Analyzing Stemming Approaches for Turkish Multi-Document Summarization

Muhammed Yavuz Nuzumlalı **Arzucan Özgür**
Department of Computer Engineering
Boğaziçi University
TR-34342, Bebek, İstanbul, Turkey
{yavuz.nuzumlali,arzucan.ozgur}@boun.edu.tr

Abstract

In this study, we analyzed the effects of applying different levels of stemming approaches such as fixed-length word truncation and morphological analysis for multi-document summarization (MDS) on Turkish, which is an agglutinative and morphologically rich language. We constructed a manually annotated MDS data set, and to our best knowledge, reported the first results on Turkish MDS. Our results show that a simple fixed-length word truncation approach performs slightly better than no stemming, whereas applying complex morphological analysis does not improve Turkish MDS.

1 Introduction

Automatic text summarization has gained more importance with the enormous growth and easy availability of the Internet. It is now possible to reach extensive and continuously growing amount of resources. However, this situation brings its own challenges such as finding the relevant documents, and absorbing a large quantity of relevant information (Gupta and Lehal, 2010). The goal of multi-document summarization (MDS) is to automatically create a summary of a set of documents about the same topic without losing the important information. Several approaches for MDS have been proposed in the last decade. However, most of them have only been applied to a relatively small set of languages, mostly English, and recently also to languages like Chinese, Romanian, Arabic, and Spanish (Giannakopoulos, 2013).

Previous studies have shown that methods proposed for languages like English do not generally work well for morphologically rich languages like Finnish, Turkish, and Czech, and additional methods considering the morphological structures of these languages are needed (Eryiğit et al., 2008). For instance, Turkish is an agglutinative language where root words can take many derivational and inflectional affixes. This feature results in a very high number of different word surface forms, and eventually leads to the data sparseness problem. Hakkani-Tür et al. (2000) analyzed the number of unique terms for Turkish and English and showed that the term count for Turkish is three times more than English for a corpus of 1M words.

There are only a few studies for text summarization on Turkish, all of which are about single-document summarization (Altan, 2004; Çığır et al., 2009; Özsoy et al., 2010; Güran et al., 2010; Güran et al., 2011). Some of these studies applied morphological analysis methods, but none of them analyzed their effects in detail.

To our best knowledge, this paper reports the first multi-document summarization study for Turkish. We used LexRank as the main summarization algorithm (Erkan and Radev, 2004), applied and analyzed different levels of stemming methods such as complex morphological analysis and fixed-length word truncation. We also created the first manually annotated MDS data set for Turkish, which has been made publicly available for future studies.

The rest of the paper is organized as follows. Section 2 presents the related work on MDS, as well as on the applications of morphological analysis on Turkish for different Natural Language Processing (NLP) and Information Retrieval (IR) problems. In Section 3, we provide a very brief introduction to the Turkish morphology and present the stemming methods that we evaluated. The details about the created data set and our experimental setup are presented in Section 4. We present and discuss the results in Section 5, and conclude in Section 6.

2 Related Work

A large number of methods have been proposed for multi-document summarization in the last 10-15 years (e.g. (Erkan and Radev, 2004; Shen and Li, 2010; Christensen et al., 2013)). While most of these approaches have only been applied to English, summarization data sets and systems for other languages like Chinese, Romanian, and Arabic have also been proposed in the recent years (Giannakopoulos, 2013).

Previous studies on automatic summarization for Turkish only tackled the problem of single-document summarization (SDS). Altan (2004) and Çığır et al. (2009) proposed feature-based approaches for Turkish SDS, whereas Özsoy et al. (2010) and Güran et al. (2010) used Latent Semantic Analysis (LSA) based methods. Güran et al. (2011) applied non-negative ma-

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 702–706,
October 25-29, 2014, Doha, Qatar.

Word	Analysis
gören *(the one who sees)*	gör+en(DB)
görülen *(the one which is seen)*	gör+ül(DB)+en(DB)
görüş *(opinion)*	gör+üş(DB)
görüşün *(your opinion)*	gör+üş(DB)+ün
görüşler *(opinions)*	gör+üş(DB)+ler
görüşme *(negotiation)*	gör+üş(DB)+me(DB)
görüşmelerin *(of negotiations)*	gör+üş(DB)+me(DB)+ler+in

Table 1: Different word forms and their morphological analysis for the stem "gör" (to see). The derivational boundaries are marked with (DB).

trix factorization (NMF) and used consecutive words detection as a preprocessing step.

The effect of morphological analysis for Turkish was analyzed in detail for Information Retrieval (Can et al., 2008) and Text Categorization (Akkuş and Çakıcı, 2013). Can et al. (2008) showed that using fixed-length truncation methods perform similarly to lemmatization-based stemming for information retrieval. Akkuş and Çakıcı (2013) obtained better results for text categorization with fixed-length word truncation rather than complex morphological analysis, but the difference was not significant. For other morphologically rich languages, there is a case study on Greek by Galiotou et al. (2013). They applied different stemming algorithms and showed that stemming on Greek texts improves the summarization performance.

3 Methodology

This section contains detailed information about the application of different levels of morphological features during the summarization process. Before diving into the details, we provide a very brief description of the morphological structure of the Turkish language.

3.1 Turkish Morphology

Turkish is an agglutinative language with a productive morphology. Root words can take one or more derivational and inflectional affixes; therefore, a root can be seen in a large number of different word forms. Another issue is the morphological ambiguity, where a word can have more than one morphological parse.

Table 1 shows an example list of different word forms for the stem "gör" (to see). All the words in the table have the same root, but the different suffixes lead to different surface forms which may have similar or different meanings. When the surface forms of these words are used in a summarization system, they will be regarded as totally different words. However, if a morphological analysis method is applied to the sentences before giving them to the summarization system, words with similar meanings can match during the sentence similarity calculations.

3.2 Stemming Policies

In this section, we explain the different stemming methods that we investigated.

Raw: In this method, we take the surface forms of words, without applying any stemming.

Root: This method takes the most simple unit of a word, namely the root form. For example, in Table 1, the words "gören", "görüşün", and "görüşmelerin" have the same root (gör), so they will match during sentence similarity calculations.

Deriv: Using the Root method may oversimplify words because some words that are derived from the same root may have irrelevant meanings. In the above example, "görüşler" and "gören" have different meanings, but they have the same root (gör). In order to solve this oversimplification issue, we propose to preserve derivational affixes, and only remove the inflectional affixes from the words. In this method, "görüşler" and "gören" will not match because when we remove only the inflectional affixes, they become "görüş" and "gören". On the other hand, the words "görüşler" and "görüşün" will match because their Deriv forms are the same, which is "görüş".

Prefix: In Turkish, affixes almost always occur as suffixes, not prefixes. Additionally, applying morphological analysis methods is a time consuming process, and may become an overhead for online applications. Therefore, a fixed-length simplification method is also tried, since it is both a fast method and can help match similar words by taking the first n characters of words which have lengths larger than n.

As the summarization algorithm, we used LexRank (Erkan and Radev, 2004), which is a salient graph-based method that achieves promising results for MDS. In LexRank, first a sentence connectivity graph is constructed based on the cosine similarities between sentences, and then the PageRank (Page et al., 1999) algorithm is used to find the most important sentences.

4 Experimental Setup

4.1 Data Set

One of the greatest challenges for MDS studies in Turkish is that there does not exist a manually annotated data set. In this study, we have collected and manually

annotated a Turkish MDS data set, which is publicly available for future studies[1].

In order to match the standards for MDS data sets, we tried to follow the specifications of the DUC 2004 data set. Our data set consists of 21 clusters, each consisting of around 10 documents. We selected 21 different topics from different domains (e.g., politics, economics, sports, social, daily, and technology), and selected 10 documents on average for each topic. The documents were obtained from the websites of various news sources. The average number of words per document is 337, and the average number of letters in a word is 6.84.

For manual annotation, we divided the 21 clusters into three groups and sent them to three annotators different from the authors. We required the human summaries not to exceed 120 words for the summary of each cluster.

4.2 Tools

4.2.1 Turkish Morphological Analysis

In order to perform different levels of morphological analysis on documents, we used a two-level morphological analyzer (Oflazer, 1994) and a perceptron-based morphological disambiguator (Sak et al., 2007), which is trained with a corpus of about $750,000$ tokens from news articles. The accuracy of the disambiguator has been reported as 96% (Sak et al., 2007). The Root and Deriv forms of words were generated from the disambiguator output.

4.2.2 MEAD Summarization Toolkit

We used MEAD (Radev et al., 2004), which is an open-source toolkit created for extractive MDS, in our experiments. MEAD handles all the necessary processes to generate a summary document (e.g., sentence ranking, selection, re-ordering, and etc.).

We used the LexRank implementation that comes with MEAD as a feature, together with the Centroid and Position features (each feature is equally weighted). We forced the generated summaries not to exceed 120 words. However, we define the following exception in order to preserve the readability and the grammaticality of the generated summary. For a candidate sentence S having n words, if the absolute difference between the threshold (which is 120) and the summary length including sentence S (say N_w) is less than the absolute difference between the threshold and the summary length excluding sentence S (say N_{wo}), and if N_w is less than 132 (which is $120*1.1$), we allow the summary to exceed the threshold and add sentence S as the last summary sentence.

We used term frequency (tf) based cosine similarity as the similarity measure during the sentence selection step. We also required sentence length to be between 6 and 50 words (which we found empirically) in order to increase the readability of the summaries. The reason behind applying this filtering is that very short sentences generally do not contain much information to become a summary sentence, whereas very long sentences decrease the readability and fill a significant percentage of the summary limit.

4.2.3 ROUGE

For evaluation, we used ROUGE, which is a standard metric for automated evaluation of summaries based on n-gram co-occurrence. We used ROUGE-1 (based on uni-grams), ROUGE-2 (based on bi-grams), and ROUGE-W (based on longest common sub-sequence weighted by length) in our experiments. Among these, ROUGE-1 has been shown to agree with human judges the most (Lin and Hovy, 2003), so we give importance to it while interpreting the results.

5 Evaluation and Results

We ran MEAD with the proposed stemming policies using different levels of cosine similarity threshold values to analyze the effect of the similarity threshold on the summarization performance. After the sentences are ranked using the LexRank method, the similarity threshold is used to decide whether to include a sentence to the summary. A sentence is not included to the summary, if its similarity to a previously picked sentence is larger than the similarity threshold.

In our preliminary experiments, we used the default similarity threshold 0.7, which was found empirically by the MEAD developers for English. However, it produced poor results on the Turkish data set.

Policy	ROUGE-1	ROUGE-2	ROUGE-W
Prefix10	**0.438**	0.194	**0.197**
Prefix12	0.433	**0.197**	0.195
Prefix9	0.432	0.194	0.194
Prefix4	0.432	0.178	0.190
Prefix7	0.431	0.189	0.190
Prefix5	0.431	0.183	0.190
Prefix6	0.430	0.185	0.189
Raw	0.428	0.189	0.191
Deriv	0.428	0.178	0.188
Prefix8	0.427	0.187	0.188
Prefix11	0.427	0.190	0.193
Root	0.420	0.186	0.185

Table 2: Best scores for different policies

Figure 1 shows the F-1 scores for the ROUGE-1 metric for policies with different thresholds. After the threshold exceeds 0.5, the performances for all policies start to decrease, so we don't report the values here to make the chart more readable. In general, Raw and Prefix10 (taking the first 10 letters of the words) achieve better performances with lower threshold values, whereas Root and Deriv operate better with relatively higher threshold values. As we stated earlier, in Turkish, words with similar meanings can occur in

[1]The data set can be retrieved from the following github repository: https://github.com/manuyavuz/TurkishMDSDataSet_alpha

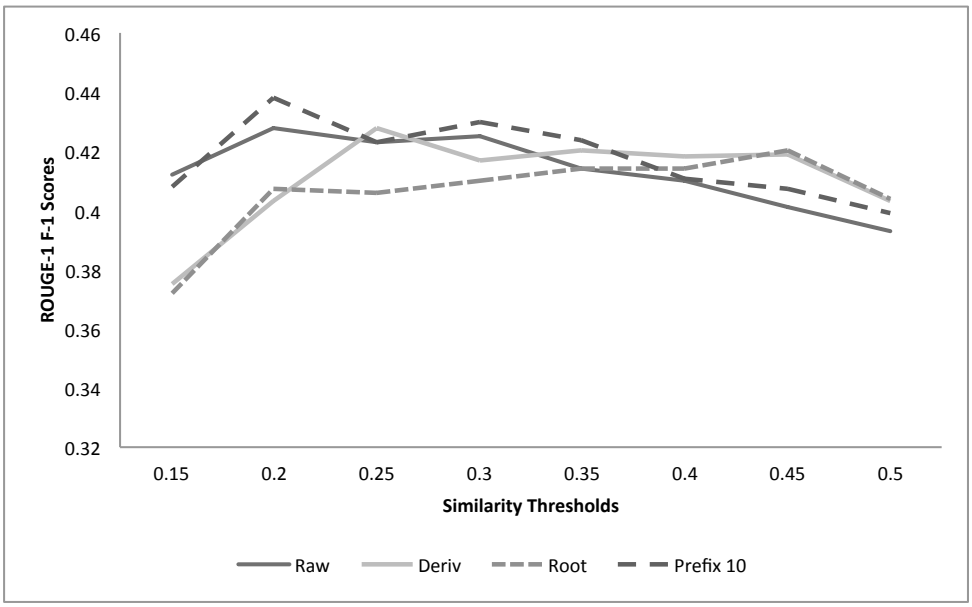

Figure 1: F-1 scores for different similarity threshold values

text with different surface forms due to their inflections. Such words can not be matched during similarity computation if morphological analysis is not performed. Therefore, using higher similarity threshold values cause very similar sentences to occur together in the summaries, and eventually, result in poor scores.

Table 2 shows the best scores obtained by each policy. The Prefix policy generally outperforms the Raw policy. The Prefix10 policy achieves the best ROUGE-1 score. On the other hand, the policies that apply complex morphological analysis (i.e. Root and Deriv) are not able to outperform the simple Prefix and Raw policies. The Deriv policy performs similarly to the Raw and Prefix policies, whereas the Root policy obtains the lowest ROUGE-1 score.

5.1 Discussion

The results show that using a simple fixed-length prefix policy outperforms all other methods, and applying complex morphological analysis does not improve Turkish MDS. The poor performance of the Root policy is somewhat expected due to the fact that, if we preserve only the roots of the words, we lose the semantic differences among the surface forms provided by the derivational affixes. On the other hand, the reason behind the observation that Deriv and Raw obtain similar performances is not obvious.

In order to further analyze this observation, we used an entropy based measure, which is calculated as shown below, to quantify the homogeneity of the clusters in the data set in terms of the variety of the surface forms corresponding to the Deriv forms of each word in the cluster. We first compute the entropy for each Deriv form in a cluster. The entropy of a Deriv form is lower, if it occurs with fewer different surface forms in the cluster. The entropy of a cluster is computed by summing the entropies of the Deriv forms in the cluster and dividing the sum by the number of words in the cluster (i.e. N).

$$D_{Deriv_i} = \{t \mid t\ inflected\ from\ Deriv\ i\}$$

$$H(Deriv_i) = \sum_{t \in D_{Deriv_i}} p(t) \log p(t)$$

$$H(C) = \sum_i \frac{H(Deriv_i)}{N}$$

To compare with the data set clusters, we generated random document clusters by randomly selecting 10 different clusters and then randomly selecting one document from each selected cluster. The average entropy value for the data set clusters and the random clusters were 4.99 and 7.58, respectively. Due to this significant difference, we can hypothesize that the documents about the same topic show a more homogeneous structure. In other words, a Deriv form is usually seen in the same surface form in a cluster of documents which are about the same topic. Therefore, the Deriv policy and the Raw policy achieve similar results for summarizing documents about the same topic.

During evaluation, we ran ROUGE with the Deriv versions of the human summaries and the system summaries in order to match semantically similar words having different surface forms. We also experimented with ROUGE using the Raw versions, but the results followed very similar patterns, so those results were not reported.

6 Conclusion

In this paper, we reported the first steps for a multi-document summarization system for Turkish. A manually annotated data set has been constructed from news

articles, and made publicly available for future studies. We utilized the LexRank summarization algorithm, and analyzed the effects of different stemming policies for Turkish MDS. Our results show that simple fixed-length truncation methods with high limits (such as taking the first 10 letters) improves summarization scores. In contrast to our expectation, using morphological analysis does not enhance Turkish MDS, possibly due to the homogeneousness of the documents in a cluster to be summarized. As future work, we plan to extend the data set with more clusters and more reference summaries, as well as to develop sentence compression methods for Turkish MDS.

Acknowledgments

We would like to thank Ferhat Aydın for his contributions during the data set corpus collection and annotation process. We would also like to thank Burak Sivrikaya and Serkan Bugur for their help in generating the human summaries for the data set.

References

Burak Kerim Akkuş and Ruket Çakıcı. 2013. Categorization of turkish news documents with morphological analysis. In *ACL (Student Research Workshop)*, pages 1–8. The Association for Computer Linguistics.

Zeynep Altan. 2004. A turkish automatic text summarization system. In *Proceedings of the IASTED International Conference Artificial Intelligence and Applications*, pages 74–83.

Fazlı Can, Seyit Koçberber, Erman Balçık, Cihan Kaynak, H Cağdaş Öcalan, and Onur M Vursavaş. 2008. Information retrieval on turkish texts. *Journal of the American Society for Information Science and Technology*, 59(3):407–421.

Janara Christensen, Stephen Soderland Mausam, and Oren Etzioni. 2013. Towards coherent multi-document summarization. In *Proceedings of NAACL-HLT*, pages 1163–1173.

Güneş Erkan and Dragomir R. Radev. 2004. Lexpagerank: Prestige in multi-document text summarization. In *EMNLP*, pages 365–371. ACL.

Gülşen Eryiğit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency parsing of turkish. *Computational Linguistics*, 34(3):357–389.

Eleni Galiotou, Nikitas Karanikolas, and Christodoulos Tsoulloftas. 2013. On the effect of stemming algorithms on extractive summarization: a case study. In Panayiotis H. Ketikidis, Konstantinos G. Margaritis, Ioannis P. Vlahavas, Alexander Chatzigeorgiou, George Eleftherakis, and Ioannis Stamelos, editors, *Panhellenic Conference on Informatics*, pages 300–304. ACM.

George Giannakopoulos. 2013. Multi-document multilingual summarization and evaluation tracks in acl 2013 multiling workshop. *MultiLing 2013*, page 20.

Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3).

Aysun Güran, Eren Bekar, and S Akyokuş. 2010. A comparison of feature and semantic-based summarization algorithms for turkish. In *International Symposium on Innovations in Intelligent Systems and Applications*. Citeseer.

A Güran, NG Bayazıt, and E Bekar. 2011. Automatic summarization of turkish documents using non-negative matrix factorization. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on*, pages 480–484. IEEE.

Dilek Z. Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2000. Statistical morphological disambiguation for agglutinative languages. In *COLING*, pages 285–291. Morgan Kaufmann.

Chin-Yew Lin and Eduard H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *HLT-NAACL*.

Kemal Oflazer. 1994. Two-level description of turkish morphology. *Literary and linguistic computing*, 9(2):137–148.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Stanford InfoLab.

Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, et al. 2004. Mead-a platform for multidocument multilingual text summarization. Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004).

Haşim Sak, Tunga Güngör, and Murat Saraçlar. 2007. Morphological disambiguation of turkish text with perceptron algorithm. In Alexander F. Gelbukh, editor, *CICLing*, volume 4394 of *Lecture Notes in Computer Science*, pages 107–118. Springer.

Chao Shen and Tao Li. 2010. Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 984–992. Association for Computational Linguistics.

Celal Çığır, Mücahid Kutlu, and İlyas Çiçekli. 2009. Generic text summarization for turkish. In *ISCIS*, pages 224–229. IEEE.

Makbule Gülçin Özsoy, İlyas Çiçekli, and Ferda Nur Alpaslan. 2010. Text summarization of turkish texts using latent semantic analysis. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING*, pages 869–876. Tsinghua University Press.