Neural Computing and Applications (2022) 34:19615–19627
https://doi.org/10.1007/s00521-022-07650-2

S.I.: DEEP LEARNING MODELLING IN REAL LIFE: (ANOMALY DETECTION, BIOMEDICAL, CONCEPT ANALYSIS, FINANCE, IMAGE ANALYSIS, RECOMMENDATION)

Multiclass sentiment analysis on COVID-19-related tweets using deep learning models

Sotiria Vernikou[1] · Athanasios Lyras[1] · Andreas Kanavos[2]

Received: 22 November 2021 / Accepted: 18 July 2022 / Published online: 6 August 2022

Abstract

COVID-19 is an infectious disease with its first recorded cases identified in late 2019, while in March of 2020 it was declared as a pandemic. The outbreak of the disease has led to a sharp increase in posts and comments from social media users, with a plethora of sentiments being found therein. This paper addresses the subject of sentiment analysis, focusing on the classification of users' sentiment from posts related to COVID-19 that originate from Twitter. The period examined is from March until mid-April of 2020, when the pandemic had thus far affected the whole world. The data is processed and linguistically analyzed with the use of several natural language processing techniques. Sentiment analysis is implemented by utilizing seven different deep learning models based on LSTM neural networks, and a comparison with traditional machine learning classifiers is made. The models are trained in order to distinguish the tweets between three classes, namely negative, neutral and positive.

Keywords Big data · COVID-19 · Deep learning · LSTM · Natural language processing · Sentiment analysis · Social media · Twitter · Word embeddings

1 Introduction

Internet growth is rapidly developing and affects every aspect of our lives. This development continues to increase day after day due to the exploding volume of data and information. Most of these data is created through human interaction in social networks where social media platforms like Twitter, Facebook and Linkedin make distant communication feasible. One of the most popular social network applications is Twitter, which provides all sorts of information and allows its users to post text messages called "tweets."

The SARS-CoV-2 virus (COVID-19) pandemic started in December 2019. The virus was first detected in the Wuhan region of China [1] and is affecting 221 countries and territories around the globe[1]. It is a new strain of coronavirus that until then had not been identified in humans[2]. This virus mainly affects the respiratory system, although other organ systems are involved. Symptoms are associated with lower respiratory tract infections, such as fever, dry cough and shortness of breath. Moreover, headache, dizziness, generalized weakness, vomiting and diarrhea were observed. It is now widely known that the respiratory symptoms of COVID-19 are highly heterogeneous, ranging from minimal symptoms to severe hypoxia with acute respiratory distress syndrome (ARDS) [2].

The coronavirus pandemic seems to have further strengthened citizens' relationship with social media. An increasing number of people now spend more time on

✉ Andreas Kanavos
akanavos@ionio.gr

Sotiria Vernikou
st1004148@ceid.upatras.gr

Athanasios Lyras
lyra@ceid.upatras.gr

[1] Computer Engineering and Informatics Department, University of Patras, Patras, Greece

[2] Department of Digital Media and Communication, Ionian University, Kefalonia, Greece

[1] https://www.worldometers.info/coronavirus/.

[2] https://www.who.int/health-topics/coronavirus.

social media platforms since the outbreak of the health crisis. Twitter has been widely used for sharing ideas, opinions and feelings related to the pandemic due to its popularity and ease of access [3]. Moreover, this platform has been exploited by many government officials worldwide as a communication channel to the general public with the ultimate purpose to regularly share policy updates and news related to the pandemic [4].

Given the unprecedented circumstances of uncertainty that the pandemic created, this paper contributes to understanding the public behavior, and specifically, it quantifies and validates the emotional and psychological conditions that prevail among citizens of different countries around the world. Its ultimate purpose is to characterize the psychological well-being during the early stage of the COVID-19 outbreak, while this infectious disease constitutes a controversial global topic in social media that is worth studying.

This study presents a number of deep learning models which aim at categorizing the sentiment found in the posts of Twitter users. The sentiment that classifiers are utilized to distinguish is either negative, neutral or positive, and the topic of these tweets concerns the COVID-19 pandemic. More to the point, the dataset used in the paper concerns the period between March and April 2020, when the disease had already spread around the world, while the cases were constantly increasing and new measures in order to limit further spread of the disease were announced. The number of tweets was 44, 955 and this was mainly due to the fact that the sentiment that prevails in each recording has been manually categorized. For the sentiment identification that prevails in the tweets, 7 different deep learning models were implemented consisting of long short-term memory (LSTM/BiLSTM) recurrent neural networks and 6 models based on traditional machine learning algorithms.

We highlight that our paper introduces a novel framework that makes use of information from social media for understanding public behavior during the most popular topic of our days, COVID-19 pandemic. Our proposed framework compares the different types of sentiments expressed in a number of months in relation to the rise of the number of cases, which impacted the economy and had different levels of lock downs. We use LSTM and bidirectional LSTM (BiLSTM) model with bag of words (BoW) and term frequency–inverse document frequency (Tf-Idf) for word representation for building a language model. Moreover, we use the BERT model as well as different machine learning models to compare the results from these classification algorithms. This framework is focused on multi-label sentiment classification, consisting of three different classes, namely negative, neutral and positive.

The rest of the paper is organized as follows. Section 2 presents related work on coronavirus tweet analyses. Section 3 overviews the basic concepts and algorithms used in this paper, while in Sect. 4, the implementation details are presented. Section 5 presents the research results, and finally, Sect. 6 depicts conclusions and draws directions for future work.

2 Related work

Social network analysis on COVID-19 tweets using machine learning techniques is considered a popular field of data mining, owing to the extensive and still growing available literature. Initially, authors in [5] used latent Dirichlet allocation (LDA) in order to recognize unigrams, bigrams, salient topics, themes and sentiments from 4 million Twitter messages between March 1 and April 21, 2020, related to the COVID-19 pandemic. The dataset was constructed by using a list of 25 popular hashtags, and the results provided useful insights about health-related emergency situations. Another research paper that uses LDA for topic modeling is the one proposed in [6]. The authors focus on identifying the sentiment that prevails during the coronavirus outbreak, with fear being the most dominant one.

Moreover, the authors in [7] conducted a social network and content analysis on tweets collected between March 27 and April 4, 2020, with the ultimate purpose of understanding what led to the conspiracy theory that 5G towers in the UK are closely related to the spread of the pandemic. Another work considers classifier ensembles formed by diversified components that are promising for tweet sentiment analysis [8]. The authors compared bag of words and feature hashing-based strategies for the representation of tweets and depicted their advantages and drawbacks, where classifier ensembles were obtained from the combination of lexicons, bag of words, emoticons and feature hashing.

Furthermore, a database of COVID-19-related tweets was analyzed aiming to provide insight toward mask usage in [9]. Classification was implemented to separate tweets into different high-level themes and topics within each theme. For each of these clusters, a sentiment profile was built and later on checked as to identify how it changed over a five-month period. Natural language processing techniques were used for applying an abstractive text summarization model.

LSTM neural networks has been widely used for forecasting COVID-19 infection for multiple countries especially in the period of lockdown [10–12]. Additionally, a sentiment analysis research paper based on posts from Sina Weibo, a popular Chinese social media platform, was presented in [13]. The posts were classified into 3 categories (negative, neutral and positive) with the use of a fine-tuned unsupervised BERT model and a Tf-Idf model for topic post identification. Sentiment classification was, also, implemented by the authors of [14]. Specifically, negative and positive sentiment classification was implemented with the use of naive Bayes and logistic regression machine learning techniques. This research paper compares the results of these classifiers on two datasets that consisted of tweets of shorter and longer length, with the former technique being the most accurate in both. Another approach that deals with sentiment classification is that of [15]. The authors examined how the lockdown at the end of March of 2020 affected people in India based on a dataset consisting of 24.000 tweets, which were extracted using two prominent hashtags. The results showed that despite the negative sentiment that appeared, positive tweets were the dominant ones.

Furthermore, the authors in [16] incorporate deep neural networks for the problem of forecasting aviation demand time series, where they utilized various models and identified the best implementation among several strategies. One of the most recent works exhibits an LSTM-CNN based system for classification [17]. Specifically, the classification task was improved as the proposed method reduced the execution time by values ranging from 30 to 42%. Thus, the effectiveness of LSTM neural network and its important contribution for specific tasks was proved.

3 Preliminaries

The classifiers presented are based on deep learning techniques and specifically on LSTM models. A detailed description of the various categorizers is presented, and the way the input data is utilized is explained. The data consist of various types, either textual, after initially received various preprocessing techniques, or purely numerical together with preprocessed textual. After training the models, they are able to categorize the sentiment into three distinct categories, which are negative, neutral and positive.

3.1 Text preprocessing

The text preprocessing phase consists of several steps as we aim to reduce the complexity of our proposed method. Specifically, all characters are converted to lowercase, and the hyperlinks are removed as they do not add any useful linguistic information [18–20]. Furthermore, mentions and hashtags that are often used on Twitter messages to attract other users attention are also eliminated. Regular expressions were also utilized in order to replace some meaningful words, such as the username. In many research papers [8, 21, 22], the removal of these words is followed, but in this case their replacement was chosen due to the fact that deep learning techniques are considered to respond better to as much information as possible, so that the meaning of the sentence is not lost when dealing with text data. This is the reason why stop word removal was not implemented.

Contractions' correction was another module to be utilized as in the English language, and especially when writing informally like in Twitter, many are the words that get converted into shortened versions. The transformation of these words into their initial form is required because the concise and the complete version of a word are treated as two completely different words, thus leading to an increase in the dimensionality and cost of calculation.

Part-of-speech (POS) tagging was in the following implemented so as to extract the useful features and enhance the proposed deep learning model. In this module, each token acquires a special tag, called the POS Tag. This tag indicates the part of speech (noun, verb, adjective, etc.) to which the token belongs, while it contains additional information about its grammatical category.

Furthermore due to the limited number of characters allowed in tweets, abbreviations are widely used on such platforms and need to be efficiently handled. Their conversion back into their initial full form was implemented by manually creating a dictionary which contained the most well-known abbreviations with their complete expression. In this dictionary, the most common emoticons were also included along with their semantic meaning.

An additional step in correcting typographical and spelling errors is the utilization of the "autocorrect" Python library. This particular library replaces words that are not considered to be correctly spelled with words that are perceived as most appropriate based on the letters that this library finds in the word of interest.

Tokenization and lemmatization were also considered; the first module includes the process of disintegrating the text into smaller sections called "tokens". Each term of each tweet is then stored within a token list, and the text's tokens appear based on their natural order. Regarding lemmatization with the utilization of previously produced POS tags and the use of the WordNet Lemmatizer, every word from the tweets was transformed into its base or dictionary form. This is implemented by reducing inflected words into a root word that exists in the vocabulary.

3.2 LSTM architecture

LSTM neural networks belong to the category of recurrent neural networks (RNNs) and were initially proposed in [23]. This variation of RNNs is widely used in the field of deep learning because it has been proved to be very effective in modeling long-term dependencies, while it eliminates the vanishing or exploding gradient problem. Moreover, this category uses a special gateway mechanism that decides which pieces of information to remember, which to update and which to pay attention to. These abilities are based on the cell and do not take into consideration the update as well as the output gate that LSTMs are composed of. The architecture described above is illustrated in Fig. 1.

The equations that govern the operation of LSTMs, based on the previous figure, are presented below.

In Eq. 1, the update gate consists of a sigmoid function that decides which of the new information should be updated or ignored by merging the $\chi^{\langle t\rangle}$ and $\alpha^{\langle t-1\rangle}$ in the memory cell $c^{\langle t\rangle}$.

$$\Gamma_u^{\langle t\rangle} = \sigma\left(W_u\left[\alpha^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_u\right) \tag{1}$$

The forget gate that is presented in Eq. 2 consists of a sigmoid function, which takes the current input of cell ($x^{\langle t\rangle}$) and the output of the previous one ($\alpha^{\langle t-1\rangle}$), deciding which parts of the old output should be removed in order to free a substantial part of memory.

$$\Gamma_f^{\langle t\rangle} = \sigma\left(W_f\left[\alpha^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_f\right) \tag{2}$$

Equation 3 depicts the output gate which consists of a sigmoid function that decides which memory cell information will be extracted.

$$\Gamma_o^{\langle t\rangle} = \sigma\left(W_o\left[\alpha^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_o\right) \tag{3}$$

Equation 4 presents $\tilde{c}$, which is a layer consisting of the hyperfunction tanh that takes the same inputs as before and creates a vector of all possible values from the new input.

$$\tilde{c}^{\langle t\rangle} = \tanh\left(W_c\left[\alpha^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_c\right) \tag{4}$$

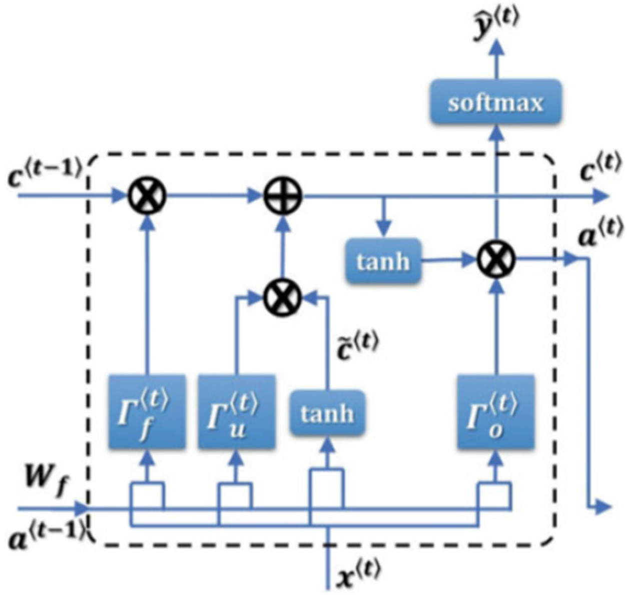
$\hat{y}^{\langle t\rangle}$
softmax
$c^{\langle t-1\rangle}$
$c^{\langle t\rangle}$
$a^{\langle t\rangle}$
tanh
$\tilde{c}^{\langle t\rangle}$
$\Gamma_f^{\langle t\rangle}$
$\Gamma_u^{\langle t\rangle}$
tanh
$\Gamma_o^{\langle t\rangle}$
W_f
$a^{\langle t-1\rangle}$
$x^{\langle t\rangle}$

Fig. 1 LSTM architecture [24]

The new cell state is presented in Eq. 5, where the outputs of the Eqs. 2 and 4 are multiplied to update the new memory cell. This is added to the old memory $c^{\langle t-1\rangle}$ multiplied by the forget gate, so that $c^{\langle t\rangle}$ occurs.

$$c^{\langle t\rangle} = c^{\langle t-1\rangle} \times \Gamma_f^{\langle t\rangle} + \tilde{c}^{\langle t\rangle} \times \Gamma_u^{\langle t\rangle} \tag{5}$$

The memory cell goes through a layer composed of a tanh function creating a vector of all possible values and multiplied with the output gate the hidden state is obtained; this information is forwarded to the next unit of the LSTM.

$$\alpha^{\langle t\rangle} = \Gamma_o^{\langle t\rangle} \times \tanh c^{\langle t\rangle} \tag{6}$$

3.3 Word embeddings

Analyzing word embeddings, we would say that they are a type of representation that allows words with similar meanings to have similar representations [25, 26]. Being considered a significant discovery, they have led to impressive performance of deep learning methods in natural language processing (NLP) problems.

The four most common word embedding techniques are presented below. Initially, the *embedding layer*, where the representation of words are learned together with a neural network model. In the following, the second is *Word2Vec*, which was proposed in [27] and is one of the most popular templates for developing pre-trained word vector representations [28]. The next method is *GloVe*, which is an extension of the previous model [29], proposed in [30–32].

The last technique is the newly proposed embedding method based on transformers, namely as *Bert* [33–36]. Bert is thought to be a state-of-the-art discovery in the field of NLP with the ability to provide better results than other methods [37, 38]. All of the aforementioned techniques are utilized for the implementation of the deep learning models of our paper.

4 Implementation

4.1 Dataset

For the scope of this research paper, the dataset used consisted of tweets related to COVID-19 pandemic[3], with the categorization of sentiment being manually

[3] https://www.kaggle.com/datatattle/covid-19-nlp-text-classification.

implemented. Initially, the dataset consisted of tweets classified into extremely negative, negative, neutral, positive and extremely positive sentiment. Our aiming at classifying the data into three categories led to merging the extremely negative and negative tweets in the same class, while the same happened for the data belonging to the extremely positive and positive class, respectively. As a result, the number of tweets per each sentiment class is presented in Table 1.

The data was collected in the period from 2/3/2020 to 14/4/2020 and the tweets included are exclusively in the English language. The period taken into consideration is when the coronavirus had already spread throughout the world and the pandemic created unprecedented situations with long-term quarantines to reduce the spread of the virus, traveling restrictions, etc. This new everyday life is depicted in this dataset, and specifically in Figs. 2 and 3 with the most common hashtags and bigrams after preprocessing.

4.2 Overview of the deep learning models

In the following, 7 different deep learning models were implemented consisting of long short-term memory (LSTM/BiLSTM) neural networks with the use of the Tensorflow and Keras libraries. The classifiers are structured with the sequential or functional APIs of the aforementioned libraries, regarding the needs of the model.

1. *Simple LSTM* The first layer of the classifier is the Keras embedding layer, which is widely used for text data. This requires the input data to be encoded to integers so that each word is represented by a single integer. After this requirement is met, the embedding layer is initialized with random weights, and during the training, a vector representation of each word of the dataset is created. The output of the previous layer is entered in the SpatialDropout1D layer. This process is performed to prevent the problem of overfitting, and subsequently, the data is introduced into 3 LSTM and 3 batch normalization layers, alternatively. Finally, a dense layer of shape 3 decides which sentiment prevails in the examined tweet. Concretely, the SpatialDropout1D layer is met in models $1-4$ and 7, while the last one is inserted in all 7 models.
2. *GloVe LSTM* GloVe pre-trained embeddings along with 3 LSTM and 3 Dropout layers are utilized. The first layer, as in the previous case, is the embedding layer. The difference, however, is that an already pre-trained vector representation of the words is introduced, as a result of which existing knowledge is transferred to the proposed method. This way, there is no need for embedding layer training with the number of learning parameters to be dramatically reduced.
3. *BiLSTM* The differentiation of this model is related to the choice of the type of the LSTM neural network, where in this case the BiLSTM was selected [39, 40]. Its ability to summarize the content of the text, both forwards and backwards, is tested in [41]. As a result, a Keras embedding layer combined with a SpatialDropout1D, a single BiLSTM and a Dropout layer is implemented.
4. *Gensim LSTM* The embedding layer used here consists of pre-trained word embeddings acquired from the data we had at our disposal. The Word2Vec algorithm was used for the implementation of the vector representation of words, and specifically, the CBoW training method was used. The rest of the model consists of a BiLSTM and a Dropout layer.
5. *Bert LSTM* The word embeddings are produced by the Keras Layer, which receives as input the *input_word_ids*, the *input_mask* and the *segment_ids* of the text data. These inputs are inserted into the pre-trained Bert encoder $bert\_en\_uncased\_L-12\_H-768\_A-12$[4]. This pre-trained encoder is derived from English language texts coming from Wikipedia and the BooksCorpus dataset, which contain about 2, 5 billion and 800 million words, respectively. Keras layer, however, produces two outputs, namely the *pooled_output* and *sequence_output*. The first output contains the vector representation for the entire input phrase (tweet), while the second one contains the representation for each token. Based on the purpose of this work, the output we are interested in is the *sequence_output*, since it contains the word embeddings needed for the rest of the model. The output of this layer is inserted into a BiLSTM and later on a Dropout layer.
6. *Bert Tokenizer LSTM* The tokens that the embedding layer receives as input are based on the Bert Tokenizer, called *WordPiece Tokenizer* [34]. This Tokenizer was chosen to be separately tested due to the innovative way in which it splits words into tokens and we aim at evaluating its behavior throughout our data. As previously, the next layers are the BiLSTM and Dropout.
7. *Text & Numerical Data LSTM* This model receives as input, in addition to the text data, a set of numerical data. The numerical data is related to the following features in Table 2. The text data is inserted into an embedding layer, whose output is inserted into a BiLSTM and then in a Dropout layer. The numerical data, after getting normalized, constitute the input of a dense layer, whose output is getting concatenated with

[4] https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/2.

Table 1 Number of tweets per sentiment

Sentiment	Number of tweets
Negative	17,031
Neutral	8332
Positive	19,592
Total	**44,955**

the output of the Dropout layer of the text data. The output of the concatenation layer forms the input for the final dense layer.

Each deep learning model receives as input preprocessed textual data and makes use of an embedding layer that transforms each word into a vector representation of size equal to 200. In the current work, different kinds of embedding layers are tested. All models, except for Bert LSTM, consist of a SpatialDropout1D layer that is used for avoiding overfitting while training. In this layer, the output of the embedding layer is given as input, while the output of SpatialDropout1D is inserted into an LSTM or BiLSTM, depending on the model. After each LSTM or BiLSTM layer, a Batch Normalization or a Dropout layer is utilized where these layers are used for overfitting avoidance.

Batch Normalization has the ability to normalize the data it receives at input. Specifically, it implements a transformation that keeps the average output close to 0 and

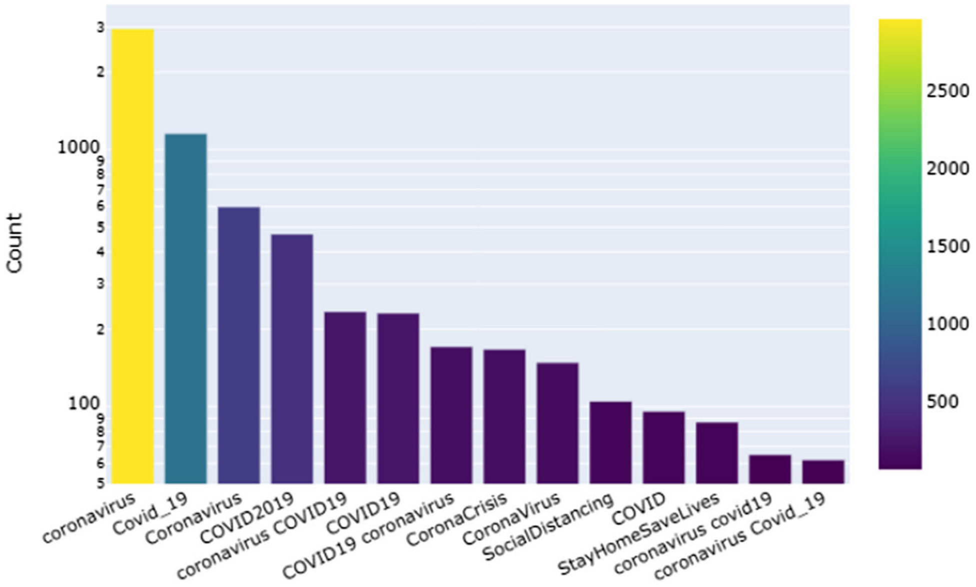

Fig. 2 Most common hashtags in dataset

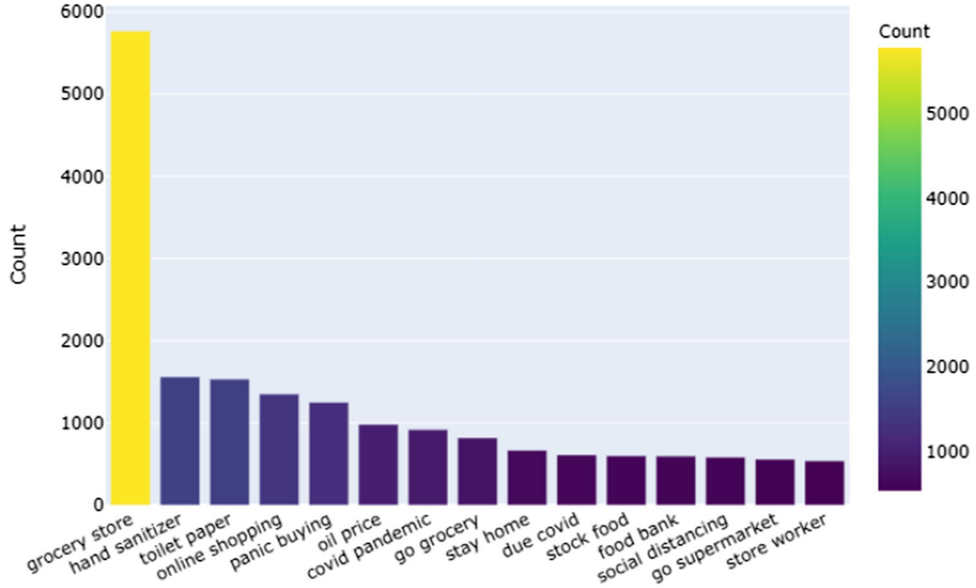

Fig. 3 Most common bigrams in dataset

Table 2 Numerical data features

Word count	Mentions count
Unique words count	Hashtags count
Letters count	URLs count
Stopwords count	Nouns count
Average length of words	Verbs count
Count of exclamation and question marks	Adjectives count
Titlecased words count	Adverbs count
Uppercased words count	Other (except nouns, verbs, adverbs, adjectives)

the standard deviation close to 1. Dropout sets to value equal to 0 different elements of each input with a certain frequency, so that while the model is trained, it does not efficiently learn the training data. Regarding the model of interest, different numbers of LSTM/BiLSTM and Batch Normalization/Dropout layers are used. The output of the last Batch Normalization/Dropout layer is given as input to a dense layer consisting of 3 neurons that is responsible for deciding the sentiment that prevails on the examined tweet.

After the complete design of all the LSTM models, the compile function of the Keras library is employed, which undertakes the configuration of the model for training. The categorical cross-entropy was included in the definitions of this function as a loss function. This function is used to create classifiers with multiple classes. At the same time, Adam was defined as the model optimizer, with the learning rate differing per model.

An example of the architecture of the LSTM models proposed is depicted in Fig. 4.

In order to identify the parameters that lead to the creation of the optimal classifier, different values of the hyperparameters for each model were tested. Regarding batch size, different values between 8 and 128 showed best results depending on the designed classifier. When it comes to embedding size, three different cases were examined, namely 100, 200 and 300. After implementing a set of tests, the results showed that when considering size equals to 100, then not much information related to each word of the tweet is captured, while an embedding dimension of 300 does not contribute to the increase in the detection accuracy.

The learning rate achieved several values ranging from 0.00001 to 0.01. The results showed that Bert Tokenizer LSTM model responds better with a very low learning rate; specifically, the learning rate in this case was equal to 0.00001. On the other hand, simple LSTM has a learning rate of 0.005, and the other models perform better with the default Keras value, that is, 0.001.

Regarding the parameters that lead to the creation of the optimal classifier, the configuration is presented in Table 3.

4.3 Overview of the machine learning models

For comparison purposes, 6 traditional machine learning models are also implemented. These are based on three classifiers, namely naive Bayes, decision tree and random forest, combined with Tf-Idf and bag of words techniques. These techniques are widely used in sentiment classification studies and it is worth taking them under consideration [42–45].

1. *Tf-Idf & Multinomial Naive Bayes* The operation of this classifier is based on the Tf-Idf technique [46] and it evaluates how relevant a word is in a document in terms of a document collection. The *TfidfVectorizer* function of the sklearn library was used to implement the Tf-Idf technique. By creating the matrix that assigned the corresponding Tf-Idf value to each word in each tweet, the data were inserted into a Multinomial Naive Bayes model [47]. This algorithm was chosen due to the fact that it is primarily used in NLP problems.
2. *Tf-Idf & Decision Tree* In this model, the same Tf-Idf technique was followed, where the produced matrix was inserted into a Decision Tree classifier [48].
3. *Tf-Idf & Random Forest* As in the two previous models, the Tf-Idf technique of converting words to numbers was implemented. Random Forest was used as the corresponding classifier [49].
4. *BoW & Multinomial Naive Bayes* This classifier utilized the conversion of words into numbers with the bag of words (BoW) technique [50]. BoW is a widely used NLP algorithm, which is based on the frequency of words in the text. The difference with Tf-Idf is that it just creates a set of vectors calculating the count of word occurrences in the document, while the Tf-Idf model contains information on the more as well as the less important words. Finally, the data were inserted into the Multinomial Naive Bayes classifier.
5. *BoW & Decision Tree* This classifier was designed using the aforementioned BoW technique in conjunction with the Decision Tree algorithm.

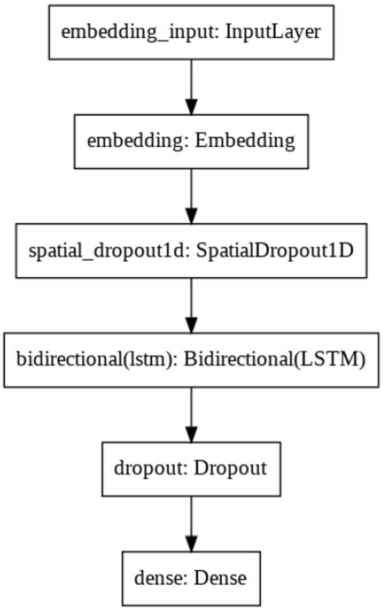
embedding_input: InputLayer
embedding: Embedding
spatial_dropout1d: SpatialDropout1D
bidirectional(lstm): Bidirectional(LSTM)
dropout: Dropout
dense: Dense

Fig. 4 BiLSTM model architecture

Table 3 Parameter configuration

Parameter	Value
Batch size	16–64 (depending on the classifier)
Embedding size	200
Learning rate	0.00001–0.001 (depending on the classifier)
Optimizer	Adam
Loss function	Categorical cross-entropy
Evaluation metric	Accuracy

6. *BoW & Random Forest* The last classifier implemented is the one based on the Random Forest algorithm in combination with the BoW technique.

5 Evaluation

The evaluation of the proposed models has been conducted with the use of the Kaggle dataset, and the models were evaluated based on their ability to classify tweets sentiment effectively. The performance of the models was measured in terms of accuracy, which is one of the most commonly used metrics for the evaluation of a system prediction. Moreover, precision, recall and F1_score were, also, calculated as presented in the following equations. Due to the fact that the problem we are dealing with is related to multi-class classification, for the determination of the values of the evaluation indices (accuracy, precision, recall and F1_score) the one vs. all approach was used.

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \tag{8}$$

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{TP + FP + FN + TN}} \tag{9}$$

$$F1 = 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP + FN})} \tag{10}$$

The number of samples that are correctly classified in terms of sentiment is those called true positive (TP). Instead, all those samples that were considered by the classifier to belong to a class when they actually do not are called false positive (FP). At the same time, those items that are classified in a class other than the one they truly belong to are considered as false negative (FN). Finally, all other elements are called true negative (TN). To the latter category belong those items that neither belong to nor were classified in the class of interest.

The accuracy, precision, recall and F1 score of the proposed models are presented in Table 4. Bert Tokenizer LSTM achieved the best performance in terms of the four metrics, followed by Text & Numerical Data LSTM, BiLSTM and Bert LSTM. The traditional machine learning approaches, e.g., Tf-Idf and BoW, as expected, performed worse than corresponding deep learning techniques. The highest value of accuracy is 90%, whereas the lowest is 61% and the same stands for the other 3 metrics as well for Bert Tokenizer LSTM and Tf-Idf & Decision Tree, respectively.

The confusion matrix of the Bert Tokenizer LSTM model is illustrated in Fig. 5. The diagonal elements represent the number of tweets for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier. The higher the diagonal values of the confusion matrix the better, indicating many correct predictions, that is the correctly classified tweets.

The next part of evaluation includes the results assessment in varying train/validation and test set percentages. Specifically, the three following configurations were utilized in order to measure the efficacy and compare the settings of the proposed models.

1. 60% train/validation - 40% test
2. 70% train/validation - 30% test
3. 80% train/validation - 20% test

Table 4 Accuracy, precision, recall and F1 score of proposed models

Model	Accuracy	Precision	Recall	F1 score
Simple LSTM	0.78	0.79	0.78	0.78
GloVe LSTM	0.80	0.80	0.80	0.80
BiLSTM	0.83	0.83	0.83	0.83
Gensim LSTM	0.67	0.70	0.67	0.68
Bert LSTM	0.82	0.82	0.82	0.82
Bert Tokenizer LSTM	**0.90**	**0.90**	**0.90**	**0.90**
Text & Numerical data LSTM	0.83	0.83	0.83	0.83
Tf-Idf & Multinomial naive Bayes	0.65	0.67	0.65	0.60
Tf-Idf & Decision tree	0.61	0.61	0.61	0.61
Tf-Idf & Random forest	0.70	0.70	0.70	0.70
BoW & Multinomial naive Bayes	0.69	0.69	0.69	0.69
BoW & Decision tree	0.63	0.63	0.63	0.63
BoW & Random forest	0.72	0.72	0.72	0.71

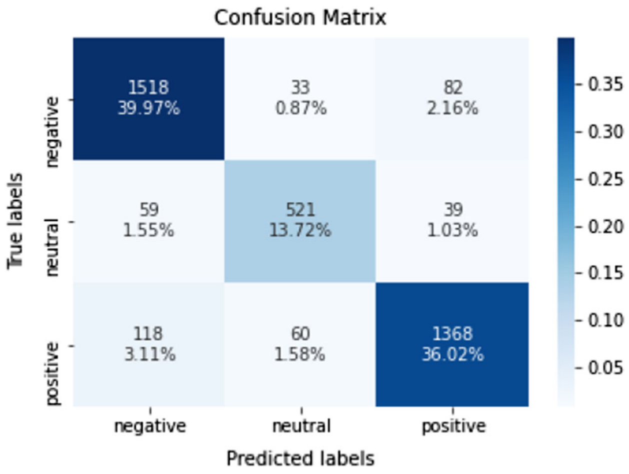

Fig. 5 Confusion Matrix of Bert Tokenizer LSTM model

The results of the three different configurations are presented in Tables 5, 6 and 7. As given in Table 4, Bert Tokenizer LSTM has the highest values for all three different data splits, followed by Text & Numerical Data LSTM, BiLSTM and Bert LSTM, whereas the worst metrics are considered for Tf-Idf and bag of words (BoW) techniques.

Finally, Table 8 presents the results for all the proposed models for each one of the three sentiments utilized in our paper. We can see that for the majority of techniques, positive tweets achieve highest percentage to be correctly classified in contrast to negative and neutral. Bert Tokenizer LSTM achieves the highest values for all three metrics, whereas Tf-Idf and bag of words (BoW) techniques have the lowest values.

Table 5 Model evaluation with data split 60/40

Model	Accuracy	Precision	Recall	F1 score
Simple LSTM	0.79	0.78	0.78	0.79
GloVe LSTM	0.79	0.79	0.79	0.79
BiLSTM	0.82	0.82	0.82	0.82
Gensim LSTM	0.63	0.67	0.63	0.63
Bert LSTM	0.81	0.81	0.81	0.81
Bert Tokenizer LSTM	**0.88**	**0.89**	**0.88**	**0.88**
Text & Numerical data LSTM	0.83	0.83	0.83	0.83
Tf-Idf & Multinomial naive Bayes	0.65	0.67	0.65	0.59
Tf-Idf & Decision tree	0.59	0.59	0.59	0.59
Tf-Idf & Random forest	0.69	0.68	0.69	0.68
BoW & Multinomial naive Bayes	0.67	0.66	0.67	0.67
BoW & Decision tree	0.60	0.61	0.60	0.61
BoW & Random forest	0.69	0.69	0.69	0.69

Table 6 Model evaluation with data split 70/30

Model	Accuracy	Precision	Recall	F1 score
Simple LSTM	0.82	0.82	0.82	0.82
GloVe LSTM	0.81	0.81	0.81	0.81
BiLSTM	0.84	0.84	0.84	0.84
Gensim LSTM	0.63	0.65	0.63	0.63
Bert LSTM	0.83	0.83	0.83	0.83
Bert Tokenizer LSTM	**0.91**	**0.91**	**0.91**	**0.91**
Text & Numerical data LSTM	0.84	0.84	0.84	0.84
Tf-Idf & Multinomial naive Bayes	0.65	0.68	0.65	0.60
Tf-Idf & Decision tree	0.60	0.60	0.60	0.60
Tf-Idf & Random forest	0.70	0.69	0.70	0.69
BoW & Multinomial naive Bayes	0.67	0.67	0.67	0.67
BoW & Decision tree	0.61	0.61	0.61	0.61
BoW & Random forest	0.71	0.71	0.71	0.70

Table 7 Model evaluation with data split 80/20

Model	Accuracy	Precision	Recall	F1 score
Simple LSTM	0.83	0.83	0.83	0.83
GloVe LSTM	0.83	0.84	0.83	0.83
BiLSTM	0.85	0.85	0.85	0.85
Gensim LSTM	0.68	0.71	0.68	0.69
Bert LSTM	0.84	0.84	0.84	0.84
Bert Tokenizer LSTM	**0.91**	**0.91**	**0.91**	**0.91**
Text & Numerical data LSTM	0.85	0.85	0.85	0.85
Tf-Idf & Multinomial naive Bayes	0.65	0.69	0.65	0.60
Tf-Idf & Decision tree	0.60	0.61	0.60	0.60
Tf-Idf & Random forest	0.71	0.71	0.71	0.70
BoW & Multinomial naive Bayes	0.67	0.67	0.67	0.67
BoW & Decision tree	0.64	0.64	0.64	0.64
BoW & Random forest	0.72	0.72	0.71	0.71

6 Conclusions and future work

In this paper, a set of models was implemented, which aims to categorize the sentiment of tweets posted from users of the Twitter platform. Specifically, the sentiment that classifiers have to forecast is either negative, neutral or positive and the topic of the tweets is focused on the COVID-19 pandemic, which appeared in December 2019. More to the point, this particular dataset used in our work, consisting of 44, 955 tweets, concerns the start of the pandemic crisis, and specifically the period March–April 2020, when the COVID-19 cases are constantly increasing with the government of each country announcing new measures to limit the spread.

In order to predict the sentiment of each tweet from the dataset, 7 different deep learning models were created consisting of LSTM neural networks and 6 models based on traditional machine learning techniques. The results obtained from the evaluation of the models based on the analysis of the sentiment vary, depending on the classifier and/or the word embedding technique, that was designed. The best classifier according to all the evaluation tests implemented is the Bert Tokenizer LSTM with appreciable difference from the second ones. Deep learning classifiers, except Gensim LSTM (due to lack of enough data for accurate word embeddings creation), are able to classify the sentiment with accuracy that exceeds 78%, whereas the maximum percentage of the traditional machine learning models is 72%. When using different train/validation and test splits, the results vary with the deep learning models being the most accurate.

Regarding future work, variations and combinations of the proposed set of models presented in this work are worth trying, in order to study whether it is possible to further improve the accuracy. Furthermore, the existing classifiers could be tested in larger datasets to verify the high levels of accuracy achieved in sentiment detection. In addition to the larger volume of the dataset, it is important to add more

Table 8 Precision, recall and F1 score of proposed models for each sentiment

Model	Sentiment	Precision	Recall	F1 score
Simple LSTM	Negative	0.79	0.81	0.80
	Neutral	0.63	**0.84**	0.72
	Positive	0.86	0.72	0.79
GloVe LSTM	Negative	0.80	0.83	0.82
	Neutral	0.69	0.79	0.74
	Positive	0.84	0.77	0.80
BiLSTM	Negative	0.83	0.85	0.84
	Neutral	0.79	0.78	0.78
	Positive	0.84	0.82	0.83
Gensim LSTM	Negative	0.72	0.70	0.71
	Neutral	0.48	0.76	0.59
	Positive	0.77	0.60	0.67
Bert LSTM	Negative	0.83	0.83	0.83
	Neutral	0.77	0.77	0.77
	Positive	0.82	0.83	0.82
Bert Tokenizer LSTM	Negative	**0.90**	**0.93**	**0.91**
	Neutral	**0.85**	**0.84**	**0.85**
	Positive	**0.92**	**0.88**	**0.90**
Text & Numerical data LSTM	Negative	0.82	0.87	0.84
	Neutral	0.83	0.74	0.78
	Positive	0.84	0.83	0.83
Tf-Idf & Multinomial naive Bayes	Negative	0.74	0.68	0.71
	Neutral	0.71	0.04	0.07
	Positive	0.58	0.85	0.69
Tf-Idf & Decision tree	Negative	0.60	0.59	0.60
	Neutral	0.50	0.52	0.51
	Positive	0.67	0.66	0.67
Tf-Idf & Random forest	Negative	0.72	0.67	0.69
	Neutral	0.69	0.53	0.60
	Positive	0.70	0.81	0.75
BoW & Multinomial naive Bayes	Negative	0.72	0.75	0.73
	Neutral	0.57	0.42	0.49
	Positive	0.70	0.74	0.72
BoW & Decision tree	Negative	0.62	0.58	0.60
	Neutral	0.53	0.59	0.56
	Positive	0.68	0.68	0.68
BoW & Random forest	Negative	0.75	0.68	0.71
	Neutral	0.66	0.57	0.61
	Positive	0.72	0.81	0.76

numerical features than those contained in the set used for the purposes of this work. Moreover, the inefficiencies of single models can be resolved by applying several combination techniques, which will lead to more accurate results as in [51, 52]. Finally, the impact of explainable machine learning can be also considered for future work as the produced explainable models will maintain a high level of learning performance (in terms of prediction accuracy) as well as will explain their rationale, characterize their strengths and weaknesses, and convey an understanding of how they will behave in the future [53, 54]. These models will be combined with state-of-the-art human–computer interface techniques capable of translating models into understandable and useful explanation dialogues for the end user.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Xu Z, Shi L, Wang Y, Zhang J, Huang L, Zhang C, Liu S, Zhao P, Liu H, Zhu L, Tai Y, Bai C, Gao T, Song J, Xia P, Dong J, Zhao J, Wang FS (2020) Pathological findings of Covid-19 associated with acute respiratory distress syndrome. Lancet Respir Med 8(4):420–422
2. Yuki K, Fujiogi M, Koutsogiannaki S (2020) Covid-19 pathophysiology: a review. Clin Immunol 215(108):427
3. Ni MY, Yang L, Leung CMC, Li N, Yao XI, Wang Y, Leung GM, Cowling BJ, Liao Q (2020) Mental health, risk factors, and social media use during the Covid-19 epidemic and cordon sanitaire among the community and health professionals in Wuhan, china: cross-sectional survey. JMIR Ment Health 7(5):e19009
4. Rufai SR, Bunce C (2020) World leaders' usage of twitter in response to the Covid-19 pandemic: a content analysis. J Public Health 42(3):510–516
5. Xue J, Chen J, Hu R, Chen C, Zheng C, Su Y, Zhu T (2020) Twitter discussions and emotions about the Covid-19 pandemic: machine learning approach. J Med Internet Res 22(11):e20550
6. Kaila RP, Prasad AVK (2020) Informational flow on twitter-corona virus outbreak-topic modelling approach. Int J Adv Res Eng Technol (IJARET) 11(3)
7. Ahmed W, Vidal-Alaball J, Downing J, Seguí FL (2020) Covid-19 and the 5g conspiracy theory: social network analysis of twitter data. J Med Internet Res 22(5):e19458
8. da Silva NFF, Hruschka ER, Hruschka ER (2014) Tweet sentiment analysis with classifier ensembles. Decis Support Syst 66:170–179
9. Sanders AC, White RC, Severson LS, Ma R, McQueen R, Paulo HCA, Zhang Y, Erickson JS, Bennett KP (2020) Unmasking the conversation on masks: natural language processing for topical sentiment analysis of Covid-19 twitter discourse. medRxiv
10. Chandra R, Jain A, Chauhan DS (2021) Deep learning via LSTM models for COVID-19 infection forecasting in India. CoRR abs/2101.11881
11. Tiwari A, Gupta R, Chandra R (2021) Delhi air quality prediction using LSTM deep learning models with a focus on COVID-19 lockdown. CoRR abs/2102.10551
12. Zeroual A, Harrou F, Dairi A, Sun Y (2020) Deep learning methods for forecasting Covid-19 time-series data: a comparative study. Chaos Solitons Fractals 140(110):121
13. Wang T, Lu K, Chow K, Zhu Q (2020) COVID-19 sensing: negative sentiment analysis on social media in china via BERT model. IEEE Access 8:162–169
14. Samuel J, Ali GGMN, Rahman MM, Esawi E, Samuel Y (2020) COVID-19 public sentiment insights and machine learning for tweets classification. Information 11(6):314
15. Barkur G, Vibha Kamath GB (2020) Sentiment analysis of nationwide lockdown due to covid-19 outbreak: evidence from India. Asian J Psychiatry 51(102):089
16. Kanavos A, Kounelis F, Iliadis L, Makris C (2021) Deep learning models for forecasting aviation demand time series. Neural Comput Appl 33(23):16329–16343
17. Savvopoulos A, Kanavos A, Mylonas P, Sioutas S (2018) LSTM accelerator for convolutional object identification. Algorithms 11(10):157
18. Kaur J, Buttar PK (2018) A systematic review on stopword removal algorithms. Int J Future Revolut Comput Sci Commun Eng 4(4):207–210
19. Luhn HP (1960) Keyword-in-context index for technical literature (kwic index). Am Doc 11(4):288–295
20. Lyras A, Vernikou S, Kanavos A, Sioutas S, Mylonas P (2021) Modeling credibility in social big data using LSTM neural networks. In: 17th international conference on web information systems and technologies (WEBIST), pp 599–606
21. Ankit Saleena N (2018) An ensemble classification system for twitter sentiment analysis. Procedia Comput Sci 132:937–946
22. Parveen H, Pandey S (2016) Sentiment analysis on twitter dataset using naive bayes algorithm. In: 2nd international conference on applied and theoretical computing and communication technology (iCATccT), pp 416–419
23. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
24. Agarwal B, Mittal N (2016) Prominent feature extraction for sentiment analysis. Springer, London
25. Kusner MJ, Sun Y, Kolkin NI, Weinberger KQ (2015) From word embeddings to document distances. In: 32nd international conference on machine learning (ICML), JMLR workshop and conference proceedings, vol 37, pp. 957–966
26. Zhao J, Zhou Y, Li Z, Wang W, Chang K (2018) Learning gender-neutral word embeddings. CoRR abs/1809.01496
27. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. In: 1st international conference on learning representations (ICLR)
28. Chang C, Lee S, Lai C (2017) Weighted word2vec based on the distance of words. In: International conference on machine learning and cybernetics (ICMLC), pp 563–568
29. Brownlee J (2017) Deep learning for natural language processing: develop deep learning models for your natural language problems. Mach Learn Mastery
30. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
31. Sharma Y, Agrawal G, Jain P, Kumar T (2017) Vector representation of words for sentiment analysis using glove. In: International conference on intelligent communication and computational techniques (ICCT), pp 279–284
32. Tifrea A, Bécigneul G, Ganea O (2018) Poincaré glove: Hyperbolic word embeddings. CoRR abs/1810.06546
33. Clark K, Khandelwal U, Levy O, Manning CD (2019) What does BERT look at? an analysis of bert's attention. CoRR abs/1906.04341
34. Devlin J, Chang M, Lee K, Toutanova K (2018) BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805
35. Sanh V, Debut L, Chaumond J, Wolf T (2019) Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR abs/1910.01108
36. Su Y, Xiang H, Xie H, Yu Y, Dong S, Yang Z, Zhao N (2020) Application of bert to enable gene classification based on clinical evidence. BioMed Research International 2020
37. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized BERT pretraining approach. CoRR abs/1907.11692
38. Tenney I, Das D, Pavlick E (2019) BERT rediscovers the classical NLP pipeline. CoRR abs/1905.05950
39. Fan Y, Qian Y, Xie F, Soong FK (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In: 15th

annual conference of the international speech communication association (INTERSPEECH), pp 1964–1968

40. Graves A, Jaitly N, Mohamed A (2013) Hybrid speech recognition with deep bidirectional LSTM. In: IEEE workshop on automatic speech recognition and understanding (ASRU), pp 273–278
41. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45(11):2673–2681
42. Karthika P, Murugeswari R, Manoranjithem R (2019) Sentiment analysis of social media network using random forest algorithm. In: IEEE international conference on intelligent techniques in control, optimization and signal processing (INCOS), pp 1–5
43. Sharma A, Dey S (2012) A comparative study of feature selection and machine learning techniques for sentiment analysis. In: Research in applied computation symposium (RACS), pp 1–7
44. Troussas C, Virvou M, Espinosa KJ, Llaguno K, Caro JDL (2013) Sentiment analysis of facebook statuses using naive bayes classifier for language learning. In: 4th international conference on information, intelligence, systems and applications (IISA), pp 1–6
45. Wang M, Cao D, Li L, Li S, Ji R (2014) Microblog sentiment analysis based on cross-media bag-of-words model. In: international conference on internet multimedia computing and service (ICIMCS), pp. 76
46. Baeza-Yates RA, Ribeiro-Neto BA (1999) Modern information retrieval. Addison-Wesley, Boston
47. Rish I (2001) An empirical study of the Naive Bayes classifier. IJCAI Workshop Empir Methods Artif Intell 3:41–46
48. Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106
49. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
50. Wallach HM (2006) Topic modeling: Beyond bag-of-words. In: 23rd international conference on machine learning (ICML), vol 148, pp. 977–984
51. Drakopoulos G, Kanavos A, Tsakalidis AK (2016) Evaluating twitter influence ranking with system theory. In: 12th international conference on web information systems and technologies (WEBIST), pp 113–120
52. Kyriazidou I, Drakopoulos G, Kanavos A, Makris C, Mylonas P (2019) Towards predicting mentions to verified twitter accounts: building prediction models over mongodb with keras. In: 15th international conference on web information systems and technologies (WEBIST), pp. 25–33
53. Gunning D, Stefik M, Choi J, Miller T, Stumpf S, Yang G (2019) XAI-explainable artificial intelligence. Sci Robot 4(37):eaay7120
54. Roscher R, Bohn B, Duarte MF, Garcke J (2020) Explainable machine learning for scientific insights and discoveries. IEEE Access 8:42200–42216

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.