

Novel Turkish Sentiment Analysis System using ConvNet

Saed Alqaraleh

Computer Engineering Department, Hasan Kalyoncu University, Turkey

Abstract: *In this paper, an efficient model for the Turkish language sentiment analysis has been introduced. As Turkish is an agglutinative language, which requires spatial processing, an efficient pre-processing model was also implemented and integrated as a part of the developed system. In addition, the Deep Convolutional Neural Networks (ConvNet) have been integrated to build an efficient system.*

Several experiments using the “Turkish movie reviews” dataset have been conducted, and it has been observed that the developed system has improved the sentiment analysis system that supports the Turkish language and significantly outperforms the existing state-of-the-art Turkish sentiment analysis systems.

Keywords: *Sentiment analysis, opinion mining, text classification, turkish language, convolutional neural networks, natural language processing.*

Received January 8, 2020; accepted February 21, 2021

<https://doi.org/10.34028/18/4/7>

1. Introduction

Due to the rapid growth of the Internet, available data in the form of text on social media, e-commerce websites, blogs, and other similar sources, where a large portion of this data can have opinions and sentiments in terms of user’s review about some topics such as products, people, government policy, research, movies, hotels, etc., Nowadays, companies increasingly use social media like Facebook, Instagram, and Twitter to market their products and services [26]. As a result, we are allowing (users) to give their opinions, write reviews and share thoughts about any available products and/or services. This leads most companies to adjust their products based on user’s opinions.

Sentiment analysis or opinion mining is a field of Natural Language Processing (NLP) that deals with analysing opinions about products, services, and even people. In general, the Lexicon-based, machine learning, and hybrid approach are the main approaches used in Sentiment Analysis (SA) tasks [21, 24]. The lexicon-based approach uses seed words or phrases, which is a set of predefined words, for identifying the Semantic Orientation (SO) of words to classify the text as positive or negative. The machine learning approach can be supervised or unsupervised learning, where the first uses a labelled dataset for training the classifier. In contrast, unsupervised machine learning methods use an unstructured dataset. In addition, the hybrid approach combines both lexicon-based and machine learning approaches [12].

1.1. SA Challenging Problems

Unfortunately, most of the existing SA techniques and studies achieved an impressive result for a limited number of languages such as English and Chinese. It is obvious that the SA system mechanism changes from a language to another. For example, it has been found that sentence-based systems are efficient for English, however, character-based is more suitable for Chinese. In addition, most SA resources such as polarity lexicons, parsers are available and well established for English. On the other hand, due to lack of resources and the complexity of some languages especially the agglutinative morphology languages such as Turkish and Korean, very few researches can be found for such a language, and even some of these researches were conducted using English resources. For instance, in [10], the English SentiWordNet [8] was used as a part of a system that classifies the French movie reviews. However, to use the SentiWordNet the authors had to translate the review’s French words to English. Also, in [28], a comparison of lexicon-based and machine learning-based sentiment analysis methods on Turkish social media was performed. The used lexicon was formed with an English lexicon that has been translated to Turkish.

In recent years, sentiment analysis related to the Turkish language attracted research interest, however, there is still a need for intensive investigations to efficiently supporting the Turkish language. In Table 1, the challenging problems related to Turkish language SA are summarized.

Table 1. Main challenging problems related to Turkish language SA.

Problem	Details
Agglutinative language	This means that by adding some suffixes to a root word new and arbitrarily long words can be generated. In addition, some suffixes may change the semantic orientation of the word.
Alphabet	Turkish has some letters that do not exist in the English alphabet “ğ”, “ç”, “ı”, “ö”, “ş”, and “ü”. In general, people tend to substitute these letters by the closest ASCII characters, for instance, “ş” and “ö” is replaced by “s” and “o” respectively. Therefore, processing Turkish is more risky to be defeated by erroneous writings.
Negations	In Turkish, the words can be negated using many ways, that each changes sentiment polarity. Words can be negated with 1) the affixes me/ma or 2) siz/siz or 3) using a separate word such as “değil” or “yok”.

1.2. Background on ConvNet

Over the last decade, Convolutional Neural Network (CNN) was able to achieve state-of-the-art performance in many tasks such as data mining (ex. translation), classification, object detection, etc. In general, CNN composed of a number of layers such as convolution, which mainly aims to detect local conjunctions of features from the previous layer and mapping their appearance to a feature map, pooling layer that responsible for reducing the size of the activation maps, and ReLu layer, that can be considered as special implementation aims to combine non-linearity and rectification layers. Currently, multiple available CNN models can be considered as good choices for improving computer-based systems. AlexNet [19], VGGNet [25], GoogleNet [27], and ResNet [11] are some of the successful models that are able to achieve performance close to the human-level in many fields.

2. Related Work

In this section, we have summarized recent researches, developments, and solutions related to the proposed system. In [22], a comparative study of results obtained by applying the Naive Bayes (NB) classifier and Support Vector Machine (SVM) classifier was presented. These algorithms have been used to classify labelled sentimental movie reviews. As a result of this study [22], it has been found that SVM can be considered as one of the best ML approaches for SA tasks on the English text. In [23], an approach that uses both machine learning and the SentiWordNet approach for sentiment classification and sentence relevance score for opinion summarization were presented. In this work, different supervised machine learning classifiers like NB, SVM, and DT were investigated, and their performance was compared with SentiWordNet approach. Overall, the results of [23, 8], showed that the supervised ML algorithms can outperform the SentiWordNet based approach for sentiment classification. In [18], a combined method using NB and SVM classifiers was proposed to

improve the performance of SA classification. Their experimental results showed that the proposed combined method outperformed the NB classifier alone. However, it has achieved almost the same as the performance of the SVM classifier alone.

In the last few years, investigating the deep learning features for the application of the Language processing problem has attracted many researchers. It has been shown in [1, 2, 9, 13, 14, 15, 30] that deep convolutional neural networks can perform and improve multiple operations in language processing, such as autocorrecting, autocompleting, and many other tasks. In [9], a library named AllenNLP, for applying deep learning methods to NLP research was introduced. The main advantage of this library, it has what they call “easy to-use” command-line tools, which as they believe can increase the rate of research experimentation and the sharing of NLP components. A CNN architecture, named as Dynamic Convolutional Neural Network (DCNN), for sentiment classification of sentences was presented in [15]. Overall, this model inspired most of the later CNN works for Sentiment Classification. On the other hand, this model can only deal with sentence classification levels.

In [30], the researchers have proposed a CNN model for Chinese language SA. Since Chinese words can be written in different ways, and they have different meanings, two different sentiment lexicons were used to help with sentiment associations. In addition, a big sentiment dataset from hotel reviews and chatting weblogs was collected. As a result, the model of [30] improves the performance as compared against Logistic Regression, SVM, and NB classifiers.

In [14], the performance of the deep CNN character-level and shallow CNN word-level was studied. Eight datasets were used in this research and their results show that the shallow word-level needs more parameters, thus it requires more storage than the character-level. However, they also found that the character-level approach is complex and much slower than the shallow word-level.

In the following, some recent studies related to the sentiment analysis in Turkish are summarized. In [7], the SVM classifier and n-grams were used for classifying Turkish reviews. In addition, the effect of part-of-speech tagging, spell-checking, and stemming was also investigated in [7]. In [3], aspect, sentence, and document levels of sentiment analysis were investigated for supporting Turkish. In addition, some linguistic issues such as conjunction and intensification were also studied. Quite good work has been performed in [16, 17] for analysing the Turkish political news using Naive Bayes, maximum entropy, SVM, and the character n-gram. In addition, in [17], the work of [16] was further improved by implementing transfer learning into the existing framework.

Recently, an important Turkish SA resource has been introduced, i.e., SentiTurkNet [4], that contains 15,000 synsets with their Part-of-Speech Tagging's (a (adjective), n (noun), v (verb) and b (adverb)), and three associated polarity values that each has a score (positive, negative and neutral/objective).

3. The Developed System

Figure 1 shows the main components of the developed system, and its details are explained below.

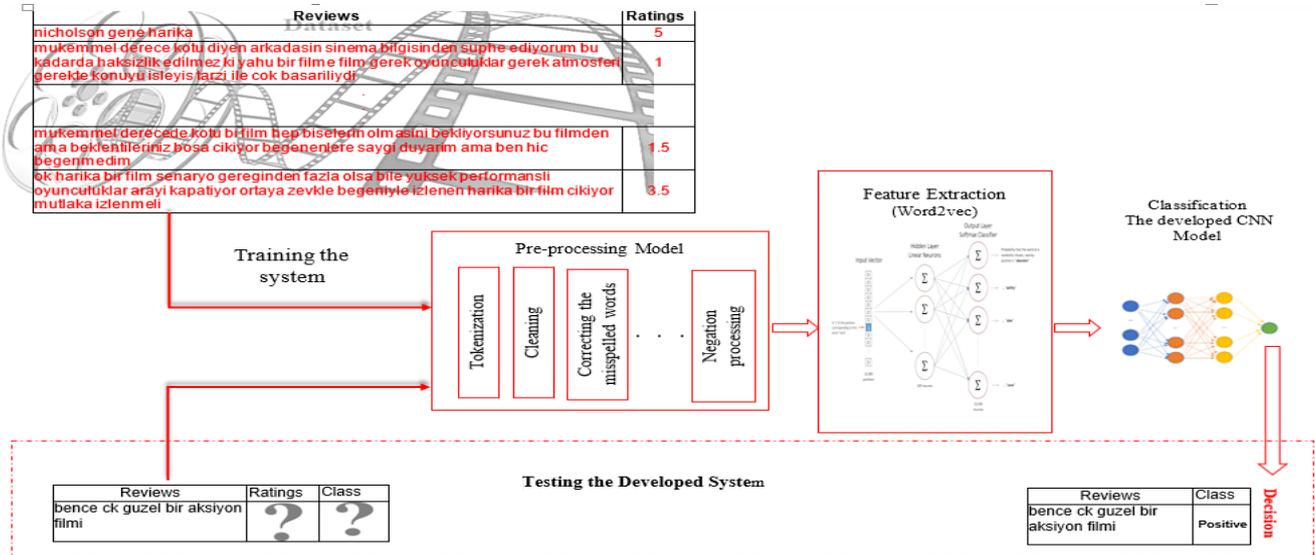


Figure 1. The main components of the developed system.

3.1. Pre-Processing

The quality of the used data is essential for building an efficient system. Hence, noisy and unwell prepared data can badly degrade the overall performance. The developed pre-processing model, in summary, consisted of:

1. Cleaning the data: This step work on removing text that has no sentiment and does not help the SA processes, for instance, special characters, stop words, URLs, punctuation, and irrelevant text must be removed. In addition, the effect of more steps such as case folding, i.e., all text is converted to the same shape (upper or lower), converting numbers to word equivalents, etc., has been studied as well.
2. Correcting the typos and misspelled words: such mistakes can change the meaning of the whole sentence. Related to the Turkish language, this step is done using the de-ASCIIfication, which works on converting the American Standard Code for Information Interchange (ASCII) English characters to their Turkish equivalents.
3. Stemming: As the Turkish language has a very rich morphological structure and multiple different words can be produced from a single stem by appending a sequence of suffixes. For instance, the verb “anlamak” (understand). When we add “-t” at the end of the verb stem “anal-”, it becomes “anlat-(mak)” which means “tell”. Then, if we add suffix “-il”, it turns into “anlatıl-(mak)” which means “being told”. In addition, Turkish has over fifty suffixes that can be affixed to verbs [8], where multiple suffixes can be used at once. For example,

the word kalamıyormuşsunuz means “You couldn't stay”. The stem of the word is “kal-” and it takes 5 different suffixes “-a” “-ma” “-yor” “-muş” “sunuz”. In this paper, the Turkish stemming algorithm developed in [8] has been integrated into the developed approach.

4. Processing negation: Related to the Turkish language, this step needs extra treatment as the negation can be expressed within the word with affixation rather than a separate individual word [7, 29].

3.2. Feature Extraction

In this step, we need to map the textual data into a real-valued vector. This can be done using the vector space model, which is an algebraic model that converts the text into a vector of words and then transform the words vector into a numerical format. In this paper, the Word2Vec word embedding pre-trained using one million Turkish Wikipedia and crawled documents has been used. As Word2vec has a two-layer neural network, the collection of Turkish movie reviews will be its input, and each word in the collection will be represented by feature vectors as shown in Equation (1).

$$\text{Word2Vec}(\text{word}_i) = [f_1, f_2, f_3 \dots f_n] \quad (1)$$

Where n is equal to 300, and f is a float number. Hence, each word is represented by a vector of 300 float numbers. Then each input is represented also 300 float numbers obtained by finding the average of its words.

3.3. Classification

In this paper, a novel system that can efficiently process the Turkish language has been introduced. This system was developed after an intensive investigation to find the most efficient layers for building our model.

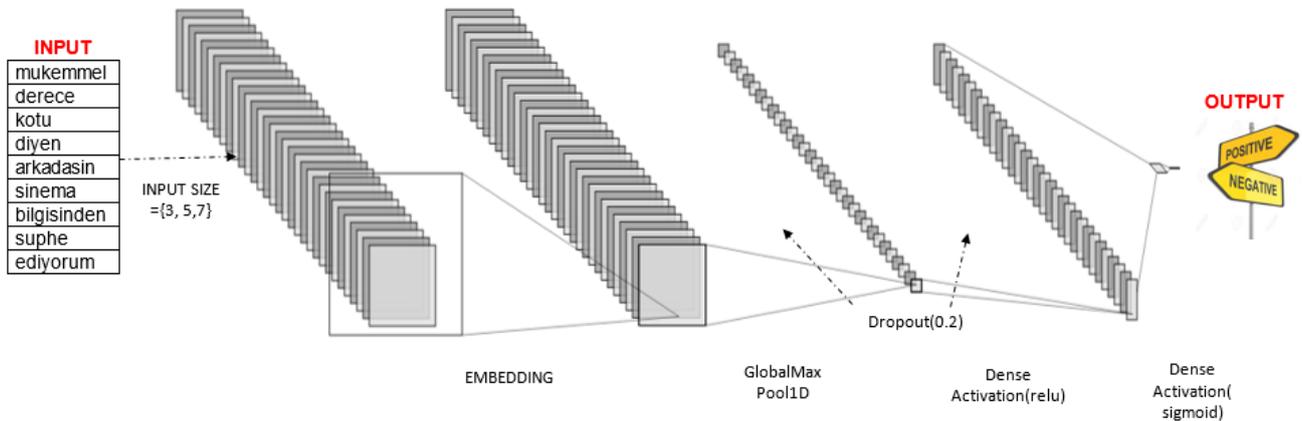


Figure 2. A visualization of the developed CNN architecture.

- a) The first used layer which mainly developed for natural language processing is the embedding layer. This layer can efficiently preserve the word’s contextual similarity, and words with similar meanings will have almost a similar representation.
- b) Pooling layer which mainly aims at reducing the size of the activation maps by down sampling the feature maps. Some of the very frequently used pooling methods are average, max, and hybrid.
- c) Then, to overcome and avoid the system overfitting, the Dropout layer can be used. In general, it performs the function of neglecting some randomly selected neurons. In this case, the neglected neurons need to be predicted, which provides multiple representations for the neural network and reduces the sensitivity to weights.
- d) Non-linearity layer: In general, artificial neural networks are designed as universal functions that need to be able to calculate and learn any function. Non-linear activation functions enable networks to learn more strongly. In this work, two of the most popular activation functions were used:
 1. Sigmoid Function, which is a nonlinear activation function that has the following two different Functions
 - a. The Uni-Polar Sigmoid. This function is particularly useful in networks that are trained by back propagation algorithms.
 - b. The Bi-Polar Sigmoid which was found to be more flexible.
 2. Rectified Logical Unit (RELU): This function can be considered as a special implementation that aims to combine non-linearity and rectification layers.

As shown in Figure 2, the developed approach consists of a set of layers such as the embedding, global max pool, dense, etc., more details about the main components (layers) of the developed CNN model are summarized below.

4. Experimental Study

The performance of the developed SA system and the pre-processing model were investigated through multiple experiments in this section. To ensure the robustness of the developed system, multiple databases, in which its details are shown in the following sub-section, have been used in this study.

4.1. Datasets and Evaluation

In this work, the last version of the Turkish movie reviews dataset introduced in [6], which is publicly available and composed of 34990 positive and negative movie reviews in Turkish is used. In more detail, the movie reviews were collected from “Beyazperde”, which is a movie site that allows users to write reviews about movies and give ratings of 0 to 5. It is worth mentioning that in our work, reviews of ratings >3 are taken as positive reviews; reviews of rating <3 are considered negative, and reviews with ratings equal to 3 are considered as neutral/objective. In addition, we have reconstructed five datasets from this dataset, the first dataset contains 5000 movie reviews, where 2500 is positive and 2500 is negative, the second dataset contains 10000 reviews (5000 is positive and 5000 is negative), the third dataset contains 15000 reviews (7500 is positive and 7500 is negative), the fourth dataset contains 25000 reviews (12500 is positive and 12500 is negative) and the fifth dataset contains all the reviews, i.e., 34990. It is worth mentioning that to ensure the robustness of the results, the 10-fold cross-validation was used and all the following experiments were performed using all the reconstructed datasets.

Related to the evaluation, we have used the following main standard metrics that can be used for evaluating classification systems:

1. Accuracy, which is the ratio of the total number of correctly classified movie reviews divide by the total number of reviews and calculated using Equation (2).
2. Precision, which is the ratio between the correct predictions and the total predictions, and can be calculated using Equation (3).
3. Recall, which is the ratio of the correct predictions and the total number of correct items in the dataset (Equation (4)).
4. F1 score, which is a weighted average of Precision and Recall, that considers both false positives and false negatives into account. F1 can be calculated using Equation (5).

$$\text{Accuracy} = \frac{TP + TN}{\text{Total}} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{F1Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (5)$$

4.2. Experiments and Results Analysis

Section #1: Turkish Language Pre-processing and Model's Parameters Specification

Related to the developed approach, it is important to note that for such a model, it is very critical to set the value of the following parameters:

1. The best window size of the input, i.e., the size of the used filters.
 2. The number of filters. In this experiment, the values 1, 3, 5, and 7 were tested for window size. While, 32, 64, 128, and 256 values were used for the number of the filter. In addition, ensuring the quality of both training and testing data is an essential request for building a system that can work efficiently in real-time. In this section, these three factors were investigated in detail.
- *Experiment 1: Investigating the Effect of the Input Window Size.*

In this experiment, the effect of changing the number of words in the processed text window was investigated. Based on our preliminary experiments, the size was set to 1, 3, 5, and 7. In more detail, the proposed CNN model was used in this experiment using all predefined values. As shown in Table 2, almost all tested values achieved the same performance, however, it is worth mentioning that increasing the window size has significantly decreased the processing time.

Table 2. The Effect of Changing the size of the used filters (Window Size).

Evaluation Matrix	Dataset	Window Size			
		1	3	5	7
Accuracy	1 st Turkish dataset	81.68	80.47	80.63	79.86
	2 nd Turkish dataset	83.84	80.02	79.90	80.00
	3 rd Turkish dataset	77.58	78.90	77.58	78.94
	4 th Turkish dataset	78.99	79.74	78.98	78.18
	5 th Turkish dataset	80.46	79.21	80.91	80.60
Precision	1 st Turkish dataset	81.63	81.87	81.77	81.88
	2 nd Turkish dataset	84.07	82.71	82.55	82.59
	3 rd Turkish dataset	77.68	78.97	77.59	79.12
	4 th Turkish dataset	78.63	79.63	78.34	78.60
	5 th Turkish dataset	78.08	77.80	79.04	78.77
Recall	1 st Turkish dataset	81.12	81.59	81.80	81.79
	2 nd Turkish dataset	84.04	82.68	82.64	82.56
	3 rd Turkish dataset	77.61	78.93	77.57	78.84
	4 th Turkish dataset	77.97	78.37	78.20	76.56
	5 th Turkish dataset	75.04	76.41	75.19	75.31
F1 Score	1 st Turkish dataset	81.31	81.70	81.79	81.83
	2 nd Turkish dataset	83.84	82.70	82.57	82.57
	3 rd Turkish dataset	77.57	78.90	77.57	78.86
	4 th Turkish dataset	78.22	78.79	78.27	77.04
	5 th Turkish dataset	76.18	77.01	76.56	76.56

- *Experiment 2: Investigating the Effect of the Number of Used Filters.*

One of the main factors that affect the overall performance of any CNN model is the number of used filters. In this experiment, the number of filters was set to 32, 64, 128, and 256. Similar to the previous experiment, the introduced CNN model was used to investigate the performance of all tested values. It has been found that in most cases 64 was the best number of filters. The results are shown in Table 3. Related to the processing time, for the first two datasets, both 32 and 64 filters require the least amount of time, however for the other datasets, as the number of samples increases, 32 filters were able to achieve the least amount of time, and 64 filters were able to achieve the second least amount of time.

Table 3. The Effect of changing the number of the used filters.

Evaluation Matrix	Dataset	Number of Filters			
		32	64	128	256
Accuracy	1 st Turkish dataset	77.62	80.15	78.53	79.57
	2 nd Turkish dataset	80.60	80.90	80.30	80.46
	3 rd Turkish dataset	77.47	77.30	77.45	77.45
	4 th Turkish dataset	78.43	78.16	78.19	78.18
	5 th Turkish dataset	79.90	79.86	79.82	79.86
Precision	1 st Turkish dataset	84.08	81.76	82.52	81.94
	2 nd Turkish dataset	81.15	81.32	81.76	81.68
	3 rd Turkish dataset	77.04	77.56	77.16	77.40
	4 th Turkish dataset	78.50	79.03	77.98	78.91
	5 th Turkish dataset	76.87	78.61	77.03	77.98
Recall	1 st Turkish dataset	84.06	81.70	82.20	81.24
	2 nd Turkish dataset	81.22	81.31	81.75	81.76
	3 rd Turkish dataset	77.01	77.54	77.16	77.34
	4 th Turkish dataset	78.20	77.61	78.00	77.93
	5 th Turkish dataset	74.68	74.48	74.74	75.23
F1 Score	1 st Turkish dataset	84.07	81.73	82.31	81.42
	2 nd Turkish dataset	81.15	81.31	81.75	81.67
	3 rd Turkish dataset	76.98	77.52	77.16	77.29
	4 th Turkish dataset	78.33	78.06	77.99	78.28
	5 th Turkish dataset	75.55	75.87	75.64	76.28

• *Experiment 3: The Effects of the Developed Pre-Processing Model*

The main aim of this experiment is to find the degree that performance can be affected by either pre-processing the data or use it directly. In other words, we will find the percentage of improvement, if existed. The results are shown in Table 4.

Table 4. The Accuracy of the proposed model with and without the pre-processing operation.

Dataset	Accuracy		Improvement (%)
	Without Pre-processing	With Pre-processing	
1 st Turkish dataset	77.62	80.24	3.37%
2 nd Turkish dataset	80	83	3.75%
3 rd Turkish dataset	77.3	80.46	4.08%
4 th Turkish dataset	79.2	83.18	5.02%
5 th Turkish dataset	78.16	85	8.75%

As shown in Table 4, it is clear that pre-processing has improved the performance when processing the

Turkish language. In addition, the effect of the pre-processing increase and became more important whenever the size of the used dataset increases. Hence, as the Turkish language is an agglutinative language, pre-processing is a must step for building an efficient system.

Section 2: Overall Performance of the Developed System

• *Experiment 4: Robustness and Scalability of the Developed System*

In this experiment, the robustness and scalability of the developed SA system were investigated using multiple different size datasets. In addition to the 10-fold cross validation that have been used in all the experiments, this experiment has been performed 5 times as shown in Table 5. Also, in addition to the accuracy, the Precision, Recall, and F1 score has also been calculated.

Table 5. The Accuracy, precision, recall, and f1 score for the developed SA system.

Iteration #	Dataset	Accuracy	Precision	Recall	F1
1 st	1 st Dataset	82.66	81.83	82.10	81.96
	2 nd Dataset	89.36	89.40	88.10	88.50
	3 rd Dataset	83.29	83.29	83.19	83.13
	4 th Dataset	81.66	81.84	80.00	80.54
	5 th Dataset	80.33	78.14	78.24	78.55
2 nd	1 st Dataset	82.80	81.94	81.78	81.85
	2 nd Dataset	82.16	82.14	82.22	82.14
	3 rd Dataset	79.52	79.64	79.53	79.50
	4 th Dataset	81.25	81.27	80.87	80.30
	5 th Dataset	80.21	77.38	76.51	76.91
3 rd	1 st Dataset	83.28	83.25	83.15	83.19
	2 nd Dataset	82.04	82.00	82.11	82.01
	3 rd Dataset	83.29	83.29	83.19	83.13
	4 th Dataset	79.09	79.10	77.57	78.03
	5 th Dataset	80.70	78.01	76.62	77.23
4 th	1 st Dataset	82.72	80.76	80.69	80.73
	2 nd Dataset	82.48	82.57	82.26	82.35
	3 rd Dataset	78.34	78.40	78.35	78.33
	4 th Dataset	89.36	89.40	88.10	88.50
	5 th Dataset	85.21	82.38	81.51	81.91
5 th	1 st Dataset	84.28	84.25	84.15	84.19
	2 nd Dataset	89.06	89.10	87.80	88.20
	3 rd Dataset	83.99	83.99	83.89	83.83
	4 th Dataset	89.86	89.90	88.60	89.10
	5 th Dataset	85.21	84.98	83.55	84.94
Average		83.29	82.73	82.16	82.36

It is clear from Table 5 that the developed system was able to process all used datasets efficiently. Also, the average of the developed SA system's accuracy, precision, recall, and F1 score is shown in Figure 3.

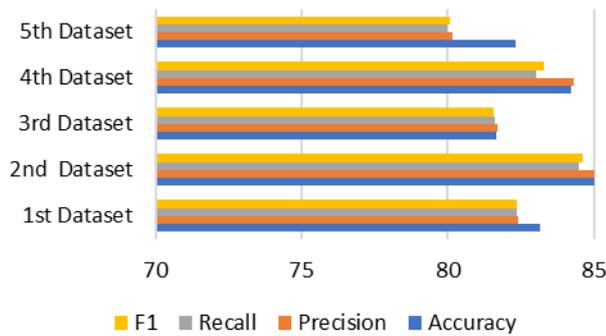


Figure 3. The average of the accuracy, precision, recall, and F1 of the developed system.

• Experiment 5: Comparing Our Approach with the State-of-the-art Turkish CNN Based Systems

It is worth mentioning that out of the existing Turkish studies, we found only two models, i.e., [5, 20], where their structure and details can be found in the related references and can be re-implemented. In this experiment, the performance of the developed approach was compared with the approaches of [5, 20].

In detail, we have re-implemented the aforementioned approaches, and their hyper-parameters were set as stated in the respected references. Figure 4 shows the experiment results, and it is clear that the developed approach has significantly outperformed all approaches, where improvement in accuracy was 35% as compared to [5] model whereas the improvement is 20% as compared to [20].

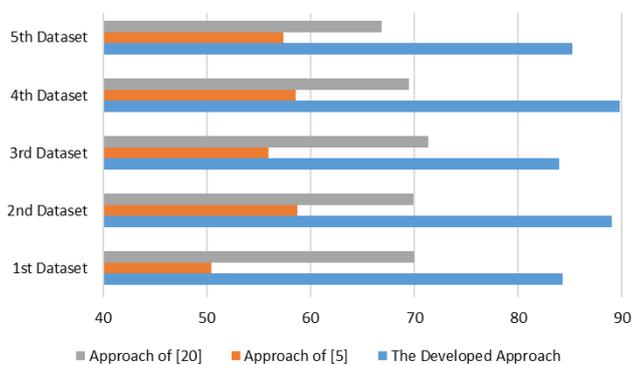


Figure 4. The accuracy for all studied models.

5. Conclusions and Future Works

Nowadays, social media analytics is an important research field and has led to improve internet-based systems. In this work, an efficient CNN sentiment analysis system for the Turkish Language was developed. The system was built after deep study that aims to find the most suitable layers and components to build efficient CNN model for Turkish language. As shown in the experiments results, the developed pre-processing model was able to improve the overall performance for Turkish SA systems. In addition, the developed approach was able to keep achieving a high performance while increasing the number of processed samples. Our experiments also show superior

performance as compared with 2 state-of-the-art CNN Turkish SA systems.

One of the directions that can be follow as future work is to investigate the performance of all available word embedding systems, another direction is to integrate multiple CNN models in parallel for improving sentiment analysis systems.

References

- [1] Alqaraleh S., Hafez A., and Tello A., "Dynamic Time Warping of Deep Features for Place Recognition in Visually Varying Conditions," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 3675-3689, 2021.
- [2] Alqaraleh S. and IŞIK M., "Efficient Turkish Tweet Classification System for Crisis Response," *The Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 28, no. 6, pp. 3168-3182, 2020.
- [3] Dehkharghani R., Yanikoglu B., Saygin Y., and Oflazer K., "Sentiment Analysis in Turkish at Different Granularity Levels," *Natural Language Engineering*, vol. 23, no. 4, pp. 535-559, 2017.
- [4] Dehkharghani R., Saygin Y., Yanikoglu B., and Oflazer K., "Sentitürknet: A Turkish Polarity Lexicon for Sentiment Analysis," *Language Resources and Evaluation*, vol. 50, no. 3, pp. 667-685, 2016.
- [5] Demirci G., Keskin Ş., and Doğan G., "Sentiment Analysis in Turkish with Deep Learning," in *Proceedings of the IEEE International Conference on Big Data*, Los Angeles, pp. 2215-2221, 2019.
- [6] Demirtas E. and Pechenizkiy M., "Cross-Lingual Polarity Detection with Machine Translation," in *Proceedings of the 2nd International Workshop on Issues of Sentiment Discovery and Opinion Mining*, New York, pp. 1-8, 2013.
- [7] Eroğul U., *Sentiment Analysis in Turkish*, Middle East Technical University, 2009.
- [8] Esuli A. and Sebastiani F., "Sentiwordnet: A Publicly Available Lexical Resource for Opinion Mining," *LREC*, vol. 6, pp. 417-422, 2006.
- [9] Gardner M., Grus J., Neumann M., Tafjord O., Dasigi P., Liu N., Peters M., Schmitz M., and Zettlemoyer L., "A Deep Semantic Natural Language Processing Platform," *arXiv preprint arXiv:1803.07640*, 2017.
- [10] Ghorbel H. and Jacot D., "Sentiment Analysis of French Movie Reviews," *Advances in Distributed Agent-Based Retrieval Tools*, vol. 361, pp. 97-108, 2011.
- [11] He K., Zhang X., Ren S., and Sun J., "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on*

- Computer Vision and Pattern Recognition*, Las Vegas, pp. 770-778, 2016.
- [12] Hussein D., "A Survey on Sentiment Analysis Challenges," *Journal of King Saud University-Engineering Sciences*, vol. 30, no. 4, pp. 330-338, 2018.
- [13] Jaouedi N., Boujnah N., and Bouhlel M., "A Novel Recurrent Neural Networks Architecture for Behavior Analysis," *The International Arab Journal of Information Technology*, vol. 18, no. 1, pp. 133-139, 2021.
- [14] Johnson R. and Zhang T., "Convolutional Neural Networks for Text Categorization: Shallow Word-Level Vs. Deep Character-Level," *arXiv preprint arXiv:1609.00718*, 2016.
- [15] Kalchbrenner N., Grefenstette E., and Blunso P., "A Convolutional Neural Network for Modelling Sentences," *arXiv preprint arXiv:1404.2188*, 2014.
- [16] Kaya M., Fidan G., and Toroslu I., "Sentiment Analysis of Turkish Political News," in *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, Macau, pp. 174-180, 2012.
- [17] Kaya M., Sentiment Analysis of Turkish Political Columns with Transfer Learning, Middle East Technical University, 2013.
- [18] Korovkinas K., Danenas P., and Garšva G., "SVM and Naïve Bayes Classification Ensemble Method for Sentiment Analysis," *Baltic Journal of Modern Computing*, vol. 5, no. 4, pp. 398-409, 2017.
- [19] Krizhevsky A., Sutskever I., and Hinton G., "Imagenet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 60, no. 6, pp. 1097-1105, 2012.
- [20] Kurt F., "Investigating the Performance of Segmentation Methods with Deep Learning Models for Sentiment Analysis on Turkish Informal Texts," Master's Thesis, Middle East Technical University, 2018.
- [21] Mäntylä M., Graziotin D., and Kuutila M., "The Evolution of Sentiment Analysis-A Review of Research Topics, Venues, and Top Cited Papers," *Computer Science Review*, vol. 27, pp. 16-32, 2018.
- [22] Miller G., Beckwith R., Fellbaum C., Gross D., and Miller K., "Introduction To Wordnet: An On-Line Lexical Database," *The International Journal of Lexicography*, vol. 3, no. 4, pp. 235-244, 1990.
- [23] Neviarouskaya A., Prendinger H., and Ishizuka M., "Attitude Sensing in Text Based on A Compositional Linguistic Approach," *Computational Intelligence*, vol. 31, no. 2, pp. 256-300, 2015.
- [24] Pradhan V., Vala J., and Balani P., "A Survey on Sentiment Analysis Algorithms for Opinion Mining," *International Journal of Computer Applications*, vol. 133, no. 9, pp. 7-11, 2016.
- [25] Simonyan K. and Zisserman A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ArxivPreprinArxiv:1409.1556*, 2014.
- [26] Stelzner M., "Social Media Marketing Industry Report," *Social Media Examiner*, 2019.
- [27] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., and Rabinovich A., "Going Deeper with Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, pp. 1-9, 2015.
- [28] Türkmenoglu C. and Tantug A., "Sentiment Analysis in Turkish Media," in *Proceedings of Workshop on Issues of Sentiment Discovery and Opinion Mining, International Conference on Machine Learning*, Beijing, pp. 1-9, 2014.
- [29] Yıldırım E., Çetin F., Eryiğit G., and Temel T., "The Impact of NLP on Turkish Sentiment Analysis," *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, vol. 7, no. 1, pp. 43-51, 2015.
- [30] Zhang L. and Chen C., "Sentiment Classification with Convolutional Neural Networks: An Experimental Study on A Large-Scale Chinese Conversation Corpus," in *Proceedings of the 12th International Conference on Computational Intelligence and Security*, Wuxi, pp. 165-169, 2016.



Saed Alqaraleh is an Assistant Professor at Hasan Kalyoncu University, Gaziantep, Turkey. He completed his Ph.D. in Computer Engineering (2015) from Eastern Mediterranean University, Cyprus. His present areas of research include Information retrieval and security, Natural Language Processing, Visual Place Recognition, and Machine learning.