Automatic Summarization of Turkish Documents Using Non-negative Matrix Factorization

Aysun Güran
Yıldız Technical University
Istanbul, Turkey
aysunguran@gmail.com

Nilgün Güler Bayazıt
Yıldız Technical University
Istanbul, Turkey
guler@yildiz.edu.tr

Eren Bekar
Yıldız Technical University
Istanbul, Turkey
erenbekar@gmail.com

***Abstract*—Automatic document summarization is a process, where a computer summarizes a document. This paper presents the performance analysis of an automatic Turkish document summarization system that applies Non-negative matrix factorization based summarization algorithm with different preprocessing methods. The preprocessing method called "Consecutive Words Detection" is an innovative approach that uses Turkish Wikipedia links to represent related consecutive words as a single term and the result of the evaluation process is promising for document summarization in Turkish.**

Keywords-component; Turkish document summarization; Turkish wikipedia; Non-negative Matrix factorization

I. INTRODUCTION

Automatic document summarization (ADS) is a process, where a computer summarizes a document. According to this process, a document is entered into the computer and a summarized document is returned, which is a non redundant extract from the original document. The summarized document is extremely useful in allowing users to quickly understand the main theme of the whole document and effectively save their searching time.

ADS can deal with extractive and abstractive summarization tasks. Extraction summarization techniques involve selecting the most important existing sentences, whereas abstraction summarization techniques involve generating novel sentences from given documents. The abstractive summarization approaches require a deeper understanding of documents. [1] proposes an abstractive summarization technique that uses five phases: *Sentence Reduction, Sentence Combination, Syntactic Transformation, Lexical Paraphrasing* and *Generalization* which replace phrases with general description. [2] describes a hybrid approach of applying sentence compression to extractive summaries to generate abstractive summaries. Abstractive approaches are restricted to specific domains due to the limitation in natural language processing technology and typically need to understand and then paraphrase the important concepts in documents. In contrast to the abstractive summarization approaches, extractive summarization approaches are more practical. Most of them represent documents with some features such as *term frequency, sentence position, cue words* etc., and combine these features to yield a salience function. For example, the extractive summary research done by [3] exploits word distribution of a given document based on the intuition that the most frequent words represent the most important concepts. The study in [4] is based on the cue phrase method that uses meta-linguistic markers (for example, *in conclusion, the paper describes*). It uses the location method which relies on the following intuition: *headings, text formatted in bold, sentences in the beginning* and *the end of the text* contain important information for a summary. The study given in [5] uses learning in order to combine several shallow heuristics such as *cue phrase, location, sentence length, word frequency* and *title*. The study in [6] proposes a learning-based approach to combine various sentence attributes that categorize sentences according to *surface, content, relevance* and *event* features. Extractive summarization systems do not have text generation, but the quality of extracts could be improved. Sentence reduction, anaphora resolution, information ordering and reducing repetition could improve summaries. [7] presents a summary revision system. [8] describes a text compression algorithm which uses discourse structure. [9] presents a system that paraphrases sentences.

ADS systems can also produce informative, indicative, query based and generic summaries. Informative type is like a substitution for the original document as it has all information in original source but condensed [10]. Indicative summary gives an overview of the original documents like headlines. Query-based summarization extracts a summary that is closely related to a given query. It is generated based on the needs of the users. Generic summarization finds the most important sentences from semantic content of a given document. Recently, many generic document summarization researches based on linear algebra methods, namely Latent Semantic Analysis and Non-negative Matrix factorization, have been proposed. Latent Semantic Analysis (LSA) is a semantic method for extracting semantic generalizations from textual passages on the basis of their contextual use. It is based on Singular Value Decomposition (SVD) of an m×n term-sentence matrix. SVD models the interrelationships among terms so that it can semantically cluster terms and sentences. The works of [11],[12],[13] propose the application of SVD in document summarization task. The major drawback of SVD is the possibility of negative value occurrences in the decomposed matrices. Non-negative matrix factorization (NMF) is assumed to provide better conceptual representation of data than SVD. As shown by [14], NMF decomposes a given matrix into two non-negative matrices (factors). [15] outlines that the semantic feature

978-1-61284-922-5/11/$26.00 ©2011 IEEE

vectors (factors) obtained by NMF are sparser than those obtained using SVD. The sparseness of the factors allows NMF to save considerable storage space. The NMF has its disadvantages too. There is no unique global minimum for the NMF whereas SVD has strengths concerning uniqueness and robust computation. The works of [16][17][18][19] propose the application of NMF in document summarization.

In contrast to the other languages, text summarization has not been studied much in Turkish language. Feature based Turkish document summarization approaches have been proposed in [20], [21], [22] and [23]. The studies [24], [25] apply LSA to Turkish text summarization. This study is the first study that applies NMF to the Turkish document summarization task. We collected a Turkish corpus that contains 100 documents and derived a human-generated extractive summary corpus. Then, we have focused on producing a generic and extractive document summarization system that applies the NMF based summarization algorithm due to Lee et al. [19].We also use several preprocessing methods in order to see their effects on Turkish text summarization. These methods include stop words removal, stemming and a new method called Consecutive Words Detection (CWD). The CWD method allows representing commonly occurring consecutive words in documents as a single term. For example, Mustafa Kemal Atatürk (The founder of the Turkish Republic), is considered as a single term. This provides semantic integration among consecutive words. In order to find these related consecutive words, we use Turkish Wikipedia [26].

The remaining parts of the paper are organized as follows: Section II describes the NMF based algorithm, Section III explains the performance evaluation process, Section IV presents experimental results and the final section gives concluding remarks and discusses future work.

II. NON-NEGATIVE MATRIX FACTORIZATION

Nonnegative Matrix Factorization is a dimension reduction method that has been widely used for various tasks including image processing [29],[30], speech recognition [31],[32], video summarization [33] and internet researches [34],[35]. More recently, this approach has found its way into the domain of text mining such as text clustering [36],[37] and text summarization [16],[17],[18],[19].

When NMF 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 NMF algorithm is an m×n term-sentence matrix $V=[v_{1i},v_{2i},\ldots v_{ni}]$, where each entry is represented by multiplying a local and a global weighting factor as follows:

$$v_{ji} = L(t_{ji}) * G(t_{ji}) \quad (1)$$

There are different weighting schemes to determine these weighting factors. In this study we have used the following weighting factors:

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

NMF approximately decomposes the matrix $V \in R^{mxn}$, into the product of two lower rank matrices $W \in R^{mxr}$ and $H \in R^{rxn}$ where r can be considered as the number of hidden topics present in a document. The factor W_{mxr} is a term-by-topic matrix whose columns are the NMF basis vectors. Assume that W_1 is the first column of W_{mxr} : in this case the nonzero elements of W_1 which is sparse and nonnegative correspond to particular terms. By considering the highest weighted terms in this vector, one can assign a topic to this basis vector. Similar interpretation holds for the other factor H_{rxn}. It becomes a topic-by-sentence matrix with sparse nonnegative columns. Assume that H_1 is the first column of H_{rxn} Then element j of H_1 measures the strength of $topic_j$ appears in $sentence_1$.

The factors W and H are generally found by solving the following optimization problem:

$$\min \Phi_f(W,H) = \frac{1}{2}\|V - WH\|_F^2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(V_{ij} - (WH)_{ij}\right)^2 \quad (2)$$

$$\text{subject to } \forall_{ij}, W_{ij}, H_{ij} \geq 0 \quad (3)$$

where $\|.\|$ denotes the Frobenius norm. Although the objective function $\Phi_f(W,H)$ is convex in W only or H only, it is not convex in both variables together. Therefore it is not possible to expect an algorithm to solve the problem in the sense of finding global minimum. However there are many numerical optimization techniques that can be applied to find local minimum. One of these techniques is proposed by Lee and Seung [14]. In order to find a local minimum of the objective function, they propose multiplicative update rules for both W and H: after initialization by two nonnegative matrices, the elements of W and H are continuously updated until $\Phi_f(W,H)$ converges under the predefined threshold or exceeds the number of repetitions. This approach is illustrated below using MATLAB array operator notation:

MULTIPLICATIVE UPDATE ALGORITHM FOR NMF

```
W= rand(m,k);  % initialize W as random dense matrix
H = rand(k,n);  % initialize H as random dense matrix
for i = 1 : maxiter
      (MU)   H = H .* (W^T V) ./ (W^T W H+10^-9);
      (MU)   W = W .* (V H^T) ./ (W H H^T+10^-9);
end
```

The parameter = 10^{-9} is added to avoid division by zero.

The NMF can readily replace low rank factorizations such as the singular value decomposition (SVD) due to its features like the nonnegativity and sparseness of the factors of the approximation. The nonnegativity allows each object to be explained by an additive linear combination of intrinsic

'parts' of the data [12]. On the other hand, unlike SVD, the sparseness of the factors W and H allows NMF to save considerable storage space.

The major drawback of NMF is a lack of guarantee of a unique global minimum. In contrast to this, SVD have strengths concerning uniqueness and robust computation.

III. GENERATING AUTOMATIC SUMMARIES BY NMF

This step selects sentences for generating summaries by using NMF. We perform NMF on V to obtain non-negative semantic feature matrix W and non-negative semantic variable matrix H. After the decomposition phase, we use the NMF based sentence selection method to get summaries of given documents by considering the highest weighted sentences. This method is proposed in [19]: The Generic Relevance of a Sentence (GRS) selects the desired number of sentences with the highest semantic weight values.

Generic relevance of jth sentence:

$$\sum_{i=1}^{r}\left(H_{ij}\,\text{weight}(H_{i.})\right) \tag{4}$$

$$\text{weight}(H_{i.})=\frac{\sum_{q=1}^{n}H_{iq}}{\sum_{p=1}^{r}\sum_{q=1}^{n}H_{pq}} \tag{5}$$

The weight ($H_{i.}$) is the relative relevance of the i^{th} semantic feature ($W_{.i}$) among all semantic features. The generic relevance of a sentence refers to how much the sentence reflects major topics, which are represented as semantic features.

IV. PERFORMANCE EVALUATION PROCESS

This section introduces data corpus, evaluation dataset, preprocessing methods and test data sets used in performance evaluation.

A. Data corpus and Evaluation Data Set

In this work, we construct a corpus that contains 100 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 is created by selecting %33 of the most important sentences from each document for each summary. Statistics of the data corpus and the evaluation data set can be found in Table 1:

TABLE I. STATISTICS OF THE DATA CORPUS

Property	Data Corpus	Evaluation Data Set
Number of documents	100	100
Avg. no. Sentences /document	21.54	7.11
Document with min. number of sentences	10	3
Document with max. number of sentences	63	20

In order to extract automatic summary documents for each document, the NMF based sentence selection system selects the same number of sentences with human summarizer.

B. Preprocessing Methods

We used three preprocessing methods as a preprocessing phase:

- Stop words Removal (SR)
- Stemming (ST)
- Consecutive Words Detection (CWD)

The first two preprocessing methods are well known in information retrieval and text mining:

The number of word forms one can derive from a Turkish root may be in the millions. Hence the dimension of the vector space of a document becomes very large. The following list was taken from [38]. It shows the list of words that can be obtained from the root uyu-, 'sleep':

uyuyorum	*'I am sleeping'*
uyuyorsun	*'you are sleeping'*
uyuyor	*'he/she/it is sleeping'*
uyuyoruz	*'we are sleeping'*
uyuyorsunuz	*'you are sleeping'*
uyuyorlar	*'they are sleeping'*
uyuduk	*'we slept'*
uyudukça	*'as long as someone sleeps'*
uyumalıyız	*'we must sleep'*
uyumadan	*'without sleeping'*
uyuman	*'your sleeping'*
uyurken	*' while someone is sleeping'*
uyuyunca	*'when someone sleeps'*
uyutmak	*'to cause someone sleep'*
uyutturmak	*'to cause someone to cause anotherperson to sleep'*

Stemming can be used as a method to reduce the number of different word forms. In this paper, to find the stems of words, we used Zemberek [27] which is a Turkish Natural Language Processing (NLP) Library. It is an open source, platform independent, general purpose NLP library and toolset designed for Turkic languages, especially for Turkish.

We also removed stop words that are too common words with not "enough content" to make any difference in summarization.

The third preprocessing phase CWD is an innovative preprocessing approach applied in this study. Commonly occurring consecutive words represent a concept that can be a named entity, a compound word, a multi-word name or commonly used terms in a specific domain. In order to find these kinds of words we use Turkish Wikipedia. The main purpose of mining Turkish Wikipedia is to extract information by analyzing web links. Web links have various

information more than just providing transfer function between pages [28].

Wikipedia has two important characteristics: the dense web link structure and concept identification by web links (Uniform Resource Locaters -URLs). Turkish Wikipedia has a dense structure, too. Articles are strongly connected to each other by this dense structure and almost every concept (article/page) has its own URL as an identifier. The 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 Turkish Wikipedia and the consecutive words in 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. These selected consecutive words are stored into a dictionary and are used as a single term in our study. This provides semantic integration between consecutive words.

C. Test Data Sets

Several test data sets are constructed by using different preprocessing methods. The following is a list of abbreviations that refers to different preprocessing methods to obtain several test datasets for performance evaluation:

- OC means the original corpus that has the original documents.
- SR means that stopwords are removed from the original corpus.
- ST means that stems of words are stored as terms.
- CWD means that consecutive words are considered as a single term by using our proposed consecutive words detection method.

V. Experimental Results

A performance analysis of algorithms is conducted on the test data sets. As a performance measure, we used precision (P), recall (R), and f-measure (F) metrics. These measures determine the coverage between the manually generated and 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:

$$P = \frac{|S \cap T|}{|T|}, R = \frac{|S \cap T|}{|S|}, F = \frac{2PR}{P+R} \quad (6)$$

Performance analysis of the system is conducted on the test data sets. For performance evaluation, eight combinations of three different preprocessing methods and two different weighting schemes are used. Table 2 shows performance evaluation results of the system.

In this table different weighting scheme representations are denoted by two letters. The first letter shows the local weighting scheme and the second one shows the global weighting scheme with the following variations:

- *First letter*: L (Logarithm)
- *Second letter*: N(No weight), I(Inverse document frequency)

TABLE II. Performance evaluation results

	F-Measure	
Test Data Sets	**LN**	**LI**
OC	0.49	0.502
SR	0.481	0.501
ST	0.497	0.48
ST+SR	0.497	0.495
CWD	0.485	**0.515**
CWD+SR	**0.506**	0.487
CWD+ST	0.47	0.492
CWD+SR+ST	0.471	0.477

As it is seen from Table 2, the first two preprocessing methods (stopwords removal (SR) and stemming (ST)) do not change the performance of the NMF based summarization algorithm: For stemming, this might be due to the presence of homonyms. For example, the word "yüz" can refer to a face, a number, or excoriation. Applying stemming to these kinds of words can produce misunderstandings while NMF finds semantically similar words. To resolve this confusion, word sense disambiguation algorithms can be used. Removal of stopwords does not affect the results but it is useful for dimensionality reduction of *term x sentence* matrix V.

When the LN weighting factor is used the best f-measure (**0.506**) is obtained with CWD+SR preprocessing method. Similarly when the LI weighting scheme is used the best f-measure (**0.515**) is depicted with CWD preprocessing method. So it can be said that the proposed CWD preprocessing method gives better performance results. The representation of consecutive words as a single term provides semantic integration among consecutive words.

These results show that the consecutive word detection can be used as a preprocessing phase in text summarization.

VI. Conclusion and Future Works

This paper presents the performance analysis of a Turkish text summarization system that applies NMF with different preprocessing methods. The proposed consecutive word detection method is one of the main and original contributions of this study. This preprocessing method gives better performance results in Turkish text summarization system. We believe that this method will also be effective in other languages.

As a future work we plan to apply other summarization methods which use semantic approaches and similarity relationships among words. We also plan to use word sense disambiguation algorithms in order to get better performance results. Another objective of our research is to extend our grammatically verified current corpus and to have larger human generated summaries created by more people. We want to make the dataset available to other researchers

working in Turkish text summarization for a comparative analysis.

REFERENCES

[1] H. Jing, "Sentence reduction for automatic text summarization" Proc. of the 6th Applied Natural Language Processing Conference, Seattle, USA, pp. 310–315, 2000.

[2] F. Liu and Y. Liu, "From extractive to abstractive meeting summaries: can it be done by sentence compression?" Proc. of the ACL-IJCNLP 2009, Suntec, Singapore, pp. 261–264, 2009.

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

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

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

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

[7] I. Mani, B. Gates, and E. Bloedorn, "Improving summaries by revising them." Proc. of ACL, 1999.

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

[9] R. Barzilay and K.R. McKeown., "Sentence fusion for multidocument news summarization." Journal of Computational Linguistics, vol. (31), MIT Press., 2005, pp. 297 – 328.

[10] N.K., Batcha and A.M., Zaki, "Algebratic Reduction in Automatic text Summarization-The state of the Art", Proc. of ICCCE 2010, Kuala, Lumpur, Malaysia, 11-13 May 2010.

[11] Y. Gong, , X. Liu, , "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis.", Proc. of ACM SIGIR, pp. 19–25, 2001.

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

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

[14] D.D. Lee and H.S., Seung , "Learning the parts of objects by nonnegative matrix factorization ," Nature, vol. 401, pp. 788-791, 1999.

[15] J.H. Lee, S. Park, A.C., Ahn and D. Kim "Automatic generic document summarization based on non-negative matrix factorization," In Information Processing and Management, vol. 45, pp 20-34, 2009.

[16] S. Park, J. H., Lee, C.Ahn, J.S., Hong, and S.J., Chun "Query based summarization using Non-negative Matrix Factorization," Proc. of the International Conference on Knowledge-Based & Intelligent Information & Engineering Systems, pp.84-87, 2006.

[17] S. Park, J.H. Lee, D. Kim, and C. Ahn, "Multidocument summarization based on cluster using Nonnegative Matrix Factorization," Proc. of Annual International Conference on Software Seminar, pp.761-770, 2007.

[18] S. Park and J.H. Lee, "Topic-based multi-document summarization using Non-negative Matrix Factorization and K-means," Journal of KIISE: Software and Applications, vol.4, pp.255-264, 2008.

[19] J.H., Lee S. Park, C.M., Ahn and D.H., Kim "Automatic generic document summarization based on non-negative matrix factorization" Journal of Information Processing and Management, vol 45, pp. 20–34, 2009.

[20] Z. Altan, "A Turkish Automatic Text Summarization System", Proc. of IASTED International Conference on AIA, Innsbruck, Austria16-18 February 2004.

[21] M. Tülek, , "Text summarization for Turkish", Master Thesis, Istanbul Technical University, Turkey, May. 2007.

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

[23] C. Cığır, M. Kutlu, and I. Cicekli, "Generic Text Summarization for Turkish", Proc. of ISCIS, Northern Cyprus, 2009.

[24] A.Güran, E.Bekar and S. Akyokuş, "A Comparison of Feature and Semantic-Based Summarization Algorithms for Turkish", Proc. of INISTA, Turkey, 2010.

[25] M. Özsoy, İ.Çiçekli and F.N. Alpaslan, "Text Summarization of Turkish Texts using Latent Semantic Analysis", In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 869–876, Beijing, August 2010.

[26] TurkishWikipedia,http://tr.wikipedia.org/wiki/Ana_Sayfa

[27] Zemberek-Turkish NLP library, https://zemberek.dev.java.net/.

[28] K. Nakayama, T. Hara, and S. Nishio, "A Thesaurus Construction Method from Large ScaleWeb Dictionaries", Proc. of International Conference on Advanced Information Networking and Applications (IEEE AINA), 2007, pp. 932-939.

[29] I. Buciu, I. Pitas, "Application of non-negative and local non negative matrix factorization to facial expression recognition." Proc. of the 17th International Conference on Pattern Recognition, Cambridge, United Kingdom, 23–26 August 2004.

[30] S.Z. Li, X. Hou, H. Zhang, Q. Cheng,"Learning spatially localized, parts-based representations." Proc. of the IEEE Conference on Computer Vision and Pattern Recognition 1, pp. 207–212, Kauai-Hawaii, 8–14 December 2001.

[31] S.Behnke, "Discovering hierarchical speech features using convolutional non-negative matrix factorization." Proc. of the International Joint Conference on Neural Networks 4, pp. 2758–2763, Portland, Oregon, 20–24 July 2003.

[32] P. Smaragdis, J.C. Brown, "Non-negative matrix factorization for polyphonic music transcription." Proc. of the IEEE Workshop on Applications ofSignal Processing to Audio and Acoustics, pp. 177–180, New Paltz- New York, 19–22 October 2001.

[33] M.Cooper, J. Foote, "Summarizing video using nonnegative similarity matrix factorization." Proc. of the IEEE Workshop on MultimediaSignal, pp. 25–28, St. Thomas- U.S. Virgin Islands, 9–11 December 2002.

[34] J. Lu, B. Xu, H. Yang, "Matrix dimensionality reduction for mining Web logs." Proc. of the IEEE/WIC International Conference on Web Intelligence, pp. 405–408, Nova Scotia- Canada, 13 October 2003.

[35] Y. Mao, L.K. Saul, "Modeling distances in large-scale networks by matrix factorization." Proc. of the ACM Internet Measurement Conference, pp.278–287, Sicily-Italy, 25–27 October 2004.

[36] F. Shahnaz, M. Berry, "Document clustering using nonnegative matrix factorization." Journal of Information Processing and Management, vol. 42, pp.373–386, 2006.

[37] M. Chagoyen,P. Carmona-Saez , H. Shatkay , J.M. Carazo, Pascual-Montano, "Discovering semantic features in the literature: A foundation for building functional associations." Journal of BMC Bioinformatics vol. 7: 41, 2006.

[38] Hakkani-Tür, D., "Statistical Modeling Of Agglutinative Languages.", Phd Thesis, Department of Computer Engineering and Information Sciences, Bilkent University, Ankara,Turkey, August 2000.