Text Summarization of Turkish Texts using Latent Semantic Analysis

Makbule Gulcin Ozsoy
Dept. of Computer Eng.
Middle East Tech. Univ.
Ankara, Turkey
e1395383@ceng.metu.edu.tr

Ilyas Cicekli
Dept. of Computer Eng.
Bilkent University
Ankara, Turkey
ilyas@cs.bilkent.edu.tr

Ferda Nur Alpaslan
Dept. of Computer Eng.
Middle East Tech. Univ.
Ankara, Turkey
alpaslan@ceng.metu.edu.tr

Abstract

Text summarization solves the problem of extracting important information from huge amount of text data. There are various methods in the literature that aim to find out well-formed summaries. One of the most commonly used methods is the Latent Semantic Analysis (LSA). In this paper, different LSA based summarization algorithms are explained and two new LSA based summarization algorithms are proposed. The algorithms are evaluated on Turkish documents, and their performances are compared using their ROUGE-L scores. One of our algorithms produces the best scores.

1 Introduction

The exponential growth in text documents brings the problem of finding out whether a text document meets the needs of a user or not. In order to solve this problem, text summarization systems which extract brief information from a given text are created. By just looking at the summary of a document, a user can decide whether the document is of interest to him/her without looking at the whole document.

The aim of a text summarization system is to generate a summary for a given document such that the generated summary contains all necessary information in the text, and it does not include redundant information. Summaries can have different forms (Hahn and Mani, 2000). Extractive summarization systems collect important sentences from the input text in order to generate summaries. Abstractive summarization systems do not collect sentences from the input text, but they try to capture the main concepts in the text, and generate new sentences to represent these main concepts. Abstractive summarization approach is similar to the way that human summarizers follow. Since creating abstractive summaries is a more complex task, most of automatic text summarization systems are extractive summarization systems.

Summarization methods can be categorized according to what they generate and how they generate it (Hovy and Lin, 1999). A summary can be extracted from a single document or from multiple documents. If a summary is generated from a single document, it is known as single-document summarization. On the other hand, if a single summary is generated from multiple documents on the same subject, this is known as multi-document summarization. Summaries are also categorized as generic summaries and query-based summaries. Generic summarization systems generate summaries containing main topics of documents. In query-based summarization, the generated summaries contain the sentences that are related to the given queries.

Extractive summarization systems determine the important sentences of the text in order to put them into the summary. The important sentences of the text are the sentences that represent the main topics of the text. Summarization systems use different approaches to determine the important sentences (Hahn and Mani, 2000; Hovy and Lin, 1999). Some of them look surface clues such as the position of the sentence and the words that are contained in the sentence. Some summarization systems use more semantic oriented analysis such as lexical chains in order to determine the important sentences. Lately, an algebraic method known as Latent Semantic Analysis (LSA) is used in the determination of the important sentences, and successful results are obtained (Gong and Liu, 2001).

In this paper, we present a generic extractive Turkish text summarization system based on LSA. We applied the known text summarization approaches based on LSA in order to extract the summaries of Turkish texts. One of the main contributions of this paper is the introduction of two new summarization methods based on LSA. One of our methods produced much better results than the results of the other known methods.

The rest of the paper is organized as follows. Section 2 presents the related work in summarization. Section 3 explains the LSA approach in detail. Then, the existing algorithms that use different LSA approaches are presented (Gong and Liu, 2001; Steinberger and Jezek 2004; Murray et al., 2005), and two new algorithms are proposed in Section 4. Section 5 presents the evaluation results of these algorithms, and Section 6 presents the concluding remarks.

2 Related Work

Text summarization is an active research area of natural language processing. Its aim is to extract short representative information from input documents. Since the 1950s, various methods are proposed and evaluated. The first studies conducted on text summaries use simple features like terms from keywords/key phrases, terms from user queries, frequency of words, and position of words/sentences (Luhn, 1958).

The use of statistical methods is another approach used for summary extraction. The most well known project that uses statistical approach is the SUMMARIST (Hovy and Lin, 1999). In this project, natural language processing methods are used together with the concept relevance information. The concept relevance information is extracted from dictionaries and WordNet.

Text connectivity is another approach used for summarization. The most well-known algorithm that uses text connectivity is the lexical chains method (Barzilay and Elhadad, 1997; Ercan and Cicekli, 2008). In lexical chains method, WordNet and dictionaries are used to determine semantic relations between words where semantically related words construct lexical chains. Lexical chains are used in the determination of the important sentences of the text.

TextRank (Mihalcea and Tarau, 2004) is a summarization algorithm which is based on graphs, where nodes are sentences and edges represent similarity between sentences. The similarity value is decided by using the overlapping terms. Cluster Lexrank (Qazvinian and Radev, 2008) is another graph-based summarization algorithm, and it tries to find important sentences in a graph in which nodes are sentences and edges are similarities.

In recent years, algebraic methods are used for text summarization. Most well-known algebraic algorithm is Latent Semantic Analysis (LSA) (Landauer et al., 1998). This algorithm finds similarity of sentences and similarity of words using an algebraic method, namely Singular Value Decomposition (SVD). Besides text summarization, the LSA algorithm is also used for document clustering and information filtering.

3 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is an algebraic-statistical method that extracts meaning of words and similarity of sentences using the information about the usage of the words in the context. It keeps information about which words are used in a sentence, while preserving information of common words among sentences. The more common words between sentences mean that those sentences are more semantically related.

LSA method can represent the meaning of words and the meaning of sentences simultaneously. It averages the meaning of words that a sentence contains to find out the meaning of that sentence. It represents the meaning of words by averaging the meaning of sentences that contain this word.

LSA method uses Singular Value Decomposition (SVD) for finding out semantically similar words and sentences. SVD is a method that models relationships among words and sentences. It has the capability of noise reduction, which leads to an improvement in accuracy.

LSA has three main limitations. The first limitation is that it uses only the information in the input text, and it does not use the information of world knowledge. The second limitation is that it does not use the information of word order, syntactic relations, or morphologies. Such information is used for finding out the meaning of words and texts. The third limitation is that the performance of the algorithm decreases with large and inhomogeneous data. The decrease in performance is observed since SVD which is a very complex algorithm is used for finding out the similarities.

All summarization methods based on LSA use three main steps. These steps are as follows:

1. *Input Matrix Creation:* A matrix which represents the input text is created. The columns of the matrix represent the sentences of the input text and the rows represent the words. The cells are filled out to represent the importance of words in sentences using different approaches, whose details are described in the rest of this section. The created matrix is sparse.
2. *Singular Value Decomposition (SVD):* Singular value decomposition is a mathematical method which models the relationships among terms and sentences. It decomposes the input matrix into three other matrices as follows:

   $$A = U \sum V^T$$

   where A is the input matrix with dimensions m x n, U is an m x n matrix which represents the description of the original rows of the input matrix as a vector of extracted concepts, $\sum$ is an n x n diagonal matrix containing scaling values sorted in descending order, and V is an m x n matrix which represents the description of the original columns of input matrix as a vector of the extracted concepts.
3. *Sentence Selection*: Different algorithms are proposed to select sentences from the input text for summarization using the results of SVD. The details of these algorithms are described in Section 4.

The creation of the input matrix is important for summarization, since it affects the resulting matrices of SVD. There are some ways to reduce the row size of the input matrix, such as eliminating words seen in stop words list, or using the root words only. There are also different approaches to fill out the input matrix cell values, and each of them affects the performance of the summarization system differently. These approaches are as follows:

1. *Number of Occurrence:* The cell is filled with the frequency of the word in the sentence.
2. *Binary Representation of Number of Occurrence:* If the word is seen in the sentence, the cell is filled with 1; otherwise it is filled with 0.
3. *TF-IDF (Term Frequency–Inverse Document Frequency):* The cell is filled with TF-IDF value of the word. This method evaluates the importance of words in a sentence. The importance of a word is high if it is frequent in the sentence, but less frequent in the document. TF-IDF is equal to TF*IDF, and TF and IDF are computed as follows:

   $$tf\,(i,j) = n(i,j) \;/\; \textstyle\sum_k n(k,j)$$

   where $n(i,j)$ is the number of occurrences of the considered word i in sentence j, and $\sum_k n(k,j)$ is the sum of number of occurrences of all words in sentence j.

   $$idf\,(i) = log(\,|D| \,/\, d_i)$$

   where $|D|$ is the total number of sentences in the input text, and d_i is the number of sentences where the word i appears
4. *Log Entropy*: The cell is filled with log-entropy value of the word, and it is computed as follows.

   $$sum = \textstyle\sum_j p(i,j)\; log_2(p(i,j))$$

   $$global(i) = 1 + (sum \,/\, log_2(n))$$

   $$local(i,j) = log_2(1 + f(i,j))$$

   $$log\text{-}entropy = global*local$$

   where $p(i,j)$ is the probability of word i that is appeared in sentence j, $f(i,j)$ is the number of times word i appeared in sentence j, and n is the number of sentences in the document.
5. *Root Type:* If the root type of the word is noun, the related cell is filled with the frequency of the word in the sentence; otherwise the cell is filled with 0.

6. *Modified TF-IDF:* First the matrix is filled with TF-IDF values. Then, the average TF-IDF values in each row are calculated. If the value in the cell is less than or equal to the average value, the cell value is set to 0. This is our new approach which is proposed to eliminate the noise from the input matrix.

4 Text Summarization

The algorithms in the literature that use LSA for text summarization perform the first two steps of LSA algorithm in the same way. They differ in the way they fill out the input matrix cells.

4.1 Sentence Selection Algorithms in Literature

4.1.1. Gong & Liu (Gong and Liu, 2001)

After performing the first two steps of the LSA algorithm, Gong & Liu summarization algorithm uses V^T matrix for sentence selection. The columns of V^T matrix represent the sentences of the input matrix and the rows of it represent the concepts that are obtained from SVD method. The most important concept in the text is placed in the first row, and the row order indicates the importance of concepts. Cells of this matrix give information about how much the sentence is related to the given concept. A higher cell value means the sentence is more related to the concept.

In Gong & Liu summarization algorithm, the first concept is chosen, and then the sentence most related to this concept is chosen as a part of the resulting summary. Then the second concept is chosen, and the same step is executed. This repetition of choosing a concept and the sentence most related to that concept is continued until a predefined number of sentences are extracted as a part of the summary. In Figure 1, an example V^T matrix is given. First, the concept *con0* is chosen, and then the sentence *sent1* is chosen, since it has the highest cell value in that row.

There are some disadvantages of this algorithm, which are defined by Steinberger and Jezek (2004). First, the reduced dimension size has to be the same as the summary length. This approach may lead to the extraction of sentences from less significant concepts. Second, there exist some sentences that are related to the chosen concept somehow, but do not have the highest cell value in the row of that concept. These kinds of sentences cannot be included in the resulting summary by this algorithm. Third, all chosen concepts are thought to be in the same importance level, but some of those concepts may not be so important in the input text.

	sent0	sent1	sent2	sent3	sent4
con0	0,557	0,691	0,241	0,110	0,432
con1	0,345	0,674	0,742	0,212	0,567
con2	0,732	0,232	0,435	0,157	0,246
con3	0,628	0,836	0,783	0,265	0,343

Figure 1. Gong & Liu approach: From each row of V^T matrix which represents a concept, the sentence with the highest score is selected. This is repeated until a predefined number of sentences are collected.

4.1.2. Steinberger & Jezek (Steinberger and Jezek 2004)

As in the Gong & Liu summarization algorithm, the first two steps of LSA algorithm are executed before selecting sentences to be a part of the resulting summary. For sentence selection, both V and $\sum$ matrixes are used.

The sentence selection step of this algorithm starts with the calculation of the length of each sentence vector which is represented by a row in V matrix. In order to find the length of a sentence vector, only concepts whose indexes are less than or equal to the number of dimension in the new space is used. The dimension of a new space is given as a parameter to the algorithm. The concepts which are highly related to the text are given more importance by using the values in $\sum$ matrix as a multiplication parameter. If the dimension of the new space is *n*, the length of the sentence *i* is calculated as follows:

$$length_i = \sqrt{\sum_{j=1}^{n} V_{ij} * \Sigma_{jj}}$$

After the calculation of sentence lengths, the longest sentences are chosen as a part of the resulting summary. In Figure 2, an example V matrix is given, and the dimension of the new space is assumed to be 3. The lengths of the sentences are calculated using the first three concepts. Since the sentence *sent2* has the highest length, it is extracted first as a part of the summary.

The aim of this algorithm is to get rid of the disadvantages of Gong & Liu summarization algorithm, by choosing sentences which are related to all important concepts and at the same time choosing more than one sentence from an important topic.

	con0	con1	con2	con3	length
sent0	0,846	0,334	0,231	0,210	0,432
sent1	0,455	0,235	0,432	0,342	0,543
sent2	0,562	0,632	0,735	0,857	0,723
sent3	0,378	0,186	0,248	0,545	0,235

Figure 2. Steinberger & Jezek approach: For each row of V matrix, the lengths of sentences using n concepts are calculated. The value n is given as an input parameter. ∑ matrix values are also used as importance parameters in the length calculations.

	sent0	sent1	sent2	sent3	sent4
con0	0,557	0,691	0,241	0,110	0,432
con1	0,345	0,674	0,742	0,212	0,567
con2	0,732	0,232	0,435	0,157	0,246
con3	0,628	0,836	0,783	0,265	0,343

Figure 3. Murray & Renals & Carletta approach: From each row of V^T matrix, concepts, one or more sentences with the higher scores are selected. The number of sentences to be selected is decided by using ∑ matrix.

4.1.3. Murray & Renals & Carletta (Murray et al., 2005)

The first two steps of the LSA algorithm are executed, as in the previous algorithms before the construction of the summary. V^T and ∑ matrices are used for sentence selection.

In this approach, one or more sentences are collected from the topmost concepts in V^T matrix. The number of sentences to be selected depends on the values in the ∑ matrix. The number of sentences to be collected for each topic is determined by getting the percentage of the related singular value over the sum of all singular values, which are represented in the ∑ matrix. In Figure 3, an example V^T matrix is given. Let's choose two sentences from concept *con0*, and one sentence from *con1*. Thus, the sentences *sent1* and *sent0* are selected from *con0*, and *sent2* is selected from *con1* as a part of the summary.

This approach tries to solve the problems of Gong & Liu's approach. The reduced dimension has not to be same as the number of sentences in the resulting summary. Also, more than one sentence can be chosen even they do not have the highest cell value in the row of the related concept.

4.2 Proposed Sentence Selection Algorithms

The analysis of input documents indicates that some sentences, especially the ones in the introduction and conclusion parts of the documents, belong to more than one main topic. In order to observe whether these sentences are important or they cause noise in matrices of LSA, we propose a new method, named as *Cross*.

Another concern about matrices in LSA is that the concepts that are found after the SVD step may represent main topics or subtopics. So, it is important to determine whether the found concepts are main topics or subtopics. This causes the ambiguity that whether these concepts are subtopics of another main topic, or all the concepts are main topics of the input document. We propose another new method, named as *Topic*, in order to distinguish main topics from subtopics and make sentence selections from main topics.

4.2.1. Cross Method

In this approach, the first two steps of LSA are executed in the same way as the other approaches. As in the Steinberger and Jezek approach, the V^T matrix is used for sentence selection. The proposed approach, however, preprocesses the V^T matrix before selecting the sentences. First, an average sentence score is calculated for each concept which is represented by a row of V^T matrix. If the value of a cell in that row is less than the calculated average score of that row, the score in the cell is set to zero. The main idea is that there can be sentences such that they are not the core sentences representing the topic, but they are related to the topic in some way. The preprocessing step removes the overall effect of such sentences.

After preprocessing, the steps of Steinberger and Jezek approach are followed with a modification. In our Cross approach, first the cell values are multiplied with the values in the ∑ matrix, and the total lengths of sentence vectors, which are represented by the columns of the V^T matrix, are calculated. Then, the longest sentence vectors are collected as a part of the resulting summary.

In Figure 4, an example V^T matrix is given. First, the average scores of all concepts are calculated, and the cells whose values are less than the average value of their row are set to zero. The boldface numbers are below row averages in Figure 4, and they are set to zero before the calculation of the length scores of sentences. Then, the length score of each sentence is calculated by adding up the concept scores of sentences in the updated matrix. In the end, the sentence *sent1* is chosen for the summary as the first sentence, since it has the highest length score.

	sent0	sent1	sent2	sent3	average
con0	0,557	0,691	**0,241**	**0,110**	0,399
con1	**0,345**	0,674	0,742	**0,212**	0,493
con2	0,732	**0,232**	0,435	**0,157**	0,389
con3	**0,628**	**0,436**	0,783	0,865	0,678
con4	0,557	0,691	**0,241**	0,710	0,549
length	1,846	2,056	1,960	1,575	

Figure 4. Cross approach: For each row of V^T matrix, the cell values are set to zero if they are less than the row average. Then, the cell values are multiplied with the values in the ∑ matrix, and the lengths of sentence vectors are found, by summing up all concept values in columns of V^T matrix, which represent the sentences.

4.2.2. Topic Method

The first two steps of LSA algorithm are executed as in the other approaches. For sentence selection, the V^T matrix is used. In the proposed approach, the main idea is to decide whether the concepts that are extracted from the matrix V^T are really main topics of the input text, or they are subtopics. After deciding the main topics which may be a group of subtopics, the sentences are collected as a part of the summary from the main topics.

	sent0	sent1	sent2	sent3	average
con0	0,557	0,691	**0,241**	**0,110**	0,399
con1	**0,345**	0,674	0,742	**0,212**	0,493
con2	0,732	**0,232**	0,435	**0,157**	0,389
con3	**0,628**	**0,436**	0,783	0,865	0,678
con4	0,557	0,691	**0,241**	0,710	0,549

⇩

	con0	con1	con2	con3	con4	strength
con0	1,248	1,365	1,289	0	2,496	6,398
con1	1,365	1,416	1,177	1,525	1,365	6,848
con2	1,289	1,177	0,732	1,218	1,289	5,705
con3	0	1,525	1,218	1,648	1,575	5,966
con4	2,496	1,365	1,289	1,575	1,958	8,683

⇩

	sent0	sent1	sent2	sent3
con0	0,557	0.691	0	0
con1	0	0,674	0,742	0
con2	0,732	0	0,435	0
con3	0	0	0,783	0,865
con4	0,557	0.691	0	0,710

Figure 5. Topic approach: From each row of V^T matrix, concepts, the values are set to zero if they are less than the row average. Then *concept x concept* similarity matrix is created, and the strength values of concepts are calculated, which show how strong the concepts are related to the other concepts. Then the concept whose strength value is highest is chosen, and the sentence with the highest score from that concept is collected. The sentence selection s repeated until a predefined number of sentences is collected.

In the proposed algorithm, a preprocessing step is executed, as in the Cross approach. First, for each concept which is represented by a row of V^T matrix, the average sentence score is calculated and the values less than this score are set to zero. So, a sentence that is not highly related to a concept is removed from the concept in the V^T matrix. Then, the main topics are found. In order to find out the main topics, a *concept x concept* matrix is created by summing up the cell values that are common between the concepts. After this step, the strength values of the concepts are calculated. For this calculation, each concept is thought as a node, and the similarity values in *concept x concept* matrix are considered as edge scores. The strength value of each concept is calculated by summing up the values in each row in *concept x concept* matrix. The topics with the highest strength values are chosen as the main topic of the input text.

After the above steps, sentence selection is performed in a similar manner to Gong and Liu approach. For each main topic selected, the sentence with the highest score is chosen. This selection is done until predefined numbers of sentences are collected.

In Figure 5, an example V^T matrix is given. First, the average scores of each concept is calculated and shown in the last column of the matrix. The cell values that are less than the row average value (boldface numbers in Figure 5) are set to zero. Then, a *concept x concept* matrix is created by filling a cell with the summation of the cell values that are common between those two concepts. The strength values of the concepts are calculated by summing up the concept values, and the strength values are shown in the last column of the related matrix. A higher strength value indicates that the concept is much more related to the other concepts, and it is one of the main topics of the input text. After finding out the main topic which is the concept *con4* in this example, the sentence with the highest cell value which is sentence *sent3* is chosen as a part of the summary.

5 Evaluation

Two different sets of scientific articles in Turkish are used for the evaluation our summarization approach. The articles are chosen from different areas, such as medicine, sociology, psychology, having fifty articles in each set. The second data set has longer articles than the first data set. The abstracts of these articles, which are human-generated summaries, are used for comparison. The sentences in the abstracts may not match with the sentences in the input text. The statistics about these data sets are given in Table 1.

Evaluation of summaries is an active research area. Judgment of human evaluators is a common approach for the evaluation, but it is very time consuming and may not be objective. Another approach that is used for summarization evaluation is to use the ROUGE evaluation approach (Lin and Hovy, 2003), which is based on n-gram co-occurrence, longest common subsequence and weighted longest common subsequence between the ideal summary and the extracted summary. Although we obtained all ROUGE results (ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-W and ROUGE-L) in our evaluations, we only report ROUGE-L results in this paper. The discussions that are made depending on our ROUGE-L results are also applicable to other ROUGE results. Different LSA approaches are executed using different matrix creation methods.

	DS1	DS2
Number of documents	50	50
Sentences per document	89,7	147,3
Words per document	2302,2	3435
Words per sentence	25,6	23,3

Table 1. Statistics of datasets

	G&L	S&J	MRC	Cross	Topic
frequency	0,236	0,250	0,244	0,302	0,244
binary	0,272	0,275	0,274	0,313	0,274
tf-idf	0,200	0,218	0,213	0,304	0,213
logentropy	0,230	0,250	0,235	0,302	0,235
root type	0,283	0,282	0,289	0,320	0,289
mod. tf-idf	0,195	0,221	0,223	0,290	0,223

Table 2. ROUGE-L scores for the data set DS1

In Table 2, it can be observed that the Cross method has the highest ROUGE scores for all matrix creation techniques. The Topic method has the same results with Murray & Renals & Carletta approach, and it is better than the Gong & Liu approach.

Table 2 indicates that all algorithms give their best results when the input matrix is created using the root type of words. Binary and log-entropy approaches also produced good results. Modified tf-idf approach, which is proposed in this paper, did not work well for this data set. The modified tf-idf approach lacks performance because it removes some of the sentences/words from the input matrix, assuming that they cause noise. The documents in the data set DS1 are shorter documents, and most of words/sentences in shorter documents are important and should be kept.

	G&L	S&J	MRC	Cross	Topic
frequency	0,256	0,251	0,259	0,264	0,259
binary	0,191	0,220	0,189	0,274	0,189
tf-idf	0,230	0,235	0,227	0,266	0,227
logentropy	0,267	0,245	0,268	0,267	0,268
root type	0,194	0,222	0,197	0,263	0,197
mod. tf-idf	0,234	0,239	0,232	0,268	0,232

Table 3. ROUGE-L scores for the data set DS2

From Table 3, it can be observed that the *Cross* approach has also the highest ROUGE scores for longer documents. The *Topic* approach has almost the same results with Gong & Liu approach and Murray& Renals & Carletta approach.

Table 3 indicates that the best F-score is achieved for all when the log-entropy method is used for matrix creation. Modified tf-idf approach is in the third rank for all algorithms. We can also observe that, creating matrix according to the root types of words did not work well for this data set.

Given the evaluation results it can be said that *Cross* method, which is proposed in this paper, is a promising approach. Also *Cross* approach is not affected from the method of matrix creation. It produces good results when it is compared against an abstractive summary which is created by a human summarizer.

6 Conclusion

The growth of text based resources brings the problem of getting the information matching needs of user. In order to solve this problem, text summarization methods are proposed and evaluated. The research on summarization started with the extraction of simple features and improved to use different methods, such as lexical chains, statistical approaches, graph based approaches, and algebraic solutions. One of the algebraic-statistical approaches is Latent Semantic Analysis method.

In this study, text summarization methods which use Latent Semantic Analysis are explained. Besides well-known Latent Semantic Analysis approaches of Gong & Liu, Steinberger & Jezek and Murray & Renals & Carletta, two new approaches, namely Cross and Topic, are proposed.

Two approaches explained in this paper are evaluated using two different datasets that are in Turkish. The comparison of these approaches is done using the ROUGE-L F-measure score. The results show that the Cross method is better than all other approaches. Another important result of this approach is that it is not affected by different input matrix creation methods.

In future work, the proposed approaches will be improved and evaluated in English texts as well. Also, ideas that are used in other methods, such as graph based approaches, will be used together with the proposed approaches to improve the performance of summarization.

Acknowledgments

This work is partially supported by The Scientific and Technical Council of Turkey Grant “TUBITAK EEEAG-107E151”.

References

Barzilay, R. and Elhadad, M. 1997. Using Lexical Chains for Text Summarization. *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 10-17.

Ercan G. and Cicekli, I. 2008. Lexical Cohesion based Topic Modeling for Summarization. *Proceedings of 9th Int. Conf. Intelligent Text Processing and Computational Linguistics (CICLing-2008)*, pages 582-592.

Gong, Y. and Liu, X. 2001. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. *Proceedings of SIGIR'01*.

Hahn, U. and Mani, I. 2000. The challenges of automatic summarization. *Computer*, **33**, 29–36.

Hovy, E. and Lin, C-Y. 1999. Automated Text Summarization in SUMMARIST. I. Mani and M.T. Maybury (eds.), *Advances in Automatic Text Summarization*, The MIT Press, pages 81-94.

Landauer, T.K., Foltz, P.W. and Laham, D. 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25, 259-284.

Lin, C.Y. and Hovy, E.. 2003. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. *Proceedings of 2003 Conf. North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL-2003),* pages 71-78.

Luhn, H.P. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research Development* 2(2), 159-165.

Mihalcea, R. and Tarau, P. 2004. Text-rank - bringing order into texts. *Proceeding of the Conference on Empirical Methods in Natural Language Processing*.

Murray, G., Renals, S. and Carletta, J. 2005. Extractive summarization of meeting recordings. *Proceedings of the 9th European Conference on Speech Communication and Technology*.

Qazvinian, V. and Radev, D.R. 2008. Scientific paper summarization using citation summary networks. *Proceedings of COLING2008*, Manchester, UK, pages 689-696.

Steinberger, J. and Jezek, K. 2004. Using Latent Semantic Analysis in Text Summarization and Summary Evaluation. *Proceedings of ISIM '04*, pages 93-100.