

Efficient feature integration with Wikipedia-based semantic feature extraction for Turkish text summarization

Aysun GÜRAN,^{1,*} Nilgün GÜLER BAYAZIT,² Mustafa Zahid GÜRBÜZ¹

¹Computer Engineering Department, Doğuş University, İstanbul, Turkey

²Mathematical Engineering Department, Yıldız Technical University, İstanbul, Turkey

Received: 06.01.2012 • Accepted: 15.05.2012 • Published Online: 12.08.2013 • Printed: 06.09.2013

Abstract: This study presents a novel hybrid Turkish text summarization system that combines structural and semantic features. The system uses 5 structural features, 1 of which is newly proposed and 3 are semantic features whose values are extracted from Turkish Wikipedia links. The features are combined using the weights calculated by 2 novel approaches. The first approach makes use of an analytical hierarchical process, which depends on a series of expert judgments based on pairwise comparisons of the features. The second approach makes use of the artificial bee colony algorithm for automatically determining the weights of the features. To confirm the significance of the proposed hybrid system, its performance is evaluated on a new Turkish corpus that contains 110 documents and 3 human-generated extractive summary corpora. The experimental results show that exploiting all of the features by combining them results in a better performance than exploiting each feature individually.

Key words: Turkish text summarization, latent semantic analysis, analytical hierarchical process, artificial bee colony algorithm, Turkish Wikipedia

1. Introduction

Automatic document summarization (ADS) is a process where a computer summarizes a document. In this process, a document is entered into the computer and a summarized document is returned. The summarized document is extremely useful in allowing users to quickly understand the main theme of the whole document and it effectively saves their searching time.

ADS can perform extractive and abstractive summarization tasks. Extractive summarization techniques involve selecting the most important existing sentences, whereas abstractive summarization techniques involve generating novel sentences from given documents. The abstractive summarization approaches require a deeper understanding of the documents. The existing abstractive summarization works have been quite limited and can be categorized into 2 types: methods using prior knowledge [1,2] and methods using natural language generation systems [3,4]. In contrast to the abstractive summarization approaches, extractive summarization approaches are more practical. Most of them represent documents with some *structural* and *semantic* sentence features that indicate sentence importance using a sentence score function. Studies [5–7] represent documents with structural features such as the term frequency, sentence position, and title words, and combine them to get an effective sentence score function. Studies [8–10] represent documents with semantic sentence features based on latent semantic analysis (LSA), probabilistic latent semantic analysis [11], and nonnegative matrix

*Correspondence: adogrusoz@dogus.edu.tr

factorization (NMF) [12], which analyze the relationships between a set of sentences and terms by producing a set of topics related to the sentences and the terms.

Algorithms for text summarization based on machine learning algorithms [13,14] consist of 2 phases: the training phase and the test phase. The training phase extracts important features from the training corpus and then generates rules using a learning algorithm. The test phase applies these rules on the test corpus and produces the corresponding summaries.

In recent years, optimization-based methods have also been proposed. The authors in [15] defined text summarization as a maximum coverage problem, whereas the authors in [16] formalized it as a knapsack problem. In [17], document summarization was modeled as a nonlinear 0–1 programming problem that covers the main content of the given documents through sentence assignment.

In contrast to other languages, automatic text summarization has not been extensively studied for the Turkish language. This is partly due to the nonexistence of standard text summarization test collections in Turkish. The previous research about Turkish summarization has been carried out in [18–24], where in [18–21], structural features were used, and [22–24] have used semantic features. In this study, we propose a new hybrid Turkish text summarization system that combines the structural and semantic sentence features. The system employs 5 structural features, 1 of which is newly proposed and 3 are semantic features whose values are extracted from Turkish Wikipedia links. The features are combined using the weights calculated by 2 novel approaches. The first approach makes use of an analytical hierarchical process (AHP) [25], which is a manual process that depends on a series of expert judgments based on pairwise comparisons of the features. The second approach makes use of the artificial bee colony (ABC) algorithm for automatically determining the weights of the features. The ABC algorithm, described in [26], imitates the foraging behavior of honey bees for numerical optimization and classification. In order to see the performance of the proposed hybrid system, we put together a Turkish corpus that contains 110 documents and derived 3 human-generated extractive summary corpora. The performance analysis of the algorithms is conducted on the human-generated extractive summary corpora. As a performance measure, we use the F-measure score that determines the coverage between the manually and automatically generated summaries. We supplement the above metric with the ROUGE evaluation toolkit [27], which is based on the n-gram cooccurrence between the manually generated and automatically generated summaries.

The main contribution of this work is the proposed hybrid text summarization system, which integrates the structural and semantic features' scores into an overall sentence score function using the methods based on the AHP and ABC algorithm. Additional contributions include the proposal of a new structural feature based on the text categorization and a modification of the semantic document features based on the LSA, which uses Turkish Wikipedia as a basis for detecting syntactically related words. The final contribution is the composition of a Turkish corpus that contains 110 news documents and human-generated summary data sets generated by 3 analysts rather than the 1 analyst used in the previous Turkish data sets. To confirm the significance of our contributions, algorithmic results are presented in detail and discussed.

The remaining parts of the paper are organized as follows: Section 2 explains the sentence features used in the extractive summarization. Section 3 outlines how the sentence features are combined via the proposed hybrid system. Section 4 presents the data corpus and the evaluation data set. Section 5 presents the experimental results, and finally, Section 6 gives the concluding remarks.

2. The framework for the proposed hybrid system

The proposed hybrid system combines the structural and semantic features based on the weights calculated by either the AHP or the ABC algorithm. Using the combined sentence score function, it then ranks the sentences in the document and extracts the highest scored sentences to generate a summary.

A detailed description of the 2 types of sentence features is presented below.

2.1. f_1 -Structural features

The structural features included in our model principally depend on the structural analysis of the sentences in the document. These features are the ' f_{11} : Length', ' f_{12} : Position', ' f_{13} : Title', ' f_{14} : Frequency', and ' f_{15} : Class relevance' features. A detailed explanation of these features is given below.

f_{11} - Length: The use of this feature is motivated by the idea that sentences are important if the number of words in them is within a certain range. After the stop words are eliminated and the stemming is applied using Zemberek [28], each sentence is given a length score, which is the number of words contained in the sentence.

f_{12} - Position: Sentences at the beginning of the documents always introduce the main topics that the documents describe. To capture the significances of different sentence positions, each sentence in a document is given a rank according to the formula shown in Eq. (1):

$$Score(f_{12})\frac{1}{P_i}, \quad (1)$$

where P_i is the position of the i th sentence.

f_{13} - Title: This feature is based on the assumption that the sentences are important if they contain the title words of a document. After the stop words are eliminated and the stemming is applied, each sentence is given a title score by summing the number of overlapping words between the title and the sentence.

f_{14} - Frequency: This feature depends on the intuition that words occurring frequently within a document usually have salient information and that sentences with a higher number of such words are important [20]. After the stop words are eliminated and the stemming is applied, each sentence is given a frequency score by summing the frequencies of the constituent words.

f_{15} - Class relevance: This sentence feature is a novel sentence feature that applies the text classification task for summary generation. In order to obtain this feature, first of all, each document to be summarized is classified using the multinomial naïve Bayes algorithm. The classifier is trained with the 1150 documents obtained from the study [29]. This data set contains 5 different classes (economy, magazine, health, political, and sports) of documents and there are 230 documents in each class. Additionally, from this data set, the most frequent unigram, bigram, and trigram word combinations in each class are stored in 5 separate dictionaries. For the document to be summarized, we count the matching unigram, bigram, and trigram words in each sentence using the N-gram word dictionary of its class. If a sentence in the document to be summarized contains these frequent N-gram words in the dictionary for the class of the document, we assume that this sentence is important for the summary generation. Therefore, we assign a sentence score to each sentence according to the number of matches between its N-gram words and the related frequent N-gram word dictionary. The above procedure is applied after the stop words are eliminated and the stemming is performed on both the text categorization and the summarization data sets.

2.2. f_2 -Semantic features

The semantic features included in our model consist of 3 LSA-based text summarization features: ‘ f_{21} : Relevance to each topic’, ‘ f_{22} : Relevance to overall topic’. and ‘ f_{23} : Relevance to other sentences’. These features use 3 main steps: creation of the input matrix, singular value decomposition, and sentence selection.

2.2.1. Creation of input matrix

In order to extract the above features for text summarization, a document is represented as an $m \times n$ term-sentence matrix $A = [a_{1j}, a_{2j}, \dots, a_{nj}]$, where each entry a_{ij} is obtained by multiplying a local and a global weighting factor as follows: $a_{ij} = L(t_{ij})G(t_{ij})$. Here, $L(t_{ij})$ is defined as $L(t_{ij}) = \log(1 + tf(t_{ij}))$ and $G(t_{ij})$ is defined as $G(t_{ij}) = \log(\frac{N}{n_i}) + 1$, where $tf(t_{ij})$ is the number of times that term t_{ij} occurs in the sentence, N is the total number of sentences in the document, and n_i is the number of sentences that contain term t_{ij} .

In this study, when the matrix A is created, instead of considering the words individually, we detect syntactically related words in the documents and treat them as a single word. For example, Mustafa Kemal Atatürk (the founder of the Turkish Republic), is considered as a single term. In order to find these syntactically related words, we use Turkish Wikipedia (Vikipedi). The main purpose of mining Vikipedi is to extract information by analyzing web links. There are several successful studies that use Wikipedia as an external knowledge resource to enrich text mining applications. In [30,31], a novel method called explicit semantic analysis (ESA) was presented to get better performance for text classification systems with Wikipedia. In their approach, they use a semantic interpreter to represent each text document as a weighted vector of Wikipedia concepts. They then add these Wikipedia concepts to a traditional bag of words approach as new features. Their results show that the ESA with Wikipedia improves the correlation of the computed semantic relatedness score with humans. The study in [32] presented a single-document summarization method that maps document sentences to semantic concepts in Wikipedia and selects sentences based on the frequency of the mapped-to concepts. Their results indicate that the Wikipedia-based summarization method is competitive with the state-of-the-art single document summarization. The study in [33] worked on categorization through syntactically related word associations and the study in [34] used syntactically related words for topic segmentation and link detection. The underlying motivation of these approaches comes from the observation that syntactically related word associations may be used to represent the gist of the semantic content of a document.

Although there are numerous studies using the English Wikipedia in semantic analysis, there are a limited number of studies using Vikipedi [35–37]. The study in [35] employed Vikipedi to discover missing links in a Vikipedi article. The study in [36] integrated semantic information into the suffix tree clustering algorithm using Vikipedi. In [37], knowledge-based word sense disambiguation methods were compared for Turkish texts, using Turkish WordNet as a primary knowledge base and Vikipedi as an enrichment resource. In another study [24], an automatic Turkish document summarization system was built. In that study, the NMF-based summarization algorithm was used with syntactically related word associations.

Wikipedia contains many different types of semantic relationships, such as synonymy, polysemy, categorical information, and hyperlinks, between articles. In our study, we only use the semantic relationship of words that cooccur literally.

Wikipedia has 2 important characteristics: the dense web link structure and the concept identification by the web links, called uniform resource locaters (URLs). Articles are strongly connected to each other by this dense structure of web links. Almost every concept (article/page) has its own URL as an identifier (i.e.

consecutive words that occur in a single URL represent a single concept or entity). In order to find these concepts or entities, all URLs are searched in Wikipedi and the syntactically related words in the links, such as Recep Akdağ (name of a person), Anayasa Mahkemesi (Constitutional Court), Sağlık Bakanlığı (Ministry of Health), and Domuz Gribi (swine flu), are selected. This modification provides semantic integration between consecutive words. In this work, all of the semantic features are extracted after the syntactically related word detection phase and we show that the performance of this modification shows promising results.

2.2.2. Singular value decomposition

Given the $m \times n$ dimensional term-sentence matrix A with rank $r \leq \min(m, n)$, the singular value decomposition (SVD) of A is defined as $A = USV^T$, where U is an $m \times r$ column-orthonormal matrix whose columns are called left singular vectors, $S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ is an $r \times r$ diagonal matrix of the singular values whose diagonal elements are nonnegative singular values sorted in descending order, and V^T is an $r \times n$ orthonormal matrix whose rows are called right singular vectors.

From a semantic perspective, the SVD indicates a breakdown of the original matrix A into r topics that contain salient patterns of word combinations in the document. In this definition, each $column_i$ of matrix A corresponding to the $sentence_i$ in the document is mapped to $column_i$ of V^T . Matrix U emphasizes the mapping between the space of r topics and the space of the m terms. The singular values of S specify the importance of the selected topics [38].

2.2.3. Sentence selection

The features f_{21} , f_{22} , and f_{23} , extracted by the SVD, are the scores assigned to each sentence of a given document. The details of these features are described below.

f_{21} - *Relevance to each topic*: When the SVD is applied to a document, the sentences of the document are represented by the columns of V^T and the extracted topics that are considered in the order from 1 through r are represented by the row of V^T . Here, the row order emphasizes the importance of the topics (i.e. the first row of V^T represents the most important topic and the last row of V^T represents the less important topic). For a $topic_k$, row $V_k^T = [v_{k,1}, v_{k,2}, \dots, v_{k,n}]$ of matrix V^T is considered [8]. The elements of this row specify the weights of $topic_k$ in $nsentences$. In our hybrid system, the $sentence_i$ with the maximum $v_{k,i}$ element is selected. This means that the $sentence_i$ matches the $topic_k$ better than others. After detecting the sentence with the maximum value, we assign a sentence score to the detected sentence using the below formula:

$$Score_{f_{21}} = \frac{1}{r_i}, \quad (2)$$

where r_i is the row order of V^T that emphasizes the importance of the sentence extracted from topic r_i .

The aim of this feature is to detect the sentences that are related to the most important topics.

f_{22} - *Relevance to overall topics*: This feature assigns a numerical value to each sentence in a document based on the study in [9]. According to this study, after performing the SVD on A , the right singular vector matrix V^T and the diagonal matrix S is obtained. Next, a modified latent vector space B is constructed:

$$B = S^2V^T. \quad (3)$$

Using the modified latent vector space B , each sentence is given a sentence score using Eq. (4):

$$S_k = \sqrt{\sum_{i=1}^r b_{i,k}^2}. \quad (4)$$

The study in [9] points out that a higher S_k value indicates that the sentence is more related to all of the important topics extracted from the document. Hence, it can be said that the aim of this feature is to detect the sentences that are related to all of the important topics.

f₂₃-Relevance to other sentences: Based on the study in [10], the feature f_{23} applies a process of dimension reduction to A , by considering only the first k largest original singular values in S . The result of the dimension reduction is a matrix $A_k = U_k S_k V_k^T$, where $k \leq \min(m, n)$.

After the dimension reduction process is applied, for each document, a sentence relationship map is established with links between the sentence vectors of the corresponding columns of A_k . The links are invalid if the sentences have a high similarity, which is computed as the inner product between the sentence vectors of the corresponding columns of A_k (in this study, to decide whether a link should be considered as a valid link, we set the similarity threshold value as 0.02). The similarity between a pair of sentences S_i and S_j is defined in Eq. (5):

$$Sim(S_i, S_j) = \frac{\vec{S}_i \vec{S}_j}{|\vec{S}_i| |\vec{S}_j|}. \quad (5)$$

For the final phase, f_{23} gives a sentence score to each sentence by counting the number of valid links that it has.

3. Combining sentence features via the hybrid system

In our proposed hybrid system, to generate a summary of a given document, first, all of the structural and semantic feature scores are normalized using z-score normalization [39], which converts the scores to a common scale with an average of 0 and a standard deviation of 1. After the normalization, the features of the sentence S are combined by the following linear model:

$$Score(S) = \sum_{j=1}^5 w_{1j} f_{1j} + \sum_{j=1}^3 w_{2j} f_{2j}, \quad (6)$$

where w_{ij} denotes the weight of feature f_{ij} .

In the above, the weights are determined by 2 different approaches. The first approach employs an AHP, which is a manual process, whereas, in the second approach, the weights are automatically learned by the ABC optimization algorithm. A detailed explanation of these approaches is given below.

3.1. Finding the optimal weight of the features with the AHP

The AHP was developed by Saaty [25] and compares criteria, or alternatives with respect to a criterion, in a pairwise fashion. It has been adopted for solving mathematical psychology and multicriteria decision making problems. It has also been used for combining classifiers in [40], where an AHP-based combined classifier produced a more robust classification performance.

In this section, we describe how the optimal weights of the features can be determined using the AHP. As a first step, we analyze each normalized sentence feature in a hierarchal structure, as shown in Figure 1. Next, we ask 3 linguistics experts to construct pairwise comparison matrices that indicate how many times more important one feature is with respect to another. A pairwise comparison matrix, $P = [p_{ij}]_{k \times k}$, with k being the number of features that will be compared, has the characteristic that $p_{ij} = 1/p_{ji}$, $p_{ii} = 1$. The experts specify the matrices using the graphical comparison module embedded in the Expert Choice software [41]. This program module enables the assignment of noninteger comparison values to the document features. To find the average pairwise comparison values for 3 separated comparison matrices, geometric mean is used.

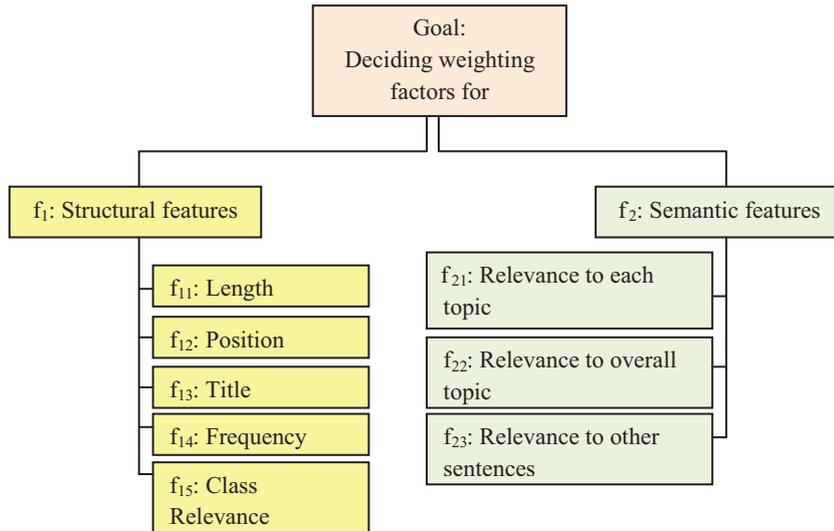


Figure 1. Hierarchal structure.

According to Figure 1, there will be 3 pairwise comparison matrices in all: the 1st is the main feature matrix with respect to the goal (between f_1 and f_2), which is shown in Table 1; the 2nd is the subcriteria matrix under f_1 (f_{11} , f_{12} , f_{13} , f_{14} and f_{15}) that is given in Table 2; and the 3rd is the subcriteria matrix under f_2 (f_{21} , f_{22} and f_{23}) that is given in Table 3.

Table 1. Average pairwise comparison matrix of the main features with respect to the goal.

Main features	f_1	f_2
f_1	1	2
f_2	1/2	1

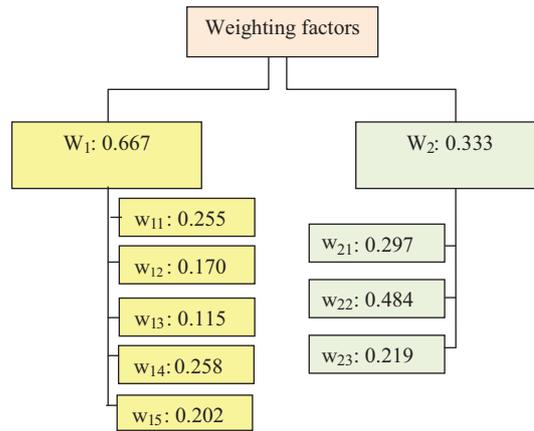
Table 2. Average pairwise comparison matrix for the features under f_1 .

f_1	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
f_{11}	1	1.6	2.4	1/1.2	1.3
f_{12}	1/1.6	1	1.6	1/1.4	1/1.3
f_{13}	1/2.4	1/1.6	1	1/2	1/1.7
f_{14}	1.2	1.4	2	1	1.3
f_{15}	1/1.3	1.3	1.7	3	1

Table 3. Average pairwise comparison matrix for the features under f_2 .

f_2	f_{21}	f_{22}	f_{23}
f_{21}	1	1/1.8	1.5
f_{22}	1.8	1	2
f_{23}	1/1.5	1/2	1

Using these comparison matrices, the AHP propagates the importance values of each node from the topmost features towards the subfeatures. Hence, each weighting score for the normalized sentence features is calculated. The results are represented in Figure 2.

**Figure 2.** AHP based weighting factors of each feature.

All of the weighting factors in Figure 3 satisfy the condition shown in Eq. (7):

$$\sum_{j=1}^5 W_1 w_{1j} + \sum_{j=1}^3 W_2 w_{2j} = 1. \quad (7)$$

3.2. Finding the optimal weight of the features with the ABC algorithm

Swarm intelligence is a research branch that models the population of the interacting agents or swarms that are able to self-organize. An ant colony, a flock of birds, or an immune system is a typical example of a swarm system. The ABC algorithm is another example of swarm intelligence. It was proposed in [26] and has been applied in the area of numerical optimization problems [42,43]. More recently, this approach has found its way into the domain of classification [45] and clustering [45,46] studies, where the experimental results show that the use of the ABC algorithm can successfully be applied in these areas. In this section, we consider the automatic determination of the optimal weights of the sentence features using the ABC algorithm.

In ABC algorithm, there are 3 kinds of bees, namely employed bees, onlooker bees, and scout bees. The employed bees go to their food source and come back to their dance area. The onlooker bees watch the dances of the employed bees and choose food depending on the dances. The scout bees search the solution space randomly and find new food sources. The position of a food source represents a possible solution corresponding to the weights of the sentence features w_{ij} . Each solution is a vector of the feature weights. The vector has length of 32 bits, since there are 8 features and each feature vale (between 0 and 15) can be represented by 4 bits. The

nectar amount of the food source corresponds to the quality of the fitness function calculated by:

$$f_i = \text{average} \left| \frac{S \cap T}{S} \right|, \quad (8)$$

where T is the manual summary and S is the machine generated summary.

Figure 3 shows the pseudo-code of the ABC algorithm. As can be seen, the ABC algorithm starts the process by generating a randomly distributed initial population of SN solutions, where SN denotes the size of the population (food source position).

```

1: Load training documents
2: Generate SN initial population
3: Evaluate the fitness ( $f_i$ ) of the population
4: set cycle to 1
5: repeat
6: FOR each employed bee{
    Produce new solution
    Calculate the value  $f_i$ 
}
7: Calculate the probability values  $p_i$  for the solutions
   By Eq. (9)
8: FOR each onlooker bee{
    Select a solution depending on  $p_{ij}$ 
    Produce new solution
    Calculate the value  $f_i$ 
}
9: If there is an abandoned solution for the scout
   then replace it with a new solution
   which will be randomly produced .
10: Memorize the best solution so far
11: cycle = cycle+1
12: until cycle = MCN

```

Figure 3. Pseudo-code of the ABC algorithm.

After the initialization, an employed bee changes the food source by testing the nectar amount (fitness value) of the new source. If the nectar amount of the new source is more than that of the old one, the bee learns the new position and forgets the old one. After all of the employed bees finish the search process, they share the nectar information and their position with the onlooker bees. The onlooker bees choose food sources with probabilities related to their nectar amount. They produce a modification on the position of the food sources and check the nectar amount of the new sources. If the nectar amounts of the new sources are more than those of the old ones, the onlooker bees memorize the new positions and forget the old ones. The onlooker bees choose new food sources depending on the probability value associated with the food source:

$$p_i = \frac{f_i}{\sum_{n=1}^{SN} f_n}. \quad (9)$$

The food source that does not progress for a certain number of cycles is abandoned. This cycle number is called the “limit”. In this case, the food source of which the nectar is abandoned by the bees is replaced with a new food source by the scout bees.

The control parameters of the ABC algorithm are the number of the SN, the value of the limit, and the maximum cycle number (MCN). In this work, we select a SN of 20, MCN of 1000, and a limit value of 100.

In order to find the optimal weights, we separate the corpus into a training set consisting of 88 documents and a test set consisting of 22 documents. We perform a 5-fold cross validation. Table 4 shows the optimal weights of each feature that allows us to reach the highest average of the precisions (Eq. (8)) calculated during the training.

Table 4. Optimal weights of each feature obtained by the ABC algorithm.

Features	Weights
w ₁₁	2
w ₁₂	2
w ₁₃	14
w ₁₄	1
w ₁₅	1
w ₂₁	12
w ₂₂	10
w ₂₃	11

4. Data corpus and the evaluation data set

We construct a data corpus that contains 110 documents collected from online Turkish newspapers. Table 5 shows the attributes of the data corpus.

Table 5. Attributes of the data corpus.

Attributes of the data corpus	Values
Number of docs	110
Total number of sentences	2487
Min sentences/doc	9
Max sentences/doc	63

To evaluate the performance of our system, 3 independent assessors are employed to conduct a manual summarization of the 110 documents. For each document, each assessor is requested to select sentences without a compression ratio for the size of the final summary. Hence, we are able to get a summary size that the assessors think is adequate. Table 6 shows the attributes of the manual summarization results and the compression ratios for each assessor.

Table 6. Attributes of the manual summarization data set and the compression rates of the assessors.

Attributes of the manual summarization data set	Values	Compression rates
Number of sentences selected by Assessor1:	775	33%
Number of sentences selected by Assessor2:	848	36%
Number of sentences selected by Assessor3:	729	31%

The average compression ratio of the 3 assessors is 33%. We also analyze the degree of disagreement among the 3 assessors. Table 7 shows the number of sentences that are selected by 1, 2, and 3 assessors.

Table 7. Number of common sentences that are selected by the assessors.

Summarization attributes	Values
Sentences selected by 1 person:	786
Sentences selected by 2 people:	451
Sentences selected by 3 people:	169

As evidenced by these results, only 169 sentences are selected by all 3 assessors. The disagreement among the 3 assessors is more than we expected.

5. Experimental results

Performance analysis is conducted on each individual sentence feature and the proposed hybrid system using the prepared Turkish data set. While analyzing the performance, the manually generated summaries are compared with the automatically generated summaries. The precision (P), recall (R), and F-measure (F) metrics that enable the evaluation of the sentence coverage among the manually and automatically generated summaries are chosen for the evaluation results. Assuming that T is the manual summary and S is the automatically generated summary, the measurements P, R, and F are defined as follows:

$$P = \frac{|S \cap T|}{|S|} \quad R = \frac{|S \cap T|}{|T|} \quad F = \frac{2PRF}{R + P}. \quad (10)$$

We supplemented the above metrics with the ROUGE evaluation toolkit that is based on the N-gram co-occurrence between the manually generated and automatically generated summaries. Suppose that a number of assessors, created manually, generated a summary set (MSS). The ROUGE-N score of a summary is calculated as follows:

$$ROUGE - N = \frac{\sum_{S \in MSS} \sum_{gram_N \in S} Count_{match}(gram_N)}{\sum_{S \in MSS} \sum_{gram_N \in S} Count(gram_N)}, \quad (11)$$

where $Count_{match}(gram_N)$ is the maximum number of N-grams occurring both in the automatic summary and in the human-generated summary, and $Count(gram_N)$ is the number of N-grams in the human-generated summary [27].

Tables 8 and 9 depict the effects of the use of the Turkish Wikipedia on the 3 semantic features (f_{21} , f_{22} , f_{23}), where the performance results with the use of the Turkish Wikipedia are encouraging. For the most part, for all 3 assessors, the use of the Turkish Wikipedia leads to an increase in the performance over the performance without its use.

Table 8. F-measure values that emphasize the effects of the use of the Turkish Wikipedia for f_{21} , f_{22} , f_{23} .

	Assessor1		Assessor2		Assessor3	
	Without Wikipedia	Use of Wikipedia	Without Wikipedia	Use of Wikipedia	Without Wikipedia	Use of Wikipedia
f_{21}	0.487	0.502	0.440	0.440	0.461	0.460
f_{22}	0.511	0.518	0.458	0.459	0.53	0.532
f_{23}	0.399	0.415	0.382	0.39	0.362	0.371

Table 9. ROUGE-1 F-measure values that emphasize the effects of the use of the Turkish Wikipedia for f_{21} , f_{22} , f_{23} .

	Assessor1		Assessor2		Assessor3	
	Without Wikipedia	Use of Wikipedia	Without Wikipedia	Use of Wikipedia	Without Wikipedia	Use of Wikipedia
f_{21}	0.674	0.680	0.636	0.632	0.658	0.660
f_{22}	0.688	0.691	0.643	0.645	0.700	0.702
f_{23}	0.596	0.602	0.569	0.575	0.560	0.564

Although the performance improvements are not very significant, which is usually the case in most of the research in this area, the obtained improvements encourage us to perform further research on the use of external resources like the Turkish Wikipedia we used to improve the text mining methods.

Tables 10 and 11 measure the effects of each feature and the proposed hybrid system on the basis of each assessor.

Table 10. F-measure values of each feature and the proposed hybrid system on the basis of each assessor.

Sentence features	Assessor1	Assessor2	Assessor3
f_{11}	0.520	0.444	0.498
f_{12}	0.407	0.416	0.372
f_{13}	0.259	0.228	0.241
f_{14}	0.522	0.470	0.513
f_{15}	0.487	0.440	0.453
f_{21}	0.502	0.440	0.460
f_{22}	0.518	0.459	0.532
f_{23}	0.415	0.39	0.371
AHP-based hybrid system	0.554	0.488	0.546
ABC-based hybrid system	0.580	0.491	0.570

Table 11. ROUGE-1 F-measure values of each feature and the proposed hybrid system on the basis of each assessor.

Sentence features	Assessor1	Assessor2	Assessor3
f_{11}	0.690	0.644	0.670
f_{12}	0.584	0.586	0.556
f_{13}	0.462	0.421	0.448
f_{14}	0.695	0.641	0.688
f_{15}	0.666	0.631	0.644
f_{21}	0.680	0.632	0.660
f_{22}	0.691	0.645	0.702
f_{23}	0.602	0.575	0.564
AHP-based hybrid system	0.710	0.666	0.707
ABC-based hybrid system	0.726	0.670	0.715

Considering the ordering of the features with respect to their performances, one can say that the assessors mostly extract: 1) sentences that contain words that frequently occur in the document (f_{14}), 2) sentences that include all of the topics of the document (f_{22}), and 3) sentences that are relatively long (f_{11}). It can also be seen that the proposed structural feature (f_{15}), which uses text categorization, outperforms the structural features like the position (f_{12}), title (f_{13}), and sentence similarity count (f_{23}) for the 3 assessors and it may be considered an acceptable feature for the text summarization task.

The last 2 rows of Tables 10 and 11 show the results that exploit the effects of all of the features by combining them with the hybrid system. The results show that exploiting all of the features by combining them resulted in a better performance than exploiting each feature individually.

The AHP- and ABC-based hybrid systems assign a weight to each of the features. The AHP generates the general feature weights depending on the expert judgment. This is a manual process and must be repeated for each different set of the features by the experts. The weights are computed automatically by the ABC method. In addition, as can be seen from Tables 10 and 11, the ABC produces the best performance results.

6. Conclusion

This paper proposed a novel hybrid Turkish text summarization system that combines the structural and semantic sentence features to yield better summarization results. A new structural feature facilitated by the document class identification and a slight enhancement of the LSA-based semantic document features by employing the Turkish Wikipedia as tool for extracting semantically related consecutive words are further contributions of the current work. For combining the features' scores into an overall score function, 2 approaches based on the AHP and the ABC algorithm have been explored. To carry out the experiments for comparing the proposed system against systems employing only structural or only semantic features, a new Turkish corpus has been constructed. The algorithmic results, which have been presented and discussed in detail, confirm the significance of our contributions.

Acknowledgments

This work is supported by the Yıldız Technical University Scientific Research Project Commission: "2012-07-03-DOP01".

References

- [1] D.R. Radev, K. McKeown, "Generating natural language summaries from multiple on-line sources", *Computational Linguistics*, Vol. 24, pp. 469–500, 1998.
- [2] S.H. Sanda, F. Lacatusu, "Generating single and multi-document summaries with gistexter", *Document Understanding Conference*, pp. 30–38, 2002.
- [3] H. Saggion, G. Lapalme, "Generating indicative-informative summaries with Su-muM", *Computational Linguistics*, Vol. 28, pp. 497–526, 2002.
- [4] H. Jing, K.R. McKeown, "Cut and paste based text summarization", *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, pp. 178–185, 2000.
- [5] H.P. Luhn, "The automatic creation of literature abstracts", *IBM Journal of Research Development*, Vol. 2, pp. 159–165, 1958.
- [6] H.P. Edmundson, "New methods in automatic extracting", *Journal of the Association for Computing Machinery*, Vol. 16, pp. 264–285, 1969.
- [7] K. Wong, M. Wu, W. Li, "Extractive summarization using supervised and semi-supervised learning", *Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, pp. 985–992, 2008.
- [8] Y. Gong, X. Liu, "Generic text summarization using relevance measure and latent semantic analysis", *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 19–25, 2001.
- [9] J. Steinberger, "Text summarization within the LSA framework", PhD Thesis, University of West Bohemia, Czech Republic, 2007.
- [10] J.Y. Yeh, H.R. Ke, W.P. Yang, I.H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis", *Journal of Information Processing and Management*, Vol. 41, pp. 75–95, 2005.
- [11] L. Hennig, "Topic-based multi-document summarization with probabilistic latent semantic analysis", *International Conference on Recent Advances in Natural Language Processing*, pp. 144–149, 2009.
- [12] J. Lee, S. Park, C. Ahn, D. Kim, "Automatic generic document summarization based on non-negative matrix factorization", *Information Processing and Management*, Vol. 45, pp. 20–34, 2009.
- [13] J. Kupiec, O.P. Jan, C. Francine, "A trainable document summarizer", *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 68–73, 1995.

- [14] S.H. Teufel, M. Moens, “Sentence extraction as a classification task”, *ACL/EACL Workshop on Intelligent Scalable Text Summarization*, pp. 58–65, 1997.
- [15] E. Filatova, V. Hatzivassiloglou, “A formal model for information selection in multi-sentence text extraction”, *Proceedings of the 20th International Conference on Computational Linguistics*, pp. 397–403, 2004.
- [16] R. McDonald, “A study of global inference algorithms in multi-document summarization”, *29th European Conference on IR Research*, pp. 557–564, 2007.
- [17] R.M. Alguliev, R.M. Aliguliyev, M.S. Hajirahimove, C.A. Mehdiyev, “MCMR: Maximum coverage and minimum redundant text summarization model”, *Expert Systems with Applications*, Vol. 38, pp. 14514–14522, 2011.
- [18] Z. Altan, “A Turkish automatic text summarization system, IASTED International Conference on Artificial Intelligence and Applications, 2004.
- [19] E. Uzundere, E. Dedja, B. Diri, M.F. Amasyalı, “Automatic text summarization for Turkish texts”, *National Conference of the ASYU*, 2008.
- [20] C. Pembe, “Automated query-biased and structure-preserving document summarization for web search tasks”, PhD Thesis, Boğaziçi University, Turkey, 2011.
- [21] C. Cığır, M. Kutlu, I. Cicekli, “Generic text summarization for Turkish”, *The Computer Journal*, Vol. 53, pp. 1315–1323, 2010.
- [22] A. Güran, E. Bekar, S. Akyokuş, “A comparison of feature and semantic-based summarization algorithms for Turkish”, *International Symposium on Innovations in Intelligent Systems and Applications*, 2010.
- [23] M. Özsoy, İ. Çiçekli, F.N. Alpaslan, “Text summarization of Turkish texts using latent semantic analysis”, *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 869–876, 2010.
- [24] A. Güran, N. Güler Bayazıt, E. Bekar, “Automatic summarization of Turkish documents using non-negative matrix factorization”, *International Symposium on Innovations in Intelligent Systems and Applications*, pp. 480–484, 2011.
- [25] T.L. Saaty, *The Analytic Hierarchy Process*, New York, McGraw-Hill, 1980.
- [26] D. Karaboga, B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”, *Journal of Global Optimization*, Vol. 39, pp. 459–171, 2007.
- [27] C.Y. Lin, E. Hovy, “Automatic evaluation of summaries using N-gram co-occurrence statistics”, *Language Technology Conference*, Vol. 1, pp. 71–78, 2003.
- [28] Zemberek- Zemberek 2 is an open source NLP library for Turkic languages 2011–2012, available at: <http://code.google.com/p/zemberek/downloads/list>.
- [29] M.F. Amasyalı, A. Beken, “Türkçe kelimelerin anlamsal benzerliklerinin ölçülmesi ve metin sınıflandırmada kullanılması”, *National Conference of SIU*, 2009.
- [30] E. Gabrilovich, S. Markovich, “Computing semantic relatedness using Wikipedia-based explicit semantic analysis”, *20th International Joint Conference on Artificial Intelligence*, pp. 1606–1611, 2007.
- [31] E. Gabrilovich, S. Markovitch, “Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge”, *21st National Conference on Artificial Intelligence*, Vol. 2, pp. 1301–1306, 2006.
- [32] K. Ramanathan, Y. Sankarasubramaniam, N. Mathur, A. Gupta, “Document summarization using Wikipedia”, *1st International Conference on Human-Computer Interaction*, pp. 254–260, 2009.
- [33] G. Williams, “In search of representativity in specialised corpora: Categorisation through collocation”, *International Journal of Corpus Linguistics*, Vol. 7, pp. 43–64, 2002.
- [34] O. Ferret, “Using collocations for topic segmentation and link detection”, *19th International Conference on Computational Linguistics*, Vol. 1, pp. 260–266, 2002.
- [35] O. Sunercan, A. Birturk, “Wikipedia missing link discovery: a comparative study”, *AAAI Spring Symposium on Linked Data Meets Artificial Intelligence*, 2010.

- [36] C. Calli, “Improving search result clustering by integrating semantic information from Wikipedia”, MS Thesis, Middle East Technical University, Department of Computer Engineering, 2010.
- [37] A. Boynuegri, “Cross-lingual information retrieval on Turkish and English texts”, MS Thesis, Middle East Technical University, Department of Computer Engineering, 2010.
- [38] I.V. Mashechkin, M.I. Petrovskiy, D.S. Popov, D.V. Tsarev, “Automatic text summarization using latent semantic analysis”, *Programming and Computer Software*, Vol. 37, pp. 299–305, 2011.
- [39] Standard Score from Wikipedia, the free encyclopedia 2001–2012, available at: http://en.wikipedia.org/wiki/Standard_score.
- [40] L. Felföldi, A. Kocsor, “AHP-based classifier combination”, *Proceedings of the 4th International Workshop on Pattern Recognition in Information Systems*, pp. 45–58, 2004.
- [41] Collaboration and Decision, Support Software for Groups and Organizations 2011–2012, available at: <http://www.expertchoice.com/>
- [42] R. Srinivasa Rao, S.V.L. Narasimham, M. Ramalingaraju, “Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm”, *International Journal of Electrical Power and Energy Systems Engineering*, Vol. 1 , pp. 116–122, 2008.
- [43] F. Kang, J. Li, Q. Xu, “Structural inverse analysis by hybrid simplex artificial bee colony algorithms”, *Computers and Structures*, Vol. 87, pp. 861–870, 2009.
- [44] S.N. Omkar, J. Senthilnath, “Artificial bee colony for classification of acoustic emission signal”, *International Journal of Aerospace Innovations*, Vol. 1, pp. 129–143, 2009.
- [45] D. Karaboga, C. Ozturk, “A novel clustering approach: artificial bee colony (ABC) algorithm”, *Applied Soft Computing*, Vol. 11, pp. 652–657, 2011.
- [46] D. Karaboga, C. Ozturk, “Fuzzy clustering with artificial bee colony algorithm, *Scientific Research and Essays*”, Vol. 5, pp. 1899–1902, 2010.