Anadolu Üniversitesi Bilim ve Teknoloji Dergisi A- Uygulamalı Bilimler ve Mühendislik
Anadolu University Journal of Science and Technology A- Applied Sciences and Engineering

2016 - Volume: 17 Number: 2
Page: 401 - 412
DOI: 10.18038/btda.31812
Received: 07 January 2016 Revised: 16 March 2016 Accepted: 21 June 2016

A HYBRID STATISTICAL APPROACH TO STEMMING IN TURKISH: AN AGGLUTINATIVE LANGUAGE

Tarık KIŞLA[1,*], Bahar KARAOĞLAN[2]

[1]Department of Computer Education and Instructional Technologies, Ege University, İzmir, Turkey
[2]International Computer Institute, Ege University, İzmir, Turkey

ABSTRACT

Finding Stem is a complicated and important issue for agglutinative languages like Turkish where theoretically infinite number of surface forms can be obtained from a single lexeme. Both analytical and statistical approaches have been tried for stemming Turkish words. Two main problems that become apparent with these approaches are the involvement of a dictionary which enforces the assumption of closed vocabulary and the disambiguation of the actual stem among the numerous candidates. Here, we present a method that exploits the simple fact that nouns and verbs have different suffix patterns. We also use statistical methods which are used for stripping off the suffixes. Based on the suffix pattern PoS is determined, which then enables the decision for the stem boundary. Thus, the presented stemming technique that does not employ a regular dictionary, is a remedy for the disambiguation problem. The performance rate of the method on golden standard PoS tagged METU-Sabancı Turkish Treebank is found to be 93.83%.

Keywords: Stemming, Natural language processing, Turkish, Agglutinative language

1. INTRODUCTION

In Turkish, while the smallest meaningful part of the word is defined as root, the stem the largest part of the word which gives the meaning to the word. Therefore we can say a word consists of two parts: the stem that carries the meaning and the inflectional suffixes that fit the word to the context of "saying", in terms of time (tense), locality (place) and arity (singular or plural). In some analytical languages like English, these attributes are specified with separate words like prepositions and are very simple in nature. In English language, a word can get limited number of suffixes, generally it is one suffix. Thus, the stemming algorithms for English are very simple. In fact, the effect of stemming on these kinds of languages for computer understanding and information retrieval is open to debate. However, in agglutinative languages countless number of lexical and surface forms can be generated from a single root making the stemming an important issue in natural language understanding. Stemming is acknowledged as a performance-enhancing element for an agglutinative language in the field of information retrieval and natural language understanding [1,2,3,4,5,6,7].

General morphological structure of a Turkish word is shown in Figure 1.

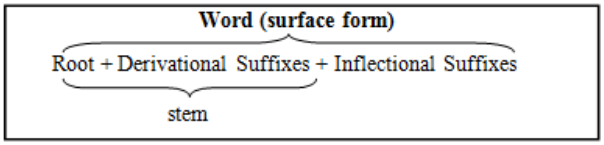


Figure 1. General morphological structure of a Turkish word

* Corresponding Author: tarik.kisla@ege.edu.tr

Roots are transformed into stems with derivational suffixes. The derivational suffixes change the meaning of the root/stem whereas the inflectional suffixes give the locality, tense or the arity of the word. In Turkish, inflectional suffixes generally come after derivational suffixes. Some exceptional derivational suffixes that come after inflectional suffixes are: like -gil, and -siz.

Table 1 shows some of the stems that can be derived from a single root "göz" (in English "eye"). In these words, the root and the derivational suffixes including gerund suffixes are separated by "#" and "/" respectively. "-" shows that there are no following inflectional suffixes. As can be seen in the examples (Table 1), the lexeme-stem contains the root and the derivational suffixes which together with inflectional suffixes constitute the surface form. If a word does not have a derivational suffix, root and stem are the same.

Table 1. Some different stems and surface forms derive from a stem “göz”(eye)

Word (surface form)	Root	Stem	Inflectional suffix(es)
göz#/ler-im (my eyes)	göz	göz	ler-im
göz#/ün-de (in your/his/her eyes)	göz	göz	ün-de
göz#cü/- (watchman)	göz	gözcü	-
göz#lük/- (eye glass)	göz	Gözlük	-
göz#lük-çü/- (optician)	göz	Gözlükçü	-
göz#lük-çü/-y-dü (once was an optician)	göz	gözlükçü (optician)	-y-dü
göz#lük-çü/-y-müş-ler (they were once opticians)	göz	gözlükçü (optician)	-mü-y-müş-ler

The number of suffixes and their numerous combinations that can be attached to a word, makes resolving of the actual stem form candidates a complex problem in agglutinative languages.

In Turkish, words may have multiple meanings according to the stem and the attached part of speech (PoS). For example “gelecek” in Turkish may mean “future” as noun or “will come” as verb depending on the context. In the “noun” case the stem is “gel#-ecek/” whereas, in the verb case the stem is “gel#/” (Figure 2).

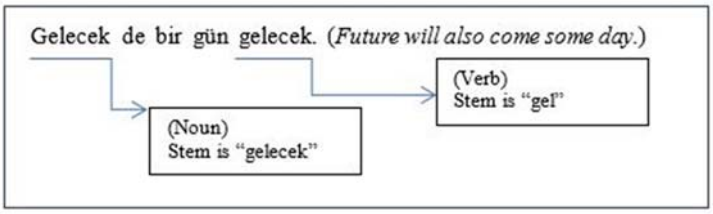


Figure 2. Stems of word "gelecek" according to different PoS

As evident from Figure 2, the stem is context dependent and PoS knowledge helps in disambiguation. In this study we aim to resolve the “stem disambiguation” problem and try to reduce the complexity in stemming that appears in the previous studies. The method employs both statistical and rule based approaches. Since the method is lexicon-independent, the reliability and accuracy is high. In addition, failure resulting from the word not being in the dictionary is thwarted.

The paper is organized as follows: In Section 2, literature related to previous work is given, in Section 3 the methodology is presented. In Section 4 and 5 results and conclusion are discussed respectively.

2. STEMMING METHODS IN TURKISH LANGUAGE

Lovins [8] was the first one to conduct a study on stemming in English language, and numerous studies have so far been performed on the field in question. Yet, the most widely accepted one among these studies is the study by Porter [9]. The algorithm introduced by Porter inspired various applications and became the de-facto standard algorithm for stemming in English. In addition to this algorithm, researchers developed various algorithms based on different methods with different performance and accuracy rates(Brute Force Algorithms, Suffix Stripping Algorithms, Lemmatization Algorithms, Stochastic Algorithms, Matching Algorithms etc.). Many studies on stemming in other languages also exist [10,11,12,13,14,15].

In this section, we summarize some stemming methods proposed for Turkish language under the titles of methodology, usage of lexicon and need for disambiguation. The methodologies of these studies are labeled as direct if either analytical nor statistical elements are employed. Studies including morphological analysis or grammatical properties are classified as analytical. Statistics/probability-based methods are referred to as statistical. Table 2 gives some stemming methods for Turkish language which are held by different authors.

Table 2. Some stemming methods for Turkish language

Description	Methodology	Use lexicon	Need disambiguation
Cut from first 5/6 letters [16]	Direct	No	No
Longest Match [17]	Direct	Yes	No
A-F Algorithm [1]	Analytical	Yes	Yes
FindStem [4]	Analytical	Yes	No
Zemberek [18]	Analytical	Yes	Yes
Suffix Stripping [19]	Analytical	No	Yes
Using n-gram statistic [20]	Statistical	No	Yes

The oldest method for stemming in Turkish was introduced by Köksal [16]. This method considers the first 5-6 letters as the stem. In another study, Kut et al. [17] developed a method named L-M (Longest Match). Using a lexicon containing the word stems and their possible forms, the method matches the stemmed word with the words found in the lexicon on the basis of the letter order from left to right. The longest matching word is considered to be the stem.

Solak and Can [1] used a dictionary of roots in their stemming work. Each root is accompanied by 64 properties compatible with stem producing methods from left to right. The letter units are matched to the roots lexicon in the order of left to right, and in case a matching root is found, the system derives the possible stems based on the accompanying rules. This study which is referred as A-F algorithm is basically an adaption of the morphological analysis method developed by Oflazer [21].

FindStem is another stemming method developed by Sever and Bitirim [4]. It basically consists of three elements: identifying the root, doing morphological analysis and identifying the stem. The method relies upon a lexicon containing the morphological and parts of speech properties of words, and syntactic rules. Sever and Bitirim reported that FindStem algorithm performs better than A-F and L-M algorithms.

Other analytical methods regarding to stemming Turkish words can be cited as "zemberek" developed by Akın and Akın [18] and "snowball" by Çilden [22].

Apart from these studies, Dinçer [20] approaches the stemming problem using n-gram statistics of letters in words being in the stem, in the suffix or in the boundary between a stem and the inflectional suffixes. As a result of this study, several stems are proposed and the performance (95.8%) is assessed according to the existence of the actual stem within the proposed ones.

Literature review reveals that the previous studies either use a lexicon which is never complete (open vocabulary), and/or deliver more than one candidate stems for a word which needs further to be disambiguated.

3. METHODOLOGY

It is agreed that vocabulary in agglutinative languages are not closed [23,24,20]. That is, new words keep coming in to the language by new terms or cultural exchange. New words are created by concatenating appropriate suffixes to the available root. The roots and the suffixes in a language change very slowly in time and can be considered as fixed. The infinitely many possible combinations of the roots and the suffixes are the reason for open vocabulary. Thus the methods that use dictionary are weak in the sense that their performance relies on the lexicon employed.

This paper elaborates an approach which use a table of possible suffixes in Turkish rather than a closed dictionary of words. The stem is disambiguated based on the PoS (Part Of Speech) of a word. Eight different parts of speech (noun, verb, adjective, pronoun, adverb, preposition, conjunction and interjection) are considered which can be classified into three categories as nouns, verbs and postposition [25,26]. In this study, postpositions are considered under the noun categories, due to the fact that they resemble nouns according to suffixes they have. In Turkish, verbs and nouns combine with different suffix patterns. Therefore, two finite state machines (FSMs) for stripping off the suffixes from nouns and verbs are designed. The processing of these FSMs with the related suffix patterns may result in several possible stems. The actual stem is determined by deciding on the stem and suffix boundaries through n-gram statistics of letters being in the stem, suffix or in the transition. Thus, the proposed approach is a hybrid method employing both analytical (stripping off the suffixes) and statistical (identifying the stem) methods (Figure 3).

Figure 3. Basic parts of suggested method

3.1. Stripping off the Suffixes

The suffix patterns for nouns and verbs are manually defined by taking into consideration the different grammatical rules for noun and verb inflectional suffixes in Turkish. 2623 verb suffix patterns are encountered in the corpus are reduced to 588 generic suffix patterns, which are categorized into 5 classes (Table 3).

Table 3. Verb inflectional suffix pattern groups

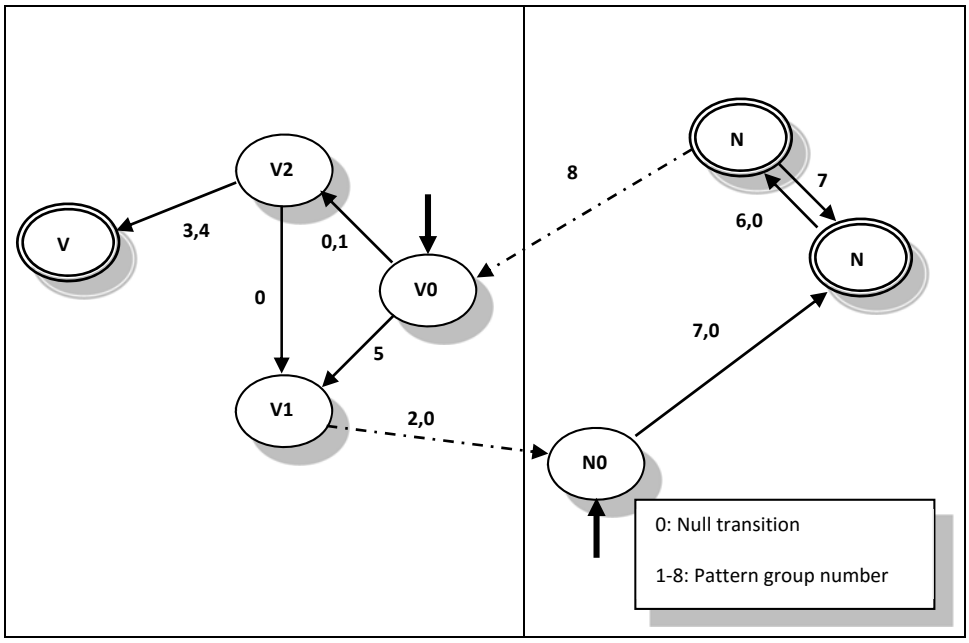
Group no	Suffix pattern	# of suffix pattern
1	Question	23
2	Gerund	24
3	Tense	480
4	Fortification / contingency compound tense	42
5	Compound question and tense	19

811 noun suffix patterns are encountered in the corpus are reduced to 130 generic suffix patterns using Turkish agglutination rules (phonetic change, elision, etc.). These suffix patterns are categorized into 3 class as shown in the Table 4.

Table 4. Noun inflectional suffix pattern groups

Group no	Suffix pattern	# of suffix pattern
6	All except "-ki", "–cesine" and "–ken"	127
7	("-ki")	1
8	Gerundium (–cesine and –ken suffixes)	2

The two interacting FSMs for identifying the suffix patterns for nouns and verbs are shown in Figure 4a and b.



a) Verb inflectional suffix FSM **b)** Noun inflectional suffix FSM

Figure 4. FSMs for stripping the suffixes

In the Figure 4, the numbers by the arrows show the state transition conditions for suffix pattern groups given in tables 3 and 4. Number 0 represents the null transition. Nodes V0-V3 represent the verb states and the nodes N1-N2 represent the noun states. V0 and N0 are the starting nodes. A terminating state is represented by a double circle.

Suffix striping is accomplished by processing either the noun or the verb FSM, scanning the letters of a word (whose part of speech is known) from right to left one by one and each time trying to find a match

for the letter sequence obtained in the suffix pattern lexicon and proceeding in the FSM according to the matched suffix pattern's group number. The process continues iteratively till all the letters in the word are exhausted. Hence, at the end of the process all possible stems are obtained.

The process is demonstrated step by step for the word "koyun" as an example (Figure 5). The word "koyun" may be analyzed in four different ways:

- koyun#/ (noun) (stem is "sheep")
- koy#/-un (noun) (stem is "bay")
- koy#/-un (verb) (stem is "put")
- koyu#/n (noun) (stem is "dark")

Before starting the analysis, the word is PoS tagged within the context of a sentence using suffix-based hidden Markov model [27]. In this example, we assume that PoS tag of the word is noun. In this case, "N0" is the starting point of the FSM for the analysis phase (Figure 4). In the proposed method, firstly the entire word is added to list of the possible stem candidates. After that, while the letters of a word are scanned from right to left, each cluster of letters is searched in the suffix pattern lexicon. In our example, firstly "-n" is stripped and searched in the suffix pattern lexicon. Once, the letter "-n" is found in the lexicon, state change is proceeding in the FSM according to the matched suffix pattern's group number (pattern's group number is 6 for "-n"). Then, "koyu" word is added to list of the possible stem candidates. The process continues recursively until all the letters (except first one) in the word are scanned.

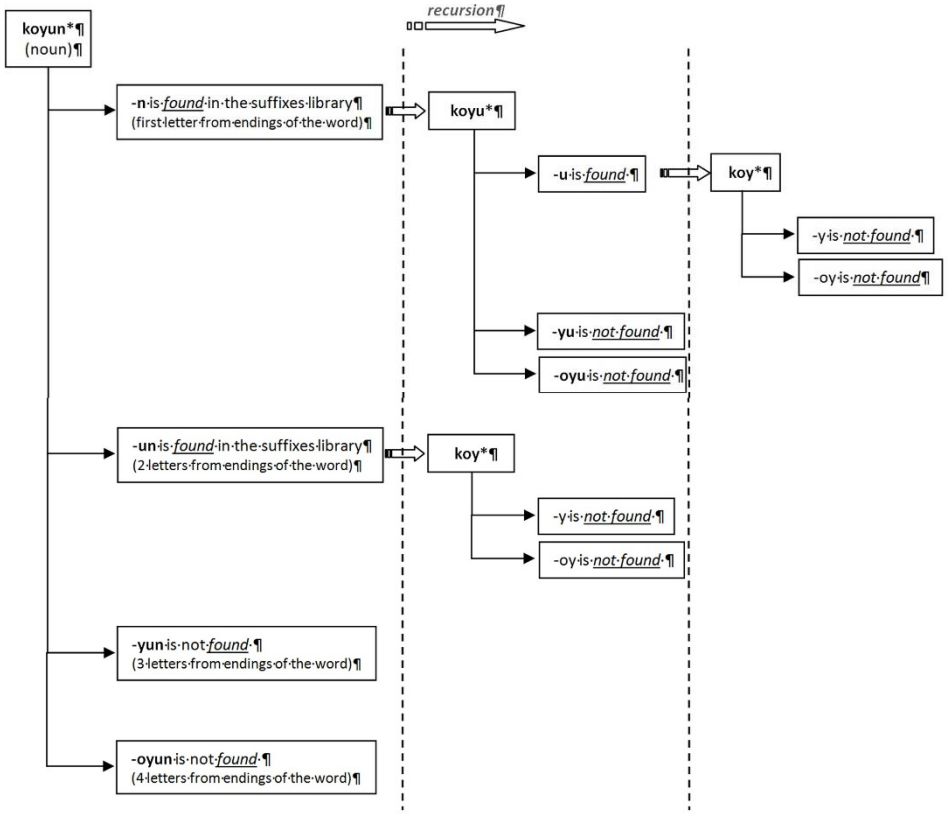


Figure 5. Analysing phases of "koyun" word

According to Figure 5, all possible stem candidates are determined using affix stripping ("koyun", "koy", "koyu") at the end of this sub process.

3.2. Identification of the Actual Stem

This is a statistics/probability-based method which selects the actual stem among the candidates identified in the previous phase. Before elaborating this process, we need to briefly introduce the notation adopted for bi-gram.

Notation of a word k:

$$k_n = h_1 h_2 \cdots h_n \ , \quad n\text{: length of word, } n > 0, h\text{: letters in the word}$$

Notation of a stem g:

$$g_m = h_1 h_2 \cdots h_m \ , \qquad m\text{: length of stem, } n \geq m > 0$$

Notation of a suffix e:

$$e_p = h_{m+1} h_{m+2} \cdots h_n \ \text{, p: length of suffix, } n \geq p > 0, n = m + p$$

The case of a letter pair being in the stem (G):

$$G = \left\{ (h_i, h_{i+1}) \ \middle| \ h_i \in g_m \wedge h_{i+1} \in g_m \wedge 0 < \mathrm{m} \leq n \right\}$$

The case of one of the letters in the pair being in the stem and the other being in the suffix (transition case - B):

$$B = \left\{ (h_i, h_{i+1}) \ \middle| \ h_i \in g_m \wedge h_{i+1} \in e_p \wedge 0 < \mathrm{p} < n \right\}$$

The probability for any letter pair (h_i, h_{i+1}) in a word ($k_n = h_1 h_2 \cdots h_n$) to be a part of the stem, to be a part of the affix system and to be in the stem-affix boundary (transition) are calculated as follows.

$$\Pr\left((h_i, h_{i+1}) \in G\right) = \mathrm{P_G}\left((h_i, h_{i+1})\right) = w_{g,i} / N$$
$$\Pr\left((h_i, h_{i+1}) \in B\right) = \mathrm{P_B}\left((h_i, h_{i+1})\right) = w_{b,i} / N$$

Here:

$w_{g,i}\,(h_i, h_{i+1})$: $f_{g,i} / (f_{g,i} + f_{e,i} + f_{b,i})$

$w_{b,i}\,(h_i, h_{i+1})$: $f_{b,i} / (f_{g,i} + f_{e,i} + f_{b,i})$

N: Total number of letter pairs (h_i, h_{i+1})

where;

$f_{g,i}$: $f_{g,i}\,(h_i, h_{i+1})$: Number of occurrences of the letter pair (h_i, h_{i+1}) in stem

$f_{b,i}$: $f_{b,i}\,(h_i, h_{i+1})$: Number of occurrences of the letter pair (h_i, h_{i+1}) in stem-suffix boundaries.

$f_{e,i}$: $f_{e,i}\,(h_i, h_{i+1})$: Number of occurrences of the letter pair (h_i, h_{i+1}) in first two letter of suffixes

The probability for a stem:

$\mathrm{P_K}(j)$, j. index of a possible stem; $1 \leq j \leq ng$ (ng: # of all candidate stems)

Assuming that the events G and B are independent phenomena, and i denotes the index of the last letter in the stem, the probability $\mathrm{P_K}(j)$ for each candidate stem of the word is calculated as follows:

$$P_K(j) = P_G((h_{i-1}, h_i)) * P_B((h_i, h_{i+1}))$$

The stem with the highest probability value "$\max(P_K(j))$" is concluded to be the actual stem.

The stem probabilities are calculated based on 4 different sets of data which are obtained by varying the PoS of words and n-grams of letters. In the data collection method (DCM) 1 and 3, each word is considered to be in one of the most generic PoS group, that is either noun or verb and in the other case all subgroups of PoS under noun and verb are considered. In the DCM 2 and 4, each word is considered to be one of the eight different PoS. Statistics of bi-gram of the letters which are explained above are using in dataset 1 and 2. 4-gram statistics are using in DCM 3 and 4. The structure of the data collection for stemming is shown in Figure 6.

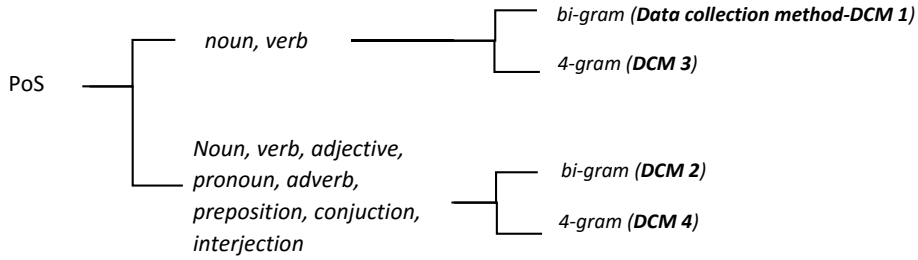


Figure 6. The structure of the data collection for stemming

4. EXPERIMENTS AND RESULTS

In order to determine success rate of the suggested method, METU (Middle East Technical University)-Sabanci University Turkish Treebank named as OSTAD [28,29] is adopted for both training and testing. OSTAD is morphologically analyzed by hand which ensures its high percentage of correctness. It consist total of 51,209 tokens (words including just letter(s) of the alphabet) and about 7,400 sentences. We used the first 6000 sentences in OSTAD as the training corpus and the remaining 1400 sentences as the test corpus.

To see the effect of the size of the training corpus on the performance, statistics are drawn from, 15 different corpus size of 250, 500, 750, 1.000, 1.250, 1.500, 2.000, 2.500, 3.000, 3.500, 4.000, 4.500, 5.000, 5500 and 6000 sentences.

The performance of stemming method for different data collection methods summarized in Figure 6, and is tested on 30 different test sets with the size of 200 sentences (approximately 1000 words) that are randomly selected from 1400 sentences of the OSTAD corpus. Mean of the performance of the methods on the 30 corpora is taken as the actual success ratio. Success rates are obtained by dividing the number of correctly identified word stems by the number of all words.

$$SuccesRate = \frac{\#\_of\_Correct\_Stemmed\_Word}{\#\_of\_Word}$$

To exclude the errors introduced by our PoS tagging and to see the net effect of the knowledge of PoS on stemming, we repeat all the experiments on previously hand tagged corpus. Figure 7 shows the graphs of the results obtained by using the training collections with different sizes for the 4 different methods.

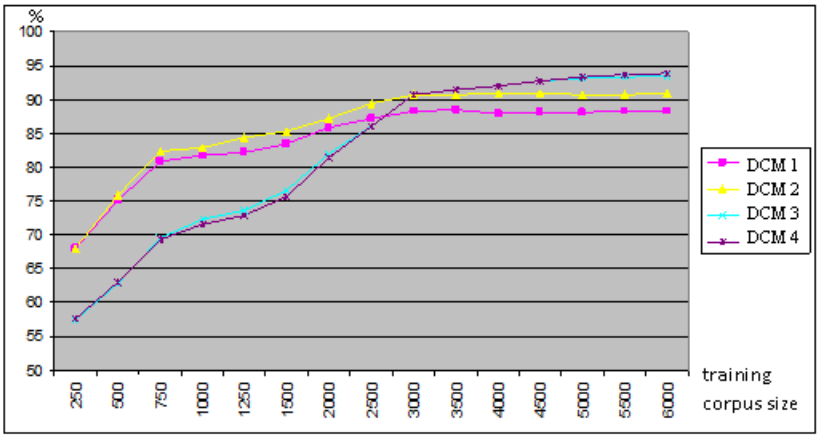


Figure 7. Results of the experiments with pre- knowledge of the PoS of the word

The graphs in figure 7 suggest that method with "DCM 4" is the best-performing one with a performance rate of 93.83%. A superficial examination of the table reveals that the performance rates of methods with "DCM 1" and "DCM 2" assume a stationary state beyond the collection size of 3000 sentences. The stabilization of the performance rate could be interpreted as indicating 3000 sentence corpus suffice for all the probabilities for the bi-gram letter units to be realized.

Our process of part of speech identification is adopted from Dinçer et al. [27] as a Hidden Markov Model (HMM) with the last 5 letter as the unit of calculation [30]. The results of the tests revealed that the highest performance rate of this method is approximately 90%.

The overall performance of our stemming method is actually the combined performance of the PoS method adopted and the performance of the stemming process exploiting the identified parts of speech of the word. The graphs in Figure 8 presents the results obtained by using the training collections with different sizes for the 4 different data collection methods.

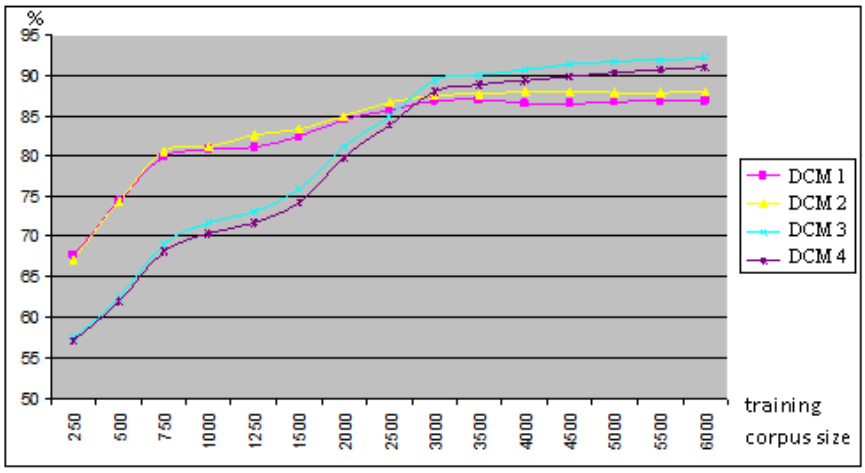


Figure 8. Results of the experiments on integrated PoS tagging and hybrid stemming method

According to graph presented above, method with "DCM 3" seems as the best-performing method (92.10%). "DCM 3" performs better than "DCM 4". It is because of this that "DCM3" operates on the generic verb and noun groups of PoS hiding the errors that might arise in using the detailed PoS (noun, verb, adjective, pronoun, adverb, preposition, conjunction and interjection). The lower performance

rates in the results compared to those in Figure 5 may be connected with the error in identifying the PoS of words in the adopted method.

5. CONCLUSION

In this paper we describe a hybrid stemming algorithm that employs both statistical and rule based approaches considering the PoS of a word, and presents a single stem avoiding the disambiguation problem. The rate of true stems identified (93.83%) on the hand tagged corpus supports our claim about the effect of PoS knowledge in stemming of Turkish words. This rate falls to 92.1% with the integrated automatic part of speech tagging and stemming method due to the error rate of the adopted PoS algorithm.

The reliability of the proposed method is high due to the fact that a closed and restricted vocabulary of suffix patterns is used and failure resulting from the word not being in the dictionary is thwarted.

These tests should be repeated on a bigger trained corpus to ensure reliability and performance increase in the results. The researchers plan to create a stemmed, PoS tagged and well organized Turkish corpus as a future work.

REFERENCES

[1] Solak A, Can F. Effects of stemming on Turkish text retrieval, in Proceedings of the Ninth Int. Symp. on Computer and Information Sciences (ISCIS'94), 1994, pp. 49-56.

[2] Ekmekçioglu FC. Lynch MF, Willett P. Stemming and n-gram matching for term conflation in Turkish texts, Information Research News 1996: 7; 2-6.

[3] Ekmekçioglu FC, Willett P. Effectiveness of stemming for Turkish text retrieval, PROGRAM-LONDON-ASLIB-2000: 34: 195-200.

[4] Sever H, Bitirim Y. FindStem: Analysis and evaluation of a Turkish stemming algorithm, in String Processing and Information Retrieval, 2003: 238-251.

[5] Pembe FC, Say ACC. A linguistically motivated information retrieval system for Turkish, in Computer and Information Sciences-ISCIS 2004, ed: Springer, pp. 741-750.

[6] Can F, Koçberber S, Balcık E, Kaynak C, Öcalan HÇ, Vursavaş OM. First large-scale information retrieval experiments on Turkish texts, in Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006, pp. 627-628.

[7] Can F, Koçberber S, Balcık E, Kaynak C, Öcalan HÇ, Vursavaş OM. Information retrieval on Turkish texts, Journal of the American Society for Information Science and Technology 2008: 59: 407-421.

[8] Lovins JB. Development of a stemming algorithm: MIT Information Processing Group, Electronic Systems Laboratory, 1968.

[9] Porter MF. An algorithm for suffix stripping, Program, vol. 14, pp. 130-137, 1980.

[10] Dolamic L, Savoy J. Stemming approaches for east european languages, in Advances in Multilingual and Multimodal Information Retrieval, ed: Springer, 2008, pp. 37-44.

[11] Savoy J. Light stemming approaches for the French, Portuguese, German and Hungarian languages, in Proceedings of the 2006 ACM symposium on Applied computing, 2006, pp. 1031-1035.

[12] Popovič M, Willett P. The effectiveness of stemming for natural-language access to Slovene textual data, Journal of the American Society for Information Science 1992: 43: 384-390.

[13] Viera AFG. Virgil J. Uma revisão dos algoritmos de radicalização em língua portuguesa, Information Research 2006: 12: 8-15.

[14] Tordai A, De Rijke M. Four stemmers and a funeral: Stemming in Hungarian, Workshop of the Cross-Language Evaluation Forum for European Languages 2005: Springer, 2006, pp. 179-186.

[15] Korenius T, Laurikkala J, Järvelin K, Juhola M. Stemming and lemmatization in the clustering of Finnish text documents, in Proceedings of the thirteenth ACM international conference on Information and knowledge management 2004, pp. 625-633.

[16] Köksal A. "Bilgi erişim sorunu ve bir belge dizinleme ve erişim dizgesi tasarım ve gerçekleştirimi," ed: Yayınlanmamış Doçentlik Tezi, Hacettepe Üniversitesi, Ankara, Turkey, 1979.

[17] Kut A, Alpkoçak A. Özkarahan E. Bilgi bulma sistemleri için otomatik Türkçe dizinleme yöntemi, Bilisim Bildirileri, 12. Ulusal Bilişim Kurultayı 1995, pp. 247-253.

[18] Akın MD, Akın AA. Türk dilleri için açık kaynaklı doğal dil işleme kütüphanesi: ZEMBEREK., Elektrik Mühendisliği 2007: 431: 38.

[19] Eryiğit G, Adalı E, An Affix Stripping Morphological Analyzer For Turkish, in *the IASTED International Conference on Artificial Intelligence and Applications*, Innsbruck, Avusturya, 2004.

[20] Dinçer T. A Statistical Information Retrieval System For Turkish, Phd. thesis, Ege Üniversitesi, İzmir, 2004.

[21] Oflazer K. Two-level description of Turkish morphology, Literary and linguistic computing 1994: 9: 137-148.

[22] Çilden EK., Stemming Turkish Words Using Snowball, ed: Retrieved, 2014.

[23] Mandelbrot BB. On the theory of word frequencies and on related Markovian models of discourse. In: Structure of language and its mathematical aspects 1961, 12. pp. 190-219,

[24] Kornai A. How many words are there? Glottometrics 2002: 4: 61-86.

[25] Uzun NE. Dünya Dillerinden Örnekleriyle Dilbilgisinin Temel Kavramları Türkçe Üzerine Tartışmalar, Türk Dilleri Araştırmalar Dizisi 39,İstanbul 2004.

[26] Hengirmen M. Türkçe temel dilbilgisi: Engin, 1998.

[27] Dincer T, Karaoglan B, and Kisla T, A suffix based part-of-speech tagger for turkish, in Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on, 2008, pp. 680-685.

[28] Oflazer K, Say B, Hakkani-Tür DZ, Tür G. Building a Turkish Treebank, In: Abeille A., Building and Exploiting Syntactically-annotated Corpora, Kluwer Academic Publishers, 2003. pp. 261-277.

[29] Atalay NB., Oflazer K, Say B. The annotation process in the Turkish treebank, in Proc. of the 4th Intern. Workshop on Linguistically Interpreteted Corpora (LINC), 2003.

[30] Kışla T. An integrated method for morphological analyse and part of speech tagging in Turkish, Phd. Thesis, Intenational Computer Institute, Ege University, Izmir Turkey, 2009.