Sentiment Analysis for People's Opinions about COVID-19 Using LSTM and CNN Models

https://doi.org/10.3991/ijoe.v19i01.35645

Maisa Al-Khazaleh[1], Marwah Alian[1](✉), Mariam Biltawi[2], Bayan Al-Hazaimeh[1]
[1]Basic Sciences Department, Faculty of Science, The Hashemite University, Zarqa, Jordan
[2]Computer Science Department, Al Hussein Technical University, Amman, Jordan
marwahm@hu.edu.jo

Abstract—The emergence of social media platforms, which contributed in activating the patterns of connection between individuals, leads to the availability of a huge amount of content such as text, images, and videos. Twitter is one of the most popular platforms of social media that encourage researchers to investigate people's feelings and opinions among through sentiment analysis studies that elicited the interest of researchers in natural language processing field. Many techniques related to machine learning and deep learning models could be used to improve the efficiency and performance of sentiment analysis, especially in complex classification problems. In this paper, different models of long short-term memory recurrent neural network are used for the sentiment classification task. The input text was represented as vectors using Arabic pre-trained word embedding (Aravec). Experiments were conducted using different dimensions of Aravec on 15779 tweets about COVID-19 collected and labeled as positive and negative. The experimental results show an accuracy value of 98%.

Keywords—Arabic sentiment analysis, Aravec word embedding, convolutional neural network, deep learning, long short term memory, COVID-19

1 Introduction

The pandemic caused by COVID-19 resulted in outbreaks and lockdowns around the world. Since its emergence in the end of 2019, the pandemic affected people's lives in different fields, such as social life, psychological, learning and teaching, healthcare, and finance [1] [47]. During this phase, people used social media platforms, such as Twitter and Facebook, to express their feelings and opinions about the current situation, thereby encouraging researchers to investigate people's feelings among these social platforms through sentiment analysis studies [2]. However, Twitter is considered as one of the most popular social platforms, because of its availability and ease of knowledge exchange [3]. During the pandemic, people have turned to social media sites to continue their social connections despite the lockdowns and restrictions imposed by governments, which in turn increased the use of these social networks by 61% [4].

Sentiment analysis that is also known as sentiment classification or opinion mining, is a text-mining approach that analyzes and extracts subjective information from a text to transform unstructured text into meaningful and valuable information [48].

It is one of NLP applications that needs machine learning algorithms as the classification process [49]. It is considered a complex process with five steps, which begins with collecting data and continues with preprocessing text, detecting sentiment, and classifying text into positive, negative, and other categories. The final step is to present the output [2], [5].

D'Andrea et al. classified the techniques applied for sentiment analysis [5] into three categories, namely, lexicon-based, machine learning (ML), and hybrid approaches that incorporate both ML and lexicon-based approaches. ML approaches split the data into training and testing sets to predict the polarity of emotions, while the lexicon-based approaches work with a predetermined list of words, and each of which is linked to a certain emotion. Furthermore, ML approaches can be referred to as supervised learning, while lexicon-based approaches are referred to as unsupervised learning [6].

Recently, deep learning (DL) is used in the domain of natural language processing (NLP). One of the dominant methods of DL is the recurrent neural networks (RNNs), and the long short-term memory recurrent neural network (LSTM) is one of its gated versions used in different NLP applications, such as classification tasks, sentiment analysis, and many others.

This research proposes the use of different DL models, such as LSTM and convolutional neural networks (CNN) with Aravec embedding [7] to represent input words, the evaluation of the approach was performed using a collected dataset from Twitter social media and different Aravec representative models, such as Skip-gram and CBOW trained on tweets with vector dimensions of 300 and 100. The rest of this research is organized as follows; Section 2 represents previous related work. Section 3 explains methodology, concepts and background methods used in the experimented models. Then, a description for the conducted experiments and utilized dataset is shown in Section 4. In Section 5, a discussion for the results is provided. Then, we conclude in Section 6.

2 Related works

Several studies and experiments have been tested in sentiment analysis for Arabic texts. Biltawi et al. [8] presented a comprehensive survey of sentiment classification that was conducted on Arabic language. They classified 32 surveyed papers into three categories, namely, the lexicon-based, ML-based, and hybrid-based methods. According to this survey, social media platforms, including Twitter, are considered the most efficient data source for Arabic sentiment analysis research. They also considered that Arabic sentiment classification remains an open area for research.

Alwehaibi et al. [9] proposed an optimized sentiment classification for dialectal short text at Arabic document level. They extracted semantic features at the word and character levels for Arabic short text. Then, they utilized LSTM, CNN, and a model that combines both CNN and LSTM to improve the efficiency. They also applied a hyper parameter tuning estimation approach. To evaluate their approach, they used a dataset of dialectal Arabic corpus and modern standard Arabic collected from Twitter to train and test the three models. The results reported an accuracy that ranged between 84% and 96.7% for all tested models. They also employ a loss value in the range of 0.29 and 3.4.

Biltawi et al. [10] proposed a hybrid model that combined the lexicon-based and the corpus-based approaches for Arabic text sentiment. They evaluated their model using two different datasets, the OCA and Twitter, and compared the results with that of the corpus-based approach. The hybrid approach outperformed the corpus-based approach with an accuracy of 96.34% using random forest with six-fold cross-validation.

In 2019, Biltawi et al. [11] proposed a fuzzy logic, lexicon-based approach to analyze sentiment in Arabic text. The authors verified their approach in two independent experiments using a large-scale Arabic book review dataset. The highest accuracy value achieved was 80.59%.

Ahmed et al. [12] analyzed Arabic tweets about COVID-19 for sentiments using five different ML models, namely, support vector machine (SVM), Naïve Bayes, random forest, logistic regression and K-Nearest neighbor. They evaluated the five models using Arabic Sentiment Twitter Corpus (ASTC) [13]. The results show that the k-NN model gains the lowest accuracy value of 63.23%, and the SVM model provides the best accuracy value of 84.14%.

Alturayeif and Luqman [2] used two transformer-based models, namely, AraBERT [14] and MARBERT [15], with a loss function that is weighted dynamically (DWLF) to analyze the sentiment of Arabic tweets. They evaluated their proposed method using SenWave and SenAIT datasets [16]. The results show that the proposed BERT-based models with emoji replacement and DWLF technique improved the sentiment classification of multi-dialect Arabic tweets with an F1-score value of 0.72.

Alhazmi and Alharbi [17] investigated the emotions twitted by Saudis during the COVID-19's final stage of lockdown. Then, they classified these emotions into eight categories such as fear, anger, trust, anticipation, surprise, joy, sadness, and disgust, as in NCR [18]. Also, they attempted to detect the changing dynamics of expressed emotions. The results show that although positive emotions predominated in the early ending stage, negative emotions were also noticed, mainly due to the uncertainty toward COVID-19.

AlZoubi et al. [19] developed several innovative techniques to analyze the emotion intensity of Arabic tweets. They used three DL models, namely, bidirectional GRU with CNN, CNN, and XGBoost regressor (XGB). To evaluate their proposed techniques, they use the dataset of SemEval-2018 Task1, which is a reference dataset with more than 1,169,075,128 tokens. The model resulted in a Pearson value of 69.2%, and an enhancement of 0.7% is also provided compared with previous best-performing state-of-the-art used models.

Albukhitan et al. [20] applied deep learning technology to produce semantic annotation for Arabic web resources. The proposed framework relies on one linking noun-phrases with concepts from a corresponding ontology. They used word embedding models and two matching verb-phrase methods and employed ontology relationships between concepts. Their approach is still emerging and needs more work to improve its performance.

However, convolutional neural networks (CNN) and long short-term memory (LSTM) have obtained extensive attention as promising methods for sentiment analysis.

For example, Heikal et al. [21] explored the performance of three DL models, namely, CNN, LSTM, and merged CNN-LSTM models with AraVec word embedding, to predict the sentiments of an Arabic Twitter dataset ASTD [22]. The ensemble CNN-LSTM model achieves the best F1 score with 64.46% value.

Also, Alayba et al. [23] attempted to study the advantages of combining two neural networks models on different Arabic sentiment datasets, Main Arabic Health Services (Main-AHS) dataset [24] and Sub-AHS dataset [25] by applying character N-Gram level (ch5gram) and word level sentiments. The proposed model achieved an accuracy value of 0.9424 when applied to the Main Arabic Health Services (Main-AHS) dataset with word-level sentiment while obtaining an accuracy value of 0.9568 when applied to the Sub-AHS dataset with Ch5-gram-level.

Meanwhile, other researchers compared the performance of traditional ML models with DL models. The results of their studies proved that DL models outperform ML models. For example, Elzayady et al. [26] compared three regular machine learning methods, K-Nearest Neighbor (KNN), Naïve Bayes, and DT with two deep learning models: LSTM and CNN. These techniques were applied to Arabic Hotel Reviews (HTL) dataset [27] and Arabic Book Reviews (LABR) dataset [28]. The results show that the combined CNN-LSTM model achieved a competitive average accuracy value of 86.88% and 85.83% when applied to LABR and HTL datasets, respectively. Oussous et al. [29] showed that the CNN and LSTM models on Moroccan Sentiment Analysis Corpus (MSAC) outperformed NB, SVM, and ME classifiers with different preprocessing techniques. Furthermore, Ombabi et al. [30] studied the performance of CNN and LSTM with different embedding models used for the input layer. The experiment was conducted on a multi-domain sentiment corpus [27] [28] where the best accuracy value (90.75%) was achieved when one CNN layer and two LSTM layers were applied with FastText skip-gram word embedding model. Alayba & Palade [31] proposed a CNN-LSTM model without the use of max-pooling layer with various word embedding models; GloVe, Word2Vec, and FastText. Also, they investigated various word normalization techniques, such as Madirma, Farasa, and Stanford. They evaluate their model using Arabic Health Services AHS dataset [25], Ar-Twitter dataset [32], and Arabic Sentiment Tweets Dataset (ASTD) [22]. Their model achieves accuracy value of 0.948 for Main-AHS dataset using Farasa Lemmatization, 0.889 for Ar-Twitter dataset using Madamira Stemming, and accuracy value of 0.8162 for the ASTD dataset using Word2Vec skip-gram embedding model with 200 dimension vectors.

A comparison among the previously mentioned related approaches in terms of reference, year, proposed method, dataset, evaluation metric, and results is shown in Table 1.

Table 1. Comparison of the methods proposed for Arabic sentiment

Paper	Year	Methods	Dataset	Data Size	Metric	Results
(AlZoubi, et al., 2020)	2020	Bidirectional GRU with CNN, CNN, and XGBoost regressor (XGB)	Arabic tweets dataset, Emotion Intensity Regression (EI-reg)	1,169,075,128 tokens	Pearson	69.2%
(Alturayeif & Luqman, 2021)	2021	Skip-Gram and CBOW	Arabic COVID-19 tweets.	13,019 tweets	F1-Score	0.72
(Alwehaibi, et al., 2021)	2021	LSTM, CNN, and an ensemble LSTM-CNN model	AraSenTi dataset	15 K balanced tweets	Accuracy	88%- 69.7%
(Biltawi,et al., 2017)	2017	Random forest, Naive Bayes, SVM, Maximum Entropy, BAGGING, BOOSTING, Neural Network, Random Forest, and Decision Tree.	Opinion Corpus for Arabic (OCA) and Twitter	1000 text files in each folder, each file consists of a single review, shorter than the reviews in the OCA corpus.	Accuracy	96.34%
(Ahmed,et al., 2021)	2021	Naïve Bayes, Support Vector Machine, Logic Regression, Random Forest, and K-Nearest Neighbor	Arabic tweets related to COVID-19 Arabic Sentiment Twitter Corpus (ASTC)	58,000 Arabic tweets	Accuracy	84%
(Albukhitan, et al., 2020)	2020	Word2Vec CBOW and Skip-gram with Mean Vectorization and Cosine similarity	A collected set of documents related to Nutrition, Food, and Health.	150 Web documents	Precision Recall	80.6 80.8
(Heikal, et al., 2018)	2018	CNN, LSTM, Ensemble (CNN-LSTM)	Arabic Twitter dataset ASTD	10,000 tweets	Accuracy, F1-measure	65.05% 64.46 %

(Continued)

Table 1. Comparison of the methods proposed for Arabic sentiment *(Continued)*

Paper	Year	Methods	Dataset	Data Size	Metric	Results
(Alayba,et al., 2018)	2018	CNN-LSTM	Main-AHS Sub-AHS Ar-Twitter ASTD	2026 tweets 732 tweets 2000 tweets 54,000 tweets	Accuracy	94.24% 95.68 % 88.10 % 79.07 %
(Elzayady, et al., 2020)	2020	ML models: NB, (KNN), and decision trees DL models: LSTM and CNN	Arabic Hotel Reviews (HTL) Arabic Book Reviews (LABR)	15,000 Arabic reviews 16,448 book reviews	Accuracy	85,83% 86,88%
(Oussous, et al., 2020)	2020	ML models: NB, SVM, ME DL models: CNN and LSTM	Moroccan Sentiment Analysis Corpus (MSAC)	2,000 reviews	Accuracy	99% with CNN
(Ombabi, et al., 2020)	2020	CNN and LSTM	multi-domain sentiment corpus	15.100 training 4,000 testing	Accuracy	90.75%
(Alayba, et al., 2017)	2021	Ensemble (CNN & LSTM)	Main-AHS Sub-AHS Ar-Twitter dataset Arabic Sentiment Tweets Dataset (ASTD)	2026 tweets. 1732 tweets 2000 tweets 10,006 tweets	Accuracy	94.83% 96.8% 88.86 81.62%

While the studies mentioned above focused on applying deep learning and machine learning techniques to Arabic datasets, other studies used other techniques for English sentiment analysis; such as the capsule network that was investigated by Demotte et al. [51]. They proposed to use shallow, deep, and ensemble capsule networks for sentiment classification with two datasets collected from Twitter. They also explored the use of static and dynamic routing methods to enhance the accuracy of text classification. The results of their experiment show accuracy of 0.869 for Stanford Twitter Sentiment Gold dataset with the shallow capsule network, dynamic routing and crawl Glove word embedding.

However, some studies focused on other low resource languages, such as Sinhala. The work of Meedeniyal and Perera [52] evaluated the categorization of Sinhala documents by proposing a model based on Latent Semantic Analysis, Gaussian Mixture model, and k-means clustering while Lenadora et al. [53] tried to investigate the behavior of Sri Lankan people posts on Facebook during COVID-19, where the behavioral patterns, topic significance, and topics co-occurrence where analyzed.

In this research, we proposed to use different deep learning models with LSTM and CNN for Arabic sentiment classification task on COVID-19 tweets with two classes, namely, positive and negative. Also, we employed CBOW and Skip-gram Aravec pre-trained vectors as input to the models.

3 Methodology

Four deep learning models are proposed to be applied to Arabic sentiment analysis based on text representation methods and DL methods. The techniques that constructed the proposed models are described in the following subsections.

3.1 Word embedding

Word embedding refers to a representation that captures the semantic relations between words. Each word is implemented as a vector of real numbers in the dimensional space where words with similar vector representations would be considered semantically similar.

AraVec refers to a pre-trained word embedding model for Arabic [18]. AraVec has 16 different learning word embedding models that have been trained using Twitter and Arabic articles from Wikipedia with vector dimensions of 100 and 300 [33] [54]. These articles and tweets are trained using an adapted version of Word2Vec models [34], the CBOW, and the skip-gram.

To obtain results with higher accuracy, we used AraVec word embedding as an embedding input layer for the tested DL models where each word is used as an input in a sequence.

3.2 Long short term memory networks (LSTM)

LSTM network is a special version of the recurrent neural network. It has been designed to overcome the problem of vanishing /exploding gradient [35] that occurs in RNN [50], and it has the ability to learn better long-term dependencies [36].

LSTM can remember information from the past through its ability to remove or add information to a memory cell state based on the context of input. The LSTM cell is controlled and regulated by three binary gates, namely, forget gate f_t, input gate i_t, and output gate o_t. Equations (1), (2), (3), (4), (5), and (6) represent the forget gate, the input gate, the activation function, cell state, output gate, and the output h_t. Having x_t is the input for each time-step, $h_{t\text{-}1}$ is the output from the previous LSTM unit also called hidden unit, and $c_{t\text{-}1}$ is the memory of previous unit.

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{2}$$

$$p_t = \tan\left(W_p \cdot \left[h_{t-1}, x_t\right] + b_p\right) \tag{3}$$

$$c_t = f_t \times c_{t-1} + i_t \times p_t \tag{4}$$

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{5}$$

$$h_t = o_t \times \tan(c_t) \tag{6}$$

Where:

f_t is the forget gate

i_t, is the input gate

o_t is the output gate

h_{t-1} is the output from the LSTM previous unit

x_t is the input for each time-step

c_t is the cell state at timestamp t

c_{t-1} is the memory of the previous unit

p_t is the activation function

W_f, W_i, W_p, W_o are the weights for the forget, input, activation, and output gate neurons respectively.

σ is the sigmoid function.

b_f, b_i, b_p, b_o are the biases for the forget, input, activation, and output gates, respectively.

Figure 1 shows the relations between these gates in a single LSTM unit where the forget gate controls how much of the old state has to be forgotten by using sigmoid activation function. The output refers to a number between 0 and 1 where the value of "zero" indicates forget while the value of "1" means keep. The input gate controls the new information that updates the memory cell state. It employs a sigmoid function to decide what values to be updated and utilizes *tanh* function to create a vector of new candidates that can be added. These two output values are combined to update the cell state. The final value on the output gate decides what information should hold to the next cell state [37].

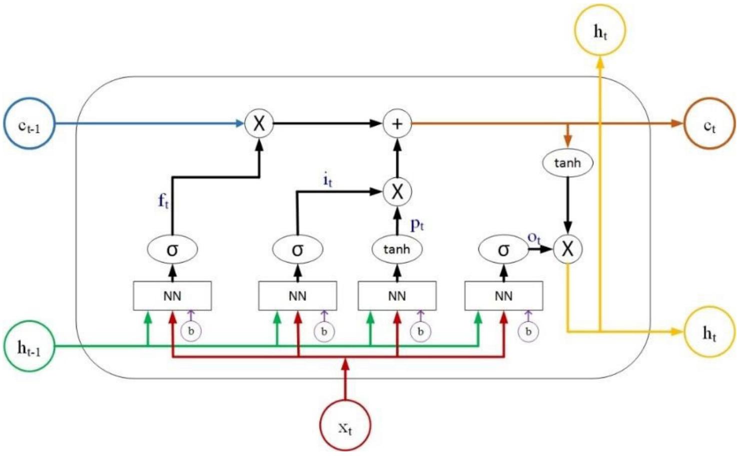


Fig. 1. LSTM unit [38]

3.3 Convolutional neural networks (CNNs)

The basic principle of CNNs or ConvNet is the convolution operation. CNNs are popularly used for image analysis, they have some type of specialization for being able to select and detect specific patterns from images, especially in sequence processing, computer vision, as well as certain NLP tasks [39] such as sentiment analysis which is a widespread used application of CNNs [40].

CNN can have more than one-dimensional convolution layer according to the type of data. When CNNs are applied to texts instead of images, one-dimensional layer is usually used to extract features because texts are considered sequential data. However, the main concept remains the same for both data types [41].

The most suitable NLP application of ConvNet is the classifications task. For example, sentiment classification can detect the patterns in a sentence regardless of their position by considering the n-grams, characters, or sequence of characters [42].

A word embedding layer and a one-dimensional convolutional network are required to use CNN for text data. In the embedding layer, each word in a sentence is converted into a word embedding vector. Then, the vector is padded to obtain equal dimensions for all vectors in the matrices [43]. The convolutional layer receives input as embedded word vectors and detects the features by applying filters to each possible window of words in the sentence. The result is one representative vector for the whole sentence. Next, convolved features are generated by passing vectors to a pooling layer for the further sampling of output and for capturing the prominent features [44].

The pooling operation is used to reduce computation power by reducing the dimensionality of features. The pooling layer combines the vectors generated from different convolution windows into one-dimensional vector by taking the maximum value or the average pooling value, which will keep the most prominent features in a sentence. Subsequently, the vector is fed into a fully connected layer to perform its intended classification task [45].

3.4 Proposed models

This section presents the four models that have been evaluated through the experiments. The differences among these models are determined by adding a new element each time to the current model, and as a result four models were experimented.

Model 1: is an LSTM model and consists of the three layers as illustrated in Figure 2.

Word embedding layer: pre-trained AraVec word embedding is used to convert the tweets into numeric format.

LSTM layer: comprises of a stack of LSTMs, with a number of hidden units equal to 100, which reads a single element of the input sequence in each time step, collects information from it and proceeds to the next time step. The input sequence is the tweet $X = \{x_1, x_2, \ldots, x_J\}$. At each time step, the hidden states $h_j^x \in \mathrm{R}$, for the tweet and illustrated in Equation 7.

$$h_t^x = f\left(h_{t-1}^x, x_t\right) \tag{7}$$

where *f* can be a non-linear function or even an LSTM. The last hidden state encapsulates a summary of the input sequence that is sent to the output layer.

Output layer: where the output is computed using Sigmoid with dense layer.

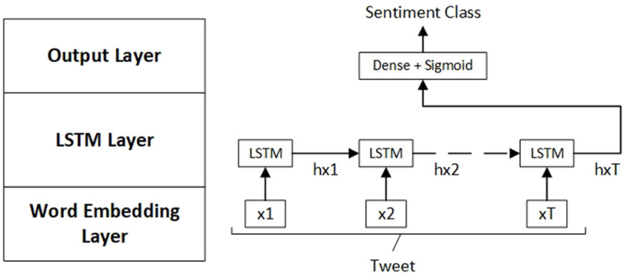


Fig. 2. Represents the architecture of model 1

Model 2: is LSTM model as well. However, dropout is added to the model to prevent it from overfitting. It consists of the same three layers. However, dropout of 0.2 is added before the LSTM layer.

Model 3: corresponds to an update of model 2 and consists of the same three layers and dropout of 0.2. The only difference is in the hidden units where it is increased into 150 units. The architectures of models 2 and 3 are illustrated in Figure 3.

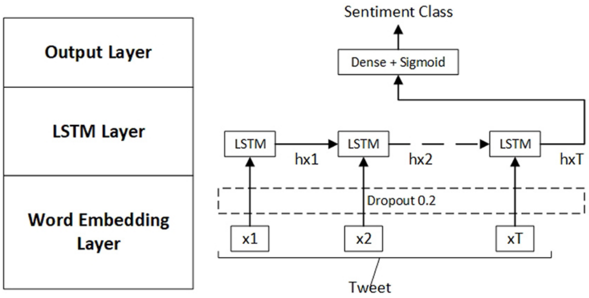


Fig. 3. Architecture of models 2 and 3

Model 4: is a four layers model that comprises LSTM and CNN as illustrated in Figure 4. The four layers are described as follows:

Word embedding layer: pre-trained AraVec is used to convert the tweets into vectors with numeric format, and a dropout of 0.2 is used to prevent the model from overfitting.

CNN layer: consists of 1D convolutional and Maxpooling operations.

LSTM layer: represents the same LSTM layers discussed in Model 1 with 150 hidden units.

Output layer: the output is computed using Sigmoid with dense layer.

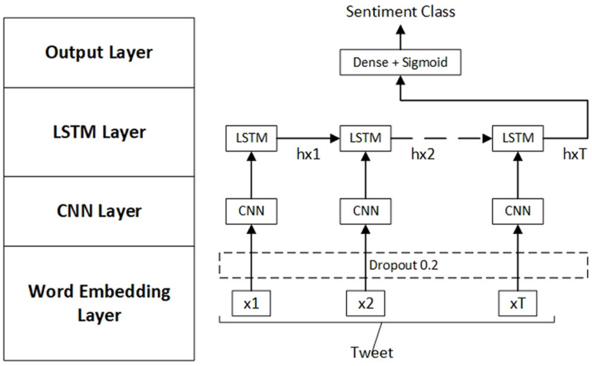


Fig. 4. Architecture of model 4

In the embedding layer of the four models and after loading the dataset, the maximum length of the tweets is computed, and a vocabulary is built where the vocab size is computed. Next, word-to-index dictionary is created to convert the tweets into vectors using the dictionary, and the short tweets were padded with zeros. Then, the embedding matrix is created using the AraVec.

4 Experiments and dataset

In this research, four DL models are applied to Arabic sentiment analysis for COVID-19 tweets dataset. The tweets are classified into two categories, namely, positive and negative. In the following subsections, more details are provided for the dataset, experimental settings and results.

4.1 Dataset

We generated our dataset by collecting Arabic tweets from Twitter regarding people's reactions to the COVID-19 pandemic. The collected dataset consists of 15779 Arabic tweets that indicate people's perceptions on the seriousness of the coronavirus. Then, the collected tweets are labeled manually by human annotators using two labels, namely, positive or negative. As a result, 12,176 tweets are labeled as positive, and 3,613 tweets are labeled as negative, which indicates an imbalanced dataset where the number of positive tweets is greater than the number of tweets in the negative class as shown in Figure 5.

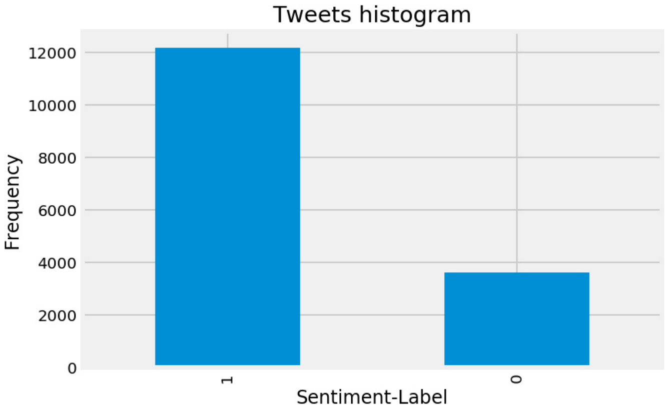
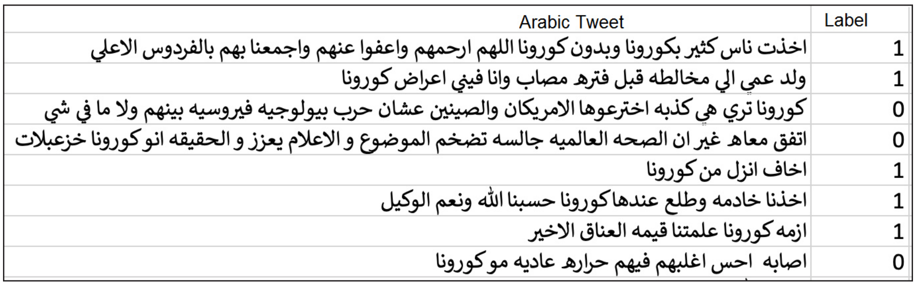


Fig. 5. Positive and negative tweets in COVID-19 labeled dataset

A sample image of the data record is shown in Figure 6 where each record consists of an Arabic tweet about COVID-19 and a label of 1 or 0, where label 1 indicates positive perception while zero indicates negative perception.



Arabic Tweet	Label
اخذت ناس كثير بكورونا وبدون كورونا اللهم ارحمهم واعفوا عنهم واجمعنا بهم بالفردوس الاعلي	1
ولد عمي الي مخالطه قبل فتره مصاب وانا فيني اعراض كورونا	1
كورونا تري هي كذبه اخترعوها الامريكان والصينين عشان حرب بيولوجيه فيروسيه بينهم ولا ما في شي	0
اتفق معاه غير ان الصحه العالميه جالسه تضخم الموضوع و الاعلام يعزز و الحقيقه انو كورونا خزعبلات	0
اخاف انزل من كورونا	1
اخذنا خادمه وطلع عندهاكورونا حسبنا الله ونعم الوكيل	1
ازمه كورونا علمتنا قيمه العناق الاخير	1
اصابه احس اغلبهم فيهم حراره عاديه مو كورونا	0

Fig. 6. A sample image of the data record in Arabic COVID-19 dataset

4.2 Data initialization and sampling

The collected tweets require a preprocessing step to fit the intended sentiment analysis task [46]. Preprocessing includes the removal of unwanted data, such as duplicate tweets, hash tags, HTML tags, URL links, numeric data, emoji, diacritics, punctuation marks, and special characters.

The next step is to divide the dataset into two sets, namely, training set and testing set. The training set represents 70% of the tweets in the dataset, which is sampled randomly from the total data while the testing set represents 30% of the dataset and sampled randomly considering the percentage of records labeled as negative and positive. The result in this stage is 4737 randomly sampled records for testing, and 11052 records are left for training. Table 2 shows the statistics of the training and testing sets in terms of the negative and positive classes.

Table 2. Training and testing sets

Class	Training	Testing	Percentage	Total
Positive	8529	3647	77%	12,176
Negative	2523	1090	23%	3,613
Total	11,052	4,737	100%	15,789

4.3 Hyperparameters settings

This subsection presents the hyperparameters' settings for all experiments conducted in this research. The baseline experiments were conducted using Adam optimizer, its default initial learning rate (0.001) and a dropout of 0.2 while four batch sizes; 32, 64, 128, and 256 were investigated and the maximum number of epochs was 20 epochs. Early stopping was determined once the model performance stopped improving which was after three training epochs.

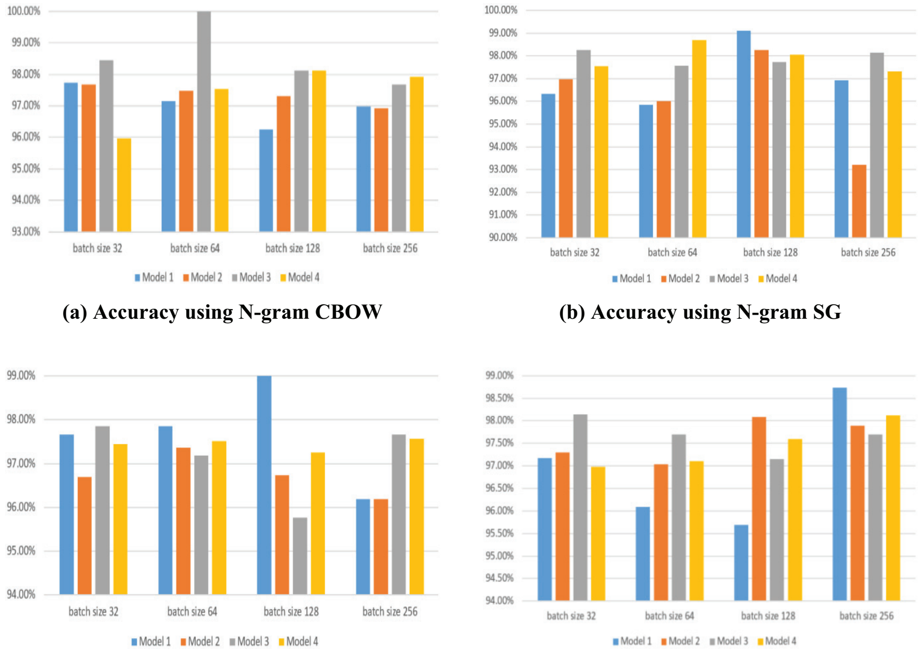
However, experimental tuning for hyperparameters was carried out, where the conducted experiments show that the maximum epoch size reached 14 while the minimum reached 4 using early stopping.

5 Results and discussion

This section illustrates the results obtained from the experiments conducted using the four models discussed in the previous section. Figure 7 shows the results of the four models using N-gram and unigram for different batch sizes with embeddings of dimension 300 while Figure 8 shows the results for these models with words embeddings of dimension 100.

It is shown in Figure 7 that when using embeddings with dimension of 300, the highest accuracy value reached 100%, while the lowest value reached 93.22%. However, with dimension of 100 as illustrated in Figure 8, the highest accuracy value reached 98.9%, and the lowest reached 91.6%.

These findings confirm that the embeddings of dimension 300 contain more information and thus provides better results. Furthermore, when comparing the CBOW to the Skip-gram (SG) model without considering the dimension, the results show that the highest accuracy value for the experiments that were implemented using CBOW reached 100% and the lowest was reached at 91.6%. Meanwhile, the highest accuracy value using SG reached 99.11%, and the lowest value was 92.15% as shown in Figure 7 and Figure 8. Therefore, there is no preference for using CBOW model over SG model or vice versa.



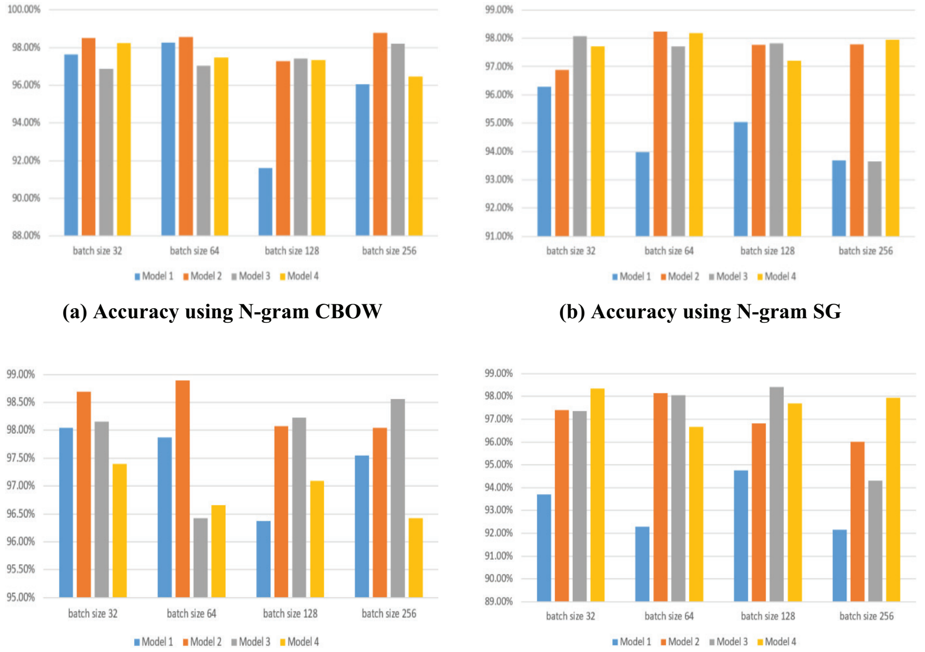
(a) Accuracy using N-gram CBOW

(b) Accuracy using N-gram SG

(c) Accuracy using unigram CBOW

(d) Accuracy using unigram SG

Fig. 7. Accuracy values using different Aravec models with dimension 300



(a) Accuracy using N-gram CBOW **(b) Accuracy using N-gram SG**

(c) Accuracy using unigram CBOW **(d) Accuracy using unigram SG**

Fig. 8. Accuracy values using different Aravec models with dimension of 100

Table 3 shows the best accuracy results while Table 4 shows the worst accuracy values for the four models. As shown in Table 3, the best accuracy among all models was obtained by model 3 and reached 100%, when using N-gram CBOW with dimension of 300, batch size of 64, and after 4 epochs. The second best accuracy value reached at 99.11% by model 1 when using N-gram SG with dimension of 300, batch size of 128, and after 6 epochs. The third best accuracy reached 98.9% by model 2, when using unigram CBOW with dimension of 100, batch size of 64, and after 6 epochs. While the last best accuracy value reached 98.69% by model 4, when using N-gram SG with dimension of 300, batch size of 64, and after 4 epochs.

Table 3. The four models sorted according to best accuracy results

Best	Accuracy	Batch Size	Epoch	Aravec Model	Dimension
Model 1	99.11%	128	6	SG	N-gram 300
Model 2	98.9%	64	6	CBOW	Unigram 100
Model 3	100%	64	4	CBOW	N-gram 300
Model 4	98.69%	64	4	SG	N-gram 300

Table 4 shows that the worst accuracy among all models was obtained by model 1 and reached 91.6%, when using N-gram CBOW dimension of 100, batch size of 128, and after 5 epochs. The second worst accuracy value reached 93.22% by model 2, when using N-gram SG dimension of 300, batch size of 256, and after 4 epochs. The explanation is that model 2 is an updated version of model 1, where a dropout is added to the model. Model 3 reached the third worst accuracy value of 93.65% when using N-gram SG with dimension of 100, batch size of 256, and after 4 epochs. Model 3 is an update of model 2, where the number of hidden layers is increased. Finally, model 4 reached the last worst accuracy value of 95.97% using N-gram CBOW with dimension of 300, batch size of 32, and after 4 epochs. The difference between models 3 and 4 is that a CNN layer was added to the latter. To recap, we can say that adding dropout, increasing hidden layers, and adding a CNN layer enhanced the performance of the sentiment classification task of the LSTM model.

Table 4. Four models sorted according to worst accuracy results with some details

Worst	Accuracy	Batch Size	Epoch	Aravec Model	Dimension
Model 1	91.6%	256	5	CBOW	N-gram 100
Model 2	93.22%	256	4	SG	N-gram 300
Model 3	93.65%	256	4	SG	N-gram 100
Model 4	95.97%	32	4	CBOW	N-gram 300

The results also show that the use of large batch sizes does not enhance the performance of the models for sentiment classification task because the worst accuracy results are obtained mostly when a batch size of 256 is used, while 64 is the batch size of the majority of the models when the best accuracy values are achieved.

6 Conclusion

In this paper, four models based on LSTM deep learning model for sentiment classification task are studied because of its ability to capture long-term dependencies to keep historical information and try to reduce the effect of vanishing/exploding gradient. Also, we attempted to test the effect of adding a one-dimension convolutional layer to the LSTM model to extract more prominent features with Aravec pre-trained word embedding model used as the input layer. The experimental results prove that the four models improve the accuracy results of sentiment classification task effectively where the best accuracy value (i.e. 100%) is achieved by model 3, which has more hidden units and applied with CBOW embedding model with dimension of 300, batch size of 64, and 4 epochs.

7 References

[1] A. Althagafi, G. Althobaiti, H. Alhakami and T. Alsubait, "Arabic tweets sentiment analysis about online learning during COVID-19 in Saudi Arabia" *International Journal of Advanced Computer Science and Applications (IJACSA),* vol. 12, no. 3, 2021. https://doi.org/10.14569/IJACSA.2021.0120373

[2] N. Alturayeif and H. Luqman, "Fine-grained sentiment analysis of Arabic COVID-19 Tweets using BERT-based transformers and dynamically weighted loss function," *Applied Sciences,* vol. 11, 2021. https://doi.org/10.3390/app112210694

[3] W. Wang, I. Hernandez, D. Newman, J. He and J. Bian, "Twitter analysis: Studying us weekly trends in work stress and emotion," *Applied Psychology,* vol. 65, no. 2, pp. 355–378, 2016. https://doi.org/10.1111/apps.12065

[4] A. Addawood, A. Alsuwailem, A. Alohali, D. Alajaji, M. Alturki, J. Alsuhaibani and F. Aljabli, "Tracking and understanding public reaction during COVID-19: Saudi Arabia as a use case," in *the 1st Workshop on NLP for COVID-19 at EMNLP 2020*, Online, 2020. https://doi.org/10.18653/v1/2020.nlpcovid19-2.24

[5] A. D'Andrea, F. Ferri, P. Grifoni and T. Guzzo, "Approaches, tools and applications for sentiment analysis implementation," *International Journal of Computer Applications,* vol. 125, no. 3, pp. 0975–8887, 2015. https://doi.org/10.5120/ijca2015905866

[6] B. Keith, E. Fuentes and C. Meneses, "A hybrid approach for sentiment analysis applied to paper reviews," in *ACM SIGKDD Conference (KDD'17)*, Halifax, Nova Scotia, Canada, 2017.

[7] A. B. Soliman, K. Eissa and S. R. El-Beltagy, "AraVec: A set of arabic word embedding models for use in Arabic NLP," *Procedia Computer Science,* vol. 117, pp. 256–265, 2017. https://doi.org/10.1016/j.procs.2017.10.117

[8] M. Biltawi, W. Etaiwi, S. Tedmori3, A. Hudaib and A. Awajan, "Sentiment classification techniques for Arabic language: A Survey," *in Proc. 2016 7th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 2016. https://doi.org/10.1109/IACS.2016.7476075

[9] J. Alwehaibi, M. Bikdash, M. Albogmi and K. Roy, "A study of the performance of embedding methods for Arabic short-text sentiment analysis using deep learning approaches," *Journal of King Saud University Computer and Information Sciences,* 2021. https://doi.org/10.1016/j.jksuci.2021.07.011

[10] M. Biltawi, G. Al-Naymat and S. Tedmori, "Arabic sentiment classification: A hybrid approach," *in Proc. 2017 International Conference on New Trends in Computing Sciences (ICTCS),* pp. 104–108, 2017. https://doi.org/10.1109/ICTCS.2017.24

[11] M. Biltawi, W. Etaiwi, S. Tedmori and A. Shaout, "Fuzzy based sentiment classification in the Arabic language," *in Proc. the 2018 Intelligent Systems Conference (IntelliSys)*, London, United Kingdom, 2018. https://doi.org/10.1007/978-3-030-01054-6_42

[12] D. Ahmed, S. Salloum and K. Shaalan, "Sentiment analysis of Arabic COVID-19 tweets," *in Proc. International Conference on Emerging Technologies and Intelligent Systems. ICETIS 2021. Lecture Notes in Networks and Systems*, vol. 322, Springer, 2021, pp. 623–632. https://doi.org/10.1007/978-3-030-85990-9_50

[13] M. Saad, "Arabic sentiment twitter corpus: positive and negative tweets collected from twitter," 2019. [Online]. Available: https://www.kaggle.com/mksaad/arabic-sentiment-twitter-corpus

[14] W. Antoun, F. Baly and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," *in Proc. the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, Language Resources and Evaluation Conference (LREC 2020)*, Marseille, France, 2020.

[15] M. Abdul-Mageed, A. Elmadany and E. Nagoudi, "ARBERT & MARBERT: Deep bidirectional transformers for Arabi," 2021. https://doi.org/10.18653/v1/2021.acl-long.551

[16] S. Mohammad, F. Bravo-Marquez, M. Salameh and S. Kiritchenko, "Semeval-2018 task 1: Affect in tweets," 2018. https://doi.org/10.18653/v1/S18-1001

[17] H. Alhazmi and M. Alharbi, "Emotion analysis of Arabic tweets during COVID-19 pandemic in Saudi Arabia," *International Journal of Advanced Computer Science and Applications,* vol. 11, no. 10, 2020. https://doi.org/10.14569/IJACSA.2020.0111077

[18] S. Mohammad, "Word affect intensities," *in Proc. of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018.

[19] O. AlZoubi, S. Tawalbeh and M. AL-Smadi, "Affect detection from Arabic tweets using ensemble and deep learning techniques," *Journal of King Saud University – Computer and Information Sciences,* 2020.

[20] S. Albukhitan, A. Alnazer and T. Helmy, "Framework of semantic annotation of Arabic document using deep learning," *Procedia Computer Science,* vol. 170, pp. 989–994, 2020. https://doi.org/10.1016/j.procs.2020.03.096

[21] M. Heikal, M. Torki and N. El-Makky, "Sentiment analysis of Arabic tweets using deep learning," in *Procedia Computer Science*, 2018. https://doi.org/10.1016/j.procs.2018.10.466

[22] M. Nabil, M. Aly and A. Atiya, "ASTD: Arabic sentiment tweets dataset," *in Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015. https://doi.org/10.18653/v1/D15-1299

[23] A. M. Alayba, V. Palade, M. England and R. Iqbal, "A combined CNN and LSTM model for Arabic sentiment analysis," *Machine Learning and Knowledge Extraction,* pp. 179–191, 2018. https://doi.org/10.1007/978-3-319-99740-7_12

[24] A. Alayba, V. Palade, M. England and R. Iqbal, "Arabic language sentiment analysis on health services," *in Proc. 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, Nancy, 2017. https://doi.org/10.1109/ASAR.2017.8067771

[25] A. Alayba, V. Palade, M. England and R. Iqbal, "Improving sentiment analysis in Arabic using word representation," *in Proc. 2018 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, London, 2018. https://doi.org/10.1109/ASAR.2018.8480191

[26] H. Elzayady, B. K. M. and S. Gouda, "Arabic opinion mining using combined CNN – LSTM models," *International Journal of Intelligent Systems and Applications,* vol. 12, pp. 25–36, 2020. https://doi.org/10.5815/ijisa.2020.04.03

[27] H. ElSahar and S. El-Beltagy, "Building large Arabic multi-domain resources for sentiment analysis," *in Proc. Computational Linguistics and Intelligent Text Processing*, 2015. https://doi.org/10.1007/978-3-319-18117-2_2

[28] M. Aly and A. Atiya, "LABR: A large scale Arabic book reviews dataset," *in Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria, 2013.

[29] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen and S. Belfkih, "ASA: A framework for Arabic sentiment analysis," *Journal of Information Science,* vol. 46, no. 4, pp. 544–559, 2020. https://doi.org/10.1177/0165551519849516

[30] A. H. Ombabi, W. Ouarda and A. M. Alimi, "Deep learning CNN-LSTM framework for Arabic sentiment analysis using textual information shared in social networks," *Social Network Analysis and Mining,* vol. 10, no. 1, pp. 1–13, 2020. https://doi.org/10.1007/s13278-020-00668-1

[31] A. Alayba and V. Palade, "Leveraging Arabic sentiment classification using an enhanced CNN-LSTM approach and effective Arabic text preparation," *Journal of King Saud University – Computer and Information Sciences,* 2021. https://doi.org/10.1016/j.jksuci.2021.12.004

[32] N. A. Abdulla, N. Ahmed, M. A. Shehab and M. Al-Ayyoub, "Arabic sentiment analysis: Lexicon-based and corpus-based," *in Proc. 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, 2013. https://doi.org/10.1109/AEECT.2013.6716448

[33] M. Alian and A. Awajan, "Factors affecting sentence similarity and paraphrasing identification," *International Journal of Speech Technology,* vol. 23, pp. 851–859, 2020. https://doi.org/10.1007/s10772-020-09753-4

[34] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," *in Proc. International Conference on Learning Representations – ICLR Workshop*, 2013.

[35] M. Roodschild, J. Gotay Sardiñas and A. Will, "Progress in artificial intelligence," *A new approach for the vanishing gradient problem on sigmoid activation,* vol. 9, no. 4, pp. 351–360, 2020. https://doi.org/10.1007/s13748-020-00218-y

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735–1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735

[37] Z. Li, D. He, F. Tian, W. Chen, T. Qin, L. Wang and T. Liu, "Towards binary-valued gates for robust LSTM training," *in Proc. of the 35th International Conference on Machine Learning*, 2018.

[38] M. Biltawi, A. Awajan and S. Tedmori, "Neural machine understanding for Arabic text," Amman: Phd Thesis, Princess Sumaya University for Techonolgy, 2021.

[39] J. Deng, L. Cheng and Z. Wang, "Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification," *Computer Speech & Language,* vol. 68, 2021. https://doi.org/10.1016/j.csl.2020.101182

[40] L. D. Medus, M. Saban, J. V. Francés-Víllora, M. Bataller-Mompeán and A. Rosado-Muñoz, "Hyperspectral image classification using CNN: Application to industrial food packaging," *Food Control,* vol. 125, 2021. https://doi.org/10.1016/j.foodcont.2021.107962

[41] D. Alsaleh and S. Larabi-Marie-Sainte, "Arabic text classification using convolutional neural network and genetic algorithms," *IEEE Access,* vol. 9, pp. 91670–91685, 2021. https://doi.org/10.1109/ACCESS.2021.3091376

[42] S. Das and A. Kolya, "Predicting the pandemic: Sentiment evaluation and predictive analysis from large-scale tweets on Covid-19 by deep convolutional neural network," *Evolutionary Intelligence,* 2021. https://doi.org/10.1007/s12065-021-00598-7

[43] Y. Yingying Liu, P. Li and X. Hu, "Combining context-relevant features with multi-stage attention network for short text classification," *Computer Speech & Language,* vol. 71, 2022. https://doi.org/10.1016/j.csl.2021.101268

[44] Y. Kim, "Convolutional neural networks for sentence classification," *in Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014. https://doi.org/10.3115/v1/D14-1181

[45] A. Fesseha, S. Xiong, E. D. Emiru, M. Diallo and A. Dahou, "Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya," *Information,* vol. 12, no. 2, 2021. https://doi.org/10.3390/info12020052

[46] M. A. El-Affendi, K. Alrajhi and A. Hussain, "A novel deep learning-based multilevel parallel attention neural (MPAN) model for multidomain Arabic sentiment analysis," *IEEE Access,* vol. 9, pp. 7508–7518, 2021. https://doi.org/10.1109/ACCESS.2021.3049626

[47] G. Ghazal, M. Alian and E. Alkhawaldeh, "E-learning and blended learning methodologies used in universities during and after COVID-19", *International Journal of Interactive Mobile Technologies*, vol. 16, no. 18, 2022. https://doi.org/10.3991/ijim.v16i18.32721

[48] K. Mrhar, L. Benhiba, S. Bourekkache and M. Abik, "A Bayesian CNN-LSTM model for sentiment analysis in massive open online courses MOOCs". *International Journal of Emerging Technologies in Learning (iJET),* vol. 16, no. 23, pp 216–232, 2022. https://doi.org/10.3991/ijet.v16i23.24457

[49] F. Hussein, S. M. El-Salhi, R. ALazazma, T. Abu-Hantash, H. Abu-Hantash and H. Thaher, "An android application using machine learning algorithm for clique detection in issues related to transportation", *International Journal of Interactive Mobile Technologies*, vol. 16, no. 14, pp. 4–22. 2022. https://doi.org/10.3991/ijim.v16i14.30625

[50] I. Nurhaida, H. Noprisson, V. Ayumi, H. Wei, E. D. Putra, M. Utami and H. Setiawan. "Implementation of deep learning predictor (LSTM) algorithm for human mobility prediction", *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 18, pp. 132–144, 2020. https://doi.org/10.3991/ijim.v14i18.16867

[51] P. Demotte, K. Wijegunarathna, D. Meedeniya and I. Perera, "Enhanced sentiment extraction architecture for social media content analysis using capsule networks", *Multimedia tools and applications,* pp. 1–26, 2021. https://doi.org/10.1007/s11042-021-11471-1

[52] D. A. Meedeniya and A. S. Perera, "Evaluation of partition-based text clustering techniques to categorize indic language documents", in *IEEE International Advance Computing Conference*, 2009. https://doi.org/10.1109/IADCC.2009.4809239

[53] D. Lenadora, G. Gamage, D. Haputhant, D. Meedeniya and I. Perera, "Exploratory analysis of a social media network in Sri Lanka during the COVID-19 virus outbreak", *arXiv preprint arXiv:2006.07855,* 2020.

[54] M. Alian and A. Awajan, "Arabic sentence similarity based on similarity features and machine learning", *Soft Computing*, vol. 25, pp. 10089–10101, 2021. https://doi.org/10.1007/s00500-021-05754-w

8 Authors

Maisa Al-Khazaleh is a faculty member at The Hashemite University, Faculty of Science, Basic Sciences Department, Zarqa, Jordan. She received her B.Sc. degree in Computer Science from Jordan University of Science and Technology in 2007, and then she received the M.Sc. degree in Computer Science from Yarmouk University in 2009. Her research interests are primarily in the area of machine learning and Natural language processing (NLP). (email: maisa@hu.edu.jo).

Marwah Alian is a faculty member at The Hashemite University, Faculty of Science, Basic Sciences Department, Zarqa, Jordan. She received her B.Sc. degree in Computer Science from The Hashemite University in 1999. Then, she received M.Sc. degree in Computer Science from The University of Jordan in 2007. In 2021, she received her PhD from Princess Sumaya University for Technology (PSUT). Her research interests are in the area of elearning systems, machine learning, data mining and Natural language processing. (email: Marwahm@hu.edu.jo).

Mariam Biltawi is a faculty member at Al Hussein Technical University, Computer Science Department, Amman, Jordan. She has a PhD in computer science from Princess Sumaya University for Technology (PSUT). She get her M.Sc. degree from Al-Balqa' Applied University while her B.Sc. from PSUT. Her research interests are primarily in the area of machine learning and Natural language processing. (email: Mariam.biltawi@htu.edu.jo)

Bayan Al-Hazaimeh is a faculty member at The Hashemite University, Faculty of Science, Basic Sciences Department, Zarqa, Jordan. She received her B.Sc. degree in Computer Science from Yarmouk University in 2002. After graduation, she worked as a teacher in Zahar high school in Jordan. Then, she received M.Sc. degree in Computer Information System from Yarmouk University in 2007. (email: Bayana@hu.edu.jo).

Article submitted 2022-09-26. Resubmitted 2022-11-15. Final acceptance 2022-11-15. Final version published as submitted by the authors.