Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal 2022, 16(4), 147–156
https://doi.org/10.12913/22998624/152453
ISSN 2299–8624, License CC-BY 4.0

Received: 2022.07.17
Accepted: 2022.08.22
Published: 2022.09.01

Experimental Comparison of Pre-Trained Word Embedding Vectors of Word2Vec, Glove, FastText for Word Level Semantic Text Similarity Measurement in Turkish

Cagatay Neftali Tulu[1*]

[1] Software Engineering Department, Adana Alparslan Turkes Science and Technology University, Balcalı, Çatalan Cd., 01250 Adana, Turkey

* Corresponding author's email: cagataytulu@gmail.com

ABSTRACT

This study aims to evaluate experimentally the word vectors produced by three widely used embedding methods for the word-level semantic text similarity in Turkish. Three benchmark datasets SimTurk, AnlamVer, and RG65_Turkce are used in this study to evaluate the word embedding vectors produced by three different methods namely Word2Vec, Glove, and FastText. As a result of the comparative analysis, Turkish word vectors produced with Glove and FastText gained better correlation in the word level semantic similarity. It is also found that The Turkish word coverage of FastText is ahead of the other two methods because the limited number of Out of Vocabulary (OOV) words have been observed in the experiments conducted for FastText. Another observation is that FastText and Glove vectors showed great success in terms of Spearman correlation value in the SimTurk and AnlamVer datasets both of which are purely prepared and evaluated by local Turkish individuals. This is another indicator showing that these aforementioned datasets are better representing the Turkish language in terms of morphology and inflections.

Keywords: semantic word similarity, word embeddings, Turkish NLP.

INTRODUCTION

Over the years, the studies in the field of Natural Language Processing (NLP) continue with increasing momentum. The academic achievements in this field provide a great convenience in the digital transformation and therefore it provides enormous benefits in people's daily life. NLP techniques are widely used in tasks such as machine translation, document classification, text summarization, sentence completion, automatic question answering, short answer grading. For a piece of text to be processed by an NLP application, it must be expressed mathematically in a way that the computer can process. Word is one of the basic semantic units in a text and it might be the starting point to express the words mathematically to process the large texts. The mathematical expression of a word is done using word vectors. Initially, the word vectors were produced with methods such as a bag of words, tf-idf [1], Latent Semantic Analysis (LSA) [2]. Processing of these vectors both increased the cost and reduced the success as a result of the growth in the corpus and the excess of the number of words in the text to be processed which exceeds billions in size. Finally, the low representation ability of these vectors and the widespread use of machine learning techniques opened the way to the generation of artificial neural network-based word vectors which then become defacto standard in a word embedding.

The discovery of generating word vectors by artificial neural network-based methods, which might be considered as a new era in Natural Language Processing history, claimed that words that have close semantic relations with each other in real life are also mathematically close

to each other in the vector space in which they are represented by artificial neural network-based methods. It became possible to prove *v("king")-v("man")+v("women")* ≅ *v("queen")* equation with these vector representations. Word2Vec [3], Glove [4], FastText [5] methods are artificial neural network-based methods that generate meaningful word vectors from large corpora without requiring any manual operations such as tagging, labeling, or any other manual intervention. While producing vectors, the morpho-syntactic structure of the words like inflection is not taken into account, only stop words cleaning and punctuation removal preprocessing steps are performed on the texts of the corpus. Considering this aspect, it is possible to produce word vectors directly without any human intervention with these methods. This is why these embedding methods are grouped in unsupervised methods.

Turkish is in the agglutinative language group in terms of its morphological structure and has similar features with languages such as Finnish and Hungarian. By the agglutinative structure of Turkish, it is possible to express a sentence that will be formed using more than one word in English with a single word. For example, the Turkish word "*göremediklerim*" is expressed in a word it is means "*the ones I could not see*" is expressed with six words in English. This distinctive structure of the Turkish language may have caused it to be a disadvantage for natural language processing studies than the other languages. The main motivation of this study is to compare Word2Vec, Glove, FastText which are named as fixed-window length neural network language models in word-level Turkish semantic similarity dataset. Experiments of this study have been conducted on three Turkish word-level similarity benchmark datasets and results of the tests showed that the Glove and FastText embeddings are far more successful than Word2Vec. Several factors might affect the quality and success of word vectors. The first one that might come to mind is the corpus used to generate those word vectors. In our study, we have used the word vectors generated through different corpora, ideally, the same corpus should be used for the fair and smooth comparison but, since it is costly develop word embedding's from huge corpora in terms of time and hardware resources, we have utilized the pre-trained vectors generated among different corpora. However, since the corpus used in word vector production are relatively close to each other in terms of the number of tokens, contents, vocabulary table length, and vector space size, it is assumed that the words are represented sufficiently at this point and it does not cause a serious deficiency in the experiments.

BACKGROUND

As stated in the hypothesis of Harris [6] that "words that occur in the same contexts tend to have similar meanings" and it is popularized by Firth [7] by underlining the idea "a word is characterized by the company it keeps" both of which are the theoretical point of the distributional semantics hypothesis which paved the way to represent the words as a mathematical unit. The representations of words as a mathematical unit have given momentum to develop highly effective NLP applications. Initially, word representations were done through the bag-of-words (bow) approach. In this approach, words in a document or a corpus are indexed with unique number values and each value is transformed into a vector that represents the words. The LSA (Latent Semantic Analysis) method [2], which represents the words in the text as vectors according to their tf-idf [1] values, is built on the bow approach. In methods that adopt the bow approach, the size of the word vector space increases as the number of unique words increases. The growth in vector space increases the cost to determine the vector of words. Parallel to this approach, which is also considered a statistical approach, word vectors were obtained through the lexical-semantic networks (i.e. WordNet [8]). The lexical-semantic network can be defined as marking all the unique words (named as sense) in a language's dictionary on a concept map in a graph structure and showing their relations with other words according to semantic relation types. With the PageRank method [9], which is based on the principle of expressing each concept on the semantic network in a vector space, the size of the word vectors created over the lexical-semantic network is the number of concepts defined on the network. For the English lexical-semantic network WordNet version 3.0, the vector size of each concept generated with PageRank method is around 118k. Besides similarity through vectorial representations of words, similarity computations through such lexical semantic networks are made by finding the concepts representing words and marking them on the network, and then the

geometric distances of these two marked points from each other such as distance between two nodes, shortest path, depth [10, 11].

The need for extra processing steps over the texts like lemmatization, POS Tagging, and the manual interventions, labeling the texts have caused the development cost to be higher for the creation of the word representations using lexical-semantic networks. Mikolov et al. (2013) [3] showed that efficient word vectors can be obtained by using all words in a text corpus on a feed-forward neural network without the need for any labeling or training and obtained the best results of that time in different natural language processing applications using the those obtained vectors. Using two methods Continuous Bag Of Words (CBOW) and SkipGram, they have claimed to generate efficient word representations named as Word2Vec from a large corpus. With CBOW, a neural network has been developed that takes the words in form of n-gram and predicts the excluded word against all the words entered except one in the window. But in SkipGram, the approach is to give a word to a feed-forward neural network and predict the words around it. With the Word2Vec method [12] developed by Google researchers, it become possible to represent any word on a corpus of billions of words with a 300-dimensional vector space. The Glove method (Global Vectors for Word Representation) [4] is another type of word vector generation method from huge corpora in an unsupervised manner. The method is developed by the researchers of Stanford University, unlike Word2Vec, the frequency of the words in a corpus being crossed with each other is also used in the training of the neural network. The best results were obtained in the word vectors obtained by this method, especially in Named Entity Recognition (NER) and semantic similarity studies. In the FastText [5] method, developed by Facebook researchers, the words are represented with n-grams at the character level, this makes it more beneficiary for generating word vectors from agglutinative languages such as Turkish. By the way, most of the Out of Vocabulary (OOV) words that are not seen during the training are expected to have word representations. With FastText, words are represented by the sum of the n-grams of characters that forms them. The FastText method is fast and can train a large corpus in a short time and generate representations for words that are not in the training set. Within this feature, p-retrained word representations are generated for many languages in the world and publicly available to download. Successful results have been gained in different natural language applications with the FastText method. This method is built based on the SkipGram which is used in Word2Vec to train the neural network. A relatively new approach to the word embedding method is presented named as ELMo [21] generated with deep learning structure using LSTM networks. The main difference of this method from the others mentioned is that this method generated word vectors according to context, polysemy words having multiple meanings have multiple embeddings to context. Theoretically, this method might provide better performance in sentence and document similarity tasks in which words might have multiple meanings according to context.

While many studies in English examine semantic similarity tasks using word vectors, a limited number of studies can be found for Turkish [12, 13, 14]. Aydogan and Karci (2019) [12] used the Beautiful Soup library to create a large corpus of 60GB Turkish texts, containing 10.5 billion words, and created word representation vectors by training this corpus with both Word2Vec (CBOW and SkipGram) and Glove methods. The created word representation vectors are examined with various criteria such as training time, similarity with other words, matching and semantic relations, and this study claimed that the Word2Vec method gave better results for both speed and performance in terms of detecting semantic similarities and training times compared to the Glove method. Dündan and Alpaydın (2019) [20] conducted an experimental evaluation study on the word embedding vectors in Turkish. Three different types of word vectors are generated from several Turkish corpora (Wikipedia, Huawei, and Bogazici University corpus). Word vectors are generated with the Word2Vec (SkipGram), FastText, and ELMo methods. Generated vectors evaluated semantically in text classification and similarity. Also, the effects of inflectional suffixes of nouns and verbs on vector generation are evaluated. Comparisons in semantic similarity are done intuitively selected Turkish words. It was concluded that FastText vectors are more successful in inflectional Turkish words and Word2Vec is better in word-level semantic similarity comparisons. In the other Turkish natural language processing studies, generated word vectors using mentioned methods are examined rather than for semantic

text similarity, but on other natural language applications [15].

MATHEMATICAL DEFINITIONS OF THE WORD EMBEDDING MODELS

Recent artificial neural network-based methods provide simple, scalable, and fast to train embedding models. The main idea in Word2Vec is to predict between every word and its context words. This is done through the SkipGram and CBOW models. In the Skipgram models, the target word is given to the neural network and it predicts the context words, it is position independent. On the other hand, CBOW predicts the target word by taking the context words around. There are two methods used to train the neural network one is the hierarchical softmax, and the other is the negative sampling.

The formal definition of the SkipGram [3], *"for each word t=1..T in the corpora, predict surrounding words –m,+m window length which is known as context words"*, the objective function is expected to maximize the probability of any context word given the center word. All the words in the corpus are initialized with one-hot vector representation.

The flow in the SkipGram with softmax model:

$$O_c \rightarrow E \rightarrow e_c \rightarrow \text{softmax} \rightarrow y' \tag{1}$$

where: O_c – the one-hot representation of the context word;
y' – the one hot representation of the predicted word;
E – the embedding matrix;
e_c – the embedding vector of the context word. The softmax function is used to predict the target word.

$$Softmax: p(t|c) = \frac{\exp(\theta_t^T e_c)}{\sum_{j=1}^{T} \exp(\theta_j^T e_c)} \tag{2}$$

where: t – the target word,
c – the context word;
e_c – the embedding vector of the context word;
θ_t – the parameter associated with output *t*. For the loss function between target and predicted value maximum log likelyhood is used;

$$L(y', y) = -\sum_{i=1}^{T} y_i log y_i' \tag{3}$$

where: y and y' are the one-hot representation of the predicted and target vectors.

Due to the computational cost of the Softmax classifier, Hierarchical Softmax is developed, which is a dramatic change in computational complexity and the number of operations needed for the algorithm. Hierarchical softmax (H-Softmax) is an approximation inspired by binary trees that were proposed by Morin and Bengio [22]. Instead of summing all the *T* numbers of the word probability, just summing log|T| number of the probability of the word for each target word. Another more beneficial approach is the Negative Sampling in the SkipGram model, in this approach, one positive pair of context, target pair feed into a neural network, and *k* number of the negative context target words feed. So instead of softmax, just the negative sampling method is applied without any exponential mathematical calculations. By the way, the model becomes to logistic regression with a sigmoid objective function. The flow in the SkipGram with negative sampling as in Eq. 4:

$$\begin{gathered} O_c \rightarrow E \rightarrow e_c \rightarrow P \rightarrow y' \\ P(y = 1 \,|c, t) = \partial(\theta_t^T e_c) \end{gathered} \tag{4}$$

where: P is the sigmoid function.

In the Glove method, the aim is to minimize the distance between the context and target vectors (through inner product) against log count of occurrence of these two words (context and target words) in the same context, and the objective function is defined as follows:

$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^{w} f(P_{ij})(u_i^T v_j - \log(P_{ij}))^2 \tag{5}$$

where: w – the number of unique words in corpora;
$f(P_{ij})$ – the weighting function showing the importance of the words by their frequency,
P_{ij} – the number of times word *i* appear in context *j*.

FasText method is generated using Skip-Gram model with negative sampling Mikolov v.d. (2013) [3]. But in addition to SkipGram, it has adopted a model called the Subword model. The intention behind the Subword model is to cover the internal structure of the words which Word2Vec ignores. For the subword information is obtained by a scoring function. Each word is represented as character n-grams. In this model, a word is represented by the sum of its character n-grams. So scoring function in Eq. 6 is generated.

$$s(w,c) = \sum_{g \epsilon G_w} z_g^T v_c \tag{6}$$

where: G_w – the set of n-grams appearing in *w,*
z_g – the vector representation to each *n*-gram *g*. This simple model allows sharing the representations across words, thus allowing to learn reliable representation for rare words.

MATERIALS AND METHOD

Turkish word level semantic similarity datasets

The benchmark datasets of semantic text similarity are generally built employing native language speaker individuals and carried out by scoring how much each word pair presented to the individuals is semantically similar to each other, the scores given to the word pairs are divided by the number of people and the real similarity score is obtained. Although the benchmark datasets are built sufficiently for the English language (such as RG65 [17], WS353 [23], SimLex-999 [24]), there is a very limited resource in Turkish for the word level semantic similarity. Sopaoglu and Ercan (2016) [16] have prepared the Turkish word-level semantic similarity dataset that we used in this study, this is one of the two datasets we found in the literature prepared particularly for Turkish. We name it as SimTurk dataset. This dataset consists of 140 different word pairs and all the pairs were scored from one to ten by ten different participants according to how similar each word pairs are, while score ten is the identical pairs, the score one is the different words and then average score taken from all participants are assigned as the semantic similarity score of a word pair. This dataset was created by considering the morphological structure of Turkish as an agglutinative language, such as its suffixes, morphemes, and structure. Some word pairs and similarity scores are given in Table 1 as an example.

Ercan and Yildiz (2018) [20] have prepared a Turkish Word Similarity and Relatedness Dataset that consists of 500 Turkish word pairs. Each pair is scored by 12 native Turkish-speaking volunteers in similarity and relatedness criteria and results are obtained with the average score among the 12 participants score. As in the TurkSim dataset, each words pairs are scored between 1–10 according to similarity and relatedness. The score of each word pairs for similarity and relatedness are given separately. In AnlamVer, word pairs are evaluated by two criteria as similarity and relatedness. Intension with similarity is to have common properties of the words. The intention behind relatedness is to have direct or indirect relations between the two words. For example, while *automobile* and *gasoline* are not similar, they are very related to each other. While this pair takes a low score in similarity, on the other hand, it takes a higher score in relatedness. For the sake of the comparison in the three datasets, the relatedness score given in the AnlamVer dataset are more in line with the similarity scores given in other datasets. So we take relatedness scores into account during the evaluation. The word pairs in AnlamVer are selected to enable the evaluation of distributional semantic models by multiple attributes of words and word-pair relations such as frequency, morphology, concreteness, and relation types (e.g., synonymy, antonymy). The similarity and relatedness scores of some sample word pairs are given in Table 2.

Another benchmark dataset is the RG65_Turkish dataset, which is prepared by translating the original RG65 [17] semantic similarity dataset produced for English into Turkish and it is used as the third benchmark dataset. It is used specifically in this study. The original RG65 dataset is used

Table 1. Examples of the word pairs and their similarity scores in SimTurk dataset

Word 1	Word 2	Similarity score
Kitaplardan	Hikayelerden	5.861
Kitaplardan	Kasaplardan	1.278
Kitaplarım	Romanlarım	7.611
Mutfaktaymış	Salondaymış	5.306
Saatlerce	Dakikalarca	6.583

Table 2. Sample word pairs and their similarity and relatedness scores from AnlamVer dataset

Word 1	Word 2	Similarity score	Relatedness score
Kırmızı	Gül	1.16	7.16
Suçlu	Şüphe	2.41	7
Laikçiler	Sekülerizmciler	9	9.41
Sevgili	Çiçek	1.16	6.5
Turizm	Ekonomi	2	5.83
Mühendis	Bilişim	2.33	6.83
Zeki	Çocuk	0.83	5.25

very effectively in evaluating the success of word-level semantic similarity studies in English. The scoring of RG65 dataset word pairs was made by native English-speaking individuals among 0–4. While 0 means two completely dissimilar words, 4 means words are identical. The score values for each word pair are assigned by averaging scores given by participants. Both Turkish and English and similarity scores of some words in the RG65_ Turkish dataset are given in Table 3.

Implementation

Neural network models prepared particularly for Turkish using Word2Vec, Glove and FastText methods are publicly available and can be downloaded from the internet [25]. Then, these downloaded models are loaded into Python using the Gensim Python library. In the vocabulary table of the loaded neural network models, word pairs of each dataset are searched, if one or both of the word pairs are not found in the vocabulary list, the similarity score is given as *Null* and those word pairs are left out of the evaluation. And the word which is not in the vocabulary is marked as OOV. For the remaining word pairs, the semantic similarity values are obtained by computing the cosine similarity (Eq. 7) of the vectors of the two words, this computation is done by calling the method named *similarity* provided by the Gensim library. The Spearman Rank correlation (Eq. 8) among the predicted similarity values and the actual similarity values of all word pairs provides the accuracy of the whole prediction, quality of the word vectors, and success of the prediction model respectively. Spearman rank correlation values are obtained between the [-1,1], the bigger Spearman correlation value shows the confidence of the predictions. Python codes, datasets can be found available publicly in the web address.

To compute the similarity between two word vectors, cosine similarity calculation is used. Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine value of the angle between two vectors.

$$sim(u,v) = \frac{u.v}{|u|.|v|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}} \quad (7)$$

where: u, v – the vectors having the same (n) dimension.

Spearman (ρ) is used to correlate word pair rankings. In case a measure returns scores instead of rankings, the ordered scores can be easily converted into ranks. Spearman correlation is computed as:

$$\rho = 1 - \frac{6\sum(x_i - y_i)^2}{n(n^2-1)} \quad (8)$$

where: x_i – the *i*. element of vector *x*;

Table 3. Sample word pairs and their similarity scores from RG65_Turkce dataset

Word 1 English	Word 2 English	Word 1 Turkish	Word 2 Turkish	Similarity score
Rooster	Voyage	Horoz	Seyahat	0.04
Fruit	Furnace	Meyve	Fırın	0.05
Food	Rooster	Yemek	Horoz	1.09
Glass	Jewel	Cam	Mücevher	1.78
Furnace	Stove	Fırın	Soba	3.11
Automobile	Car	Otomobil	Araba	3.92

y – the i. element of vector y;
n – refers to the dimension of the vectors.

RESULTS

Both three datasets SimTurk, AnlamVer, and RG65_Turkce are fed one by one into three different models and the results are shown in Table 4. The similarity predictions made using Turkish word vectors generated by the Glove and FastText methods had a higher correlation value than the Word2Vec, this shows the accuracy and confidence of the Glove and FastText vectors.

While the Glove vectors have made the most successful predictions for the SimTurk dataset, the FastText method has made the best predictions for the RG65_Turkce dataset. When we examine the RG65 and SimTurk datasets, we see that the words in the SimTurk dataset are more eligible to the Turkish word-level morphological structure. Based on this, we might conclude that Turkish word vectors created by the Glove method are more successful in word similarity. Another evaluation point can be made on the number of single words, the number of tokens, and the size of the embedding space. These values are shown in Table 5.

Table 4. Spearman rank correlation values showing the success of similarity prediction of word pairs in data sets for each method

Dataset \ Metod	Word2Vec	Glove	FastText
SimTurk	0.56	0.83	0.79
AnlamVerSim	0.44	0.60	0.57
AnlamVerRel	0.52	0.77	0.80
RG65_Türkçe	0.42	0.59	0.65

Note: *Evaluation of the AnlamVer dataset is done seperately for similarity scores and relatedness scores one by one

According to Table 5. It might be concluded that the reason for the greater number of unique words for FastText is that this method creates words using n-grams of characters, and this greater number does not indicate the volume of the unique words. Also, the greater number in token count in FastText should be related to the character n-gram representations of the words. The effect of the vector space dimension on the success of the model is not encountered because values are very close to each other. Although there is a reverse correlation among the number of single words in Word2Vec and Glove, this is not a unique reason for Glove's success.

In case a word in a word pair is not found in the vocabulary table of the model, the missing word has been marked as out of vocabulary (OOV) and that word pair was excluded from the evaluation. The high number of words marked as OOV can be seen as one of the indicators that the scope of the training corpus is not large enough. The number of OOV words and the number of evaluated word pairs for each pre-trained method are shown in Tables 6, 7, 8, and Figures 1, 2, 3, respectively.

The reason for the high number of OOV words in the SimTurk dataset is that the words in the dataset are generally extended with affixes in inline with the morphological structure of the Turkish language. In particular, the number of left out word pairs and the number of OOV words are lower in FasText this shows the success

Table 5. The metadata of the corpuses used for each for each embeding method and correlation values

Metod	Unique word count	Token count	Word vector dimension	Correlation in SimTurk dataset	Correlation in RG65_Türkçe dataset	Correlation in AnlamVer similarity	Correlation in AnlamVer relatedness
Word2Vec	412 K	73 M	400	0.56	0.42	0.44	0.52
Glove	253 K	2.76 B	300	0.83	0.59	0.60	0.77
FastText	2 M	14 B	300	0.79	0.65	0.57	0.80

Table 6. Number of OOV encountered in each method for SimTurk dataset

Metod	Number of word pairs	Unique words count	Number of OOV words	Number of ignored pairs	Number of tested pairs	Correlation value in SimTurk
Word2Vec	140	272	107	68	72	0.56
Glove	140	272	86	61	79	0.83
FastText	140	272	40	28	112	0.79

Table 7. Number of OOV encountered in each method for AnlamVer dataset

Metod	Number of word pairs	Unique words count	Number of OOV words	Number of ignored pairs	Number of tested pairs	Correlation value in AnlamVer for similarity	Correlation value in AnlamVer for relatedness
Word2Vec	500	317	71	143	357	0.44	0.52
Glove	500	317	65	127	373	0.60	0.77
FastText	500	317	47	93	407	0.57	0.80

Table 8. Number of OOV encountered in each method for RG65_Turkce dataset

Metod	Number of word pairs	Unique words count	Number of OOV words	Number of ignored pairs	Number of tested pairs	Correlation value
Word2Vec	65	49	2	3	62	0.42
Glove	65	49	2	3	62	0.59
FastText	65	49	0	0	65	0.65

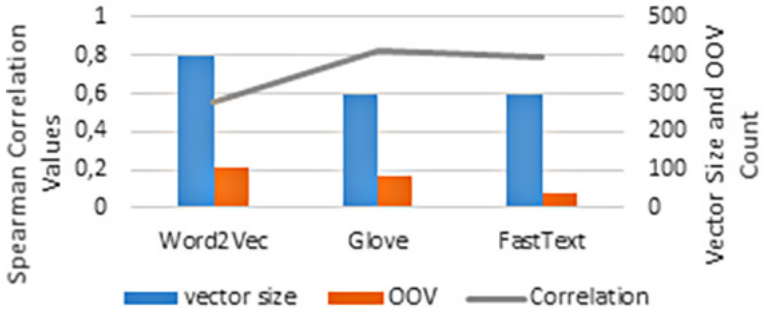

Figure 1. Comparison chart of vector size, OOV count, and correlation values of Word2Vec, glove and FastText methods in SimTurk dataset

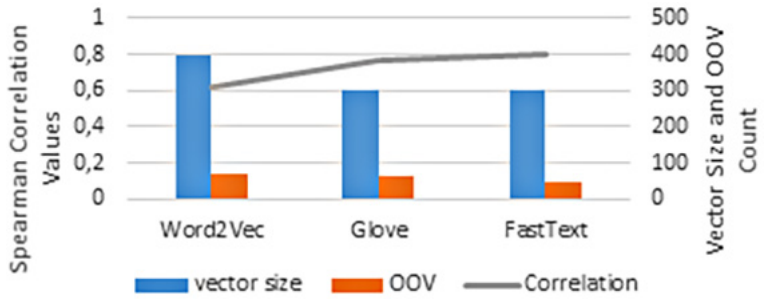

Figure 2. Comparison chart of vector size, OOV count, and correlation values of Word2Vec, glove and FastText methods for AnlamVer dataset (for relatedness)

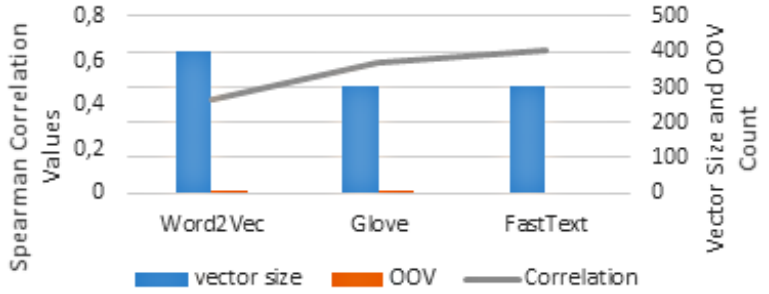

Figure 3. Comparison chart of vector size, OOV count, and correlation values of Word2Vec, glove and FastText methods for RG65_Turkce dataset

of the word embedding vectors produced by this method for the Turkish language, while on the other hand, it can also be concluded that the corpus used and the words on the corpus are more suitable for Turkish. The common OOV words obtained for all three methods in the SimTurk dataset are as follows:

"mutfaktaymış, salondaymış, karalılardan, duraktakiler, kedilerdenmiş, komedilerdenmiş, küpelerimizi, akranlarımızdan, güleceklerdi, çiçeklerdenmiş, eteklerdenmiş, firmadakilerden, şirkettekilerden, neşelendiğimiz, mavililerinki, evlilerinki, okuyacaklarmış, kapayacaklarmış, kağıttakiler, bardaktakiler, boyayacakları, kapayacaklardı, çalışkanlıktan, kazmıştım, korkusuzlardı, cesurlardı, kazananlardı, yenilenlerdi, eskilerimize, kazarlardı, etaplarım, korkusuzmuş, kokusuzmuş, sarılıların, altılıklar, dallandırdık, babalardaymış, annelerdeymiş, rendemiz, teflonlarından".

The common OOV words for three models in the AnlamVer dataset are as follows:

"kemalizmcilerden, tecrübelenme, dostçasına, iktidarcılar, ebediyeten, pratikleştirsin, sekülerizmciler, gençmişçesine, kavramsallaştırsın, cezalandırışı, biçimlendirilmişlerdir, arkadaşımsı, yaşlıymışçasına, mezarhane, kıtlıklarda, bitirdilerse, başlamadılarsa, öldürülüşündeki, bağışlayışı, yüreklicesine, sosyalleştirdikleri, atatürkist, erkekçene, sabahlardaki, davadaş, kemalci, iyiliyorsun, anormalleşiyor, alacaklanmak, üşengen, muhalefetçiler, korumaktaydılar, barıştırılırken, çenebazlığına, bolluklarda, asosyalleştirdikleri, saldırmaktaydılar, maymunsuları, affedişi, atatürkçülerden, savaştırılırken, manavınkiler, maymungilleri, kirlice, primatçaları, konuşkanlığına".

When we examine the given OOV words, we see that those common OOV words are not used very often in Turkish and therefore they may have not been trained in the corpus. Almost all the words have multiple inflections representing the agglutinative structure of Turkish which allows expressing a sentence with one word. Even if we increase the size of the corpus, it will not be possible to cover all words that have been generated by adding suffixes. Therefore, there will be always OOV words in Turkish, because theoretically, it is possible to generate an infinite number of words just by adding suffixes. As a result, using word representations that can fit this deficiency might solve this OOV problem in Turkish.

CONCLUSIONS

In this study, comparative experiments were conducted with three different word embedding methods on three different Turkish word similarity datasets. While Turkish word embeddings of Glove and FastText word have gained a higher Spearman correlation value, the success of Word-2Vec has been far behind these two methods. Meanwhile, when the results are examined based on the datasets, the correlations obtained with SimTurk and AnlamVer datasets had a higher correlation in comparison to Rg65_Turkce. Since the RG65_Turkce dataset is directly translated, it might not be suitable for semantic similarity measurement. Because words are generally without suffixes, most of the words are in stem form and some words are rarely used in Turkish. On the other hand, the SimTurk and AnlamVer datasets are completely Turkish-specific, the words used in daily life are preferred in these datasets which are in line with the morphological structure of Turkish. Another factor affecting the success of the FastText and Glove methods might be the size and the content of the corpus trained for the model, but the corpora used for the training of those specific methods are generally extracted from web platforms such as Google and Common Crawl, which are semantically close to each other. It is assumed that the diversity of the corpus content does not affect the vector representation quality. On the other hand, these three methods evaluated in this study do not support words with more than one meaning (polysemy). For example, there is only one vector representing one word. In case, the same words are used for different semantical meanings, these embedding methods do not support it, because they are context insensitive. Representing one word with one embedding vector is also called "meaning conflation deficiency" or ambiguity in the word meaning. If a semantic similarity study is carried out at the sentence or document level, this deficiency can cause lower success, particularly in context-sensitive datasets. Another point to take into account is the measurement of the contribution of inflectional suffixes to the semantic word similarity in Turkish. By separating the same words from inflectional affixes and measuring semantic similarity, the contribution of affixes to semantic word similarity can also be measured. And this kind of measurement should be supported by the new studies. While deciding to choose the best fit

for word embedding methods in Turkish natural language processing studies, the results obtained in this study will be a light to the researchers and will eventually give positive contributions to the Turkish NLP research ecosystem.

REFERENCES

1. Sammut, C., Webb, G.I. TF–IDF. Springer US, 2010. https://doi.org/10.1007/978-0-387-30164-8_832
2. Dumais, S.T. Latent Semantic Analysis. Annual Review of Information Science and Technology. 2005; 38: 188–230. https://doi.org/10.1002/aris.1440380105
3. Mikolov, T., Chen, K., Corrado, G., Dean, J. Efficient Estimation of Word Representations in Vector Space; 2013.
4. Pennington, J., Socher, R., Manning, C.D. GloVe: Global Vectors for Word Representation. In Empirical Methods in Natural Language Processing (EMNLP). 2014; 1532–1543.
5. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. Enriching Word Vectors with Subword Information. 2016. arXiv preprint arXiv:1607.04606.
6. Harris, Z. 1954. Distributional structure. Word, 10(23), 146–162.
7. Firth, J.R. 1957. A synopsis of linguistic theory 1930–1955. In Studies in Linguistic Analysis,. Oxford: Philological Society. Reprinted in F.R. Palmer (ed.), Selected Papers of J.R. Firth 1952–1959, London: Longman, 1968; 1–32.
8. Christiane Fellbaum (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.
9. Brin, S., Page, L. The anatomy of a large-scale hypertextual Web search engine (PDF). Computer Networks and ISDN Systems. 1998; 30 (1–7): 107–117. https://doi.org/10.1016/S0169-7552(98)00110-X
10. Budanitsky, A., Hirst, G. Evaluating WordNet-based measures of semantic distance. Comput. Linguistics. 2006; 32(1): 13–47.
11. Wu Z., Palmer, M. 1994. Verb Semantics and Lexical Selection. In Proceedings of the 32nd Annual Meeting ofthe Associations for Computational Linguistics.
12. Aydogan, M., Karci, A. Kelime Temsil Yöntemleri ile Kelime Benzerliklerinin İncelenmesi. 2019; 34, 181–195. https://doi.org/10.21605/cukurovaummfd.609119
13. Amasyalı, M.F., Balcı, S., Mete, E., Varlı, E.N. 2012. Türkçe Metinlerin Sınıflandırılmasında Metin Temsil Yöntemlerinin Performans Karşılaştırılması EMO Bilimsel Dergi, 2(4), 95–104.
14. Arabacı, M.A., Esen, E., Atar, M.S., Yılmaz, E., Kaltalıoğlu, B. 2018. Detecting Similar Sentences Using Word Embedding. In 2018 26th Signal Processing and Communications Applications Conference.
15. Esen, E., Özkan, S. Analysis of Turkish Parliament Records in Terms of Party Coherence. In 2017 25th Signal Processing and Communications Applications Conference (SIU) 1–4. IEEE 2017.
16. Sopaoglu, U., Ercan, G. 2016. Evaluation of Semantic Relatedness Measures for Turkish Language. In Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2016), Konya, Turkey 2016.
17. Rubenstein, H., Goodenough J.B. Contextual correlates of synonymy. Communications of the ACM. 1965; 8(10): 627–633.
18. Camacho-Collados J., Pilehvar, T. From word to sense embeddings: A survey on vector representations of meaning. Journal of Artificial Intelligence Research. 2018; 63: 743–788.
19. Ercan, G., Yıldız, O.T.. AnlamVer: Semantic Model Evaluation Dataset for Turkish – Word Similarity and Relatedness. In Proceedings of the 27th International Conference on Computational Linguistics, 2018, 3819–3836.
20. Dündar, E.B., Alpaydın, E. Learning Word Representations with Deep Neural Networks for Turkish. 27th Signal Processing and Communications Applications Conference (SIU), 2019, 1–4. https://doi.org/10.1109/SIU.2019.8806491
21. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee K., Zettlemoyer, L. Deep contextualized word representations. In Proc. of NAACL, 2018.
22. Morin, F., Bengio, Y. 2005. Hierarchical probabilistic neural network language model. In Robert G. Cowell and Zoubin Ghahramani, editors, AISTATS'05, 2005, 246–252.
23. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E. Placing Search in Context: The Concept Revisited. ACM Transactions on Information Systems. 2002; 20(1): 116–131.
24. Hill, F., Reichart, R., Korhonen, A. SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. Computational Linguistics. 2015; 41(4): 665–695. https://doi.org/10.1162/COLI_a_002
25. https://github.com/cagataytulu/TurkishWordSimilarity/