CNN vs. LSTM for Turkish Text Classification

Melih Yayla
Computer Engineering Department
Hasan Kalyoncu University
Gaziantep, Turkey
melihyayla@gmail.com

Mustafa Diyar Demirkol
Electrical-Electronics Engineering Department
Hasan Kalyoncu University
Gaziantep, Turkey
mustafadiyar.demirkol@gmail.com

Saed Alqaraleh
Computer Engineering Department
Hasan Kalyoncu University
Gaziantep, Turkey
saed.alqaraleh@hku.edu.tr

2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) | 978-1-6654-3603-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/INISTA52262.2021.9548407

***Abstract*—In this paper, the efficiency of two states of the art text classification techniques, i.e., Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) for supporting the Turkish text classification has been investigated. In addition, the effect of the main preprocessing steps such as Tokenization, Stop Word Elimination, Stemming, etc. has also been studied. Several experiments using "TTC-3600" dataset were performed, and it has been observed that both CNN and LSTM can efficiently support the Turkish language and can achieve quite good performance. Related to data preprocessing, results indicated that such a process improves the performance, however, for the Turkish language, it is preferred to exclude stemming. Also, by comparing the performance of feature extraction techniques for processing Turkish language, Word2Vec outperforms TF-IDF.**

***Index Terms*—Text Classification, Turkish Language, Convolutional Neural Networks, Long Short-Term Memory, Natural Language Processing.**

I. Introduction

In today's world, the possession of knowledge or information holds an important place for people, companies, or even states. However, the extraction of this information is quite an essential and hard task. To overcome this problem and to obtain the requested information, information retrieval (IR) systems were developed. In this paper, we are going to investigate two text classification techniques, i.e., Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), for supporting the Turkish text classification.

In general, text classification can be defined as the process of stating previously declared categories to text documents. Text classification can be exemplified with the classification of e-mail messages as spam or not. Another example is that it will automatically tag all incoming news on a subject for example "art", "football" or "movies". Text classification is also one of the most popular study topics in the field of Natural Language Processing (NLP), which aims to classify the tagged texts into the related categories (classes). Nowadays, Naive Bayes, Support Vector Machine [3], Neural Network [4] and K-nearest neighbor [5] are frequently used for text classification. However, the impressive performance archived by Neural Networks especially CNN and LSTM in fields such as image classifications, content-based image retrieval, self-derived cars, and many others fields, has attracted researchers to use such approaches for text processing tasks such as translations, classification, etc.

II. Related Work

In this section, we have summarized recent researches, developments, and solutions related to text classification. The study of Çelenli et al., aims to develop a centroid based classifier. In this study, documents are represented by vectors using the Paragraph Vector model (Doc2Vec). The results of their experiments indicate that using Distributed Bag of Words (DBOW) architecture with five epochs of classifiers paired with document embedding vectors obtains the best accuracy. Also, it has been observed that using more epochs decreases the classification accuracy of the Doc2Vec interestingly. On the other hand, using scarce data amount leads the Doc2Vec to outperform the SVM classifiers that use tf-idf representations [6].

Another study was done by Şahin, which compares the use of word2vec in the classification of seven different categories of Turkish texts with the classical bag of words (BoW) text representation. Here, each sample was expressed by a vector that has the average of the sample's words, then, SVM was used as the classifier. The experiments were conducted for different parameter settings of word2vec and its effect on classification success was examined. The study observed the accuracy of word2vec which is at the best-measured value was 0.92F is better than tf-idf weighted BoW method which is at the best-measured value of 0.89F [2].

The classification performance of heterogeneous classifier ensembles for Turkish and English languages was investigated by Kilimci et al. For this purpose, some base learners such as multinominal naive Bayes (MNB), support vector machine (SVM), multivariate Bernoulli naive Bayers (MVBN), convolutional neural network (CNN), and random forest (RF) were used. Here, to merge the determination of these base learners, both majority voting and stacking methods were used. Also, Word2vec and TF-IDF were used for feature representation. By applying base learners and heterogeneous ensemble systems with majority voting and stacking methods on 8 different datasets represented by TF-IDF or Word2vec, RF and CNN obtained the best results, and stacking outperforms majority voting [7].

Similar to the previous study, in [8], the effect of ensemble models while classifying Turkish texts using some classification algorithms such that naive bayes (NB), J48 – Decision Tree, K – Nearest Neighbor (K-NN), and support vector machine (SVM) as base classifiers was investigated. For the ensemble learning models. In this study, TTC-360 dataset which is consisted of 13 categories such as economy, sport, art, etc. was used. Results of [8], showed that base classifiers with Boosting and Rotation Forest ensemble models were able to achieve the best accuracy rate.

On the other hand, Torunoglu et al. [9] studied the effect of different preprocessing steps on Turkish texts classification. For preprocessing, stemming, stop word filtering and word weighting steps were applied. For the classification, Naïve Bayes, Naïve Bayes Multinomial (mnNB), Support Vector Machines (SVM), And K-Nearest Neighbor were used. According to their results, stemming has the lowest impact on text classification. However, they stated that stemming is more appropriate for information retrieval tasks.

III. Mechanism of Text Classification

In general, the following components can be considered as the main ones of Text Classification.

A. Text Gathering

This step is working on collecting the samples and datasets that can be used for building the classification system and also for investigating the performance of such a system. In this study, as we aim to process the Turkish language, the “TTC-3600” Turkish data set that was constructed using 3600 Turkish news and articles and humanly annotated to the following topics Ekonomi (Economy), Kültür-Sanat(Art and Culture), Sağlık (Health), Siyaset (Politics), Spor (Sport), Teknoloji (Technology), where each one has 600 articles are used.

B. Text Pre-Processing

In general, text preprocessing is one of the important steps in information retrieval and analysis systems. It basically prepares the text into more useful, workable, and proper form. Turkish language belongs to a branch of the Altai language family, and it is an additive language, in which words are made and withdrawn by suffixes. Also, Turkish language has some specific characteristics such as 1) There is no masculinity or femininity feature like in Arabic and German languages. 2)The names which came after the numbers do not take the plural suffix. 3) There are thickness-thinness and flatness-roundness harmony in Turkish. According to the first harmony, vowels in a word are either thick or thin, and according to the second harmony, they are always flat or round. 4) The consonants f, j, and h do not exist in the ordinal Turkish words, while they exist in words that were included from other languages. 5) The number of consonants that can be found at the beginning of the word is limited. These consonants are “b, c, d, g, k, s, t, v, y”. 6) In the case the consonant c is at the beginning of the word, it will be changed to another consonant ç. 7) The n consonant letter contains only ”what” and its derivative words: what, when, why, how, and where(In Turkish, they mean ne, ne zaman, neden, nasıl, and nerede respectively). 8)The consonant p is found at the beginning of some words was obtained by changing ”b”. 9)In Turkish alphabet, there are no “x,w,q” letters whereas there are the letters “ç, ğ, ı, ö, ü” different from English alphabet. 10) Turkish words are read as written [11].

Some of the text preprocessing steps that are investigated in order in this study are:

Tokenization: The first step of preprocessing is tokenization, i.e., the input text is turned into word tokens [14].

Stop Word Elimination: Stop words can be defined as the most used words in a language. However, most of the stop words have no meaning by themselves. If these words are eliminated, it would be easier to use the most meaningful and semantic words. For the stop word elimination, there are several libraries like *sklearn* that can be used to eliminate such words [10].

Lowercasing: One of the common text preprocessing techniques is lowercasing all characters in the text. This method helps us to increase the stability of inevitable outcomes. It is an appropriate technique for most of the NLP issues [10].In other words, as shown in Table I, lowercasing basically creates a standard for the datasets. For example, it assists search engines to create search indexes in a standard way which improves the effectiveness [10].

TABLE I: Lowercase Example.

Raw	Lowercased
İsTanBuL İSTANBUL İsTaNbUl	istanbul
KiTAP KitAp KiTaP	kitap

Stemming: Another text preprocessing technique is stemming. Stemming is basically a method that finds the root of the words [10]. Some different techniques are used to perform this process. For the Turkish language, the most common algorithm is the SnowBall algorithm. In this algorithm, there are some rules that the coder has followed due to the Turkish language morphology [12]. The rules are;

- Turkish language has only one affix type which is the suffix.
- In Turkish, it is not possible to have a plural suffix after a possessive suffix.
- In Turkish, a suffix can have more than one allomorph to have sound harmony.
- In Turkish, each vowel expresses a different syllable.
- In Turkish, most of the monosyllabic words are the stem.
- In Turkish, if a word possesses nominal verb suffixes, it comes at the end of the word.
- In Turkish, a suffix can be treated as a noun suffix and a nominal verb suffix [12].

The different sound structures of a morpheme (although it comes to mind at the first moment as a word, we can say that it is a fragmented form of the word in a sense) is called

TABLE II: Suffix Allomorphs [12].

Letter	Allomorph
U	I,i,u,ü
C	c,ç
A	a,e
D	d,t
I	ı,I

allomorph. There are some different versions of allomorphs in Table II.

TABLE III: Derivational Suffixes [12].

a/a	Suffixes
1	-lUk
2	-CU
3	-Cuk
4	-lAş
5	-lA
6	-lAn
7	-CA
8	-lU
9	-sUz

Derivational suffixes create nouns like the suffixes -tion or -ness in English. Different types of suffixes are shown in Table III.

TABLE IV: Nominal Verb Suffixes [12].

a/a	Suffix
1	-(y)Um
2	-sUn
3	-(y)Uz
4	-sUnUz
5	-lAr
6	-md
7	-n
8	-k
9	-nUz
10	-Dur
11	-cAsInA
12	-(y)DU
13	-(y)sA
14	-(y)mUş
15	-(y)ken

Also, there are some verb suffixes that are used to create time tenses. Some of the verb suffixes are shown in Table IV.

On the other hand, there are some noun suffixes. These suffixes change words and meanings. Some of the noun suffixes are shown in Table V.

Normalization: One of the important processes of text preprocessing is the normalization step. Normalization is a method that transforms a text into a standard form [9]. Normalization is substantial for the text processes, especially in informal writing where miswrites, abbreviations occur too much. This process affects the analysis of text dramatically [9], where people are generally using the letters "c, g, i, o, u" instead of dotted ones "ç, ğ, ı, ö, ü". This situation may cause some problems when classifying such samples. Also, most people do not use vowels when they are texting or posting something. This is another problem that it is preferred to

TABLE V: Noun Suffixes [12].

a/a	Suffixes
1	-lAr
2	-(U)m
3	-(U)mUz
4	-(U)n
5	-(U)nUz
6	-(s)U
7	-lArl
8	-(y)U
9	-nU
10	-(n)Un
11	-(y)A
12	-nA
13	-DA
14	-Nda
15	-Dan
16	-nDAn
17	-(y)lA
18	-ki

be solved. There are some text normalization methods such as dictionary mappings, statistical machine translation, and spelling – correction based approaches that can be used in such case [10]. For the Turkish language, there is an open-source library named Zemberek. This library is using a spelling – correction based approach to check if a word is correctly written and gives proposals for a word. In other words, Zemberek uses some heuristics look-up tables and language models for text normalization [13]. It is worth mentioning that based on our ongoing experimental work, it has been noticed that word correction in general, and using the Zemberek tool, in particular, can improve the overall performance of Turkish text classification systems by approximately 5%.

C. Feature Extraction

In the processing of texts, the words in the text show categorical and discrete features. It is important to encode such data to use it in the preferred algorithms. The process of subtracting a list of words from the text, and mapping them to the feature set which can be used by a classifier is called text feature extraction. Different types of feature extraction methods are mentioned below.

1) Traditional Methods: Count Vectorization, TF-IDF Vectorizer, and HashingVectorizer are the traditional methods of the feature extraction for Text Classification [15]. In this study, TF-IDF which is considered as the state of the art traditional feature extraction method is used.

TF-IDF Vectorizer: Term Frequency can be explained as the number of appearances of a word in the related text document [15]. Equation 1 can be used as the calculation of Term frequency.

$$TF(w_i) = \frac{O(w_i)}{N} \tag{1}$$

$O(w_i)$ represents the occurrence of the ith word, N is the total number of words that existed in the used vector.

Inverse document frequency (IDF) measures how important a term is. In general, it is proved that stop words appear in most texts frequently but have little importance. Hence, in the case of IDF, the highest score is assigned to the rare words, and the low score is assigned to the frequent words [15]. Inverse document frequency can be calculated using Equation 2.

$$IDF(w_i) = log\frac{N}{T} \tag{2}$$

T represents number of documents that includes i[th] word. To get the overall score, i.e., TF-IDF, as shown in Equation 3, IDF is multiplies by the TF [15].

$$TF - IDF(w_i) = TF(w_i) * IDF(w_i) \tag{3}$$

2) Word Embedding Methods: Word embedding is a natural language modeling technique that matches words or expressions to an equivalent numerical vector(s). This process helps machine learning methods to understand the given inputs by contributing the vector representation of the inputs. Also, this method has some other advantages like reducing the dimension of words and prevent similarity of contextual words [20]. Word2vec, GloVe, and Fasttext are examples of the Word embedding approaches. Word2vec which is considered the most suitable option for the Turkish language as stated in [1] is used in this study.

Word2vec: It is an unsupervised and prediction-based model that expresses words in vector spaces. It was invented in 2013 by Google researcher Tomas Mikolov and his team. Word2vec has two sub-methods: CBOW (Continuous Bag of Words) and Skip-Gram. Both methods are similar in general [23], and its output is represented by Equation 4.

$$Word2vec(W_i) = [F_1, F_2, F_3, ...F_m] \tag{4}$$

Where, in our case, m is set to 300, and F is a float number.

D. Classification

A document's automatic classification according to predefined categories is currently attracted researchers' attention. Unsupervised, supervised and semi-supervised are the three main methods for text classification. In the last decade, the automatic text classification task has some significant improvements using artificial intelligence algorithms such as Neural Networks, Bayesian classifiers, Decision Tree, support vector machines (SVMs), etc. In this study, the performance of CNN and LSTM was investigated, and their details are summarized below.

1) Convolution Neural Network: Convolutional Neural Network which is a kind of Multilayered Perceptron is a feed-forward neural network, was inspired by the visual center of the animals [18] and its mathematical convolution process can be considered as the response of a neuron to stimuli from the stimulus field [17], [19]. The architecture of CNN sets in one or more convolutional layers, sub-sampling layers pursued by fully connected one(s) [16]. In convolutional layers, the input is filtered and feature maps are obtained. In the sub-sampling layers, feature maps are sampled. Finally, the fully connected layer works on generating output based on the representation(vector) from the previous layers. In other words, each layer produces some features based on the result of the previous layer(s) and the overall structure(model) can learn the feature hierarchy by combining and training all layers. The aim here, starting with the low-level details, is to achieve effective learning up to high-level details [16].

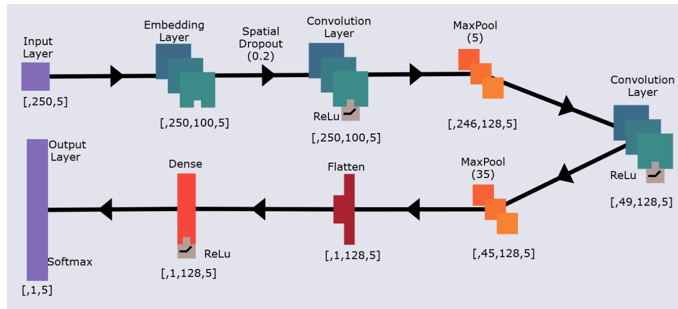

Fig. 1: Structure of the used CNN

2) Long Short-Term memory: Recurrent Neural Network (RNN) is a concomitant and collaborate neuron networks, where neurons are connected with each other by weights. These kinds of neural networks are very helpful in the event of inputs of changing sizes, self-acting translation, self-acting pattern recognition, etc. The orientation of transmission of knowledge in These kinds of neural networks are bidirectional, which withholds the order of the data, and can connect with high sequence inputs as such network is grounded on a loop by courtesy of interior memory. In 1997, Hochreiter and Schmidhuber proposed a new method named Long Short Term Memory (LTSM)that can be defined as an extension(improved version) of RNN. LTSM can deal with the problem of fading gradient by the virtue of its memory that enables deleting, writing, and reading the info through three gates; Input gate that permits or obstruct the updates; Forget gate that deactivates an insignificant neuron depending on weights learned from the algorithm; and output gate which is the control gate of neurons [20] [21] [22].

E. Test & Result

In this study, multiple experiments were performed using the previously mentioned dataset "TTC-3600", In the first experiment, a comparison between the studied feature extraction methods, i.e., Word level TF-IDF, N-gram level TF-IDF, Characters level TF-IDF, and the word2vec Word embedding, to find out the most suitable approach for the Turkish language was performed. In the second one, the effect of text preprocessing on the performance of both CNN and LSTM was investigated. Finally, the comparison between the two states of the art CNN and LSTM classification approaches is studied as well.

Experiment 1: Comparing the Performance of Feature Extraction Techniques for Processing Turkish Language

In this experiment, the accuracy of feature extraction techniques was measured. For this experiment, four feature ex-

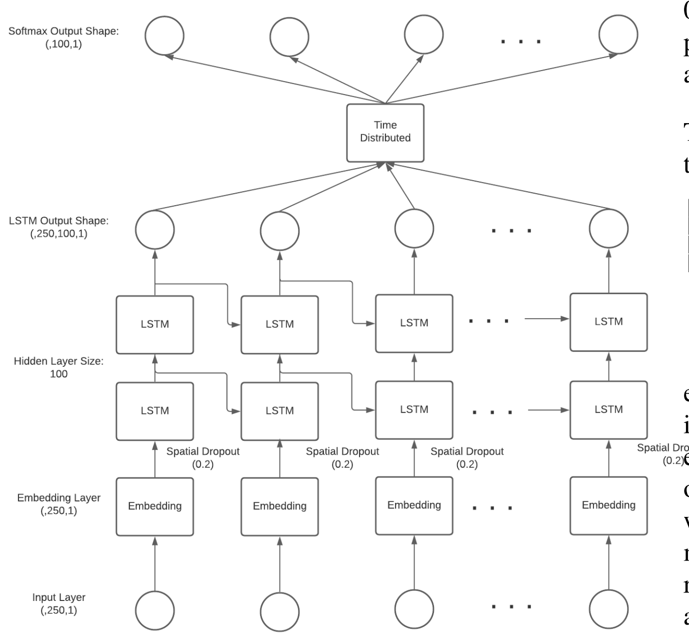

Fig. 2: Structure of the used LSTM

traction methods which are Word level TF-IDF method, N-gram level TF-IDF method, Characters level TF-IDF method, and Word2vec word embedding were used. As a first attempt, Word level TF - IDF method was used. After applying this method, CNN had 0.2 accuracy and LSTM had 0.178 accuracy. After that, N-gram level TF-IDF method was used. By using this approach, CNN and LSTM had the same accuracy which is 0.2. As a third attempt, the characters level TF- IDF method was used. By applying this approach, CNN had 0.178 accuracy whereas LSTM had 0.26 accuracy. As a last attempt, Word2vec method was used. After applying this method, CNN had 0.861 accuracy and LSTM had 0.822 accuracy. Overall, as shown in TABLE VI, it is clear that CNN with Word2vec method has the highest accuracy.

TABLE VI: The accuracy of feature extraction with the pre-processing operations.

Approach	Accuracy			
	Word level TF-IDF	N-gram level TF-IDF	Characters level TF-IDF	Word2Vec
CNN	0.2	0.2	0.178	0.861
LSTM	0.178	0.2	0.26	0.822

Experiment 2: The Effects of Pre-Processing the Turkish Text

In the second experiment, the effect of pre-processing on the accuracy of the classification process was measured. As shown in TABLE VII, in the first step, none of the pre-processing methods were used. According to this process, CNN had 0.896 accuracy and LSTM had 0.8 accuracy. Secondly, full pre-processing steps were used. After applying full pre-processing, CNN had 0.913 accuracy whereas LSTM had 0.9 accuracy. As the last part, pre-processing without stemming was also investigated and CNN had 0.922 accuracy while LSTM had 0.91 accuracy. After all these attempts, it is clear that pre-processing without stemming allows CNN to have the highest accuracy.

TABLE VII: The accuracy of classification with and without the pre-processing operation.

Approach	Accuracy		
	Without Pre-processing	Full Pre-processing	Pre-processing Without Stemming
CNN	0.896	0.913	0.922
LSTM	0.8	0.9	0.91

Experiment 3: CNN vs. LSTM

In this experiment, based on the result of the previous two experiments, two systems for CNN and LSTM have been implemented. In addition, these systems use Word2vec feature extraction technique, which was found to be best among the other studied methods. Also, pre-processing without stemming was used which provided the best accuracy among the other methods. To obtain more accurate results, this experiment was repeated in five iterations. The results of all the iterations and their average are shown in Table VIII. Overall, LSTM had an accuracy of 0.9294 whereas CNN had an accuracy of 0.9278. However, even that the difference between the obtained accuracies is not much, the execution time of CNN is almost 1/3 the time of LSTM.

TABLE VIII: The Accuracy Comparison of CNN and LSTM.

Iteration	Accuracy	
	CNN	LSTM
1st	0.93	0.926
2nd	0.935	0.939
3rd	0.922	0.917
4th	0.926	0.93
5th	0.926	0.935
avg	**0.9278**	**0.9294**

Overall, based on our results and the results obtained by Doğru H.B. et al. [24], where the two studies were conducted using the same dataset, i.e., ”TTC-3600”, using deep learning such as CNN and LSTM improves the performance of Turkish text classification. In more detail, the traditional methods such as Support Vector Machines(SVM), Naive Bayes, and Random Forest were able to archive an accuracy of 86.39%, 85.00%, 84.17% respectively. Hence, on average CNN increases the accuracy by at least 6.39%, similarly, LSTM increases the accuracy by 6.55%.

F. Conclusion and Future Work

In this study, various experimental examinations were performed to observe the effect of different text classification steps and methods on the accuracy rates of Turkish text classification. The TTC-3600 dataset, which contains the news collected from six different agencies and news portals, and is also available online is used.

Also, two popular classifiers, CNN and LSTM, are used. To find the best accuracy rates, different versions of pre-processing and feature extraction methods are used with the mentioned classifiers. First of all, different feature extraction methods, which are word level TF-IDF, N-gram level TF-IDF, characters level TF-IDF, word2vec, are used. After this experiment, it was found that word2vec method has the best accuracy rate. After that, the effect of pre-processing methods is evaluated by calculating the accuracy of classifiers with pre-processing, without pre-processing, and pre-processing without stemming, it is clear that pre-processing without stemming has the best accuracy rate.

As the last step, five iterations were made with CNN and LSTM with pre-processing without stemming and word2vec method to find the best approach. As a result, it is seen that the average accuracy of CNN is 0.9278, and the average accuracy of LSTM is 0.9294. It is observed that the accuracies are closed, but related to the execution time of the CNN requires almost 1/3 the time of LSTM.

For future work, our study can be extended by using some state-of-the-art word embedding methods like ELMo and XL-Net. Also, for the system classifier, some other artificial intelligence algorithms such that Support Vector Machine (SVM), Decision Tree, Bayesian Classifier can be integrated with deep features(i.e., extracting the output of selected layer(s) of the used deep learning model).

REFERENCES

[1] Alqaraleh, Saed, and MERVE IŞIK.(2020). Efficient Turkish tweet classification system for crisis response. Turkish Journal of Electrical Engineering & Computer Sciences 28, no. 6 (2020): 3168-3182.

[2] Sahin G., “Turkish document classification based on Word2Vec and SVM classifier”, 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018, Antalya, Turkey.

[3] Sebastiani, F..” Machine Learning in Automated Text Categorization”, ACM Computing Survey. pp. 1-47, 2002

[4] S. N.Sivanandam, S. N. Deepa “Principles of Soft Computing”

[5] Gonde Guo, Hui Wang, David Bell,Yaxin Bi and Kieran Greer “KNN Model Based Approach in classification. pp .986-996 , 2003

[6] Celenli H. I. , Ozturk S. T. , Sahin G., Gerek A., GANİZ M. C., “Document Embedding Based Supervised Methods for Turkish Text Classification”. 3rd International Conference on Computer Science and Engineering (UBMK), 20 - 23 September 2018, p.477-482, Sarajevo, Bosnia-Hercegovina.

[7] Kilimci, Z., Akyokus, S.,: Deep Learning- and Word Embedding-Based Heterogeneous Classifier Ensembles for Text Classification. Complexity 2018, 1-10 (2018).

[8] Kılınç, D . (2016). The Effect of Ensemble Learning Models on Turkish Text Classification. Celal Bayar University Journal of Science , 12 (2).

[9] Torunoğlu, D., Çakırman, E., Ganiz, M.C., Akyokuş, S., Gürbüz, M.Z. (2011). Analysis of Preprocessing Methods on Classification of Turkish Texts. INISTA 2011, June, 2011, Istanbul, Türkiye

[10] KDnuggets. (2019). All you need to know about text preprocessing for NLP and Machine Learning - KDnuggets. [online] Available at: https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html [Accessed 8 Nov. 2019].

[11] Eteration. (2019). Türkçe Doğal Dil İşlemede Zemberek – eteration. [online] Avaliable at: https://www.turkedebiyati.org/turkcenin-ozellikleri.html

[12] O. Tunçelli, (2019). Turkish Stemmer for Python – GitHub. [online] Avaliable at: https://github.com/otuncelli/turkish-stemmer-python

[13] TH. Tuna, (2019). Turkish Text Normalization – GitHub. [online] Avaliable at: https://github.com/ahmetaa/zemberek-nlp/tree/master/normalization

[14] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.

[15] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436–444, 2015.

[17] K. . N. Fukushima, “A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.,” Biol. Cybern., vol. 36, no. 4, pp. 193–202, 1980.

[18] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” J. Physiol., vol. 195, no. 1, pp. 215–243, Mar. 1968.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] Naili, Marwa and Habacha, Anja and Ben Ghezala, Henda. (2017). Comparative study of word embedding methods in topic segmentation. Procedia Computer Science. 112. 340-349. 10.1016/j.procs.2017.08.009.

[21] Rhanoui, Maryem and Mikram, Mounia and Yousfi, Siham and Barzali, Soukaina. (2019). A CNN-BiLSTM Model for Document-Level Sentiment Analysis. Machine Learning and Knowledge Extraction. 1. 832-847. 10.3390/make1030048.

[22] Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735–1780.

[23] Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, G.s and Dean, Jeffrey. (2013). Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems. 26.

[24] Doğru, Hasibe Büşra and Hameed, Alaa and Tilki, Sahra and Jamil, Akhtar. (2021). Comparative Analysis of Deep Learning and Traditional Machine Learning Models for Turkish Text Classification.