A novel COVID-19 sentiment analysis in Turkish based on the combination of convolutional neural network and bidirectional long-short term memory on Twitter

Abdullah Talha Kabakus

Correspondence
Abdullah Talha Kabakus, Department of Computer Engineering, Faculty of Engineering, Duzce University, Duzce, Turkey. Email: talhakabakus@duzce.edu.tr

Abstract
The whole world has been experiencing the COVID-19 pandemic since December 2019. During the pandemic, a new life has been started by necessity where people have extensively used social media to express their feelings, and find information. Twitter was used as the source of what people have shared regarding the COVID-19 pandemic. Sentiment analysis deals with the extraction of the sentiment of a given text. Most of the related works deal with sentiment analysis in English, while studies for Turkish sentiment analysis lack in the research field. To this end, a novel sentiment analysis model based on the combination of convolutional neural network and bidirectional long short-term memory was proposed in this study. The proposed deep neural network model was trained on the constructed Twitter dataset, which consists of 15k Turkish tweets regarding the COVID-19 pandemic, to classify a given tweet into three sentiment classes, namely, (i) positive, (ii) negative, and (iii) neutral. A set of experiments were conducted for the evaluation of the proposed model. According to the experimental result, the proposed model obtained an accuracy as high as 97.895%, which outperformed the state-of-the-art baseline models for sentiment analysis of tweets in Turkish.

KEYWORDS
COVID-19, deep neural network, sentiment analysis, text classification, Twitter

1 INTRODUCTION

The whole world has been experiencing the heavy effects of the coronavirus disease 2019 (COVID-19) pandemic since December 2019. According to the World Health Organization (WHO), as of June 5, 2021, more than 172 million cases and more than 3.7 million deaths have been confirmed on a global scale. When it comes to Turkey, there have been more than 5.2 million confirmed cases of COVID-19 with 48 million deaths. Twitter, one of the most popular microblogging platforms with having 192 million daily active users as of the fourth quarter of 2020, lets registered users post status messages, which are called “tweets.” A tweet can contain characters up to 280 characters and can be concerned with everything related to life, including but not limited to economics, politics, personal life, products, health, and entertainment. In addition to the public, companies use social media (e.g., Twitter) to promote products, brand names, and services. Additionally, many governments use Twitter to communicate with the public. Especially during the pandemic, people have used social media to express their feelings, find information, and also calm themselves which is unsurprising as it was reported that people feeling lonely were more inclined to use social media to cope with lacking social contact during the COVID-19 lockdown. According to a recent report, Twitter saw a record number of users, 186 million, during the pandemic. Therefore, the proposed study
intentionally used Twitter as the source of the data. This huge number is self-representative regarding the interpretation of the data posted on Twitter is highly time-consuming and labor-intensive. Sentiment analysis is one of the hottest research topics of computer science and is accepted as a subfield of natural language processing (NLP). Sentiment analysis is basically defined as the task of extracting emotions or opinions from a given text for a given topic. Sentiment analysis is an increasingly popular instrument for the analysis of social media. The underlying technology of sentiment analysis is a part of artificial intelligence (AI) as the essence of it is a text classification task. Twitter sentiment analysis is a subfield of sentiment analysis that deals with identifying sentiments in tweets that combines the power of sentiment analysis with the popularity of Twitter. Due to these advantages, Twitter sentiment analysis has been widely adopted in many different domains, including but not limited to politics, economics, product sales, medical, sports analytics, security informatics, and consumer satisfaction and brand analysis. Twitter sentiment analysis is considered a more challenging task than sentiment analysis on conventional text (e.g., review documents) due to (i) the short length of tweet messages, (ii) the frequent use of informal and irregular words, slang, and misspelled words (e.g., repeated characters), (iii) the usage of Twitter-specific keywords such as hashtags, and usernames, (iv) the usage of emojis, and emoticons, and (v) the rapid evolution of the language in Twitter. Most of the related works deal with sentiment analysis in English, while studies for Turkish sentiment analysis lack in the research field. To this end, a novel Twitter sentiment analysis model based on the combination of convolutional neural network (CNN) and bidirectional long short-term memory (BiLSTM) for Turkish was proposed with this study. The proposed deep neural network (DNN) model was trained and evaluated on the constructed dataset that consists of the collected Turkish tweets regarding the COVID-19 pandemic. The main contributions of the proposed study are listed as follows:

- A novel deep neural network, which is a combination of CNN and BiLSTM, was proposed for Twitter sentiment analysis in Turkish. The proposed model was finalized after the model optimization task, which was performed in an automated way thanks to the employed Grid Search technique.
- Unlike the related work, both emojis and emoticons were not cleared during the preprocessing since they were used within the proposed sentiment analysis model.
- The tweet annotation process was performed in an automated way instead of manual annotation, which is highly time-consuming and labor-intensive. Additionally, instead of labeling tweets with respect to whether they contain a predefined set of symbols or words (e.g., emojis, emoticons, keywords, and hashtags), the employed approach uses a lexicon-based tweet annotation.

The rest of the article is organized as follows: Section 2 briefly reviews the related work. Section 3 describes the material and method used for the proposed study. Section 4 describes the experimental result and discussion in the light of conducted experiments. Finally, Section 5 concludes the article with future directions.

2 RELATED WORK

Coban et al. proposed an approach based on traditional machine learning (ML) algorithms for the Turkish sentiment analysis task in Twitter. They classified the tweets into two sentiment classes, namely, (i) positive, and (ii) negative. They queried the Twitter API through the predefined set of emoticons where containing the ""; (""; "") and ""; D; and ""; −""; (""; ("; and ""; (" emoticons label the corresponding tweets as the positive and negative, respectively. Based on this assumption, they constructed a dataset that consisted of 20k Turkish tweets. Our study intentionally did not use this assumption since (i) a tweet may contain emoticons from both classes, and (ii) a tweet does not have to contain any emoticons. During the preprocessing, they remove emoticons from tweets. Unlike this study, we intentionally kept emoticons as well as emojis as they are sentimentally meaningful (as discussed in Section 3). They employed four traditional ML algorithms, namely, the (i) support vector machine (SVM), (ii) Naïve Bayes, (iii) Multinomial Naïve Bayes, and (iv) k-Nearest Neighbors (k-NN). According to their experimental result, the Multinomial Naïve Bayes classifier obtained the highest accuracy, an accuracy of 66.06% when the features of vector space were extracted through the employed n-gram method. Akgun et al. proposed an approach based on lexicon and n-grams. The proposed method was evaluated on the constructed dataset that was labeled into three classes, namely, (i) positive, (ii) negative, and (iii) neutral through a given lexicon. According to the experimental result, the lexicon-based method slightly outperformed the n-gram based method by obtaining an F1-score of 70% compared to 69%. One drawback of this study is that, unlike our study, emotional expressions in tweets were removed during the preprocessing. Another drawback of this study is that they assigned the thresholds for the emotion classes in a not-formulaic way. Demirci proposed an approach for the emotion analysis of Turkish tweets and collected tweets for six emotions, namely, (i) anger, (ii) disgust, (iii) fear, (iv) joy, (v) sadness, and (vi) surprise. The constructed dataset contained 6000 tweets, 1000 tweets per emotion. The effects of various traditional ML algorithms, namely, (i) Naïve Bayes, (ii) Complement Naïve Bayes, (iii) SVM, and (iv) k-NN. According to the experimental result, the SVM outperformed the other algorithms by obtaining an accuracy of 69.92%. Ileri proposed a user-centric approach for Turkish emotion analysis in Twitter. The motivation behind this study was that connected users may be more likely to hold similar emotions. They considered the same six emotions that were considered by Demirci. The dataset that was used for this study was constructed by querying the Twitter API through the predefined Turkish keywords for the covered emotions as follows (English translations are given in the parenthesis): (i) “șinir” (“anger”), (ii) “korku”
("fear"), (iii) "mutülük" ("happiness"), (iv) "üzüntü" ("sadness"), (v) "iğren c" ("disgusting"), and (vi) "sa skunluk" ("surprise"). Consequently, a total of 72,000 tweets were collected, 1,200 tweets for each emotion. According to their experimental result, they reported that multi-class emotional domains, ideally with more than two emotions, cause lots of biases in class predictions unless the classes are greatly separable in terms of features. Unlike this study, we intentionally did not remove emotional expressions (e.g., emojis and emoticons) from tweets during preprocessing since they are sentimentally meaningful (as discussed in Section 3). Also, weighted edges were ignored in this study as all edge weights were assumed to equal to 1.

Tocoglu et al.23 proposed an approach based on DNNs for the emotion analysis of Turkish tweets. The six emotion classes they considered were the same as the emotion classes considered by Demirci.21 Since the dataset constructed by Demirci21 is not publicly available and large enough to be used in DNNs. Tocoglu et al.23 constructed their own dataset that consists of 205,278 tweets. They labeled these tweets using a lexicon-based method. When it comes to the classification task, they proposed various models based on three different DNN architectures, namely, (i) artificial neural network (ANN), (ii) CNN, and (iii) LSTM. According to the experimental result, while the proposed CNN obtained the highest accuracy, an accuracy of 74%, the proposed ANN obtained the worst accuracy. The authors mentioned that they did not perform cross-validation due to the computational constraints of using a large dataset. Unlike this work, we performed k-fold cross-validation, which is a technique that ensures each observation in the dataset is used for both training and validation (more detail is provided in Section 4). Demirci et al.24 proposed a sentiment analysis approach for Turkish using both traditional ML and deep learning techniques. They collected 3,000 tweets regarding the coup attempt that happened on July 15, 2016. They covered two sentiment classes, namely, (i) positive, and (ii) negative. To classify the tweets, they employed three traditional ML techniques, namely, (i) logistic regression, (ii) SVM, and (iii) random forest, and a DNN model that consisted of six Dense and three Dropout layers. According to the experimental result, their DNN provided better performance than the traditional ML techniques by obtaining an accuracy of 81.86%. One drawback of this study is that, unlike our study, neither emojis nor emoticons were used within this study. Another drawback of this study is that their constructed dataset is too small to effectively train a DNN. Ucan et al.25 proposed a Turkish emotion analysis based on four pre-trained language models, namely, (i) BERT, (ii) DistilBERT, (iii) BERTurk, and (iv) DistilBETurk. They employed various traditional ML such as SVM, and Naive Bayes and deep learning techniques such as gated recurrent unit (GRU), and LSTM. According to conducted experiments on various well-known datasets, the proposed model based on BERTurk/BERT obtained the best performance. One drawback of this study is the absence of the employment of the features that target the emotional contexts of tweets. In addition to the related work that was proposed for sentiment analysis in Turkish, there exist some studies that employ the combination of CNN and LSTM for sentiment analysis in various languages such as English, Vietnamese, and Arabic. Onan26 proposed an architecture that combines TF-IDF weighted Glove word embedding with CNN-LSTM architecture for sentiment analysis on product reviews. The proposed architecture was trained and evaluated on a total of 6940 tweets written in English. The three sentiment classes that were covered in this study were, namely, (i) positive, (ii) negative, and (iii) neutral. This architecture obtained an accuracy of 93.85% on the test set of the employed dataset, which consisted of 692 tweets. Unlike this study, we have employed a hyperparameter optimization task to reveal the best-performing hyperparameters for the proposed DNN model. Vo et al.27 proposed an approach based on the combination of CNN and LSTM for sentiment analysis in Vietnamese. To this end, they have collected comments/reviews from Vietnamese commercial web pages. The constructed corpus consisted of 17,500 reviews and was annotated by three human annotators. They covered three sentiment classes, namely, (i) positive, (ii) negative, and (iii) neutral. According to the experimental result, the proposed model outperformed SVM, LSTM, and CNN baseline models. One drawback of this study is that the proposed model does not employ text preprocessing techniques despite that it is necessary for Vietnamese. Ombabi et al.28 proposed an approach based on the combination of CNN-LSTM and SVM for sentiment analysis in Arabic. They employed fastText,29 an open-source library by Facebook AI Research, to construct the word embeddings. The feature maps were learned by the proposed CNN-LSTM model. Then, these features maps were yielded into the employed SVM to classify the given input into two sentiment classes, namely, (i) positive, and (ii) negative. According to the experimental result, the proposed model obtained an accuracy of 90.75% on a multi-domain corpus. One drawback of this study is that they empirically (without employing any well-known techniques) determined the values of hyperparameters. Instead of this approach, we employed a widely-used technique for this critical task (which is described in detail in Section 3.4). The comparison of the related work is given in Table 1.

3 MATERIAL AND METHOD

In this section, we start by describing the details of the dataset construction process that was used by the proposed method. Then, the employed preprocessing process that was applied to the constructed dataset, and how this unlabeled dataset was annotated were described. Finally, the details of the proposed method are discussed.

3.1 Dataset construction

The constructed dataset contains Turkish tweets related to the COVID-19 pandemic from Twitter. These tweets were fetched through the Twitter Standard Search API30 that is provided by Twitter. To this end, an open-source Python library, namely, tweepy,31 was employed. The criteria that were
### Table 1: The comparison of the related work in terms of (i) employed technique(s), (ii) covered emotions, and (iii) drawback(s)

| Related work | Employed technique(s) | Covered emotions | Drawback(s) |
|--------------|-----------------------|------------------|-------------|
| Coban et al. | Traditional ML algorithms | (i) Positive, and (ii) negative | Labeling tweets through the predefined set of emotions |
| Akgun et al. | Lexicon-based techniques | (i) Positive, (ii) negative, and (iii) neutral | Removal of emotional expressions during preprocessing |
| Demirci | Traditional ML algorithms | (i) Anger, (ii) disgust, (iii) fear, (iv) joy, (v) sadness, and (vi) surprise | Not employing DNNs alongside the traditional ML algorithms |
| Ileri | User-centric techniques | (i) Anger, (ii) disgust, (iii) fear, (iv) joy, (v) sadness, and (vi) surprise | Removal of emotional expressions during preprocessing |
| Tocoglu et al. | Lexicon-based techniques and DNNs | (i) Anger, (ii) disgust, (iii) fear, (iv) joy, (v) sadness, and (vi) surprise | Not employing the cross-validation technique |
| Demirci et al. | Traditional ML algorithms, and DNNs | (i) Positive, and (ii) negative | Removal of emotional expressions during preprocessing |
| Ucan et al. | Pre-trained language models, traditional ML algorithms, and DNNs | (i) Anger, (ii) disgust, (iii) fear, (iv) joy, (v) sadness, and (vi) surprise | Absence of the employment of the features that target the emotional contexts of tweets |
| Onan | Pre-trained language models, and DNNs | (i) Positive, (ii) negative, and (iii) neutral | Absence of the hyperparameter optimization task |
| Vo et al. | DNNs | (i) Positive, (ii) negative, and (iii) neutral | Absence of the employment of text preprocessing techniques |
| Ombabi et al. | Pre-trained language models, traditional ML algorithms, and DNNs | (i) Positive, and (ii) negative | Hyperparameters of the proposed DNNs were empirically determined |

set to filter tweets from the Twitter Standard Search API were as follows: (i) the tweets should be written in Turkish, (iii) the tweets should contain at least one of the following keywords: "covid", "covid-19", "covid19", "koronavirüs", and "pandemi", and (iii) the tweets should not be "retweets", which are the tweets that were broadcast by users to inform their network, to prevent duplications. Consequently, a total of 50k tweets were collected. The sequence length distribution of these tweets is presented in Figure 1.

### 3.2 Preprocessing

Since the collected tweets are raw, a preprocessing process is key to preparing them to be ready to be yielded into the proposed DNN. The preprocessing process converts the data to a more meaningful form, which eventually makes the ML model better represent the data. As a natural result of this improvement, the accuracy of the ML model improves. The employed preprocessing process performs the following operations on the collected raw tweets:

- The capital letters of tweets were folded to lowercase.
- The hyperlinks, which are sentimentally meaningless, were cleared the tweets.
- The stop words, which are commonly used words in a natural language that do not convey polarity, were cleared from the tweets to reduce the noise in textual data. The stop words, that are specific to Turkish, were retrieved from an open-source Python library, namely, stop-words. The keywords that were used to query the Twitter Standard Search API were also included in the list of stop words.
- A Twitter mention is a keyword that is a part of Twitter’s internal communication mechanism and contains a username of another Twitter user by preceding a "@" character. It is not a "must" for a tweet to contain mention(s) but a tweet may contain mentions as long as its character limit, 280
Characters, allows. Since Twitter mentions are sentimentally meaningless, they were cleared from the tweets. It is worth mentioning that the "@" character is not the only one that was cleared from tweets; but also the succeeding username since they may be sentimental despite they actually do not aim to be (e.g., "@happiness").

- A Twitter hashtag is a keyword that tweets may contain to declare the topic of a tweet. A Twitter hashtag contains one word or combined words (without any space between these words) by preceding a "#" character. Since Twitter hashtags do not convey a sentimental polarity, they were cleared from tweets.

- The words, that contain a single character, were cleared from the tweets.

- The punctuation marks were cleared from the tweets after the extraction of emojis and emoticons.

- Unlike the related work, emojis and emoticons were intentionally not cleared from the tweets since they are sentimentally meaningful.

Six of the fetched 50k tweets were eliminated from the dataset since becoming empty Strings ("""") after the preprocessing process. Therefore, the final dataset consisted of 49,994 unique tweets. Any stemming approaches were not applied since stemming algorithms do not work well with tweets. The number of samples that each sentiment class of the final dataset consisted of is listed in Table 2.

### 3.3 Dataset annotation

Since the constructed dataset is not annotated, it is necessary to annotate it as the proposed model is based on a supervised ML technique. In literature, the datasets were commonly annotated based on either using a set of keywords or emoticons. Unlike these works, a lexicon-based approach was employed for generalization. When it comes to the state-of-the-art lexicons, TextBlob was intentionally not opted since it does not...
utilize social media-specific symbols such as emojis, and emoticons while analyzing the sentiment of a given sentence. To this end, VADER (Valence Aware Dictionary and sEntiment Reasoner),\textsuperscript{42} which is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, was employed. VADER basically reveals a polarity of a given sentence in the range of $[-1, +1]$ where $-1$ and $+1$ mean extremely negative and extremely positive sentiments, respectively. Even though VADER was originally developed for sentiment analysis in English, we have employed it on Turkish tweets since (i) it also uses emotional expressions (e.g., emojis, and emoticons), slang, hashtags, and punctuation marks, which are all common in tweets, alongside words, and (ii) these features are language-independent. The labels were assigned based on the approach proposed by Bandi and Fellah\textsuperscript{43} as follows: If the calculated polarity of a tweet is less than $-0.4$, it is labeled as negative. Symmetrically, if the calculated polarity of a tweet is greater than $+0.4$, it is labeled as positive. The tweets whose polarity values are in the range from $-0.4$ to $+0.4$ are labeled as neutral. This approach is formulated in Formula \ref{eq:1} where $T_i$, $P_i$, and $S_i$ indicate the given tweet $T_i$, the calculated polarity of $T_i$, and the assigned sentiment label of $T_i$, respectively.

\begin{equation}
S_i = \begin{cases} 
\text{negative}, & P_i < -0.4 \\
\text{neutral}, & -0.4 \leq P_i \leq +0.4 \\
\text{positive}, & P_i > +0.4 
\end{cases}
\end{equation}

3.4 Proposed model

The proposed model is a novel DNN that is a combination of CNN and BiLSTM to classify a given tweet's sentiment into one of the three sentiment classes, namely, (i) negative, (ii) neutral, and (iii) positive. CNNs are basically composed of different types of layers. The proposed model was implemented with Keras,\textsuperscript{44} a widely used deep learning API that was written in Python. Keras acts as a wrapper for various DNN backends. TensorFlow,\textsuperscript{45} a widely used open-source ML platform that was developed by Google, was opted as the backend of the proposed model since it is the one recommended by the developer of Keras.\textsuperscript{46} The proposed model consists of 13 layers as an overview of its architecture is presented in Figure 2. The model started with an Embedding layer, which creates vector space for the given input (tweets). Then, a 1-dimensional Convolutional layer (Conv1D) was employed to perform convolution operations on the output of the Embedding layer. While $n$, $l_{en}$, and $m$ denote the number of input sequences, the number of tokens of the input sequence $i$, and the mean of the lengths of the input sequences, the calculation of the

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{architecture.png}
\caption{An overview of the architecture of the proposed DNN model, which is a combination of CNN and BiLSTM}
\end{figure}
length of the input sequences of the Embedding layer (denoted by max_len) was formulized in Formula 2. The max_len was calculated as 24 for the constructed dataset.

\[
m = \frac{\sum_{i=1}^{n} \text{len}_i}{n}
\]

\[
\text{max}_{\text{len}} = m + \sqrt{\frac{\sum_{i=1}^{n} (\text{len}_i - m)^2}{n}}
\]

Following the Conv1D layer, a 1-dimensional Max Pooling (MaxPool1D) layer was employed to reduce the spatial size of the representation that helps to reduce the number of parameters and computation in the network.\(^{47}\) Recurrent neural network (RNN) is a type of feedforward ANN, which is capable of handling variable-length sequence inputs.\(^{48}\) LSTM\(^ {48}\) is a special type of RNN that contains gates to maintain memory for long periods.\(^ {51}\) Therefore, LSTM is capable of remembering the past data and producing output with respect to the past and current data, which are key necessities for our case. The main advantages of LSTM over RNN are as follows: (i) LSTM has the ability to solve the general problem of gradient descent,\(^ {50}\) and (ii) it contains a long-term memory which is a key necessity for sequence processing. Bidirectional LSTM (BiLSTM) is a further development of LSTM that has the ability to access both preceding and succeeding contexts. Therefore, BiLSTM can solve sequential modeling better than LSTM.\(^ {51}\) Due to the aforementioned advantages of BiLSTM, it was intentionally employed in the proposed model instead of RNN and LSTM. After these typical layers of CNN, a BiLSTM layer was employed. Then, a Dropout\(^ {52}\) layer was employed to randomly drop a defined ratio of neurons from the neural network. Therefore, Dropout is a widely used technique to prevent the “overfitting” problem, which is one of the major challenges of DNNs because of increased depth and complexity.\(^ {53,54}\) This pattern (sequentially added BiLSTM and Dropout layers) was employed one more time. Then, a Flattening layer was employed to reshape the input (a matrix) into a vector. Following the Flattening layer, two Dense layers, which are fully connected neural network components, were employed for classification purposes. Between these two Dense layers, a Dropout layer was employed. The Softmax was employed as the activation function of the last Dense layer, which is responsible for the classification of the given tweet into one of the three sentiment classes. Except for the last Dense layer, the ReLU was employed as the activation function for all Conv1D layers and the other Dense layer to introduce non-linearity to the proposed DNN. The Adaptive Moment Estimation (Adam)\(^ {55}\) algorithm, which is an extension to Stochastic Gradient Descent (SGD), was employed as the optimization algorithm of the proposed neural network to update the neural network weights more efficiently by computing adaptive learning rates for each network parameter from estimates of the first and second moments of the gradient.\(^ {56}\) A loss function in a neural network is responsible for estimating the loss in each iteration (a.k.a. epoch) in order to reduce it on the next iteration. The Categorical Cross-entropy function was employed as the loss function since the task of the proposed neural network is a multi-class classification problem.

Hyper-parameters of a neural network are the parameters of a DNN model that affect the learning process and are set empirically. Therefore, the Grid Search technique, which is an extensive way of hyper-parameter optimization since all possible combinations of hyper-parameter values were literally experimented with,\(^ {57}\) was employed to reveal the best value for each hyper-parameter. To this end, each hyper-parameter has experimented with various values as they are listed in Table 3. This technique was realized thanks to the scikit-learn library as it provides an easy-to-use integration with Keras to programmatically evaluate the given values for the hyper-parameters and then reveal the best combination of them that produces the highest accuracy, which was regarded as the success metric of the proposed model. As a result of this process, the employed hyper-parameters were determined. The layers of the proposed model including the corresponding hyper-parameters are listed in Table 4.

| Hyper-parameter | Evaluated values |
|-----------------|------------------|
| Number of filters of the Conv1D layers | 32, 64, 128 |
| Kernel size of the Conv1D layers | 3, 5, 7 |
| Number of units of the Dense layers | 32, 64, 128, 256 |
| Optimization algorithm | Adam, SGD, RMSprop, Adamax |
| Learning rate of the optimization algorithm | 1 \times e^{-2}, 1 \times e^{-3}, 1 \times e^{-2} |
| Size of vector space of the Embedding layer | 32, 64, 128, 256 |
| Activation function (except for the last Dense layer) | ReLU, tanh, sigmoid, softmax |
| Batch size | 32, 64, 128 |
| Dropout rate | 0.25, 0.3, 0.4, 0.5, 0.6, 0.7 |
| Layer # | Layer type | Hyper-parameters with their values |
|---------|------------|-----------------------------------|
| 1       | Embedding  | - Vocabulary size: 91,012          |
|         |            | - Dimension of embedding: 64      |
|         |            | - Length of input sequences: 24   |
| 2       | Conv1D     | - Number of filters: 64           |
|         |            | - Kernel size: 3                  |
|         |            | - Activation function: ReLU       |
| 3       | MaxPool1D  | - Pool size: 2                    |
| 4       | Conv1D     | - Number of filters: 64           |
|         |            | - Kernel size: 3                  |
|         |            | - Activation function: ReLU       |
| 5       | MaxPool1D  | - Pool size: 2                    |
| 6       | BiLSTM     | - Number of units: 10             |
| 7       | Dropout    | - Dropout rate: 0.6               |
| 8       | BiLSTM     | - Number of units: 10             |
| 9       | Dropout    | - Dropout rate: 0.6               |
| 10      | Flattening | -                                  |
| 11      | Dense      | - Number of units: 32             |
|         |            | - Activation function: ReLU       |
| 12      | Dropout    | - Dropout rate: 0.6               |
| 13      | Dense      | - Number of units: 3              |
|         |            | - Activation function: Softmax     |

Adam  
- Learning rate: $1 \times 10^{-3}$

## EXPERIMENTAL RESULT AND DISCUSSION

The whole dataset was split as follows: 70% and 30% of the whole dataset were used as training and validation sets (which consisted of 34,960 samples), and test sets (which consisted of 14,984 samples), respectively, due to being the commonly used ratio in data mining. When it comes to the split of the training and validation sets, the *stratified k-Fold cross-validator*, which ensures that generated test sets contain the same (or as close as possible) distribution of target classes, was employed to create folds. The *stratified k-Fold cross-validator* was intentionally opted instead of the *k-Fold cross-validator* since the distribution of the target classes in the constructed dataset was imbalanced. The *stratified k-Fold cross-validator* was implemented thanks to the *scikit-learn* library. The number of folds ($k$) was set to 10 which means that the training and validation sets were divided into ten equal parts where nine parts (90% of the training and validation sets, which equals 31,464) were used for training, and the remaining one part (10% of the whole set, which equals 3496) was used for validation. The training of the proposed DNN was started with the *Early Stopping* callback, which helps to prevent overfitting by stopping the training when the monitored criterion (a.k.a. monitor) has not been getting better for the predefined number of epochs (a.k.a. patience). The monitor, and patience of the proposed model were defined as the calculated loss of the validation set (a.k.a. validation loss) and 2 epochs. The batch size of the training set, which defines the number of training samples used to estimate the gradient direction before the model’s internal parameters are updated, was set to 64 as a result of the employed *Grid Search*. The training had continued for 50 epochs as the plots of the calculated accuracy values for the training and validation sets over the epochs were given in Figure 3. As it can be seen from this figure, it is safe to say that the proposed model does not overfit.

After the training of the proposed model was completed, it was evaluated on the test set whose samples were used for neither training nor validation purposes. Since the proposed model handles a classification problem, its efficiency was measured through a de-facto standard technique, namely, *confusion matrix*. This technique defines several evaluation metrics including but not limited to *accuracy*, *precision*, *F1-score*, and *recall*. While $T$, $F$, $P$, and $N$ denote correctly classified samples, incorrectly classified samples, samples labeled as *positive*, and samples not labeled as *positive*, respectively, the aforementioned evaluation metrics were formulated in *Formula 3*. 

---

[59] Article reference.
[60] Article reference.
[61] Article reference.
[62] Article reference.
[63] Article reference.
[64] Article reference.
[65] Article reference.
FIGURE 3  The plots of the calculated accuracy values for the training and validation sets over the epochs

FIGURE 4  The confusion matrix of the evaluation of the proposed model on the test set

Accuracy = \( \frac{TP + TN}{P + N} \)

Precision = \( \frac{TP}{TP + FP} \)

F1-score = \( \frac{2 \times TP}{2 \times TP + FP + FN} \)

Recall = \( \frac{TP}{P} \)

(3)
The confusion matrix of the evaluation of the proposed model on the test set is presented in Figure 4. According to the experimental result, the accuracy, F1-score, precision, and recall of the proposed model were obtained as high as 97.895%, 98.116%, 98.372%, and 97.895%, respectively. The inference time of the proposed model, which is the duration to evaluate a given single sample, was calculated as low as 0.214 ms.

The proposed model was finalized after a series of experimental modifications as follows: First, the effect of Batch Normalization, which is a method to normalize the output of each activation by the mean and standard deviation of the outputs calculated over the samples in the minibatch, was experimented by adding a Batch Normalization layer after each Max Pooling layer. This modification decreased the original accuracy by 16.762%. Second, the depth of the proposed DNN was increased by adding another BiLSTM after the employed Convolutional layers. This modification decreased the original accuracy by 1.999% as well. Third, the number of units in BiLSTM layers was increased from 10 to 20 to see the effect of increasing the width of the proposed DNN. This modification decreased the original accuracy by 0.478%. Fourth, the Dense layer prior to the final Dense layer was removed. This modification decreased the original accuracy by 5.737% as well. As another contribution to the research topic, these modifications with their effects on the accuracy obtained by the proposed model are listed in Table 5.

In addition to this experiment, the most frequently used words in the constructed dataset were extracted to reveal the most frequently posted terms by the public. To this end, an open-source Python library, namely, WordCloud, was employed to generate word clouds of the most frequently used 100 words in the given text with excluding the stop words of Turkish, and the keywords used to query the Twitter Standard Search API. To this end, the word clouds of (a) all tweets, (b) the tweets labeled as positive, and (c) the tweets labeled as negative were generated as they are presented in Figure 5.

Keras provides pre-trained state-of-the-art models for 2-dimensional data such as InceptionV3, ResNet50V2, and Xception; but it does not provide any pre-trained models that can be applied to 1-dimensional data which is the case for the proposed study. Therefore, as baselines to the proposed model, the two state-of-the-art DNN models proposed by Naseem et al. were implemented from scratch. One of these models was based on CNN (hereby called "COVIDSenti\_CNN"), and the other one was based on RNN, more precisely Bidirectional LSTM (hereby called "COVIDSenti\_LSTM"). Both these models were trained exactly under the same training configuration (e.g., the same loss function, the same optimization algorithm and learning rate, the same Embedding layer, the same dataset, and the same training/validation/test ratios) with the proposed model to reveal the performance difference between these models. Similarly, these baseline models were evaluated on the same set of the proposed model (which is called the test set in the manuscript). According to the experimental result, the proposed model’s all evaluation metrics, namely, accuracy, F1-score, precision, and recall were obtained higher than the baseline models as the comparison of these models in terms of their efficiencies of classifying the given tweets is listed in Table 6.

### Table 5 The experimental modification with their effects on the accuracy obtained by the proposed model

| Modification on the proposed model                                      | Change on accuracy (%) |
|------------------------------------------------------------------------|------------------------|
| Adding a Batch Normalization layer after each Max Pooling layer        | −16.762                |
| Adding another BiLSTM layer                                            | −1.999                 |
| Increasing the number of units in BiLSTM layers from 10 to 20          | −0.478                 |
| Removing the Dense layer prior to the final Dense layer                | −5.737                 |

FIGURE 5 The generated word clouds of (A) all tweets, (B) the tweets labeled positive, and (C) the tweets labeled as negative
TABLE 6  The comparison of the proposed model with the baseline models in terms of their efficiencies of classifying the given tweets

| Related work     | Accuracy (%) | F1-score (%) | Precision (%) | Recall (%) |
|------------------|--------------|--------------|---------------|------------|
| COVIDSenti_CNN⁵  | 89.442       | 93.015       | 98.073        | 89.442     |
| COVIDSenti_LSTM⁶ | 97.691       | 97.805       | 98.015        | 97.691     |
| Proposed model   | 97.895       | 98.116       | 98.372        | 97.985     |

5 | CONCLUSION

Social networks have been an essential part of people’s daily lives, especially during the COVID-19 pandemic as many countries have declared full or partial lockdown for long periods. Given a vast number of posts that have been shared daily on social networks, it is highly time-consuming and labor-intensive to interpret their content by hand. Sentiment analysis based on text mining has demonstrated its efficiency in many domains thanks to the advances in AI and NLP. In this study, a novel Twitter sentiment analysis model for Turkish, which is a DNN based on a combination of CNN and BiLSTM, was proposed. The proposed model can be easily adopted to various AI-powered tasks, including but not limited to (i) public opinion pooling, (ii) brand/product confidence surveying, (iii) understanding election tendencies, and customer analysis, and (iv) marketing optimization. To train the proposed DNN, a dataset, that consisted of 15k unique (RTs were excluded) Turkish tweets, was constructed. The proposed model was both trained and evaluated on the constructed dataset after splitting the whole dataset into training, validation, and test sets. According to the experimental result, the proposed model obtained an accuracy as high as 97.895% which outperformed the state-of-the-art baselines. The experimental result proves that the combination of CNN and BiLSTM provides a promising architecture for a challenging text analysis task such as Twitter sentiment analysis. As future work, the word embedding techniques can be employed before yielding the input data (the preprocessed tweets) into the proposed DNN. Also, an attention mechanism can be integrated into the proposed model in order to try to further improve the classification accuracy of the proposed model.

ACKNOWLEDGMENT

We would like to thank Google for providing free computational resources through the Colab platform.

CONFLICT OF INTEREST

The author declares that there is no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ENDNOTES

1. https://twitter.com
2. Turkish translation of “coronavirus”.
3. Turkish translation of “pandemic”.

ORCID

Abdullah Talha Kabakus  https://orcid.org/0000-0003-2181-4292

REFERENCES

1. WHO coronavirus (COVID-19) dashboard. World Health Organization (WHO); 2021. Accessed November 11, 2021. https://covid19.who.int
2. Q4 and fiscal year 2020 letter to shareholders. Twitter; 2021. Accessed November 11, 2021. https://s22.q4cdn.com/826641620/files/doc_financials/2020/q4/FINAL-Q4%20-TWTR-Shareholder-Letter.pdf
3. He W, Wu H, Yan G, Akula V, Shen J. A novel social media competitive analytics framework with sentiment benchmarks. Inf Manag. 2015;52(7):801-812. doi:10.1016/j.im.2015.04.006
4. Kim SK, Park MJ, Rho JJ. Effect of the Government’s use of social media on the reliability of the government: focus on twitter. Public Manag Rev. 2015;17(3):328-355. doi:10.1080/14719037.2013.822530
5. Naseem U, Razzak I, Khushi M, Eklund PW, Kim J. COVIDSenti: a large-scale benchmark twitter data set for COVID-19 sentiment analysis. IEEE Trans Comput Soc Syst. 2021;1-13:1003-1015. doi:10.1109/TCSS.2021.3051189
6. Cauberghe V, van Wesenbeeck I, de Jans S, Hudders L, Ponnet K. How adolescents use social media to cope with feelings of loneliness and anxiety during COVID-19 lockdown. Cyberpsychol Behav Soc Netw. 2021;24(4):250-257. doi:10.1089/cyber.2020.0478
7. Rojas JPF. Coronavirus: lockdowns drive record growth in Twitter usage. Sky News; 2021. Accessed November 11, 2021. https://news.sky.com/story/coronavirus-lockdowns-drive-record-growth-in-twitter-usage-12034770
8. Puschmann C, Powell A. Turning words into consumer preferences: how sentiment analysis is framed in research and the news media. Soc Med Soc. 2018;4:1-12. doi:10.1177/2056305118797724
Kabakus AT. A novel COVID-19 sentiment analysis in Turkish based on the combination of convolutional neural network and bidirectional long-short term memory on Twitter. Concurrency Computat Pract Exper. 2022;34(22):e6883. doi: 10.1002/cpe.6883

How to cite this article: Kabakus AT. A novel COVID-19 sentiment analysis in Turkish based on the combination of convolutional neural network and bidirectional long-short term memory on Twitter. Concurrency Computat Pract Exper. 2022;34(22):e6883. doi: 10.1002/cpe.6883