INISTA 2010
International Symposium on Innovations in Intelligent Systems and Applicaitons, 21-24June 2010, Kayseri & Cappadocia,TURKEY

A Comparison of Feature and Semantic-Based Summarization Algorithms for Turkish

Aysun Güran
Yıldız Technical University,
dogrusozaysun@hotmail.com

Eren Bekar
Yıldız Technical University,
erenbekar@gmail.com

Selim Akyokuş
Doğuş University,
sakyokus@dogus.edu.tr

Abstract

In this paper we analyze the performances of a feature-based and two semantic-based text summarization algorithms on a new Turkish corpus. The feature-based algorithm uses the statistical analysis of paragraphs, sentences, words and formal clues found in documents, whereas the two semantic-based algorithms employ Latent Semantic Analysis (LSA) approach which enables the selection of the most important sentences in a semantic way. Performance evaluation is conducted by comparing automatically generated summaries with manual summaries generated by a human summarizer. This is the first study that applies LSA based algorithms to Turkish text summarization and its results are promising.

1. Introduction

The World Wide Web is a massive source for storing and accessing textual information. The amount of information available in this source grows rapidly. People are beset with unprecedented difficulties because of this rapid growth. One of these difficulties is the lack of an efficient method to reach the required information in a reasonable time. Text Summarization (TS) is considered as an essential task to overcome this problem. Summaries are extremely useful in allowing users to quickly understand the main theme of the whole document. In addition, summaries significantly improve their search experience and effectively reduce their searching time.

TS is the process of identifying the most salient information in a document or a set of related documents. There are two main approaches to the task of summarization: *extraction* and *abstraction*. Extraction involves selecting the most important existing sentences, whereas abstraction involves generating novel sentences from a given document. The abstractive summarization approaches use information extraction, ontological information, information fusion and compression [1],[2],[3],[4],[5]. These approaches require a deeper understanding of documents and they are limited to small domains. In contrast to the abstractive summarization, extractive summarization approaches are more practical.

Many different types of extractive text summarization approaches can be found in literature. These approaches extract sentences that contain the most important concepts in a document or a set of related documents. The study done by [6] exploits word distribution of a given document based on the intuition that the most frequent words represent the most important concepts. The study in [7] is based on the cue phrase method that uses meta-linguistic markers (for example, \in conclusion", \the paper describes") and uses the location method which relies on the following intuition: headings, text formatted in bold, sentences in the beginning and end of the text contain important information for a summary. The study given in [8] uses learning in order to combine several shallow heuristics (cue phrase, location, sentence length, word frequency and title). The study in [9] proposes a learning-based approach to combine various sentence features that categorize sentences according to surface, content, relevance and event features. All the methods given above depend on formal clues found in documents.

Latent Semantic Analysis (LSA) is a method for extracting semantic generalizations from textual passages on the basis of their contextual use [10], [11], [12], [13], [14]. It has found applications in a number of areas such as text retrieval, text segmentation and more recently, single or multiple text summarization systems. It is based on Singular Value Decomposition (SVD) of an $m\times n$ term-document (or term-sentence) matrix. SVD models the interrelationships among terms so that it can semantically cluster terms and sentences.

Turkish language is one of the most commonly used 20 languages in the world. It has agglutinative morphology which means that various new words can be derived by adding suffixes to a root of a word. Working with an agglutinative language such as Turkish is a real and important research issue in the context of text summarization. In contrast to the other languages, there are not many researches done on TS in Turkish language. This is partly due to the lack of a standard summarization test collection in Turkish. The previous studies about Turkish summarization are done by [15], [16], [17], [18] and [19]. These studies principally depend on the statistical analysis of paragraphs, sentences and words in documents by considering some specific weighting factors.

In this study, we apply three different algorithms on a new Turkish corpus that contains 50 documents. These algorithms are belong to [10],[11],[19]. The first study is done by Kılcı and Diri [19]. It is based on formal clues and statistical analysis of documents. The other two algorithms are based on LSA. These two algorithms [10],[11] represent the term-document matrix with different weighting schemes and differ from each other on the criteria of important sentence selection. We compare the performances of these algorithms on our data corpus and propose future work to improve their performances.

The outline of the paper is as follows: Section 2 describes the summary of algorithms. Section 3 explains data corpus and the evaluation data set. Section 4 demonstrates the performance analysis of three algorithms. Finally, Section 5 presents the conclusion and suggested future work directions.

2. Summary of Algorithms

The three different algorithms used in this study are described in this section.

2.1 Algorithm I

This algorithm [19] principally depends on the statistical analysis of paragraphs, sentences and words of a given document. The algorithm proposes a method that creates summaries by selecting the most important sentences with the help of a score function. In order to generate summaries, this function uses several kinds of document features, such as *key phrases, term frequencies, the positions of sentences in original text, the average lengths of sentences, the existence of title keywords, positive and negative cue words and adverbs, the existence of some punctuation marks, day-month names, numeric literals* and *proper nouns*. In this algorithm, first of all, a document is scanned and decomposed into individual sentences. Later on, sentences are ranked by their score which emphasize their significance. Finally, the top scored sentences are selected according to sequential order that appears in the original document.

A detailed explanation of list of features used in the score calculation of Algorithm I is given below [19]:

- *Title* – The sentences that contain the title and subtitle words of a document increase the value of score function.
- *Positive (Cue) Words*: These are words such as *"özetle", "sonuç olarak", "kısacası", "neticede"* etc., that semantically emphasize the importance of sentences. The score of sentences is increased whenever these cue words occur in sentences.
- *Paragraph Location*: The first and last paragraphs of documents are assumed to have higher importance.
- *Proper Nouns:* The words that begin with a capital letter are assumed to be proper nouns. The sentences with proper nouns have higher importance.
- *Term Frequency:* The frequency of terms (words) in a document is calculated except stopwords. The sentences with higher term frequencies terms have higher importance.
- *Adverbs:* Certain adverbs, referred as collocations in [x], such as *"açık açık"*, "*adeta", "kelimesi kelimesine", "iyiden iyiye",* etc., emphasize the importance of sentences. The sentences with this kind of adverbs have higher importance.
- *Negative (Cue) Words* – These are words such as "*öyleyse", "çünkü",* etc., that semantically explains the reason of the previous sentence and has less significance. The score of sentences is reduced whenever these cue words occur in sentences.
- *Numeric Literals:* The sentences that contain numeric literals are assumed to have higher importance.
- *Average Length* – The average number of words for each sentence is calculated. The sentences that are close to this average length are assumed to have higher importance.
- *Day/Month* - The sentences that contain the names of week days and months are assumed to have higher importance.
- *Keywords:* This is an optional parameter specified by a user. If the user enters certain keywords, then the sentences with these key words are assumed to have higher importance.
- *Ending Mark:* The punctuation symbols *(?,!)* at the end of sentences emphasize the importance of the sentences and the score of sentences with these marks is increased.
- *Quotations:* The sentences with quotation marks are assumed to have higher importance.

Each of these features has a weighing factor that can be specified by a user. Table 1 shows default values of these weighting factors. In performance analysis default values are used. The evaluation of score function can be found in [19].

Table 1. Default values of weighting factors

Properties	Weights
Title	20
Positive (Cue) Words	15
First Paragraph	20
Proper Nouns	5
Term Frequency	7
Adverbs	5
Negative (Cue) Words	-10
Numeric Literals	3
Average Length	10
Day/Month	5
Keywords	20
Ending Mark	2
Quotations	5
Last Paragraph	10

2.2. LSA Based Algorithms

2.2.1 Data Representation

When LSA is applied to text summarization, a document is represented as a term-sentence matrix in which each row stands for a unique word and each column stands for a sentence on a given document. The input of LSA algorithm is an $m \times n$ term-sentence matrix $A_i=[a_{1i},a_{2i},\ldots,a_{ni}]$, where each entry a_{ji} represents some weights. In the study done by [10], each term a_{ji} in *matrix A* is represented by multiplying a local and a global weighting factor as follows:

$$a_{ji}=L(t_{ji})*G(t_{ji}) \qquad (1)$$

where $L(t_{ji})$ is the local weighting factor for term *j* in sentence *i*, and $G(t_{ji})$ is the global weighting factor for term *j* in the whole document. Local weighting $L(t_{ji})$ has the following four possible alternatives:

- *No weight*: $L(t_{ji})$ = $tf(t_{ji})$ where $tf(t_{ji})$ is the number of times term t_{ji} occurs in the sentence.
- *Binary weight*: $L(t_{ji})$ = 1 if term t_{ji} appears at least once in the sentence; otherwise, $L(t_{ji})$ = 0.
- *Augmented weight:* $L(t_{ji})=0.5+0.5*tf(t_{ji})$ / $tf(max)$ where tf(max) is the frequency of the most frequently occurring term in the sentence.
- *Logarithm weight*: $L(t_{ji})=\log(1+tf(t_{ji}))$

And global weighting $G(t_{ji})$ has the following two possible alternatives:

- *No weight*: $G(t_{ji})$ = 1 for any term i.
- *Inverse Document Frequency*: $G(t_{ji})=\log\left(\frac{N}{n_i}\right)+1$, where N is the total number of sentences in the document, and n_i is the number of sentences that contain term i.

2.2.2 Algorithm II-A

Algorithm II-A is based on the study done by [10]. It uses LSA that depends on SVD. In SVD, a *matrix A* is decomposed into the product of three other matrices:

$$A=USV^T \qquad (2)$$

where $U=[u_{ij}]$ is an $m \times n$ column-orthonormal matrix whose columns are called left singular vectors, $S=\mathrm{diag}(\sigma_1,\sigma_2,\ldots,\sigma_n)$ is an *nxn* diagonal matrix, whose diagonal elements are nonnegative singular values sorted in descending order and $V=[v_{ij}]$ is an $n \times n$ orthonormal matrix, whose columns are called right singular vectors. If rank (A) is r, then S and A will satisfy [10]:

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_n = 0 \qquad (3)$$

$$A=\sum_{i=1}^{r}\sigma_i\, u_i\, v_i^T \qquad (4)$$

$$A=\sigma_1\, u_1\, v_1^T+\ldots+\sigma_k\, u_k\, v_k^T+\ldots+\sigma_r\, u_r\, v_r^T \qquad (5)$$

In order to extract the most important *s* sentences, the following process is applied *s* times starting with k=1:

- Select kth right singular vector in matrix V^T with the highest value. The sentence with the highest value is used in the summary. Increase k by 1.
- If k reaches the predefined number *s*, terminate the process; otherwise, go to the previous step again.

In this method, the greater k value means that the selected sentence is less significant. As a result, the summary may include sentences which are not particularly important. In order to solve this problem, Algorithm II-B given in the next section is proposed.

2.2.3. Algorithm II-B

Algorithm II-B is based on the study done by [11]. Algorithm II-B improves Algorithm II-A by considering the fact that the statistical significance of each LSA dimension is approximately the square of its singular value [20]. The study in [11] exploits this fact and change sentence selection criteria. In this method, first, a matrix B is calculated:

$$B=S^2*V^T \qquad (6)$$

Then, the significance score, S_k, value of each sentence vector in the modified latent vector space B is determined using the following equation:

$$S_k=\sqrt{\sum_{i=1}^{r} b_{i,k}^2} \qquad (7)$$

The Algorithm II-B selects the sentences with the highest S_k score. It is demonstrated in [11] that this modification gives a significant improvement over the Algorithm II-A in [10].

3. Data corpus and the evaluation data set

We constructed a corpus that contains 50 documents collected from the online Turkish newspapers and some news portals. To evaluate the performance of our system, we manually derived an evaluation data set. This evaluation data set was created by a human summarizer with a compression rate. Human summarizer selected %30 of the most important sentences from each document for each summary.

Table 2 shows statistics of the data corpus and the evaluation data set that is constructed by a human summarizer:

Table 2. Statistics of the data corpus and manually created evaluation data set

Property	Data Corpus	Evaluation data set
Sentences /document	24,48	8,12
Words / document	364,6	146,66
Words / sentence	15,15	18,52
Document with min. number of sentences	12	4
Document with max. number of sentences	65	20

The average number of sentences selected by a human summarizer is 8,12. Therefore, approximately, 8 sentences are selected from a given document by each of automatic summarization algorithms used in this study.

4. Experimental Results

Performance analysis of three algorithms is evaluated on the corpus collected for this study. For the performance analysis, we choose an intrinsic evaluation method and used precision (P), recall (R), and f-measure (F). These measures determine the coverage between the summaries constructed manually by a human and the automatically generated summaries. Assuming that T is the manual summary and S is the automatically generated summary, the measurements P, R and F are defined as:

$$P=\frac{|S\cap T|}{|S|},\ R=\frac{|S\cap T|}{|T|},\ F=\frac{2PR}{R+P} \qquad (8)$$

Table 3 shows performance evaluation results of Algorithm I on the evaluation data set. The f-value is calculated as 0,490.

Table 3. Performance evaluation results of Algorithm I

Algorithm I	P	R	F
	0,485	0,526	**0,490**

Table 4 shows performance evaluation results of Algorithm II-A on the evaluation data set. The Algorithm II-A uses eight different weighting schemes as explained in Section 2.2.1. The best f-value (0,4954) is obtained when weighting scheme parameters are set to LI.

Table 4. Performance evaluation results of Algorithm II-A

Algorithm II-A			
Local-Global Weighting Scheme	**P**	**R**	**F**
NN	0,4775	0,5267	0,488
NI	0,4825	0,5346	0,4943
BN	0,48	0,5239	0,4877
BI	0,485	0,5303	0,493
AN	0,4775	0,5203	0,4858
AI	0,455	0,4968	0,4623
LN	0,485	0,5293	0,4929
LI	0,4875	0,5328	**0,4954**

Table 5 depicts performance analysis of Algorithm II-B on the evaluation data set. The best f-value (0,5135) is found with LN weighting scheme.

Table 5. Performance evaluation results of Algorithm II-B

Algorithm II-B			
Local-Global Weighting Scheme	**P**	**R**	**F**
NN	0,4875	0,5382	0,4984
NI	0,4975	0,549	0,5087
BN	0,495	0,549	0,5082
BI	0,4925	0,5445	0,5045
AN	0,485	0,5372	0,4974
AI	0,4775	0,5213	0,4859
LN	0,5	0,5554	**0,5135**
LI	0,4975	0,5482	0,5084

The obtained experimental results show that LSA based algorithms perform better than Algorithm I. It might be expected that Algorithm I should produce better results due to the many different types of features. However, the LSA based algorithms perform better. The main reason for this can be the semantic nature of the LSA algorithm. It allows textual passages to be compared to each other more intelligently than by directly comparing the words they share. It also enables a meaningful comparison of words that never appear together.

The LSA based Algorithm II-B produces the best results because of the modified sentence selection criteria that is related to the fact that the statistical significance of each LSA dimension is approximately the square of its singular value [20].

5. Conclusion and Future Work

This paper presents the performance analysis of a Turkish text summarization system that applies different algorithms. This is the first study that applies LSA based algorithms to Turkish text summarization and its results are promising.

As a future work we plan to apply other summarization methods. We believe that the features used in Algorithm I can be used to improve the performance of LSA based algorithms which can find the best similarity between small groups of terms in a semantic way. Another objective is to extend our current corpus and make it available to other researchers working in this area.

6. References

[1] Dejong, G., "Fast Skimming of News Stories: The FRUMP System", Ph.D. thesis at Yale University, New Haven, CT, 1978.

[2] Rau, L. and Jacobs, P., "Creating segmented databases from free text for text retrieval", *In Proceedings of the 14th ACM-SIGIR Conference*, New York, 1991, pp. 337–346.

[3] Witbrock, M. and Vibhu M., "Ultra-summarization: A statistical approach to generating highly condensed non-extractive summaries", *In Proceedings of the 22nd Annual International ACM SIGIR Conference,* Berkeley, 1999, pp. 315–316.

[4] Jing, H., "Using hidden Markov modeling to decompose human-written summaries", *Computational Linguistics*, 2002, 28(4), pp. 527–543.

[5] Knight, K., and Daniel M., "Statistics-based summarization—Step one: Sentence compression.", *In Proceedings of the 17th National Conference of AAAI*, 2000, pp. 703–710.

[6] Luhn, H. P. "The automatic creation of literature abstracts.", *IBM Journal of Research Development*, 2(2):159–165, 1958.

[7] Edmundson, H. P., "New methods in automatic extracting.", *Journal of the Association for Computing Machinery*, 1969, 16(2), pp. 264–285.

[8] Kupiec, J., Jan O. P., and Francine C., "A trainable document summarizer.", *In Research and Development in Information Retrieval*, pp. 68–73, 1995.

[9]Wong, K., Wu, M. and Li, W., "Extractive Summarization Using Supervised and Semi-Supervised Learning", *In Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, August 18-22, 2008, pp 985-992.

[10] Gong, Y., Liu, X., "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis.", *In the proceeding of ACM SIGIR*, 2001, pp. 19–25.

[11] Steinberger, J., "Text Summarization within the LSA Framework", PhD Thesis, University of West Bohemia in Pilsen, Czech Republic, January 2007.

[12] Li, W., Li, B. and Wu, M., "Query focus guided selection strategy for DUC 2006", *In Proceedings of the DUC Conference,* 2006.

[13] Yeh, J.Y., Ke, H.R., Yang, W.P. and Meng, I.H., "Text summarization using a trainable summarizer and latent semantic analysis", *Information Processing and Management,* 41 (2005), pp. 75–95.

[14] Zha, H., "Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering.", *In Proceedings of the 25th annual international ACM SIGIR conference,* Tampere, Finland, 2002, pp. 113–120.

[15] Altan, Z., "A Turkish Automatic Text Summarization System", *IASTED International Conference on* AIA, 16-18 February 2004, Innsbruck, Austria.

[16] Tülek, M., "Text summarization for Turkish", master thesis, Istanbul Technical University, Turkey, May. 2007.

[17] Uzundere, E., Dedja, E., Diri, B., Amasyalı, M.F, " Automatic Text Summarization for Turkish", *In Proceedings of ASYU 2008*, Isparta, Türkiye.

[18] Cığır, C., Kutlu, M. and Cicekli, I., "Generic Text Summarization for Turkish", *In Proceedings of ISCIS*, Northern Cyprus, 2009.

[19] Kılcı, Y. and Diri,B., "Turkish Text Summarization System",Senior Project, Yıldız Technical University in Turkey, 2008. (*www.kemik.yildiz.edu.tr)*

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