Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 57-63, Nisan 2020

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 57-63, April 2020

Research Article

Spelling Correction with the Dictionary Method for the Turkish Language Using Word Embeddings*

Murat Aydoğan[1**], Ali Karcı[2]

[1] Bingol University, Genc Vocational School, Department of Computer Technologies, Bingol, Turkey (ORCID: 0000-0002-6876-6454)
[2] Inonu University, Faculty of Engineering, Department of Computer Engineering, Malatya, Turkey (ORCID: 0000-0002-8489-8617)

(Conference Date: 5-7 March 2020)

(**DOI:** 10.31590/ejosat.araconf8)

ATIF/REFERENCE: Aydoğan, M. & Karcı, A. (2020). Spelling Correction with the Dictionary Method for the Turkish Language Using Word Embeddings. *Avrupa Bilim ve Teknoloji Dergisi*, (Special Issue), 57-63.

Abstract

Today, a massive amount of data is being produced, which is referred to as "big data." A significant part of big data is composed of text data, which has made text processing all the more important. However, when text processing studies are examined, it can be seen that while there are many world language-oriented studies, especially the English language, there has been an insufficient level of studies published specific to the Turkish language. Therefore, Turkish was chosen as the target language for the study. A Turkish corpus of approximately 10.5 billion words was created, consisting of unlabeled data containing no spelling errors. Word vectors were trained using the Word2Vec method on this corpus. Based on this corpus, a new method was proposed called the "dictionary method," with a dictionary created covering almost all known Turkish words. Then, text classification was applied to a multi-class Turkish dataset. This dataset contains 10 classes and approximately 1.5 million samples. Vector values of the token words in this dataset were transferred from the dictionary by transfer learning. However, words not found in the created dictionary were considered as incorrect; then, using LSTM (Long Short-Term Memory), which is a deep neural network (DNN) architecture, the proposed method attempts to predict correct or similar words as replacement words. Following this process, it was seen that the accuracy rate improved by 8.68%. Turkish dataset that is created, corpus and dictionary will be shared with researchers in order to contribute to Turkish text processing studies.

Keywords: *Word Embedding, Turkish Text Processing, LSTM, Dictionary Method*

Kelime Gömmelerini Kullanarak Türkçe Dili İçin Sözlük Metodu ile Yazım Düzeltme

Öz

Günümüzde oldukça büyük miktarda veri üretilmektedir. Üretilen bu büyük verinin çok önemli bir kısmı ise text verilerinden oluşmaktadır. Bu durum, text processing çalışmalarının daha da önem kazanmasını sağlamıştır. Ancak yapılan çalışmalar incelendiğinde başta İngilizce olmak üzere birçok dünya dili odaklı çalışmalar yapılırken Türkçe diline özgü çalışmaların yeterli sayıda olmadığı görülmüştür. Bu nedenle bu çalışmada hedef dil olarak Türkçe seçilmiştir. Etiketsiz verilerden oluşan ve yazım yanlışı bulunmayan yaklaşık 10.5 milyar kelimeden oluşan etiketsiz ve büyük Türkçe bir derlem üretilmiştir. Word2Vec metodu kullanılarak bu derlem üzerinde kelime vektörleri eğitilmiştir. Bu derlemi temel alarak "Sözlük Metodu" adı verilen yeni bir yöntem önerilmiştir, üretilen derlem içindeki kelimeler ile hemen hemen tüm Türkçe kelimeleri kapsayan bir sözlük oluşturulmuştur. Daha sonra çok sınıflı Türkçe bir dataset üzerinde metin sınıflandırma işlemi uygulanmıştır. Bu veriseti içerisindeki token kelimelerin vektörel değerleri sözlükten transfer öğrenme ile aktarılmıştır. Ancak sözlükte bulunmayan kelimelerin hatalı kelimeler olduğu düşünülerek bir derin sinir ağı mimarisi olan LSTM (Uzun Kısa Süreli Bellek) yöntemi ile bu kelimelerin yerine doğru veya yakın anlamlı kelimeler tahmin edilmeye çalışılmıştır. Bu işlemin ardından metin sınıflandırma uygulamasının doğruluk oranında %8.68

* This paper was presented at the ***International Conference on Access to Recent Advances in Engineering and Digitalization (ARACONF 2020)***.

**Corresponding Author: Bingol University, Genc Vocational School, Department of Computer Technologies, Bingol, Turkey, ORCID: 0000-0002-6876-6454, maydogan@bingol.edu.tr

oranında gelişme olduğu görülmüştür. Üretilen Türkçe veriseti, derlem ve sözlük Türkçe metin işleme çalışmalarına katkı sağlamak amacıyla araştırmacılarla paylaşılacaktır.

Anahtar Kelimeler: *Kelime Gömme, Türkçe Metin İşleme, LSTM, Sözlük Metodu*

1. Introduction

Today, due largely to the widespread use of the Internet, the amount of data being produced each day has increased rapidly. According to IBM, about 2.5 billion data items are produced in a single day in large datasets known as "Big Data." The majority of this data (90-%95) is composed of text data. According to Forbes, the amount of text data produced within just the next 2 years will reach 40 zettabytes. In addition, Google processes more than 40,000 search results every second, which corresponds to approximately 3.5 billion searches per day [1]. In the literature, text classification problems are one of the most studied natural language processing problems.

While conducting various text analyses in many global languages, primarily English, the number of studies based on the Turkish language have been at an inadequate level. Kılınç et al. created a Turkish dataset called TTC-3600, comprising of six categories with data collected from various news sites. They also applied a text classification process to the dataset using different machine-learning algorithms [2]. The convolutional neural network architecture is a method of deep learning recommended for multidimensional inputs, and particularly for two-dimensional visual data. However, successful results have also been seen within studies in the field of natural language processing [3].

Another important development in the field of natural language processing has been the development of artificial neural network-based word embedding methods. The most popular method in this field is Word2Vec, as developed by Mikolov et al. [4], [5]. There are two different approaches for the Word2Vec method, which are the CBOW and Skip-gram algorithms. Another important word embedding method is the Glove method, which was developed by Stanford University [6].

The organization of this paper is as follows. In the next chapter, the methodology will be introduced and the datasets used in this study will be explained. The proposed method and experimental results will be explained in chapter 3. The study will be concluded in chapter 4.

2. Material and Methods

2.1. Deep Neural Networks

Deep learning is a special state of the topic of machine learning, and therefore falls under the concept of artificial intelligence. When reviewing the literature, the first neural network is the perceptron algorithm [7]. This network was formed from an input and output layer. With the addition of more than one level to these neural networks, the goal was to determine more complex connections [8]. This newly created architecture was called a deep neural network. It can also be known as the multilayered or multi-neuron state of classic artificial neural networks for deep neural networks. In recent years, along with the use of graphic processing units that can process faster and more powerfully, the cost of fast processing has fallen considerably. This development made deep neural networks that much more popular [9].

2.1.1. Long Short-Term Memory (LSTM)

The LSTM method brings solutions to the exploding and vanishing gradients problems faced by the simple RNN architecture, and is also an approach preferred in recent years as it generally provides better results than the basic RNN architecture. The Fig. 1 illustrates the internal structure of the LSTM cell [10].

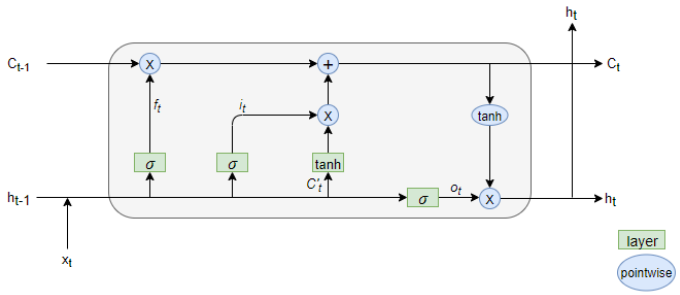

Fig. 1 LSTM architecture

C_t ***(Cell State):*** This is the channel to which information flows uninterruptedly from one cell to another. If considered that the LSTM cells are sequenced successively, the flow of information between the cells is ensured through these means [11]:

$$C'_t = \tanh(W_c \,.\, [h_{t-1}, x_t] + b_c \quad (1)$$

$$C_t = f_t * c_{t-1} + i_t * C'_t \quad (2)$$

f_t (Forget Gate): LSTM decides which information will be forgotten and therefore unable to be transferred to the other cells with the cell states of this gate. The information to be forgotten is decided based on implementing the Sigmoid process for the input information and the information coming from previous states [12].

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad (3)$$

i_t (Input Gate): This gate decides which information will be stored with the cell states and transferred to the other cells. The cell states are also updated after the processes in this step are completed, and are then prepared for transfer to a subsequent cell [13].

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (4)$$

o_t (Output Gate): This gate decides the output value. However, the entire value coming from the cell state is not provided as the output. For this, a filter is implemented, being passed through a cell state $tanh$ process [14].

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

2.1.2. Word Embedding

How the texts within a corpus are embedded is one of the most critical points for text-processing studies and the most important input for a network. The conversion of text into numerical data is called, in its simplest definition, word embedding. The same text can be converted to different numerical values in different manners [15]. It is a Word2Vec model essentially based on the principle of learning through artificial neural networks and words that was developed in 2013 by Mikolov et al. from the estimation-based word embedding method. The model is based on the principle of estimating the target word with reference to words taken as input.

Words not at the core of the window size in the CBOW (Continuous Bag of Words) model are taken as input, and the words in the core are attempted to be estimated as the output. This process continues until the sentence ends (see the Fig. 2). The value shown as w(t) is the output value to be estimated and is located at the center of the sentence, while the values shown as w(t-2). . . ..w(t+2) are the output values not at the center (window-size), based on the preferred dimension of the window [16], [17].

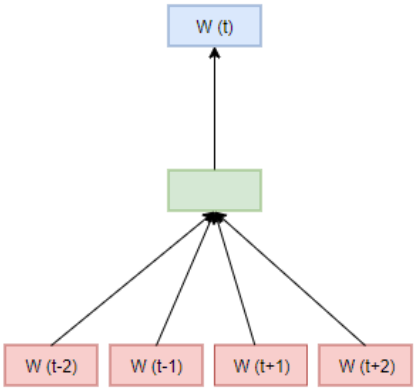

Fig. 2 CBOW method

2.2. Dataset

Two Turkish language datasets were created in the scope of the current study. The first dataset is a large corpus created from unlabeled Turkish texts, named as the Dictionary, and was used to train word vectors. The second dataset was created for the purposes of classification, and used together with the acquired word vectors in deep neural networks.

2.2.1. Dictionary

A large corpus was created from Wikipedia articles, and which required a storage capacity of about 60 GB. The reason for this corpus being generated was to create a kind of vocabulary dictionary. The corpus produced for this dictionary covers almost all known Turkish words, with Wikipedia used as its source in order to avoid word errors. The words in the corpus were then converted to vector form using the Word2Vec method. The resulting corpus contains 10.5 billion token words and 850,000 unique words. In addition, it is the largest Turkish dataset ever created, based on a review of the literature by the researchers, and includes approximately 22 million lines and 528 million sentences. The reason behind creating such a large dataset was to cover almost all known words of the Turkish language. Considering that there are approximately 115,000 words held in the Turkish Language Institution's dictionary, the produced dataset is considered to be comprehensive and therefore fit for purpose.

2.2.2. Dictionary Method

The word vectors were then transferred by transfer learning, and then text classification was applied on dataset. With this method, the accuracy value was improved [18]. In the study, a real dataset was used for the text classification created by the records kept by the call center employees and the messages sent by the users using the web page. Since the generated dataset is a real dataset, there are many spelling errors. Therefore, it has a negative effect on the performance of the model. The purpose of this method is to detect words that do not exist within a vocabulary. As mentioned in the Materials section, Dataset1 (for word embedding training) is the largest corpus ever created in Turkish language studies, and was also found to contain no spellchecking errors. Therefore, it is accepted that the words excluded from the vocabulary were either incorrect or misspelled. For the detection of out-of-vocabulary words, the following method was proposed. After the words are converted to tokens on the dataset where the classification process will be made, the numerical values of the words in the form of vectors were taken from the vocabulary and transferred by transfer learning method. The critical point at this stage is that the values in the vectors are initially set to 0 (zero). Therefore, the words in the vocabulary are replaced with vectors defined as 0 by changing their values and the words that are not in the vocabulary will remain as zero (0). In other words, when the process of converting words into vectors is complete, the vectors that are made up of zeros (representation of words) denote words that are considered out-of-vocabulary. The Fig 3 illustrates the general architecture of the dictionary method.

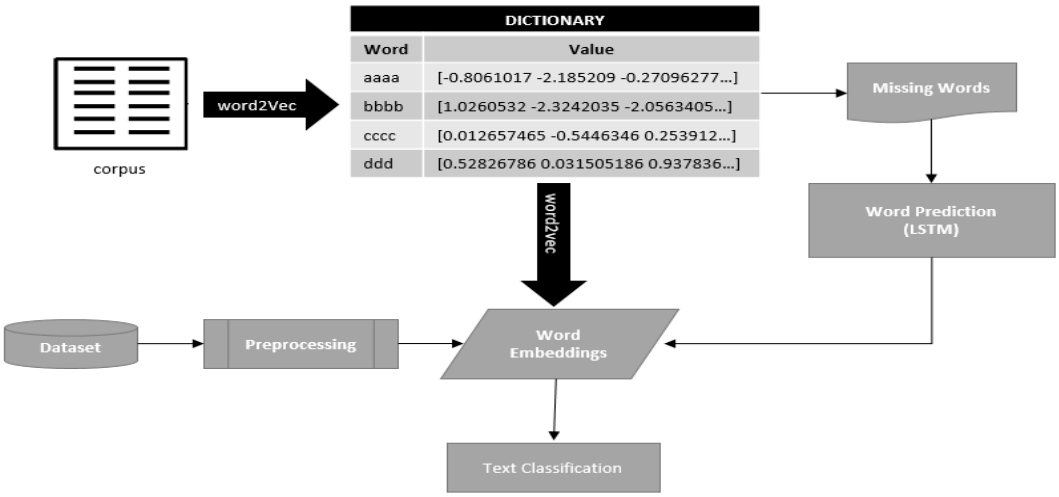

Fig. 3 Overview of our proposed method

3. Results and Discussion

When Table 1 is examined, it can be seen that a total of 197,288 words were taken from the dataset, but that only 142,047 words were found in the dictionary, representing 28% of unique words in the dataset that could not be taken from the dictionary and were therefore considered misspelt. The classification was applied on the dataset using the LSTM network, one of the deep neural network models.

Table 1. Results without using the dictionary method

Word Vectors in Dictionary	847766
Word Vectors in Dataset	197288
Read Word Vectors From Dictionary	142047
Missing Words	55241
Read Rate From Dictionary	%72

As an example, the sentence “Ben üniversite de ögrenciyim” was examined. The English equivalent of this sentence is “I am a student at university.” However, in this sentence, the word “university” which has the Turkish equivalent of “üniversite,” was misspelled as “üniverste.” The sentence was converted into tokens and the vector equivalent of each word was taken from the dictionary by transfer learning method. Therefore, the values defined as zero at first were then taken from the dictionary. When Figure 4 is examined, it shows that the word “üniverste” was not found in the dictionary because it was a misspelled word, and that the word’s vector remained as zero.

Fig. 4 Using of dictionary with transfer learning.

However, for each of the other words, a match was found in the dictionary and therefore the corresponding vectors were taken from the dictionary. This same process was applied for all sentences in the dataset. Any misspelled words were found when the vectors remained as zero after the conversion of the words into vectors. Then, instead of these words, words that are correct or considered to be closest to the correct word were attempted to be predicted by way of the LSTM method.

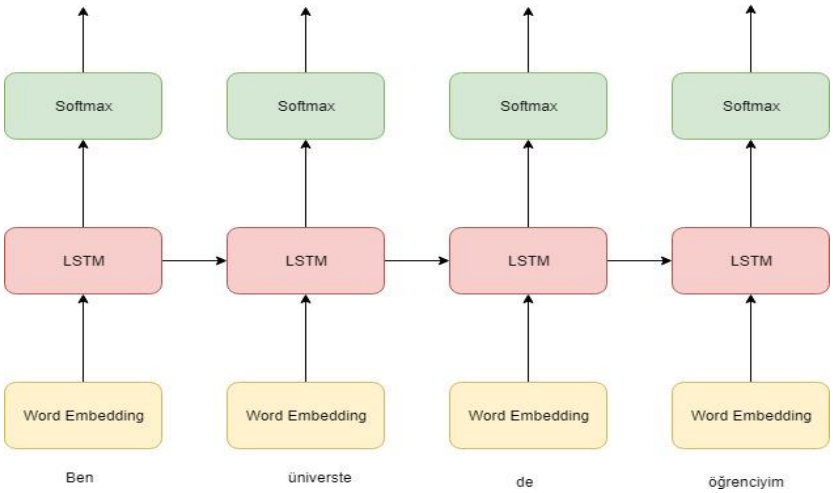

Fig. 5 Using of dictionary with transfer learning.

The structure of the LSTM model used in the current study was developed as shown in Figure 5. The number of neurons used in the developed model was 256. Dropout was added after the LSTM layer in order to prevent overfitting [14]. The added dropout value was implemented as 0.2, and the softmax function was used in the calculation of the output value.

model.similar_by_word("üniverste")	
[('üniversite', 0.6658480167388916), ('fakülte', 0.621586263179779), ('okul', 0.6204925179481506), ('konservatuar', 0.6177264451980591), ('ilahiyat', 0.6013895869255066)]	[('university'), ('faculty'), ('school'), ('conservatory'), ('theology')]

Fig. 6 List of similarity words

As shown in the Fig 6, the misspelled word “universte” was detected and prediction of words that could replace it was attempted based on the LSTM method. The purpose of this step is to replace a misspelled word with the correct word or one of the nearest synonymous words to the correct word. Figure 8 shows the words predicted as an alternative to the word “universte” and the English meanings of these words in the right column. With the proposed method, the word üniversite (university), which is the correct spelling of this word, was first guessed. In the words that came after that, it can be seen that they had similar meanings to this word. So, the model has shown a successful performance in this case. This method, called the dictionary method, was applied to 55,241 words from the dataset where the word could not be taken from the dictionary due to spelling errors. Instead of the misspelt words, correct or near-synonymous words were attempted to be predicted. The results of this process are shown in Table 2.

Table 2 Results with using the dictionary method

Accuracy without Dictionary Method	% 77.12
Replacing Words Using Dictionary Method	55241
Accuracy with Dictionary Method	% 85.80
Performance Improvement	**% 8.68**

When Table 2 was examined, LSTM was applied without the dictionary method and a 77% accuracy rate was obtained. Then, the 55,241 words that could not be matched were processed using the proposed dictionary method, resulting in an improved accuracy rate of 85% after obtaining vector equivalents. Thus, the performance of the proposed model was shown to improve at a significant rate of 8.68%.

4. Conclusions and Recommendations

The process of converting words into numerical form is one of the most critical points of text processing. A word embedding method proposed by Mikolov et al., known as Word2Vec, is still popular today. In the literature, word embedding is frequently mentioned as a very important input for neural networks. In the current study, a method based on word embedding was proposed, named the dictionary method.

Data collected from real environments is often found to be unclean. These impurities are often noted as spelling errors in text processing studies, and is a situation that significantly reduces the accuracy rate of proposed methods detailed in such studies. As a solution developed to this problem, an application named the dictionary method was developed in the current study and applied to an unclean dataset that contained numerous spelling errors.

As a result of this application, the accuracy rate was improved by 8.66%. It is thought that this method significantly contributes to the text processing literature, considering that much of today's data is derived from the Internet and therefore may be considered to be unclean data containing spelling errors.

In future studies, the researchers are planning to extend the developed corpus and dictionary. Additionally, they are planning to develop spelling correction using auto encoders.

References

[1] Wu, X., Zhu, X., Wu, G. Q., & Ding, W. (2013). Data mining with big data. IEEE transactions on knowledge and data engineering, 26(1), 97-107. D. Kılınç, A. Özçift, F. Bozyigit, P. Yıldırım, F. Yücalar, E. Borandag, TTC-3600: A new benchmark dataset for Turkish text categorization. Journal of Information Science, 43(2), 174-185, (2015).

[2] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278- 2324 (1998).

[3] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space. Proceedings of Workshop at ICLR. Scottsdale, (2013).

[4] Q. Le, T. Mikolov, Distributed representations of sentences and documents. 31st International Conference on Machine Learning, China, 2014 (2002)

[5] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Qatar, 2014.

[6] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review, (1958) 65(6), 386.

[7] Y. Qi, S. G. Das, R. Collobert, J. Weston, Deep learning for characterbased information extrac

[8] R. Socher, Y. Bengio, C. D. Manning, Deep learning for NLP (without magic), Tutorial Abstracts of ACL, (2012) 5.

[9] M. Tan, C. D. Santos, B. Xiang, B. Zhou, LSTM-based deep learning models for non-factoid answer selection, (2015) arXiv preprint arXiv:1511.04108.

[10] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, (2016) arXiv preprint arXiv:1607.04606.

[11] M. Sundermeyer, R. Schlüter, H. Ney, LSTM neural networks for language modeling. In Thirteenth annual conference of the international speech communication association, (2012).

[12] Y. Wang, M. Huang, L. Zhao, Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 conference on empirical methods in natural language processing, (2016) (pp. 606-615).

[13] K. S. Tai, R. Socher, C. D. Manning, Improved semantic representations from tree-structured long short-term memory networks, (2015) arXiv preprint arXiv:1503.00075.

[14] C. Zhou, C. Sun, Z. Liu, F. Lau, A C-LSTM neural network for text classification, (2015) arXiv Preprint. arXiv1511.08630.

[15]T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space. Proceedings of Workshop at ICLR. Scottsdale, (2013).
[16]Q. Le, T. Mikolov, Distributed representations of sentences and documents. 31st International Conference on Machine Learning, China, 2014.
[17]N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, JMach Learn Res. 2014, (2014) 15(1):1929-1958.
[18]Aydogan, Murat & Karci, Ali. (2019). Improving the accuracy using pre-trained word embeddings on deep neural networks for Turkish text classification. Physica A: Statistical Mechanics and its Applications. 541. 10.1016/j.physa.2019.123288.