Automatic Text Summarization
(The state of the art 2007 and new challenges)

Karel Ježek[1], Josef Steinberger[1]

[1]Katedra informatiky a výpočetní techniky, FAV, ZČU – Západočeská Univerzita v Plzni,
Univerzitní 22, 306 14 Plzeň
{jstein, jezek_ka}@kiv.zcu.cz

Abstract. The headline of this paper names a research area originating from the late 50's but not loosing its popularity until the present time. Moreover, one of the most relevant today's problems caused by the rapid growth of the Web, which is called *information overloading*, has increased the necessity of more sophisticated and powerful summarizers. This paper shortly introduces a taxonomy of summarization methods and an overview of their principles from classical ones, over corpus based, to knowledge rich approaches. We consider various aspects which can affect their classification. A special attention is devoted to application of recent information reduction methods, based on algebraic transformations. Further, we introduce experiences with the development of our own summarizing method. Finally, some new ideas and a conception for the future of this field are mentioned.

1 Introduction

Enormous increasing and easy availability of information on the World Wide Web have recently resulted in brushing up the classical linguistics problem - the condensation of information from text documents. This task is essentially a data reduction process. It was manually exerted from time out of mind and firstly computerized in late 50^{th} . Resulted summary has to inform by selection and or by generalization on important content and conclusions in the original text. Recent scientific knowledge and more efficient computers form a new challenge giving the chance to solve *the information overload problem* or at least to postpone its decision and decrease its negative impact.

There are plenty of various definitions what actually text summarization means. Except this mentioned a few lines above e.g.:

- "A brief but accurate representation of the contents of a document",
- "A distilling the most important information from a source to produce an abridged version for a particular user/users and task/tasks",

The quantitative features, which can characterize the summary include:

- semantic informativeness (can be viewed as the measure of ability to reconstruct from the summary the original text),
- coherence (express the way how the parts of the summary create together an integrated sequence)
- compression ratio.

The history of automatic i.e. computerized summarization began 50 years ago. As the oldest publication, describing an implementation of an automatic summarizer is often cited [1]. Luhn's method uses term frequencies to appraise eligibility of sentences for the summary. Its main idea is based on knowledge that significant words carrying most information are not too frequent nor too seldom in the text. Establishing boundaries of words significance by the help of their frequency would be a matter of experience. The next step is ranking of sentences, reflecting the number of significant words and their distance in a sentence. After it remains only to choose one or several highly ranked as a result. It should be mentioned (it seems nowadays funny) that Luhn's motivation was as well information overload.

The next remarkable progress was done ten years later [2]. Edmundson's work introduced hypothesis concerning high information value of title phrases, sentences from the beginning and from the conclusion of the article, sentences containing cue words and phrases as "important", "results are", "paper introduces", etc.

Even if next years brought further results, the renaissance of this field and remarkable progress came in 90^{th}. We should take notice [3] or [4]. It is the time of broader use of artificial intelligence methods in this area and combination of various methods in hybrid systems. New millennium due to WWW expansion shifted the interest of researches to summarization of groups of documents, multimedia documents and application of new algebraic method for data reduction.

This paper is organized following way. 2. Chapter describes the basic notions and typology of summarizers. Chapter 3 is devoted to a short overview of classical methods. Chapter 4 is on new approaches with impact on algebraic reduction methods, including our own LSA-based approach. The last chapter concludes the paper and the further research is proposed.

2 Taxonomy of Summarizing Methods

There are several, often orthogonal views which can be used to characterize summarizers. The list and description of the most often cited follows.

Comparing the form of summary we recognize:

- *Extracts*, they are summaries completely consisting of word sequences copied from the original document. As the word sequences can be used phrases, sentences or paragraphs. As expected, extracts suffer from

inconsistencies, lack of balance, and lack of cohesion. Sentences may be extracted out of context, anaphoric reference may be broken.

- *Abstracts*, they are summaries containing word sequences not present in the original. Up to now it is too hard task for computer research to solve it successfully.

The view coming from the level of processing distinguishes:

- *Surface-level* approaches, in which case information is represented in notions of shallow features and their combination. Shallow features include e.g. statistically salient terms, positionally salient terms, terms from cue phrases, domain-specific or a user's query terms. Results have the form of extracts.
- *Deeper-level* approaches may produce extracts or abstracts. The later case uses synthesis involving natural language generation. They need some semantic analysis e.g. can use entity approaches and build a representation of text entities (text units) and their relationships to determine salient parts. Relationships of entities include thesaural relations, syntactic relations, meaning relations and others. They can as well use discourse approaches and model the text structure on the base of e.g. hypertext markup or rhetorical structure.

Another typology comes from the purpose the summary serves:

- *Indicative* summaries give abbreviated information on the main topics of a document. They should preserve its most important passages and often are used as the end part of IR systems, being returned by search system instead of full document. Their aim should be to help a user to decide whether the original document is worth reading. The typical lengths of indicative summaries range between 5 till 10% of the complete text.
- *Informative* summaries provide a substitute ("surrogate", "digest") for full document, retaining important details, while reducing information volume. Informative summary is typically 20-30 % of the original text.
- *Critical or Evaluative* summaries capture the point of view of the summary author on a given subject. Reviews are typical example, but they are little bit out of scope of nowadays automatic summarizers.

It should be noted, that all three mentioned groups are not mutually exclusive and they are common summaries serving both indicative and informative function. It is quite usual to hold informative summarizers as a subset of indicative ones.

When distinguished by the audience we can recognize:

- *Generic* summaries, when the result is aimed at a broad community of readers, all major topics are equally important,
- *Query-based* summaries, when the result is based on a question e.g. "what are the causes of the high inflation?"
- *User focused or Topic focused* summaries, which are tailored to the interest of particular user or emphasize only particular topics.

There are some other views we can use for taxonomy of summarizers e.g.:
Span of processed text:

- *single document* or *multi-document* summarization.

Language:

- *monolingual* versus *multilingual*.

Genre:

- scientific article or report or news …

3 Overview of Methods Based on Classical Principles

3.1 Pioneering Works

The first approaches of the automatic text summarization used only simple (surface level) indicators to decide what parts of a text include into the summary. The oldest sentence extraction algorithm was developed in 1958 [1]. It used frequencies of terms as the sentence relevance criterion. The basic idea was that a writer will repeat certain words when writing about a given topic. The importance of terms is considered proportional to their frequency in summarized documents. The frequencies are used in the next step to score and select sentences for the extract. Other indicators of relevance used in [5] are the position of a sentence within the document and the presence of certain *cue-words* (i.e., words like "important" or "relevant") or words contained in the title. The combination of cue-words, title words and the position of a sentence was used in [2] to produce extracts and was demonstrated their similarity with human written abstracts.

3.2 Statistical Methods

In [4] was proved that the relevance of document terms is inversely proportional to the number of documents in the corpus containing the term. The formula for term relevance evaluation is given by $tf_i \times idf_i$, where tf_i is the frequency of term i in the document and idf_i is the inverted frequency of documents containing this term. Sentences can be subsequently scored for instance by summing relevance of terms in the sentence.

An implementation of a more ingenious statistical method was described in [3]. It uses a Bayesian classifier to compute the probability that a sentence in a source document should be included in a summary. To train the classifier the authors used a corpus of 188 pairs of full documents/summaries. The characteristic features used in Bayesian formula include except of word frequency also uppercase words, sentence length, phrase structure, in-paragraph position.

An alternative way how to measure term relevance was proposed in [6]. Instead of rough term counting the authors used *concept relevance* which can be determined

using WordNet. E.g. the occurrence of the concept “car” is counted when the word “auto” is found as well as when, for instance, “autocar”, “tires”, or “brake” are found.

3.3 Methods Based on Text Connectivity

Anaphoric expressions[1] that refer to previously mentioned parts of the text need to know their antecedents in order to be understood. Extractive methods can fail to capture the relations between concepts in a text. If a sentence containing an anaphoric link is extracted without the previous context the summary can become difficult to understand. Cohesive properties comprise relations between expressions of the text. They have been explored by different summarization approaches.

Let us mention a method called *Lexical chains,* which was introduced in [7]. It uses the WordNet thesaurus for determining cohesive relations between terms (i.e., repetition, synonymy, antonymy, hypernymy, and holonymy) and composes the chains by related terms. Their scores are determined on the basis of the number and type of relations in the chain. Only those 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 [8]. 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. The sentences where the occurrence of frequently mentioned objects overcomes the given limit are included into the summary.

Into the group of methods based on text connectivity we can include the methods utilizing *Rhetorical Structure Theory* (RST). RST is a theory about text organization. It consists of a number of rhetorical relations that connect together text units. The relations tie together a *nucleus* – which is central to the writer’s goal, and a *satellite* - less central or marginal parts. From relations is composed a tree-like representation which is used for extraction of text unit into the summary. In [9] sentences are penalized according to their rhetorical role in the tree. In the concrete 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 [10], 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.

3.4 Iterative Graph Methods

Iterative graph algorithms, such as HITS [11] or Google’s PageRank [12] have been originally developed as exploring tools of the link-structure to rank Web pages. Later

[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).

on they were successfully used in other areas e.g. citation analysis, social networks etc. In graph ranking algorithms, the importance of a vertex within the graph is iteratively computed from the entire graph. In [13] the graph-based model was applied to natural language processing, resulting in an algorithm named TextRank. The same graph-based ranking principles were applied in summarization [14]. A graph is constructed by adding a vertex for each sentence in the text. 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 as the number of common tokens between lexical representations of two sentences. The iterative part of algorithm is consequently applied on the graph of sentences. When its processing is finished, vertices (sentences) are sorted by their scores. The top ranked sentences are included in the result.

3.5 Coming Close to Human Abstracts

There is a qualitative difference between the summaries produced by current automatic summarizers and the abstracts written by human abstractors. Computer systems can identify the important topics of an article with only a limited accuracy. 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 extract, the result can be incoherent and sometimes even misleading.

Lately, some non-sentence-extractive summarization methods have started to appear. Instead of reproducing full sentences from the summarized text, these methods either compress the sentences [15, 16, 17, 18], or re-generate new sentences from scratch [19]. In [20] a *Cut-and-paste strategy* was proposed. The authors have identified 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. Extraction is robust towards all irregularities of input text. It is failure-proof and less language dependent.

4 New Approaches Based on Algebraic Reduction

Several approaches based on algebraic reduction methods have appeared in the last couple of years. The most widely used is *latent semantic analysis* (LSA) [21], however other methods, like *non-negative matrix factorization* (NMF) [22] or *semi-discrete matrix decomposition* (SDD) [23] look promising as well.

4.1 LSA in Summarization Background

LSA is a fully automatic algebraic-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 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:

$$A = U\Sigma V^T, \tag{1}$$

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[2].

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

[2] U', resp. Σ', V'^T, denotes matrix U, resp. Σ, V^T, reduced to r dimensions.

vector can be viewed as representing such a topic [25], the magnitude of its singular value representing the importance degree of this topic[3].

4.2 LSA-Based Single-Document Approaches

The summarization method proposed in [26] 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 $\Sigma^2 \cdot V^T$, which represents its summarization score as well (for details see [27]).

In [28] 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 [29]. 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 $A'=U'\Sigma'V'^T$. 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 3.4).

---

[3] In [25] it was shown that the dependency of the significance of each particular topic on the magnitude of its corresponding singular value is quadratical.

4.3 LSA-Based Multi-Document Approaches

In [30] we proposed the extension of the method to process a cluster of documents written about the same topic. Multi-document summarization is one step more complex task than single-document summarization. It brings into new problems we have to deal with. The first step is again to create a term by sentence matrix. In this case we include in the matrix all sentences from the cluster of documents. (In the case of single-document summarization we included the sentences from the one document.) Then we run sentence ranking. Each sentence gets the score, which is computed in the same way as when we summarize a single document – vector length in the matrix $\Sigma^2 \cdot V^T$. Now, we are ready to select the best sentences (the ones with the greatest score) for the summary.

However, two documents written about the same topic/event can contain similar sentences and thus we need to solve redundancy. We propose the following process: before adding a sentence into the summary, look if there is a similar sentence already in the summary. The similarity is measured by the cosine similarity in the original term space. We determine a threshold here. Extracted sentence should be close to the user query. To satisfy this, query terms get a higher weight in the input matrix.

Another problem of this approach is that it favours long sentences. It is natural because a longer sentence probably contains more significant terms than a shorter one. We solve this by dividing the sentence score by $number-of-terms^{lk}$, where lk is the length coefficient.

Experiments showed good results with a low dimensionality. It is enough to use up to 10 dimensions (topics). However, the topics are not equally important. The magnitude of each singular value holds the topic importance. To make it more general we experimented with different power functions in the computation of the final matrix used for determination of sentence score: $\Sigma^{power} \cdot V^T$.

In [31], an interesting multi-document summarization approach based on LSA and *maximal marginal relevance* (MMR) was proposed. A common approach for determining relevance and redundancy in multi-document summarization is to use MMR, in which candidate sentences are represented as weighted term-frequency vectors which can thus be compared to query vectors to gauge similarity and alreadyextracted sentence vectors to gauge redundancy, via the cosine of the vector pairs. While this has proved successful to a degree, the sentences are represented merely according to weighted term frequency in the document, and so two similar sentences stand a chance of not being considered similar if they don't share the same terms. One way to rectify this is to do LSA on the matrix first before proceeding to implement MMR, but this still only exploits term co-occurrence *within* the documents at hand. In contrast, the system described in [31] attempts to derive more robust representations of sentences by building a large semantic space using LSA on a very large corpus.

5 Conclusion

We presented the history and the state of the art in the automatic text summarization research area. We paid the most attention to the approaches based on algebraic reduction methods. Their strong property is that they work only with the context of terms and thus they do not depend on a particular language. The evaluation of summarization methods has the same importance as the own summarizing. The annual summarization evaluation conference DUC (Document Understanding Conference) set the direction in the evaluation processes. However, still the only fully automatic method for the comparison of summarizers' quality is ROUGE [32], which compares human-written abstracts and system summaries by matches of n-grams. We plan to participate at DUC'08 with our new summarizer, whose core will be based on tensor LSA. Three dimensions, instead of two, will be used – terms, sentences and documents. The idea of the method is that two sentences will be projected close to each other if they contain the same terms. Similarly, documents will be projected close to each other if they contain the same terms. Terms will be closer if they are contained in the same sentence/document. This way, the topics should be created more accurately when compared with matrix LSA. In the resulting space either MMR or our previous vector length approach can be used.

Acknowledgement

This work was supported by grant no. 2C06009 Cot-Sewing.

References

1. Luhn, H.P.: The Automatic Creation of Literature Abstracts. In IBM Journal of Research Development 2(2). (1958) 159–165.
2. Edmundson, H.P.: New Methods in Automatic Extracting. In Journal of the Association for Computing Machinery 16(2). (1969) 264–285.
3. Kupiec, J., Pedersen, J.O., Chen, F.: A Trainable Document Summarizer. In Research and Development in Information Retrieval. (1995) 68–73.
4. Salton, G.: Automatic Text Processing. Addison-Wesley Publishing Company, (1988).
5. Baxendale, P.B.: Man-made Index for Technical Literature - an experiment. In IBM Journal of Research Development, 2(4), 1958, pp. 354–361.
6. 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.
7. 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.
8. Boguraev, B., Kennedy, C.: Salience-based content characterization of text documents. In I. Mani and M.T. Maybury, eds., Advances in Automatic Text Summarization, 1999, The MIT Press.

9. 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.
10. Marcu, D.: From Discourse Structures to Text Summaries. In Proceedings of the ACL97/EACL97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain, 1997, pp. 82–88.
11. Kleinberg, J.M.: Authoritative sources in a hyper-linked environment. In Journal of the ACM, 46(5), 1999, pp. 604-632.
12. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. In Computer Networks and ISDN Systems, 30, 1998, pp. 1–7.
13. Mihalcea, R., Tarau, P.: Text-rank - bringing order into texts. In Proceeding of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 2004.
14. Mihalcea, R., Tarau, P.: An Algorithm for Language Independent Single and Multiple Document Summarization. In Proceedings of the International Joint Konference on Natural Language Processing, Korea, 2005.
15. Jing, H.: Sentence Reduction for Automatic Text Summarization. In Proceedings of the 6th Applied Natural Language Processing Conference, Seattle, USA, 2000, pp. 310–315.
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. Sporleder, C., Lapata, M.: Discourse chunking and its application to sentence compression. In Proceedings of HLT/EMNLP, Vancouver, Canada, 2005, pp. 257–264.
18. 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.
19. 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.
20. 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.
21. Landauer, T.K., Dumais, S.T.: A solution to platos problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. In Psychological Review, 104, 1997, pp. 211–240.
22. Lee, D.D., Seung, H.S.:. Learning the parts of objects by non-negative matrix factorization. In Nature, 401 (6755), 1999 pp. 788–791.
23. Kolda, T.G., O'Leary, D.P.: A semidiscrete matrix decomposition for latent semantic indexing information retrieval. In ACM Transactions on Information Systems 16(4), 1998, pp. 322–346.
24. Berry, M.W., Dumais, S.T., O'Brien., G.W.: Using linear algebra for intelligent IR. In SIAM Review, 37(4), 1995.
25. Ding, Ch.: A probabilistic model for latent semantic indexing. In Journal of the American Society for Information Science and Technology, 56(6), 2005, pp. 597–608.
26. Yihong Gong, Xin Liu: Generic text summarization using relevance measure and latent semantic analysis. In Proceedings of ACM SIGIR. New Orleans, USA, 2002.
27. Steinberger, J., Ježek, K.: Text Summarization and Singular Value Decomposition. In Lecture Notes for Computer Science vol. 2457, Springer-Verlag, 2004, pp. 245-254.
28. Murray, G., Renals, S., Carletta J.: Extractive Summarization of Meeting Recordings. In Proceedings of Interspeech, Lisboa, Portugal, 2005.

29. 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, 41(1), 2005, pp. 75–95.
30. Steinberger, J., Křišťan, M.: LSA-Based Multi-Document Summarization. Proceedings of 8th International PhD Workshop on Systems and Control, Balatonfured, Hungary, 2007.
31. B. Hachey, G. Murray, and D. Reitter. The embra system at duc 2005: Query-oriented multi-document summarization with a very large latent semantic space. In Proceedings of the Document Understanding Conference (DUC) 2005, Vancouver, Canada, 2005.
32. Lin, Ch.: Rouge: a package for automatic evaluation of summaries. In Proceedings of the Workshop on Text Summarization Branches Out, Barcelona, Spain, 2004.