0, and δ_j is equal to 1 for the top e sentences extracted by the system, otherwise its value is 0. For details, see [32].

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.

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

3.3.1 Cosine Similarity

A basic content-based similarity measure is Cosine Similarity [35]:

$$\cos(X, Y) = \frac{\sum_i x_i \cdot y_i}{\sqrt{\sum_i (x_i)^2} \cdot \sqrt{\sum_i (y_i)^2}}, \tag{4}$$

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

3.3.2 Unit Overlap

Another similarity measure is Unit Overlap [34]:

$$overlap(X, Y) = \frac{\|X \cap Y\|}{\|X\| + \|Y\| - \|X \cap Y\|}, \tag{5}$$

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

3.3.3 Longest Common Subsequence

The third content-based measure is called Longest Common Subsequence (LCS) [33]:

$$lcs(X, Y) = \frac{length(X) + length(Y) - edit_{di}(X, Y)}{2}, \tag{6}$$

where X and Y are representations based on sequences of words or lemmas, $\mathrm{lcs}(X, Y)$ 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 [33].

3.3.4 N-gram Co-occurrence Statistics – ROUGE

In the last edition of DUC 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 [20].

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

$$\text{ROUGE-}n = \frac{\sum_{C \in RSS} \sum_{gram_n \in C} \mathrm{Count}_{match}(gram_n)}{\sum_{C \in RSS} \sum_{gram_n \in C} \mathrm{Count}(gram_n)}, \tag{7}$$

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

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

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) – and ROUGE-SU4 – a bigram measure that enables at most 4 unigrams inside bigram components to be skipped [21].

3.3.5 Pyramids

The Pyramid method is a novel semi-automatic evaluation method [30]. 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 related subparts more tightly. SCUs 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 agrees between the peer summary and manual summary. However, this promising method still requires some annotation work.

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. We mention the three most important tasks – document categorization, information retrieval and question answering.

3.4.1 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 those obtained by categorizing full documents (an upper bound) or random sentence extracts (lower bound). Categorization can be performed either manually [23] or by a machine classifier [12]. 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 that by a summarizer. It is often done only by comparing the system performance with the upper and lower bounds.

In SUMMAC evaluation [23], 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 3.2.1).

3.4.2 Information Retrieval

Information Retrieval (IR) is another task appropriate for the task-based evaluation of a summary quality. *Relevance correlation* [33] 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 [36]. 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 [33]).

3.4.3 Question Answering

An extrinsic evaluation of the impact of summarization in a task of *question answering* was carried out in [28]. 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.

4 LSA IN SUMMARIZATION FRAMEWORK

Latent Semantic Analysis (LSA) [19] is a fully automatic mathematical/statistical technique for extracting and representing the contextual usage of words' meanings in passages of discourse. The basic idea is that the aggregate of all the word contexts in which a given word does and does not appear provides mutual constraints that determine the similarity of meanings of words and sets of words to each other. LSA has been used in a variety of applications (e.g., information retrieval, document categorization, information filtering, and text summarization).

The heart of the analysis in summarization background 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[5].

If there are m terms and n sentences in the document, then we will obtain an $m \times n$ matrix A. 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:

$$A = U\Sigma V^T \tag{8}$$

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*. The dimensionality of the matrices is reduced to r most important dimensions and thus, U is $m \times r$, Σ is $r \times r$ and V^T is $r \times n$ matrix.

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 [4], if a word combination pattern is salient and recurring in a document, this pattern will be captured and represented by

[5] The best performing weighting in our experiments was a simple Boolean weight: 1 if the sentence contains a particular word and 0 if it does not (see Section 6.2).

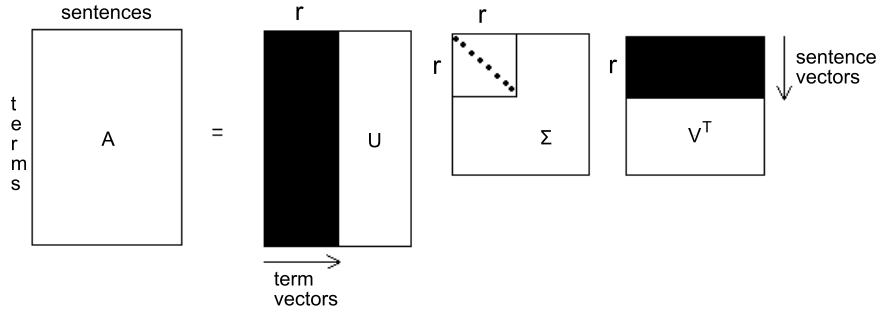


Fig. 2. Singular Value Decomposition

one of the left 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 represents this pattern best will have the largest value with this vector. Assuming that each particular word combination pattern describes a certain topic in the document, each left singular vector can be viewed as representing such a topic [7], the magnitude of its singular value representing the importance degree of this topic.

The summarization method proposed in [10] 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.

In order to fix the problem, we changed the selection criterion to include in the summary the sentences whose vectorial representation in the matrix $\Sigma^2 \cdot V$ 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 important topics, possibly including more than one sentence about an important topic, rather than one sentence for each topic. More formally: after computing the SVD of a term by sentences matrix, we compute the length of each sentence vector in $\Sigma^2 \cdot V$, which represents its summarization score as well (for details see [39]).

In [29] an LSA-based summarization of meeting recordings was presented. The authors followed the Gong and Liu 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 Σ. The number 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 [43]. 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 $A' = U'\Sigma' V'^T$.[6] 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 in the resulting map (see Section 2.5).

5 EVALUATION BY LATENT SEMANTIC ANALYSIS

The ability to capture the most important topics is used by the two evaluation metrics we propose. The idea is that a summary should contain the most important topic(s) of the reference document (e.g., full text or abstract). It 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 represents the degree of term importance in salient topics. The methods measure 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 we have proposed two measures.

5.1 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 the summary. We 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:

$$\cos\varphi = \sum_{i=1}^{n} ur_i \cdot us_i, \tag{9}$$

where ur is the first left singular vector of the reference text SVD, us is the first left singular vector of the summary SVD[7] and n is the number of unique terms in the reference text.

[6] U', or Σ', V'^T, A', denotes matrix U, or Σ, V^T, A, reduced to r dimensions.

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

5.2 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.

5.2.1 Dimensionality Reduction

If we perform SVD on a $m \times n$ 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[8]. 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, $p = \min(sw/fw \cdot 100, 100)$, where sw is the number of words in the summary and fw is the number of words in the reference text.[9]
- We reduce the latent space to r dimensions, where $r = p/100 \cdot$ *total number of dimensions*. 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 and the full text contains 30 sentences. Thus, SVD creates a space of 30 dimensions and we choose the 3 most important dimensions (r is set to 3).

However, $p\,\%$ dimensions contain more than $p\,\%$ information. It is possible to estimate each dimension's significance from the magnitude of its singular value. In [7] it was proved that the statistical significance of each LSA dimension is approximately the square of its singular value.

We performed an experiment with DUC2002 data in which we tried to find out how much information is contained in the top $p\,\%$ dimensions. In [7] it was shown that the magnitudes of the squares of singular values follow a Zipf-like distribution:

$$\sigma_i^2 = a \cdot i^b, \tag{10}$$

where b is very close to -1 and a is very large.

[8] It is the first left singular vector.

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

Suppose, for example, we have singular values $[10, 7, 5, \ldots]$, that their significances (squares of singular values) are $[100, 49, 25, \ldots]$, and that the total significance is 500 (sum of the singular value squares). Then the relative significances are $[20\,\%, 9.8\,\%, 5\,\%, \ldots]$: i.e., the first dimension captures 20 % of the information in the original document.

Figure 3 illustrates the logarithm dependency of the significance of r most important dimensions used for evaluation on the summary length (both quantities are shown in percents). For instance, when evaluating a 10 % summary, the 10 %

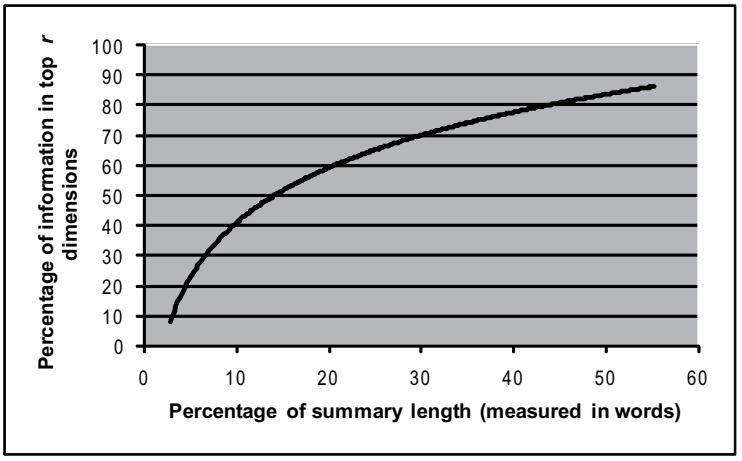


Fig. 3. The dependency of the significance of r most important dimensions on the summary length

most important dimensions used for evaluation deal with 40 % of document information, or when evaluating 30 % summary, the top 30 % dimensions deal with 70 % of document information.

5.2.2 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 that correspond to the most important topics. The result is labeled B:

$$B = \begin{pmatrix} u_{1,1}\sigma_1^2 & u_{1,2}\sigma_2^2 & \ldots & u_{1,r}\sigma_r^2 \\ u_{2,1}\sigma_1^2 & u_{2,2}\sigma_2^2 & \ldots & u_{2,r}\sigma_r^2 \\ \ldots & \ldots & \ldots & \ldots \\ u_{m,1}\sigma_1^2 & u_{m,2}\sigma_2^2 & \ldots & u_{m,r}\sigma_r^2 \end{pmatrix}. \tag{11}$$

Then we take matrix B and measure the length of each row vector:

$$|b_i| = \sqrt{b_{i,1}^2 + b_{i,2}^2 + \ldots + b_{i,r}^2}. \tag{12}$$

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

$$s = \begin{bmatrix} |b_1| \\ |b_2| \\ \ldots \\ |b_n| \end{bmatrix} \tag{13}$$

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

6 EXPERIMENTS

To assess the usefulness of our evaluation measures, we used the DUC 2002 corpus. This gave us the opportunity to compare the quality of the systems participating in DUC from an angle of several evaluation measures. Furthermore, we were able to compare the system rankings provided by our measures against human rankings.

In 2002 the family of ROUGE measures had not yet been introduced. However, now we were able to perform ROUGE evaluation. This gives us another interesting comparison of standard evaluation measures with our LSA-based ones. We 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, we computed the Pearson correlation between these rankings and human ones.

6.1 DUC 2002 Corpus

DUC 2002 included a single-document summarization task, in which 13 systems participated[10]. 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[11], the DUC organizers also distributed baseline summaries (the first 100 words of a document). The coverage of all the summaries was assessed by humans. For assessing the quality

[10] 2002 is the last version of DUC that included the evaluation of single-document informative summaries. In later years only headline-length single-document summaries were analysed.

[11] Two systems produced only headlines. Therefore, we did not include them in the evaluation.

of each evaluation method, we computed the Pearson correlation between system rankings and human ones.

6.2 Term Weighting Schemes for SVD

We analysed various term weighting schemes for SVD input matrix. The vector $A_i = [a_{1i}, a_{2i}, \ldots, a_{ni}]^T$ is defined as:

$$a_{ij} = L_{ij} \cdot Gij, \tag{14}$$

where L_{ij} denotes the local weight for term j in sentence i, and G_{ij} is the global weight for term j in the whole document.

Local weighting $L(t_{ij})$ has the following four possible alternatives [8]:

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

Global weighting G_{ij} has the following four possible alternatives:

- No weight (NW): $G_{ij} = 1$ for any term j.
- Inverse sentence frequency (ISF): $G_{ij} = \log(N/n_j) + 1$, 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): $G_{ij} = \frac{gf_j}{sf_j}$, 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): $G_{ij} = 1 - \sum_i \frac{p_{ij} \log(p_{ij})}{log(nsent)}$, 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 evaluation methods are compared in Figures 4 (reference document is an abstract) and 5 (reference document is the 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 statistical significant. The best performing weightings are used for the comparison of evaluators in Tables 1 and 2.

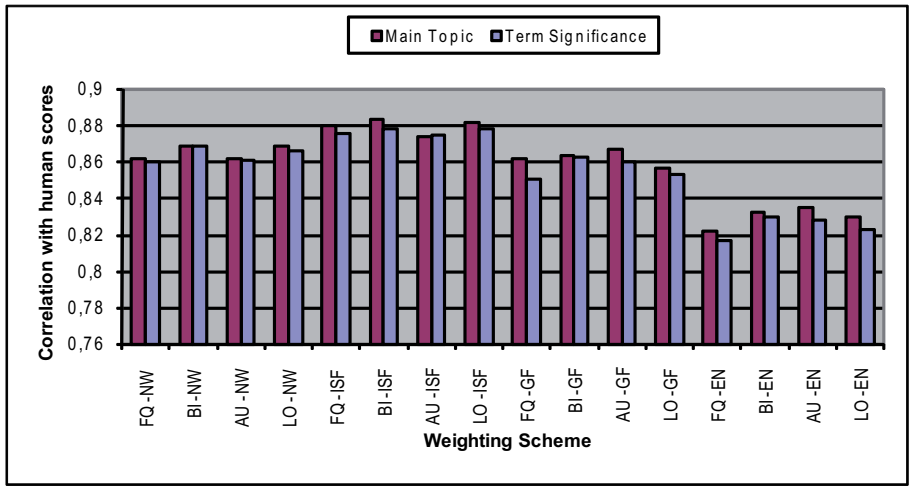


Fig. 4. The influence of different weighting schemes on the evaluation performance measure by the correlation with human scores. The meaning of the letters is as follows: [Local weight]–[Global weight]. The reference document is abstract.

6.3 Baseline Evaluators

We included two baseline evaluators in the evaluation. The first one – cosine similarity – was described in Section 3.3.1. 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. The best performing value of n for the 100-word summaries was 30. This setting is used in Tables 1 and 2.

6.4 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 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 correlation. The first reason is that abstractors usually put in the abstract some words 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,

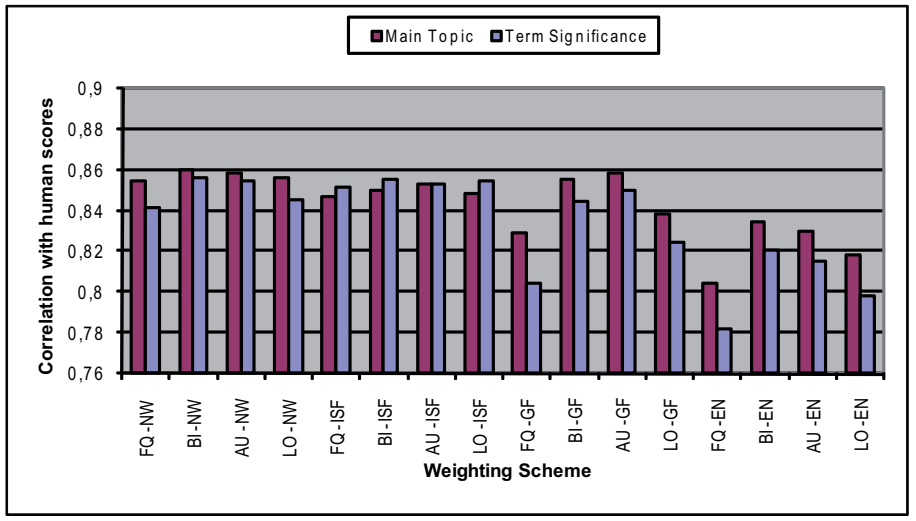


Fig. 5. The influence of different weighting schemes on the evaluation performance measure by the correlation with human scores. The meaning of the letters is as follows: [Local weight]–[Global weight]. The reference document is full text.

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 1. Correlation between evaluation measures and human assessments – the reference document is an abstract

results in better performance. The differences between LSA measures and baselines were not statistically significant at 95 % confidence.

6.5 Summary and Full Text Similarity

In the second experiment we took the full text as a reference document. We compared Cosine similarity, top n keywords, and LSA-based measures with human rankings. ROUGE is not designed for comparison with full texts. We report the results in Table 2.

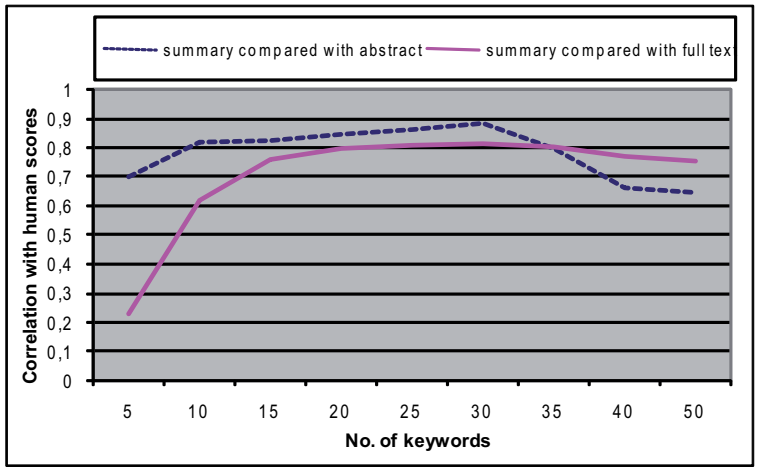


Fig. 6. The dependency of the performance of the keyword evaluator on the number of keywords

Score	Correllation
LSA – Main Topic Similarity	0.85988
LSA – Term Significance Similarity	0.85573
Keywords	0.80970
Cosine similarity	0.27117

Table 2. Correlation between evaluation measures and human assessments – the reference document is a full text

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 measure correlates even significantly better. The difference between 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 LSA-based similarity is appropriate for the evaluation of extractive summarization where abstracts are not available.

7 CONCLUSIONS

We have covered the basic ideas of recent approaches to text summarization. The exact taxonomy of evaluation methods was presented. Moreover, we introduced our metrics, which are based on latent semantic analysis that can capture the main topics of an article. We experimentally compared the approach with state-of-the-art

ROUGE evaluation measures. We demonstrated that the system ranking provided by ROUGE correlates well with the human ranking when comparing summaries with abstracts. The appropriate usage of our LSA-based evaluation measures is to compare summaries with full texts. The method works well on extractive summaries. If abstracts are included in a corpus we recommend using the ROUGE family, however, if not then LSA-based comparison with the source is a good choice. For the future we plan to apply our evaluation method in multi-document summarization.

Acknowledgement

This research was partly supported by National Research Programme II, project 2C06009 (COT-SEWing).

REFERENCES

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

[2] Baxendale, P. B.: Man-Made Index for Technical Literature – An Experiment. In IBM Journal of Research Development, Vol. 2, 1958, No. 4, pp. 354–361.

[3] Benbrahim, M.—Ahmad, K.: Text Summarisation: The Role of Lexical Cohesion Analysis. In The New Review of Document & Text Management, 1995, pp. 321–335.

[4] Berry, M. W.—Dumais, S. T.—O'Brien, G. W.: Using Linear Algebra for Intelligent IR. In SIAM Review, Vol. 37, 1995, No. 4.

[5] Boguraev, B.—Kennedy, C.: Salience-Based Content Characterization of Text Documents. In I. Mani and M. T. Maybury, eds., Advances in Automatic Text Summarization, The MIT Press, 1999.

[6] Brin, S.—Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Computer Networks and ISDN Systems, Vol. 30, 1998, pp. 1–7.

[7] Ding, Ch.: A Probabilistic Model for Latent Semantic Indexing. In Journal of the American Society for Information Science and Technology, Vol. 56, 2005, No. 6, pp. 597–608.

[8] Dumais, S. T.: Improving the Retrieval of Information from External Sources. In Behavior Research Methods, Instruments & Computers, Vol. 23, 1991, No. 2, pp. 229–236.

[9] Edmundson, H. P.: New Methods in Automatic Extracting. In Journal of the Association for Computing Machinery, Vol. 16, 1969, No. 2, pp. 264–285.

[10] Gong, X.—Liu, X.: Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In Proceedings of ACM SIGIR, New Orleans, USA, 2002.

[11] Hovy, E.—Lin, C.-Y.: Automated Text Summarization in SUMMARIST. In I. Mani and M. T. Maybury, eds., Advances in Automatic Text Summarization, 1999, The MIT Press, pp. 81–94.

[12] Hynek, J.—Ježek, K.: Practical Approach to Automatic Text Summarization. In Proceedings of the ELPUB'03 Conference, Guimaraes, Portugal, 2003, pp. 378–388.

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

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

[15] Kleinberg, J. M.: Authoritative Sources in a Hyper-Linked Environment. In Journal of the ACM, Vol. 46, 1999, No. 5, pp. 604–632.

[16] Knight, K.—Marcu, D.: Statistics-Based Summarization – Step One: Sentence Compression. In Proceeding of The 17th National Conference of the American Association for Artificial Intelligence, 2000, pp. 703–710.

[17] Kovaľ, R.—Návrat, P.: Intelligent Support for Information Retrieval of Web Documents. In Computing and Informatics, Vol. 21, 2002, No. 5, pp. 509–528.

[18] Kupiec, J.—Pedersen, J. O.— Chen, F.: A Trainable Document Summarizer. In Research and Development in Information Retrieval, 1995, pp. 68–73.

[19] Landauer, T. K.—Dumais, S. T.: A Solution to Platoís Problem: the Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge. In Psychological Review, Vol. 104, 1997, pp. 211–240.

[20] Lin, Ch.—Hovy, E.: Automatic Evaluation of Summaries Using n-Gram Co-Occurrence Statistics. In Proceedings of HLT-NAACL, Edmonton, Canada, 2003.

[21] Lin, Ch.: ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of the Workshop on Text Summarization Branches Out, Barcelona, Spain, 2004.

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

[23] Mani, I.—Firmin, T., House, D.—Klein, G.—Sundheim, B.—Hirschman, L.: The TIPSTER Summac Text Summarization Evaluation. In Proceedings of the 9th Meeting of the European Chapter of the Association for Computational Linguistics, 1999, pp. 77–85.

[24] Marcu, D.: From Discourse Structures to Text Summaries. In Proceedings of the ACLI97/EACLI97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain, 1997, pp. 82–88.

[25] McKeown, K.—Klavans, J.—Hatzivassiloglou, V.—Barzilay, R.—Eskin, E.: From Discourse Structures to Text Summaries. In Towards Multidocument Summarization by Reformulation: Progress and Prospects, AAAI/IAAI, 1999, pp. 453–460.

[26] Mihalcea, R.—Tarau, P.: Text-Rank – Bringing Order Into Texts. In Proceeding of the Comference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 2004.

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

[28] Morris, A.—Kasper, G.—Adams, D.: The Effects and Limitations of Automatic Text Condensing on Reading Comprehension Performance. In Information Systems Research, Vol. 3, 1992, No. 1, pp. 17–35.

[29] Murray, G.—Renals, S.—Carletta J.: Extractive Summarization of Meeting Recordings. In Proceedings of Interspeech, Lisboa, Portugal, 2005.

[30] Nenkova, A.—Passonneau, R.: Evaluating Content Selection in Summarization: The Pyramid Method. In Document Understanding Conference, Vancouver, Canada, 2005.

[31] Ono, K.—Sumita, K.—Miike, S.: Abstract Generation Based on Rhetorical Structure Extraction. In Proceedings of the International Conference on Computational Linguistics, Kyoto, Japan, 1994, pp. 344–348.

[32] Radev, D.—Jing, H.—Budzikowska, M.: Centroid-Based Summarization of Multiple Documents. In ANLP/NAACL Workshop on Automatic Summarization, Seattle, USA, 2000.

[33] Radev, D.—Teufel, S.—Saggion, H.—Lam, W.—Blitzer, J.—Qi, H.—Celebi, A.—Liu, D.—Drabek, E.: Evaluation Challenges in Large-Scale Document Summarization. In Proceeding of the 41st meeting of the Association for Computational Linguistics, Sapporo, Japan, 2003.

[34] Saggion, H.—Radev, D.—Teufel, S.—Lam, W.—Strassel, S.: 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.

[35] Salton, G.: Automatic Text Processing. Addison-Wesley Publishing Company, 1988.

[36] Siegel, S.–Castellan, N. J.: Nonparametric Statistics for the Behavioral Sciences. Berkeley, CA: McGraw-Hill, 2nd edn., 1988.

[37] Spark Jones, K.—Galliers, J. R.: Evaluating Natural Language Processing Systems: An Analysis and Review. In Lecture Notes in Artificial Intelligence, No. 1083, Springer, 1995.

[38] Sporleder, C.—Lapata, M.: Discourse Chunking and Its Application to Sentence Compression. In Proceedings of HLT/EMNLP, Vancouver, Canada, 2005, pp. 257–264.

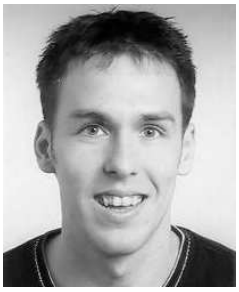
[39] Steinberger, J.—Ježek, K.: Text Summarization and Singular Value Decomposition. In Lecture Notes for Computer Science, Vol. 2457, pp. 245–254, Springer-Verlag, 2004.

[40] Steinberger, J.—Kabadjov, M. A.—Poesio, M.: Improving LSA-Based Summarization with Anaphora Resolution. In Proceedings of HLT/EMNLP, Vancouver, Canada, 2005, pp. 1–8.

[41] Steinberger, J.—Ježek, K.: Sentence Compression for the LSA-Based Summarizer. In Proceedings of the 7th International Conference on Information Systems Implementation and Modelling, Přerov, Czech Republic, 2006, pp. 141–148.

[42] Toman, M.—Steinberger, J.—Ježek, K.: Searching and Summarizing in Multilingual Environment. In Proceedings of the 10th International Conference on Electronic Publishing, Bansko, Bulgaria, 2006.

[43] Yeh, J.-Y.—Ke, H.-R.—Yang, W.-P.— Meng, I-H.: Text Summarization Using a Trainable Summarizer and Latent Semantic Analysis. In Special issue of Information Processing and Management on An Asian digital libraries perspective, Vol. 41, 2005, No. 1, pp. 75–95.



Josef Steinberger works at Department of Computer Science and Engineering at University of West Bohemia in Pilsen. His research focuses on automatic multilingual text analysis, especially summarization, latent semantic analysis and coreference resolution.

Karel Ježek works at Department of Computer Science and Engineering at University of West Bohemia in Pilsen. His research interests are in theory of formal languages and compilers, operating systems theory, programming languages, database and knowledge-base systems. Recently he is interested in data mining.