Spell Checking and Error Correcting Application for Turkish

E. Yılmaz İnce

***Abstract*—Spell checking and error correcting applications for agglutinative languages is quite different from other languages, such as English. Thus, to develop spell checking and error correcting software, morphological analysis and mathematical preliminaries are required in agglutinative languages as Turkish. In this study, an application is developed for spell checking and error correcting for Turkish. The application uses Turkish corpus and morphological structure. nZemberek is used for Turkish dictionary suitability of the roots of words and compliance of suffixes to Turkish rules were determined using. The application utilizes n-gram depending on word length and the edit distance method for the correction of words. The application completes the spell check with a 95% success rate and suggests the correct options for spelling errors with an 86% success rate.**

***Index Terms*—Agglutinative languages, edit distance, n-gram, spell checking, spell error correcting.**

I. INTRODUCTION

Nowadays, obtaining and storing individual and corporate information is conducted in an online environment, thanks to the possibilities of Internet technology. Entering information to the computer by humans may cause spell errors. Increasing the value of the received text in the computer and manipulating the text with natural language processing algorithms require spell checking and correction [1]. In order to resolve spelling errors, misspelled words must be detected. The misspelled words must be corrected in order to be used in various fields, such as cryptology [2], [3], data compression [4], [5], and optical character recognition [6], [7]. The application mentioned in this paper used the spell checking and error correcting application in Turkish, in the project named automated assessment software for short answer question in learning management systems, to find and correct misspelled words in the student exam answer [8].

There are several methods for misspelled word detection. Various types of auto-correction techniques of text has noted [9], such as the minimum edit distance [10], similarity key [11], simple N-gram vector distance [12], singular value decomposition n-gram vector distance [13], probabilistic [14], and neural net [15]. Most of these methods are used in natural language processing. The methods of natural language processing are mostly performed in English and similar languages. While performing these methods in other languages that are different from English, various problems can arise [16], [17]. Spell checking and the error correcting process for agglutinative languages is quite different from other languages such as English. The studies conducted on misspelled word detection are not suitable for Turkish if they are enhanced according to keyword-based approaches [18]-[21] because Turkish is an agglutinative language, similar to Finnish and Hungarian. This affects the length of Turkish words. For instance the word "Başarısızlaştırıcılaştırıveremeyebileceklerimizdenmişsinizcesineyken" has 68 letters, whose root is Başarı (which means successful in English), has been proposed as the longest word in Turkish. Translation of this word in English is: “When as if you would be among those we cannot easily/quickly make a maker of unsuccessful ones”.

Spell check and error correcting applications are more complicated for Turkish due to the length of the words. Therefore morphological analysis and mathematical preliminaries are required for spelling error checking of Turkish [22]. The simplified nominal and verbal grammars of Turkish are [22], [23];

- Nominal root+ plural suffix+ possessive suffix+ case suffix+ relative suffix
- Verbal root+ voice suffix+ negation suffix+ compound verb suffix+ main tense suffix+ question suffix+ second tense suffix+ person suffix

When the studies are analyzed about spelling error checking in Turkish, there seems to be very few studies in this area. Certain studies have been revealed that perform error checking by examining the morphological structure of the language [24]-[27] by using ngram and edit distance algorithms [28]-[32] and by using text similarity [33]. Some of these studies have generated candidate words for all misspelled words after spell check, in order to correct misspelled words [28]-[30]. The developed software [28] comprises two-steps, determining all the roots from the dictionary that can be the root of the misspelled word, and generating (systematically) all the possible words that "resemble" the given character string, from these roots, realizes the correct word suggestions with a 95% success rate when the count of incorrect spelling letters is one. Also it is found that the accuracy of choosing among the corrections, the contextually salient correction as the first option is 74% through a statistical ranking process that takes context into account. Fig. 1 shows the program output for the incorrect form of “çalışmalarıyla” as “çaışmalarıyla”, with threshold 2. In another study [34] the software that has distribution of errors of edit distance 79.6% rate for one, 15% rate for two and 5.4% rate for three misspelled letters respectively

Manuscript received December 25, 2016; revised March 12, 2017. This work was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant EEEAG 114E952.

E. Yılmaz İnce is with the Computer Technologies Department, University of Suleyman Demirel, Isparta, 32200 Turkey (e-mail: ebruince@sdu.edu.tr).

doi: 10.18178/ijiee.2017.7.2.663

observed in a database of misspelled Turkish words. It claimed that 500 misspelled words are checked per second and generates all candidate words in less than 20 milliseconds when defining the performance of the application [34]. Another method, the finite machine, was used in a spelling error correcting program [29] examines an average of 10 words per second and the suggestion of candidate word takes 1 to 5 seconds for the misspelled words, according to word length. Another software program [30] realizes spelling correction with a 71% success rate and suggests the right candidate for spelling error correction with a 98% success rate. Similarly, the software [31] uses n-gram with the value 2, and corrects spelling with an 86.13% success rate.

```
Misspelled word:    çaışmalarıyla
Threshold t:        2
Solutions:          yazışmalarıyla            yatışmalarıyla
on left edge        yapışmalarıyla            yakışmalarıyla
                    ...                       ...
                    kaşışmalarıyla            çıkışmalarıyla
Candidate Roots:[4] çağ çakı çal çalı çam çan çap çar çat çatı
                    çağ çak çakış çal çalış çap çat çatış çav
                    çav çay

Solutions:[5]       Lexical                   Surface
-----------------------------------------------------------------
Edit distance 1     çat+Hş+mA+lArH+ylA        çatışmalarıyla
                    çap+Hş+mA+lArH+ylA        çapışmalarıyla
                    çalış+mA+lArH+ylA         çalışmalarıyla (corr.)
Edit Distance 2     çav+mA+lArH+ylA           çavmalarıyla
                    ...                       ...
                    çat+mA+lArH+ylA           çatmalarıyla
```

Fig. 1. The program output [28].

In this study, the studies about checking spelling errors are examined; spelling error checking is completed based on the Turkish corpus and morphological structure. In the current study, for the detected misspelled words, candidate words were generated by spelling corrector software that uses n-gram [12] and edit distance [35] methods.

II. Material and Methods

It is realized that there are few studies about Turkish spelling error checking and correction in the light of the literature view. Moreover, the n-gram method is the most widely used method in studies. For these reasons, in the current study, spell check was completed based on the Turkish corpus and morphological structure. Turkish dictionary suitability of the roots of words and compliance of suffixes to Turkish rules was determined by nZemberek [36]. The Turkish error checking software was developed in C#.NET language with Microsoft Visual Studio 2010 platform. The n-gram and edit distance methods were used in the software. nZemberek is used for spelling error checking and correcting in the application of the morphological structure of Turkish words. nZemberek contains a letter file, a suffix file, a text file (root words and special case tags), and optional syllable finder (shown in Fig. 2). Moreover, nZemberek has a suffix production class, a class for root word special cases, a helper for parsing operation, and a class containing basic information about the language specific classes. Therefore, Turkish morphological analysis can be completed with nZemberek according to Turkish suffix order rules. Once nZemberek reads a root word, and the related special cases are attached to the root object and the resulting object and stored into a special direct a cyclic word graph tree [37] (shown in Fig. 3).

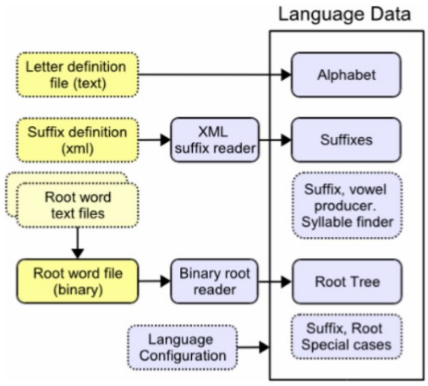

Fig. 2. Framework of nZemberek [36].

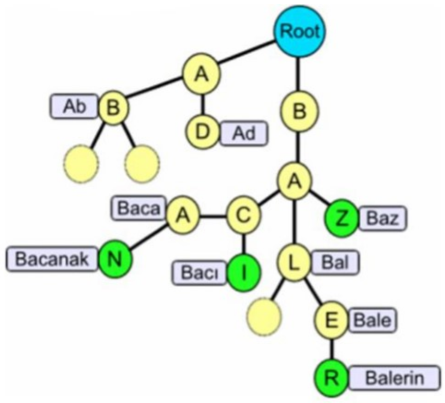

Fig. 3. Graph tree of nZemberek [36].

The n-gram is used in order to learn the rules of the letters of the word sequence on the whole words. N-gram is a contiguous sequence of n items from a given sequence of text or speech [38]. A n-gram of size 1 is referred to as a unigram, size 2 is a bigram, size 3 is a trigram, and larger sizes are sometimes referred to by the value of n. For example, the Turkish word "bilgisayar" that means computer, exemplifies bigram and trigram.

- bigram of "bilgisayar" ; bi-il-lg-gi-is-sa-ay-ya-ar
- trigrams of "bilgisayar" ; bil-ilg-gis-isa-say-aya-yar

When the studies that perform error checking in Turkish using n-gram and edit distance were analyzed, the unigram, bigram, and trigram types of n-grams were found to be used as text attributes [28]-[32], [34]. In their study [30], error checking is studied, 40 volunteers inscribed 15 different texts of at least 500 words through computer. According to the findings obtained in the research, the types of errors were indicated as follows: incorrectly typed character(s) (kirap instead of kitap), character substitution (kalm instead of kalem), character deletion (deft instead of defter), adjacent character swap (silig instead of silgi) and character insertion (örnerk instead of örnek).

In this study, firstly misspelled word length is determined and this value saved as n value. Different from other n-gram based studies in Turkish, in this study, n-grams, (n-1) grams, and (n-2) grams are used as attributes of the text that are based on the misspelled word length, n. In order to increase the performance of the program during the suggestion of candidate words, considering the study [30], three is used as the threshold value for the n-grams. Then, using the attributes obtained by the method of modified n-grams and directs a cyclic word graph tree obtained by nZemberek, and the selection of the suggestion of candidate words from the corpus is made by the edit distance algorithm (shown in Fig. 4). Edit distance is an algorithm that quantifies how dissimilar two strings (*X* and *Y*) are to one another by counting the minimum number of operations (insertion, deletion, substitution, etc...) required to transform one string into the other. When the specified words *X* and *Y* lengths are taken as m and n are respectively, ed ($X[m]$ and $Y[n]$) [35]:

$$\text{ed}(X[0], Y[j]) = j \quad 1 \le j \le n \tag{1}$$

$$\text{ed}(X[i], Y[0]) = j \quad 1 \le i \le m \tag{2}$$

In this study, the software was tested by using 2 as the threshold value for the edit distance value in the creation of the candidate words in order to select the neighboring words.

Edit distance

$ed(X[i+1], Y[j+1]) = ed(X[i], Y[j]) \qquad if\ x_{i+1} = y_{j+1}$

$= 1 + \min\{ed(X[i-1], Y[j-1]), \qquad if\ both\ x_i = y_{j+1}$

$ed(X[i+1], Y[j]), \qquad and\ x_{i+1} = y_j$

$ed(X[i], Y[j+1])\}$

$= 1 + \min\{ed(X[i], Y[j]), \qquad otherwise$

$ed(X[i+1], Y[j]),$

$ed(X[i], Y[j+1])\}$

Fig. 4. Edit distance algorithm.

III. Results

In this study, Turkish spell check and corrector software were developed. The software was developed in the C#.NET Windows application platform to provide user input of the word to be checked. When a user enters the word, the level of word correctness is tested. If correctness of the word is true, "Spelling is right" is indicated to the user. Otherwise, "Spelling is wrong" is indicated to the user. nZemberek is used for spelling error checking and correcting, which contains a letter file, a suffix file, a text file (root words and special case tags), and optional syllable finder. nZemberek reads the root word, and related special cases, which are attached to the root object and the resulting object and stores them in a special direct acyclic word graph tree. Right or wrong written word is checked by this framework [37]. After error checking the word, word suggestions are presented to the user for the identified misspelled words. N-gram and edit distance algorithms are used for spell checking and correction with 3 and 2 used as the threshold values, respectively. The misspelled word's length is used for n-gram calculating as n, (n-1), and (n-2). The candidate words selected by the edit distance algorithm with the nearest 2 distances were added to the suggestion list.

Thereafter, the suggestion list was sorted according to the frequency of candidate words. For example "çaışmalarıyla" word error checking that was typed by the user with "çaışmalarıyla…………Spelling is wrong" error checking alert and candidate words "çalışmalarıyla, çatışmalarıyla, çakışmalarıyla, acışmalarıyla, atışmalarıyla". The error is in the root structure, "çalış," which is the verb and means "work" in English, the neighboring words are found with the help of nZemberek direct acyclic word graph tree, n-gram, and edit distance.

The performance of the software that functions as spelling error checking and suggests of candidate words was performed on a computer that has Intel®Core™i3 CPU 2.53GHz 64 bite 4GB RAM features. As in the example of the word, "çaışmalarıyla", 14 microseconds were calculated as algorithm run time, as taken with a stopwatch.

IV. Discussion

In the current study, the software that utilizes n-gram depending on word length and edit distance algorithm, realizes the spell check with a 95% success rate and suggests the right candidate for spelling error correction with an 86% success rate. The software performance was also analyzed, and the program checks 10000 words per second.

TABLE I: Misspelled Word Examples

Misspelled word	Correct form of the word	n and(n-1) and(n-2) grams(µs)	1and2and3 grams (µs)
çaışmalarıyla	çalışmalarıyla	14	22
kitappp	kitap	11	18
örnerk	örnek	7	11
bağımlıl	bağımlılık	17	30
çiçekçimiş	çiçekçiymiş	11	18
oldurrmaktadırlar	oldurtmakdırtadırlar	11	19
bırakabileekmişcesine	bırakabilecekmişcesin e	17	26
gençleşşşmiş	gençleşmiş	18	46
algortimalarıyla	algoritmalarıyla	6	15

Table I shows misspelled words and the correct form of the words in Turkish that were analyzed with two different parameters. All words are tested with n, (n-1), and (n-2) grams to find the correct form of the word, then the time of the process was determined in terms of microseconds (µs) (shown in Table I). Furthermore, all misspelled words were tested with 1, 2, and 3 grams, and the results were obtained. According to the findings, n, (n-1) and (n-2) grams performances were higher than 1, 2 and 3 n-grams.

V. Conclusion

In this study, spell check and correction software was developed with n-gram and edit distance algorithms for Turkish, which is an agglutinative language. This study differs from the other studies that have reviewed candidate words for all misspelled words in order to correct misspelled words in Turkish [28]-[31], [34] taking into account the length (n) of analyzed words n, (n-1), and (n-2) grams were used to correct the word. This makes the application optimized for agglutinative languages as if they have a corpus with direct a cyclic word graph tree that has inputs as root word, suffixes and related special cases. This is a state of the art study due to the fact that this is the first time that an algorithm has been used on the Turkish language and results were successful. Spell check and error correcting methods can be used in situations where the text-based search is used to find accurate results instead of incorrect results. Moreover, the application can be used in web based environments by programmers in order to save Turkish texts to the other projects correctly. The application is used in project [8] to find and correct misspelled words in the student exam answer. Furthermore, the application can be used in many fields, such as cryptology, data compression, and optical character recognition for Turkish texts.

REFERENCES

[1] K. Oflazer, "Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction," *Computational Linguistics,* vol. 22, no. 1, pp. 73-89, 1996.

[2] M. Agarwal, "Text steganographic approaches: Comparison," *International Journal of Network Security and Its Applications*, vol. 5, no. 1, p. 91, 2013.

[3] K. F. Rafat and M. Sher, "Communication in veil: Enhanced paradigm for ASCII text files," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 7, 2013.

[4] J. L. Dolby, "An algorithm for variable length proper-name compression," *Information Technology and Libraries,* vol. 3, no. 4, pp. 257-275, 2013.

[5] B. A. Lipetz, P. Stangl, and K. F. Taylor, "Performance of Ruecking's word compression method when applied to machine retrieval from a library catalog," *Information Technology and Libraries,* vol. 2, no. 4, pp. 266-271, 2013.

[6] V. Kulyukin, A. Vanka, and H. Wang, "Skip Trie matching: A greedy algorithm for real time OCR error correction on smartphones," *International Journal of Digital Information and Wireless Communications*, vol. 3, no. 3, pp. 56-65, 2013.

[7] D. P. Bryant and B. R. Bryant, "Assistive technology for individuals with learning disabilities," in *Assistive Technologies for People with Diverse Abilitie*, Springer Press, New York. Cohen, A. (1986). Introduction to Computer Theory, John Wiley and Sons Inc., Newyork, pp. 251-276, 2014.

[8] TÜBİTAK Project Number: 114E952, Öğrenme Yönetim Sistemi Kısa Cevaplı Sorular için Otomatik Değerlendirme Yazılımı: Türkçe Örneği, 2016.

[9] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys (CSUR),* vol. 24, no. 4, pp. 377-439, 1992.

[10] R. A. Wagner, "Order-n correction for regular languages," *Commun. ACM*, vol. 17, no. 5, pp. 265-268, May 1974.

[11] M. K. Odell and R. C. Russell, U.S. Patent Numbers, 1,261,167 and 1,435,663 U.S. Patent Office, Washington, D.C., 1918.

[12] E. M. Riseman and A. R. Hanson, "A contextual postprocessing system for error correction using binary N-Grams," *IEEE Trans. Comput.*, C-23, pp. 480-493, May 1974.

[13] K. Kukich, "A comparison of some novel and traditional lexical distance metrics for spelling correction," *in Proc. INNtT* 90-Paris (Paris, France, July), pp. 309-313, 1990.

[14] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," *in Proc. the Eastern Joint Computer Conference*, vol. 16, pp. 225–232, 1959.

[15] T. Garaas, M. Xiao, and M. Pomplun, "Personalized spell checking using neural networks," 2015.

[16] Z. Güngördü and K. Oflazer, "Parsing Turkish using the Lexical-Functional Grammar Formalism," *Machine Translation*, vol. 10, no. 4.

[17] K. Oflazer, "Turkish and its challenges for language processing," *Language Resources and Evaluation*, 2014.

[18] L. Barari and B. Qasemi, "Spell checker adaptive, language independent spell checker," in *Proc. AIML 05 Conference,* CICC, Cairo, Egypt, pp. 19-21, December 2005.

[19] X. Tong and D. A. Evans, "A statistical approach to automatic OCR error correction in context," in *Proc. the Fourth Workshop on Very Large Corpora,* pp. 88-100, Copenhagen, Denmark, 1996.

[20] S. S. Kang and C. W. Woo, "Automatic segmentation of words using syllable bigram statistics," in *Proc. the Sixth Natural Language Processing Pacific Rim Symposium*, Tokyo, Japan, pp. 729-732, November 27-30, 2001.

[21] S. Deorowicz and M. G. Ciura, "Correcting spelling errors by modeling their causes," *International Journal of Applied Mathematics and Computer Science*, vol. 15, no. 2, pp. 275-285, 2005.

[22] H. Sak, T. Güngör, and M. Saraçlar, "Resources for Turkish morphological processing," *Language Resources and Evaluation*, vol. 45, no. 2, pp. 249-261, 2011.

[23] TDK. (2016). [Online]. Available: http://www.tdk.org.tr/

[24] A. Solak and K. Oflazer, "Parsing agglutinative word structures and its application to spelling checking for Turkish," *in Proc. the 14th Conference on Computational linguistics*, vol. 1, pp. 39-45, August 1992.

[25] A. Solak and K. Oflazer, "Design and implementation of a spelling checker for Turkish," *Literary and Linguistic Computing*, vol. 8, no. 3, pp. 113-130, 1993.

[26] S. Kuru and H. L. Akin, "Spelling checking in Turkish," in *Proc. the DECSYM 92 Latest Trends in Computing Symposium*, Antalya, 1992.

[27] H. L. Akın, S. Kuru, T. Gungor, I. Hamzaoglu, and D. Arbatli, "A spelling checker and corrector for Turkish," in *Proc. the Second Turkish Symposium on Artificial Intelligence and Neural Networks,* Boğaziçi University, 1993.

[28] K. Oflazer and C. Güzey, "Spelling correction in agglutinative languages," in *Proc. the Fourth Conference on Applied Natural Language Processing*, pp. 194-195, October 1994.

[29] S. Kaygın and M. M. Bulut, "Türkçe metinlerdeki Yazım Yanlışlarını Bilgisayar Ortamında Bulma ve Düzeltme," *Elektrik, Elektronik, Bilgisayar Mühendisliği 7. Ulusal Kongresi*, pp. 43-46, 1997.

[30] Ü. Çakıroğlu and Ö. Özyurt, "Türkçe Metinlerdeki Yazım Yanlışlarına Yönelik Otomatik Düzeltme Modeli," in *Proc. Elektrik, Elektronik ve Bilgisayar Mühendisliği Sempozyumu ve Fuarı (ELECO 2006)*, Aralık 6-10, Bursa/TURKEY, pp. 322-326, 2006.

[31] K. Günel and R. Aşlıyan, "Hece 2 gram İstatistikleri ile Türkçe Sözcüklerde Hata Tespiti," *IEEE 14. Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Belek, Antalya, 2006.

[32] R. Aşlıyan, K. Günel, and T. Yakhno, "Detecting misspelled words in Turkish text using syllable N gram frequencies," *in Pattern Recognition and Machine Intelligence*, pp. 553-559, Springer Berlin Heidelberg, 2007.

[33] B. Dursun and A. C. Sönmez, "Türkçe metin benzerlik Hesaplaması için Yeni Bir Yöntem," in *Proc. Signal Processing, Communication and Applications Conference,* April 2008, pp. 1-4.

[34] K. Oflazer, "Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction," *Computational Linguistics*, vol. 22, no. 1, pp. 73-89, 1996.

[35] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of the Association for Computing Machinery*, vol. 7, no. 3, pp. 171-176, 1964.

[36] A. A. Akın and M. D. Akın. (2016). Zemberek an open source NLP framework for Turkic languages. [Online]. Available: http://zemberek.googlecode.com

[37] Zembereknlp. (2016). [Online]. Available: http://zembereknlp.blogspot.com.tr/

[38] M. Damashek, "Gauging similarity with N grams: Language-independent categorization of text," *Science,* vol. 267, no. 5199, pp. 843-848, 1995.

Ebru Yılmaz İnce was born in Antalya, 1986. She received the B.S. degree in 2008 from Technical Education Faculty of Süleyman Demirel University, Turkey. She received M.S. degree in 2011 from the Department of Educational Technologies and completed Ph.D. degree in 2016 from the Department of Computer Engineering at Süleyman Demirel University. She is currently an instructor at the Department of Computer Technologies in Süleyman Demirel University. Her research interest includes educational technologies, natural language processing and engineering education.