# ANALYSIS OF TEXT CLASSIFICATION WITH PRE-TRAINED WORD2VEC EMBEDDING MODELS OBTAINED BY USING DIFFERENT PREPROCESSING METHODS

[1]AYSUN GURAN, [2]SELIM AKYOKUS, [3]HAKKI YAGIZ ERDINC

[1]Doğuş University, Istanbul, Turkey
[2]Medipol University, Istanbul, Turkey
[3]Doğuş University, Istanbul, Turkey

**Abstract -** Semantic relationships between words are not considered in the frequency-based word embedding methods such as TF-IDF vectorization and one-hot encoding. Word2Vec is a predictive embedding model that uses a simple neural networks with a single hidden layer to generate words vectors that can identify semantic relationships between words represented with a lower number of dimensions. The main aim of this work is to analyze the effects of different preprocessing techniques on Word2Vec algorithm for Turkish news categorization task. To be able to reach this aim, a collection of approximately 5 million Turkish news documents was used as a training corpus and 2 million news documents were used for testing. We remove punctuation marks and stop words from the analyzed documents. We also apply stemming to the words of the documents. After the preprocessing phases, we use Naïve Bayes, Support Vector Machines, Artificial neural networks, K-nearest neighbors, logistic regression, and majority based voting ensemble classifier to classify our corpus. The experimental results show that the application of preprocessing techniques on the predictive embedding models increase the accuracy of text classification.

**Keywords -** Preprocessing Techniques, Word2vec, Turkish Text Categorization, Max Voting Classifier

## I. INTRODUCTION

Word embedding methods can be examined under two main categories: frequency-based word embedding methods and prediction-based word embedding methods. In the simple frequency-based word embedding methods, words can be represented with different vector structures. The most commonly used structures in the literature are the count vectors, the TF-IDF vectors, the Co-occurrence matrices, one-hot encoding representations [1][2][3]. The frequency-based word embedding methods assume that the words in a given document are independent of each other and they don't capture any semantic relations between words. On the other hand, the prediction-based word embedding methods such as Word2Vec[4], Doc2Vec [5], Fast Text [6]use shallow neural networks that can learn semantic relationships among nearby words, and they can predict a word from its context or predict the surrounding words (context) of a given word. These methods were prediction based in the sense that they assigned probabilities to the words based on their contexts and proved to be state of the art representation method for many different types of natural language processing tasks.Word2vec, developed by Tomas Mikolov in 2013,is a combination of two techniques – Continuous bag of words (CBOW) and Skip-gram model. Both of these methods use shallow neural networks which map words of similar meanings to nearby points in a continuous vector space. These techniques are based on the assumption that similar words should have similar context words. CBOW tends to predict the probability of a word given a context that may be a single word or a group of words. Skip – gram flips CBOW's architecture. It aims to predict the context of given a word. For example, assume that we have a following consecutive words: "artificial neural networks". CBOW model tries to predict the word "neural" when given "artificial" and "networks" as inputs, on the other hand Skip-gram model tries to predict "artificial" and "networks" when given the word "neural" as input. CBOW and Skip-gram techniques learn weights that act as word vector representations. In order to reduce the cost of learning phase two different algorithm can be used: Negative Sampling (NS) and Hierarchical Softmax (HS). The NS needs a few negative cases rather than the whole vocabulary, on the other hand, HS uses a binary tree to directly represent the probability of outputting each word in the vocabulary.

As it is known, preprocessing methods such as tokenization, stopword filtering, stemming are important and critical steps that effects the performance of text mining/processing application. These methods enable the transformation of unstructured data into structured format that the text mining algorithms can work on it. Hence these methods generally increase the performance of text mining algorithms. This paper aims to examine the impact of preprocessing methods on the Word2vec representation model used in text classification applications. In text mining domain, most of the studies on texts categorization examined the effects of preprocessing methods that use the frequency-based word embedding representation models. These studies report that preprocessing methods such as stemming and stopword filtering have significant effect on the accuracy results of the algorithms.

However, there are not so many studies to observe the effects of preprocessing methods on the prediction-based word embedding methods. Motivated by the lack of studies on this area, we analyze the effect of preprocessing techniques on pre-trained Word2Vec modelsfor the classification of Turkish texts. We remove punctuation marks and stop words from the analyzed documents. We also apply stemming to the words of the documents. After the preprocessing phases, we use Naïve Bayes (NB), Support Vector Machines (SVM), Artificial neural networks (ANN), K-nearest neighbors (KNN) and logistic regression (LR) classifiers to classify our corpus. The corpus in this study consists of 4.76 million unlabeled and approximately 2 million labeled Turkish new documents. We also use maximum voting ensemble classifier to combine the decisions of multiple classifiers to improve the classification accuracies.
This paper is organized as follows: Related work is discussed in Section 2. We give the details of applied methods we used in our experiment in Section 3. Concluding remarks and future work are included in Section 4.

## II. RELATED WORK

Text classification is the task of labeling natural language text documents with classes from a predefined set. Generally, a text classification task contains four different phases: preprocessing, document representation, feature extraction/selection, and classification. The preprocessing phase consists of activities such as punctuation marks removal, tokenization, stop-word removal and stemming. The document representation phase applies the frequency based and the predictive based word embedding models in order to represent documents with vectors. The feature extraction/selection phase tries to reduce the number of features by utilizing the methods such as Gini index [7], Information Gain [8], chi-square [9] and the final phase uses the classification algorithms that classify the unlabeled documents into a set of predefined categories.

In the literature, there are some research studies that point out that the use of preprocessing methods can provide performance improvements on the text classification task. Mulki et al. [10]applies stop-words removal, stemming, negation detection and common emoji recognition for the categorization of positive and negative multiple-domain reviews. They state that the preprocessing techniques improves the performances of the classification algorithms. Uysal and Gunay[11] examine the impact of preprocessing extensively on text classification. They state that choosing appropriate combinations of preprocessing tasks provides significant performance improvements. Torunogluet al.[12]denotes that the effect of stop-word removal and stemming have little effects on classification accuracy. Toraman et al. [13] concludes their study by emphasising that the preprocessing phases increase classification accuracies in most cases. Toman et al. [14], Pomikalek and Rehurek[15], and Duwairi et al. [16] also report that preprocessing tasks improve the performances of the classification algorithms. These studies use the frequency-based word embedding methods when they represent the documents.
There are few studies that investigate the role of text preprocessing in the text classificationarchitectures that use the predictive based word embedding methods to represent the documents. Weng[17] applies stop-words filtering, parts of speech tagging, negation tagging and stemming on sentiment analysis task and reports that preprocessing of input data represented with Word2vec does not improve the accuracy rating.Camacho-Collados and Pilehvar[18]focus on the effects of text preprocessing methods such astokenization, lemmatization and multi-word grouping for the neural network architectures. Their experimental results show that word embeddings trained on multiword-grouped corpora perform surprisingly well.
In this study, our objectiveis to analyze the effect of preprocessing techniques on the classification of text represented with pre-trained Word2Vec word embeddingmodels which learn representation of text using a prediction based modelusing neural network architectures. In addition to single classifiers, we also used an ensemble of classifiers that integrates the decisions of single classifiers using Maximum voting strategy.

## III. OVERVIEW OF APPLIED PREPROCESING METHODS AND DATASET

As it is stated in the previous section, the preprocessing methods are applied as the first steps in a text classification application in order to increase the performance of classification algorithms. During the experiments, the following pre-processing methods are utilized on our datasets:

- Removal of punctuation marks (RPM)
- Removal of stopwords (RSW)
- Stemming with ZEMBEREK(SWZ)
- RPM and RSW
- RPM and RSW and SWZ

The stop word list given in the reference [19] is used for the removal of stop words. Inorder to get the stems of words we use Zemberek[20] that is an open source, platform independent, general purpose natural language processing toolset designed forTurkish. By removing punctuation marks, we obtain datasets that only consist of alphabetic and alphanumeric characters. We also applied several preprocessing methods simultaneously to our dataset. For example, we applied both RPM and RSW methods to our dataset.

Within the scope of our study, two comprehensive data sets were collected from various online news websites. The first corpus has approximately 4.76 million unlabeled news documents and the second data set consist of 1,968,491 million labeled documents.The principal purpose of the use of unlabeled data set is to obtain word vectorsby applying the Word2Vec algorithm. In order to achieve this purpose, the Word2Vec algorithm with Skip-Gram modelis run on the unlabeled data sets.Since we use 5 different preprocessing methods individually or together,we applied Word2vec algorithm 5 timesto obtain five different pre-trained word embedding models.During the application of Word2vecalgorithm, the vector length isset to 100, the window size is determined as 5 and the words occur at least 5 times are taken into consideration. In addition, iteration number is accepted as 5 and Negative sampling methodis used in order to speed up training time.With the application of Word2vec algorithm using these parameters, the total number of words in each dataset and memory requirements of pre-trained word embedding models obtained from original unlabeled dataset without any preprocessing andfrom 5 different datasetsobtained with the application of preprocessing methods are shown in Table I.

| | ***Total number of words*** | ***Word2Vec (in GB)*** |
|---|---|---|
| The word vector model from unlabeled corpus | 1,674,166 | 2.03 |
| The word vector model after RPM is applied | 1,192,704 | 1.44 |
| The word vector model after RSV is applied | 1,673,708 | 2.03 |
| The word vector model after SWZ is applied | 737,042 | 0.89 |
| The word vector model after RPM + RSV is applied | 1,192,246 | 1.44 |
| The word vector model RPM + RSV + SWZ is applied | 607,501 | 0.74 |

**Table I. Memory Requirement for Word2vec Model**

The labeled data set used in our study consists of 1,968,491 new documents. 1,644,883 of them are used in training phase and the remaining 323,608 documents are used for testing. The train-test-validation split ratio is acceptedas 80%.A document vector is obtained by taking average of all vectors of word embeddings appearing in a document by using pre-trained models that are obtained from the unlabeled data corpus. Table II shows the total number of labeled documents under different categories. As it can be seen from the Table II, there are 11 different categories of documents in our labeled corpus.

| **Categories** | **Training Dataset** | **Test Data Set** | **Total** |
|---|---|---|---|
| World | 170640 | 39068 | 209708 |
| Education | 18293 | 1562 | 19855 |
| Finance | 294545 | 58853 | 353398 |
| Agenda Topics | 437259 | 91775 | 529034 |
| Art and culture | 30628 | 6119 | 36747 |
| Magazine | 67453 | 13487 | 80940 |
| Health | 47220 | 8233 | 55453 |
| Politics | 20745 | 4136 | 24881 |
| Sport | 371688 | 73690 | 445378 |
| Technology | 116969 | 13088 | 130057 |
| Life | 69443 | 12727 | 82170 |
| Total (%80 Trainset-%20 Testset) | 1,644,883 | 323,608 | 1,968,491 |

**Table II. Corpus Statistics**

## IV. CLASSIFICATION OF NEWS DOCUMENTS

After being passed from different preprocessing phases, the news documents are classified by using Naïve Bayes (NB), Support Vector Machines (SVM), Artificial neural networks (ANN), K-nearest neighbors (KNN) and logistic regression (LR) classification algorithms. In addition, an ensemble classifier is also used with Maximum Voting (MV) integration strategy to combine decisions of the best multiple classifiers. In MV algorithm, hard voting criteria is used in decision making phase that predicts the class label by taking the majority vote of decisions of multiple classifiers.

Table III shows the accuracy results of each classifiers on datasets represented with pre-trained word embedding models that are obtained after applying RPM, RSV, SWZ, RPM + RSV and RPM +RSV + SWZ preprocessing methods.

As it can be seen from the table, the highest performance result (0.779) is obtained byusing the SVM classifier when both the removal of punctuation marks (RPM) and the removal of stop words (RSV) preprocessing methods are applied at the same time. As it can be seen the second and third rows of Table III, both RPM and RSV methods improve the performances of each of classifiers. Stemming (SWZ) doesn't improve performances of all the classifiers. But the combined use of RPM, RSV, and SWZ methods improves the accuracy values of each classifier. By considering the average performances of the preprocessing techniques given in the last of Table III, it can be said that the classifiers generally produce better accuracy results on datasets encoded with pre-trained word embedding models obtained with Word2Vec algorithm after the applications of the preprocessing techniques. From Table III, we can also observe that SVM, ANN and LR classifiers produce best performances. We also combine the

decisions of these top three classifiers with Maximum Voting ensemble integration method.

| | NB | SVM | ANN | KNN | LR |
|---|---|---|---|---|---|
| Original corpus | 0.558 | 0.775 | 0.771 | 0.616 | 0.771 |
| The corpus after RPM is applied | 0.573 | 0.778 | 0.775 | 0.619 | 0.775 |
| The corpus RSV is applied | 0.573 | 0.776 | 0.771 | 0.619 | 0.773 |
| The corpus SWZ is applied | 0.563 | 0.771 | 0.768 | 0.613 | 0.767 |
| The corpus RPM + RSV is applied | 0.589 | **0.779** | 0.775 | 0.624 | 0.776 |
| The corpus RPM +RSV + SWZ is applied | 0.578 | 0.777 | 0.778 | 0.620 | 0.774 |
| Average | 0.572 | 0.776 | 0.773 | 0.618 | 0.772 |

**Table III. Performance results of the classifiers**

The performance of the ensemble classifier with Maximum Voting integration strategy can be seen in Table IV.

| | Maximum Voting Classifier |
|---|---|
| Original labeled corpus | 0.779 |
| The labeled corpus after RPM is applied | 0.780 |
| The labeled corpus RSV is applied | 0.781 |
| The labeled corpus SWZ is applied | 0.769 |
| The labeled corpus RPM + RSV is applied | **0.782** |
| The labeled corpus RPM +RSV + SWZ is applied | 0.773 |

**Table IV. Performance results of Ensemble Classifier with Maximum Voting integration strategy**

As it can be seen from Table IV, we obtained a better accuracy result (**0.782**) with the ensemble classifierwhen compared with single classifiers. The best resultis achieved when RPM and RSV preprocessing methods are applied before generating the pre-trained word embedding models. When compared with the accuracy results of each of single classifiers given in Table III, the ensemble classifier withthe maximum voting classifier produces better performance results after almost the application of any preprocessing methods.

## V. DISCUSSIONS AND FUTURE WORK

The preprocessing phase is a very important first step of any text mining application that often increase the performance and efficiency of text mining algorithms. Within the scope of our study, two comprehensive data set are collected from various online news websites. The first corpus consists of approximately 4.76 million *unlabeled* news documents and the second one includes 1,968,491 million *labeled* documents. By using the unlabeled document set, we obtained 5 different pre-trained word embedding models after the application of the different preprocessing methods. These word embedding models are used to represent the labelled dataset. Then, a set of classification algorithms are applied on the labelled dataset in order to observe the effect of the preprocessing methods on predictive word embedding models obtained with Word2vec algorithm. The experimental results show that the application of preprocessing techniques on the predictive embedding models increase the classification accuracies of text classification algorithms on Turkish datasets. As a future work, we plan to apply other word predictive embedding algorithms like Glove and FastText in order to obtain pre-trained models after the application of different preprocessing methods. We also plan to analyze the effect of preprocessing techniques on predictive word embedding models for different languages like English and German.

## REFERENCES


[1] Arroyo-Fernández, I., Méndez-Cruz, C. F., Sierra, G., Torres-Moreno, J. M., &Sidorov, G. (2019). Unsupervised sentence representations as word information series: Revisiting TF–IDF. *Computer Speech & Language*, *56*, 107-129.

[2] Chen, S., Chen, Y., Yuan, F., & Chang, X. (2018). Establishment of herbal prescription vector space model based on word co-occurrence. *The Journal of Supercomputing*, 1-12.

[3] Naiboğlu, H.S:,Kaptıkaçtı, O., Sardal, C., Güran, A., Uysal, M.,"Unsupervised And Supervised Term Weigthing Methods For Character N-Gram Based Author Categorization", CIE44&IMSS'14 Proceedings, 14-16 October 2014, İstanbul/Turkey.

[4] Mikolov T, Chen K, Corrado G, Dean J. (2013), "Efficient estimation of word representations in vector space". Proceedings of Workshop at ICLR. Scottsdale, Arizona 2-4 Mayıs 2013.

[5] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML 2014), pages 1188–1196, Beijing, China.

[6] Joulin, A., Grave, E., Bojanowski, P., &Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

[7] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, Z. Wang A novel feature selection algorithm for text categorization Expert Systems with Applications, 33 (2007), pp. 1-5.

[8] Lee, C., & Lee, G. G. (2006). Information gain and divergence-based feature selection for machine learning-based text categorization. Information Processing &Management, 42, 155–165.

[9] Chen, Y.-T., & Chen, M. C. (2011). Using chi-square statistics to measure similarities for text categorization. Expert Systems with Applications, 38, 3085–3090.

[10] Mulki, H., Haddad, H., Ali, C. B., &Babaoglu, I. OnIslemeninTurkceDuygusallıkAnalizineEtkisi Preprocessing Impact on Turkish Sentiment Analysis.

[11] Uysal, A.K., Gunal, S.: The impact of preprocessing on text classification. Inf. Process. Manage. 50(1), 104–112 (2014)

[12] Torunoglu, D., Cakirman, E., Ganiz, M. C., Akyokus, S., & Gurbuz, M. Z. (2011). Analysis of preprocessing methods on classification of Turkish texts. In International Symposium on

Innovations in Intelligent Systems and Applications (INISTA) (pp. 112–117).

[13] Toraman, C., Can, F., &Kocberber, S. (2011). Developing a text categorization template for Turkish news portals. In International symposium on innovations in intelligent systems and applications (INISTA) (pp. 379–383).

[14] Toman, M., Tesar, R., &Jezek, K. (2006). Influence of word normalization on text classification. In Proceedings of the 1st international conference on multidisciplinary information sciences & technologies (Vol. 2, pp. 354–358). Merida, Spain.

[15] Pomikálek, J., &Rehurek, R. (2007). The Influence of preprocessing parameters on text categorization. International Journal of Applied Science, Engineering and Technology, 4, 430–434.

[16] Duwairi, R., Al-Refai, M. N., &Khasawneh, N. (2009). Feature reduction techniques for Arabic text categorization. Journal of the American Society for Information Science and Technology, 60, 2347–2352.

[17] Wang, Andrew Jen. "Tongs: Tldr; opinion network guide system." (2017).

[18] Camacho-Collados, J., &Pilehvar, M. T. (2017). On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. *arXiv preprint arXiv:1707.01780*.

[19] https://github.com/hakkiyagiz/SIU2019 (Accessed September 2019).

[20] Zemberek. <http://code.google.com/p/zemberek/>(Accessed September 2019).

★★★