Khalil *et al. J Big Data* (2021) 8:136
https://doi.org/10.1186/s40537-021-00523-w

Journal of Big Data

RESEARCH Open Access

Deep learning for emotion analysis in Arabic tweets

Enas A. Hakim Khalil[1*], Enas M. F. El Houby[1] and Hoda Korashy Mohamed[2]

*Correspondence:
enaskhalil@gmail.com
[1] Present Address: Systems & Information Department, Engineering Research Division, National Research Centre NRC, Dokki, Giza 12311, Egypt
Full list of author information is available at the end of the article

Abstract

Currently, expressing feelings through social media requires great consideration as an essential part of our lives; besides sharing ideas and thoughts, we share moments and good memories. Social media such as Facebook, Twitter, Weibo, and LinkedIn, are considered rich sources of opinionated text data. Both organizations and individuals are interested in using social media to analyze people's opinions and extract sentiments and emotions. Most studies on social media analysis mainly classified sentiment as positive, negative, or neutral classes. The challenge in emotion analysis arises because humans can express one or several emotions within one expression. Human beings can recognize these different emotions well; however, it is still not easy for an emotion analysis system. In most cases, the Arabic language used through social media is of a slangy or colloquial form, making it more challenging to preprocess and filter noise since most lemmatization and stemming tools are built on Modern Standard Arabic (MSA). An emotion analysis model has been implemented to categorize emotions. The model is a multiclass and multilabel classification problem. However, few studies have been adapted for this emotion classification problem in Arabic social media. Nearly the only work is the one of SemEval 2018 task1- sub-task E-c. Several machine learning approaches have been implemented in this task; a few studies were based on deep learning. Our model implemented a novel multilayer bidirectional long short term memory (BiLSTM) trained on top of pre-trained word embedding vectors. The model achieved state-of-the-art performance enhancement. This approach has been compared with other models developed in the same tasks using Support Vector Machines (SVM), random forest (RF), and fully connected neural networks. The proposed model achieved a performance improvement over the best results obtained for this task.

Keywords: Deep learning, BiLSTM, Multi label emotion classification, Word embedding, Aravec, ARLSTEM

Introduction

With the rapid growth of web applications, such as E-commerce platforms and substantial social media comments in various fields, an urgent need to deal with this massive amount of web data and automatically extract helpful information has arisen. Sentiment analysis models play a significant role in this task. Sentiment analysis is a computational field within natural language processing (NLP) concerned with people's sentiments and opinions toward objects such as services, persons, products, events, organizations,

and topics. Thanks to the availability of high-performance computational computers, which allows using different machine learning techniques, especially deep learning, to build high-performance, robust automatic sentiment analysis models. Sentiment analysis detects positive, neutral, or negative opinions from the text. Emotion analysis is one of the most common sentiment analysis tasks for recognizing different feelings through text expression.

Emotions are mainly expressed using language; however, some emotions, such as joy, fear, and sadness, are more fundamental than others and can be expressed differently. For example, our degree of utterances can reveal that we are so sad and slightly angry. The term affect refers to various categories of emotions as joy, fear, arousal, and valence. Emotions detection is significant in many fields, such as public health, marketing, disaster management, public policy, and political issues [1]. There are two categories of representations for emotions; Ekman model [2] and Plutchik model [3, 4]. The Ekman representation includes anger, happiness, disgust, surprise, fear, and sadness. However, the Plutchik includes Ekman's six emotions and two labels: trust and anticipation. Emotion recognition systems from facial expressions and images have been widely used [5–8]. Hand gesture recognition has been used as a Human action recognition (HAR) [9]. Emotion recognition models can also be explored based on human–computer interaction [10–12]. Emotion-rich textual data from social networks can be processed for various real-world applications, including [13–17].

The research interest in Arabic sentiment analysis increases drastically due to the Internet's large number of Arabic language users. However, emotion recognition from Arabic text still needs enormous efforts to develop more accurate emotion mining models in MSA and dialectal Arabic using a large-scale emotion lexicon.

The fast rise of social media platforms (e.g., Twitter and Facebook) attracted the researchers' attention to the technique of "Affect detection from text," which enabled users to communicate their sentiments, emotions, and ideas via text. SemEval, an international workshop on semantic evaluation developed from SensEval (Evaluation Exercises for the Semantic Analysis of Text, Organized by ACL-SIGLEX), produced an excellent value task, "The SemEval-2018 Task 1: Affect in Tweets" [1] in its 12th workshop on semantic evaluation in 2018. This task consists of five subtasks in which automatic systems infer a person's emotional state from their tweet for each task in English, Arabic, and Spanish. Therefore, this study performed a multilabel emotion classification in the Arabic language using SemEval-2018 Task1-datasets. Several preprocessing steps are implemented, including removing non-Arabic words, removing digits, removing stop words, and applying a robust stemmer, Arabic light stemmer (ARLSTM). Then, the tweets are represented as feature vectors using AraVec (a set of Arabic word embedding models for use in Arabic NLP) [18]. These embedded vectors are fed to a multilayer bidirectional long short-term memory (BiLSTM) network. The contribution of this study can be summarized as follows:

- We proposed an optimization BiLSTM network for multilabel Arabic emotion analysis. However, building a multilabel multiclass classification model in emotion analysis is still considered an issue that must be tackled more and enhanced, especially in Arabic.

- Different preprocessing procedures that suit the nature of social media colloquial text have been implemented.
- We employed a CBOW word embedding model for word representation. Word embedding is proved to give the best results over other word representations.
- We used a deep neural network of multilayers BiLSTM network with its great ability to extract the context information from Arabic text. We also investigated the effect of changing the number of BiLSTM layers to improve the performance.
- Additionally, we investigated model hyperparameters tuning effect along with different optimizers on the model's performance.
- The model outperformed other machine learning models built for the same task.

The remainder of the paper is organized as follows: "Related work" section discusses related studies, including several Arabic text emotion analysis methods. "Methodology" section describes the proposed approach for evaluating the emotional content in tweets. "Experimental setup and results" section summarizes our findings and examines the most significant findings. "Discussion" discusses the results. Finally, "Conclusions and future work" section presents the conclusion and future studies.

Related work

Learning users' emotions is essential in many applications. Emotions can be inferred through text expressions and HAR as facial expressions, hand gestures, body posture, and voice. In [9], social robots were used as communication assistance for education and entertainment. The study described an acceptable and natural interaction between social robots and children. The thermal facial reaction of youngsters, i.e., the nose tip temperature signal, was recorded and classified in real-time using the Mio Amico Robot during an experimental session. The categorization was performed by comparing the thermal signal analysis-classified emotional state to the emotional state recorded by Face reader 7. An empathic robot in [6] recognized human emotions through facial expressions and automatically responded to these specific emotional states. It produced a state-of-the-art accuracy rate of 95.58%. Additionally, it implemented a convolutional neural network (CNN) and a bank of Gabor filters in different experiments for feature representation. It also employed SVMs and multilayer perceptron as classifiers. Customer feedback detection in [12], a multimodal affect recognition system was developed to classify whether a customer likes or dislikes a product examined at a counter by analyzing the consumer's facial expression, hand gestures, body posture, and voice after testing the product. Hand gesture recognition is widely used in scientific research; it is crucial for interacting with deaf individuals. Constantine et al. [10] proposed a transfer-learning approach using AlexNets and hyperparameter tuning using ABC, GA, and PSO algorithms. The methodology produced effective outcomes with an average accuracy of 98.09%, outperforming the best work in the medical sector. In [19], computational analysis techniques measured the emotional facial expression of people with Parkinson's disease (PD). It is essential to examine an experimental pilot work for masked face detection in PD since PD experiences hypomania, which often reduces facial expression. This experiment achieved an accuracy of 85% on the testing images using a deep learning-based model.

Multilabel emotion classification is a hot topic in emotion analysis tasks since it represents real-life situations where the human may express a mixture of emotions in their text simultaneously. For example, the text may express happiness, love, optimism, sadness, or pessimism. Thus, building models with more than one output emotion for each input text is more beneficial. The following few paragraphs present some recent efforts in Arabic multilabel emotion classification.

Emotion mining in Arabic (EMA) [16] performed emotion and sentiment mining on Arabic tweets. First, preprocessing steps are performed by applying normalization rules adopted in [20], including diacritics or tashkeel, hamza, elongations, and non-Arabic letter removals. Then, most frequent emojis are replaced with the corresponding Arabic word using a manually created lexicon to replace each emoji. Finally, ARLSTEM [21] is used for stemming. In the feature selection stage, the author tried different features separately. However, the word embedding from AraVec proved to be the best feature. The tweet is finally classified either as neutral or as one or more of the 11 emotions (anger, disgust, anticipation, joy, love, optimism, fear, pessimism, sadness, trust, and surprise). A linear support vector classifier (SVC) achieved the best performance among all classifiers tested, with a test accuracy of 0.489.

TW-Star [22] used different preprocessing stemming (Stem), lemmatization (Lem), stop words removal, and common emoji recognition (Emo). The preprocessed tweets are classified by a binary relevance (BR) multilabel classifier using SVM with term frequency inverse document frequency (TF-IDF) features. Several experiments with different combinations of preprocessing achieved the best results of accuracy 0.465 by combining Emo, Stem, and Stop (Emo + Stem + Stop).

TeamUNNC [23] performed tokenization, white space removal, and punctuation treatment as individual words. In the second stage, word2vec embedding AraVec [18] were combined with Affective Tweets Weka-package features. Finally, the classification is implemented with a fully connected NN with three dense hidden layers and stochastic gradient descent (SGD) optimizer. The model achieved an accuracy of 0.446, exceeding the baseline model accuracy.

In [24], feature vectors were developed using the Doc2Vec model. Then, the random forest (RF) algorithm was used for classification, and Doc2Vec size varied from 10 to 1000 with an increment of 10 iterations. The number of decision trees used in the forest ranged from 10 to 150, with an increment of 10 with each iteration. The maximum tree depth in the algorithm varied from 2 to 20, with an increment of 1 with each iteration. This model achieved an accuracy of 0.25.

TeamCEN [25] Uses Globe vector representation [26] for representing the words into vectors.; it depends on word-word co-occurrence statistics. Then the presentation of the tweet is made by using aggregated sum and dimensionality reduction of the glove vectors of the words in that tweet. The classification is then done using RF and SVM.

Therefore, we developed a novel BiLSTM deep learning model for multilabel emotion classification in Arabic tweets using the same dataset as the models described above. The model performance surpassed the above models owing to the ability of LSTM networks to handle the sequential data as texts better than other machine learning and deep learning models.

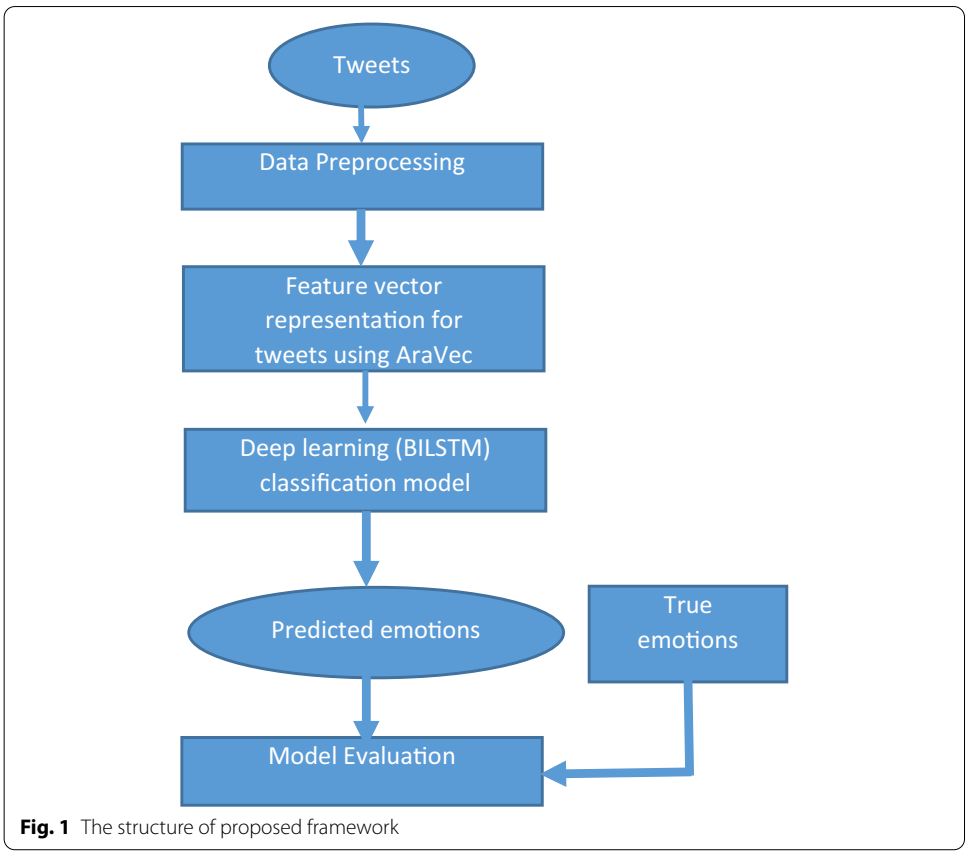

Fig. 1 The structure of proposed framework

Methodology

This section describes the approach we followed to develop a framework for predicting users' emotions from their tweets; the framework is shown in Fig. 1. The framework includes the following pipeline:

1. Data preprocessing:

   First, the tweet dataset has been preprocessed. The performed preprocessing steps are presented in the following steps:

   - Initial preprocessing:

     Text normalization has been applied, including removing elongation (Tatweel) in Arabic words like هـــــي changed to هي, يكتبـــــون changed to يكتبون, removing repeated characters and digit removal. Additionally, trimming special characters, removing English characters (a–z A–Z), French characters (àéèæoeç), and replacing emotions into اسم_حساب_شخصي.

   - Stop word removal: In natural language, stop words are those words that do not add to the meaning or have very little sense. Stop words are usually removed from

the text before training the model. Stop words occur more frequently in the text; thus, they do not add valuable information for classification or clustering. Our Arabic stop word list is updated from the NLTK Arabic stop word[1],[2]. Our updated list considered the change in stop words resulting from removing Hamza and Yaa as 'ان', 'اما', 'اليكن', 'اليكما' , 'اليكم', 'انى ,اليكم', 'انه', 'انما', 'انتما','انتن', 'انتم', 'انت', 'انا'). Some ambiguous words from this list as كان instead of كأن are not considered in our updated list of stop words for not to increase ambiguity.

- Creating of emoji lexicon:
  A lexicon with the most shared tweet emojis is manually created, where each emoji is transcribed to its corresponding Arabic word. Then, emojis are replaced with related meanings to emotions.
- Stemming:
- Further normalization step is stemming. It reduces the word to its standard form. Here, the stemming process is performed using ARLSTEM [21]. With social media data (tweets data), using a word stem is more valuable than its lemma because tweets are mostly dialectal Arabic not MSA form; most Arabic morphological analyzers are trained using MSA [17]. The ARLSTEM normalizes the word by removing diacritics; this may cause ambiguity in word semantics. However, it is interesting since it facilitates the stemming process. ARLSTEM replaces hamzated Alif with Alif, Alif Maqsura with Yaa, and removes Waaw at the beginning. Prefixes from the words' beginning and suffixes from the word's end are trimmed. Stemming also includes transforming the word from the feminine form to the masculine form, Stem the verb prefixes and suffixes, or both. In the experiments below, the proposed model has been examined with different variations of stemming either to use the stemmer in its basic form or to use it and exclude some special words frequently repeated in tweets and not stemming that words, such as the word "الله" and all the derivations as "لله", "اللهم", "بالله", "فالله" and replace all these word derivations with one standard form "الله". This minor modification has a good impact on the performance since the count of occurrence of this word represents about 16% of the number of words in our dataset, and normalizing these words not stemming them reduces the ambiguity.

2. Feature extraction:
   Feature selection and extraction are two major techniques used to improve the performance of different machine learning models by removing redundant and irrelevant features and reducing the dimensionality space [27]. Rostami et al. [28] presented the most recent feature selection approaches that involve using a genetic algorithm for feature selection. It is proved to be efficient in many machine learning classifiers that do not have high computational complexity. Rostami et al. [29] produced a machine learning classification model that increased the classification

[1] https://www.nltk.org/.

[2] The python command to download Stopwords list of NLTK package is << nltk.corpus import stopwords >> and to assign arabic_stopwords << arabic_stopwords = stopwords. Words("arabic") >>.

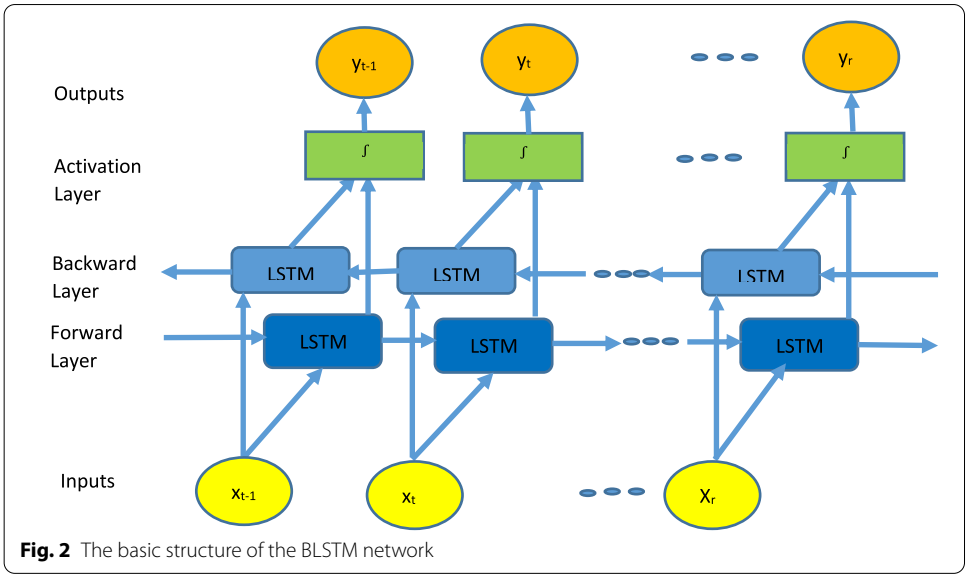

Fig. 2 The basic structure of the BLSTM network

accuracy and succeeded in computational complexity reduction using a constrained feature selection. In our sentiment analysis problem, word embedding proved to be a more effective feature extraction technique than n-gram and TF-IDF. In this study, we employed the AraVec embedding model [28]. AraVec is a large-scale dataset (about 205,000 words) that consists of different Arabic dialects, and it is trained on the Twitter data domain. Word embedding is proved to be decisive since it overcomes the sparsity problem in n-grams models and simplifies semantics by providing exact representations for words that may exist in the same context. The pre-trained model, "Twitter-CBOW/tweets_cbow_300," loaded by gensim libraries in Python has been used. A 300 dimensions real number vector represents the word; the tweet embeddings are calculated by taking the average embedding for all words in the tweet. Then, the average embedded vector of each tweet is fed to the classifier and classified either as neutral or as one or more of the 11 emotions (disgust, anger, fear, pessimism, anticipation, joy, love, optimism, sadness, trust, and surprise).

3. Network architecture
   We build a deep learning model of recurrent neural network (RNN), BiLSTM layers. A BiLSTM is a sequence model with two LSTMs: forwarding and backward direction input. BiLSTM increases the amount of information available to the network and improves the context (e.g., knowing the following and the preceding word in a sentence). As a result, it learns faster than the one-directional approach, although it depends on the task. The basic structure of BiLSTM is shown in Fig. 2.

The proposed model contains three BiLSTM layers with 300, 200, and 50 inputs. Each layer has a Relu activation function. A dropout layer follows each BiLSTM with three dropout layers, one after each BiLSTM layer. Following the first BiLSTM is a repeater layer. The last layer in the model is a dense layer with 11 outputs

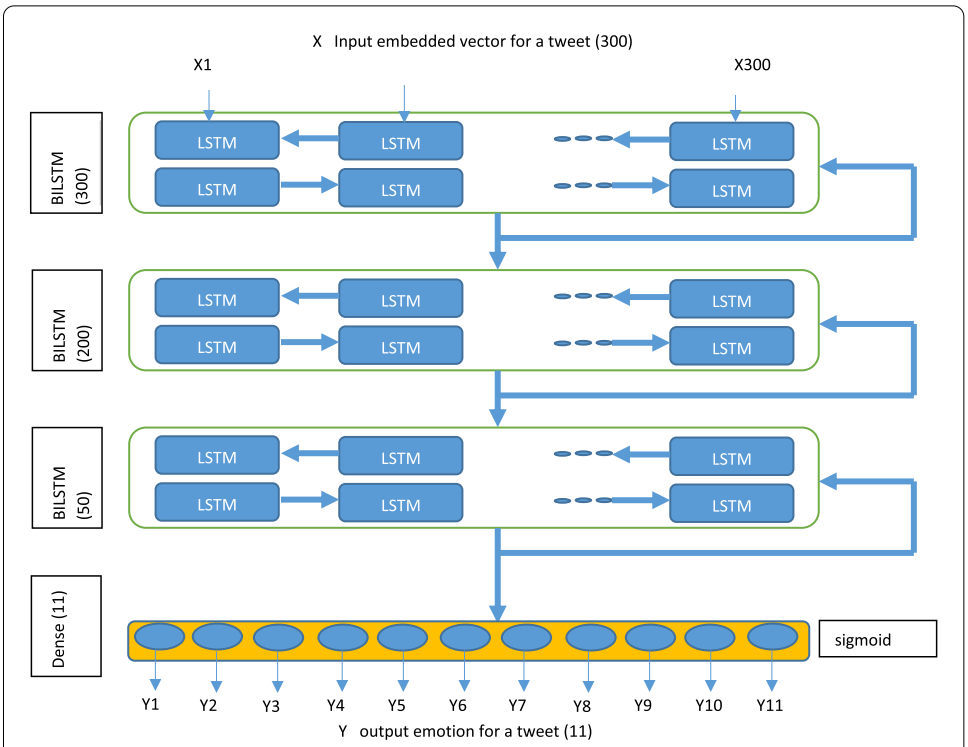

Fig. 3 Deep learning (BILSTM) classification model

corresponding to the 11 emotions. The activation is a sigmoid function; it gives a probability of each emotion. We approximate the values to 0, 1 class for each 11 output representing the 11 emotions. Fig. 3 shows the network structure.

Experimental setup and results

The work has been implemented on a Dell G5 15 laptop Intel i7 10th Gen, CPU 2.6 GHz with 6 GB GPU Nvidia Geoforce RTX2060. Libraries of Scikit learn 0.24.2, genism 3.8.3, and Keras 2.3.0 libraries under the tensorflow2.1.0 platform in Python 3.6.13 have been used. The pre-trained word embedding model "Twitter-CBOW/tweets cbow 300" is loaded by gensim libraries in Python. Dataset is provided publicly by SemEval 2018 task1, the E-C subtask for the Arabic language [1]. This task has 2278 tweets for training, 585 tweets for development, and 1518 tweets for test data. We concatenated three datasets with a total size of 4381 tweets for the cross-validation process. The proposed model has been evaluated using the following metrics.[3]

– Multilabel accuracy (or Jaccard index): "it is the size of the intersection of the predicted and actual label sets divided by the size of their union." It is computed for each tweet t and averaged over all tweets in the dataset T:

[3] https://competitions.codalab.org/competitions/17751.

$$\text{Accuracy}\frac{1}{|\text{T}|}\sum_{\text{t}\in\text{T}}\frac{\text{Gt}\cap\text{Pt}}{\text{Gt}\cup\text{Pt}} \tag{1}$$

Here, Pt is the set of the predicted labels for tweet *t*, *Gt* is the set of the actual (gold) labels for tweet *t*, and *T* is the set of tweets

- Precision: It is the number of true positive results divided by the number of positive results, including those unidentified correctly,

$$\text{PPV} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{2}$$

- Recall: It is the number of true positive results divided by the number of samples that should have been identified as positive. Recall is also known as sensitivity in diagnostic binary classification.

$$\text{Recall}\left(\text{sensitivity or true Positive Rate TPR}\right)\ \text{TPR} = \frac{\text{TP}}{\text{P}} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{3}$$

where TP is true positive, FP False Positive, and FN is False Negative.
F1 score is the harmonic mean of precision and recall

$$\text{F1 score} = 2.\frac{\text{PPV}\times\text{TPR}}{\text{PPV}+\text{TPR}} \tag{4}$$

- Micro-averaged metrics differ from the overall accuracy when the classifications are multilabel. Thus, it is essential to clarify the difference between micro and macro averages (precision, recall, and F1-score). In macro-average, the metrics are independently computed for each class. Then, the average is calculated (all classes are treated equally), whereas micro-average combines the contributions of all classes to compute the average metric. Thus, micro-average is preferred in a multiclass classification setup, especially in imbalance class cases (i.e., having more examples of one class than other classes).
- Micro-averaged F-score is computed as follows:

$$\text{Micro}-\text{avg Precision(micro}-\text{P)} = \frac{\sum_{\text{e}\in\text{E}}\text{number of tweets correctly assigned to emotion class e}}{\sum_{\text{e}\in\text{E}}\text{number of tweets assigned to emotion class e}} \tag{5}$$

$$\text{Micro}-\text{avg Recall(micro}-\text{R)} = \frac{\sum_{\text{e}\in\text{E}}\text{number of tweets correctly assigned to emotion class e}}{\sum_{\text{e}\in\text{E}}\text{number of tweets in emotion class e}} \tag{6}$$

$$\text{Micro}-\text{avg F} = \frac{2\text{Xmicro}-\text{P Xmicro}-\text{R}}{\text{micro}-\text{P}+\text{micro}-\text{R}} \tag{7}$$

where E is the given set of emotions(eleven in our model)

- Macro-averaged F-score is calculated as follows:

$$\text{Precision(Pe)} = \frac{\text{number of tweets correctly assigned to emotion class e}}{\text{number of tweets assigned to emotion class e}} \tag{8}$$

Table 1 Results of the proposed model with and without emoji lexicon

Optimizer	Adam				Nadam			
	F1	P	R	Jacc-index	F1	P	R	Jacc-index
With Emoji Lexicon								
Stem	0.613	0.696	0.549	0.494	0.614	0.691	0.553	0.496
Modified stem	**0.615**	**0.695**	**0.551**	**0.498**	0.615	0.69	0.555	0.497
Without Emoji Lexicon								
Stem	0.591	0.669	0.529	0.471	0.593	0.665	0533	0.476
Modified stem	0.591	0.67	0.528	0.471	0.593	0.664	0.536	0.475

The bold values reflected the best results obtained

$$\text{Recall(Re)} = \frac{\text{number of tweets correctly assigned to emotion class e}}{\text{number of tweets in emotion class e}} \tag{9}$$

$$\text{Fe} = \frac{2\text{XPeXRe}}{\text{Pe} + \text{Re}} \tag{10}$$

$$\text{Macro} - \text{avg F} = \frac{1}{|\text{E}|} \sum_{\text{e}\in\text{E}} \text{Fe} \tag{11}$$

Experimental setup

All experiments are conducted using cross-validation with the number of splits k = 10. Each time one split is taken as a test, another nine splits are for training, and the number of repeats = 3, giving a total of 30 trials in each experiment.

After several experiments, the parameters are tuned as follows. The learning rate is adjusted to a value of 0.001, optimizer algorithm "Adam" with loss "mse" and "accuracy" metrics.

Preprocessing variations

First, the effect of using or not using the emoji lexicon stated in the methodology section has been investigated. These experiments have been tested with different stemming. Different optimizers are commonly used for optimizing SGD; we tried two optimizers: Adam and Nadam. These optimizers belong to adaptive learning rate methods and have been proven to provide the best results in deep learning sentiment analysis systems. The number of epochs is set to 50. The batch size is set to 200; three dropout rates are set to 0.5, 0.5, and 0.5 for the three dropout layers. The learning rate is set to 0.001. Additionally, the activation functions for all BiLSTMs are set to a Relu function to avoid gradient-vanishing problems and overfitting. The loss function is "mse," and the output dense layer activation is set to sigmoid function. The experiments are conducted with and without emoji lexicon (Table 1). Table 1 presents the impact of whether or not to exclude the word "الله" and the derivations as "الله", "اللهم", "بالله", "فالله" from any text normalization or stemming (modified stemming), this positively impacts the accuracy of about 0.1% to 0.3% in the Jaccard index using the emoji lexicon. Best results obtained are when using Emoji lexicon with modified stem and Adam optimizer as shown in bold.

Table 2 Results of the proposed model using different batch sizes and epochs

	Epoch = 20				Epoch = 50			
Batch size	**Mic F1**	**Mic P**	**Mic R**	**Jacc**	**Mic F1**	**Mic P**	**Mic R**	**Jacc**
60	0.605	0.712	0.526	0.483	0.61	0.681	0.553	0.491
100	0.605	0715	0.524	0.482	0.612	0.683	0.556	0.494
200	0.603	0.725	0.516	0.479	**0.615**	**0.695**	**0.551**	**0.498**
300	0.594	0.733	0.50	0.467	0.612	0.708	0.539	0.491

The bold values reflected the best results obtained

Table 3 the drop out layers effect on performance

Model	**Number of Dropout layers**	**Dropout rates**	**Jaccard Acc**
Model1	No dropout	–	0.467
Model2	3	0.2,0.2,0.2	0.488
Model3	3	0.3,0.3,0.3	0.492
Model4	**3**	**0.5,0.5,0.5**	**0.498**
Model5	3	0.8,0.8,0.8	0.439

The bold values reflected the best results obtained

Hyperparameter tuning

Manual grid search has been implemented to choose the best hyperparameters. Different batch sizes with different epochs are investigated and reported through experiments (Table 2). The learning rate is adjusted to a value of 0.001. The optimizer algorithm is set to "Adam," and loss is "mse" and "accuracy" metrics. This experiment used the modified stemmer and the manually built emoji lexicon. The best results are obtained with a batch size of 200 and 50 Epochs as shown in bold font in Table 2.

Dropout effect

Dropout is a regularization method that randomly ignores selected neurons during training. This makes these ignored neurons have no contribution to the activation of downstream neurons, and any weight updates are not applied to the neuron on the backward pass. Thus, when using dropout regularization, the network becomes less sensitive to the specific weights of neurons, leading to better generalization and less over fitting for the training data. Table 3 presents different variations of our model with and without dropout layers; the best results were obtained with the dropout rate of 0.5, smaller dropout is not sufficient for over fitting removal, and greater dropout rate results in the loss of important features. Hence, both cases result in performance degradation. The best results are shown in bold font in Table 3.

Other optimizers

Four optimizers, Adam, Nadam, Adamax, and RMSprop, have been tried for the proposed model while fixing other parameters' model to batch size = 200. The learning rate value is set to 0.001, and the number of epochs is 50, with loss "mse" and

Table 4 Effect of changing the optimizer with epochs of 50

Optimizer	Micro-F1	Micro_P	Micro_R	Jaccard
Adam	**0.615**	**0.695**	**0.551**	**0.498**
Nadam	0.615	0.69	0.555	0.497
RMSprop	0.609	0.69	0.546	0.491
Adamax	0.609	0.726	0.524	0.485

The bold values reflected the best results obtained

Table 5 different Models tested by varying number of BiLSTM

Model	Number of BiLSTM layers	BiLSTM layers- units	Dropout layers	Jaccard Acc
Model1	1	300	1	0.477
Model2	2	300,200	2	0.485
Model3	**3**	**300,200,50**	**3**	**0.498**
Model4	4	300,200,50,50	4	0.481

The bold values reflected the best results obtained

Table 6 Proposed model best parameters

Dropout rates	0.5,0.5,0.5
Optimization	Adam
Loss	"mse"
Number of BiLSTM layers	3
Activation Function in BiLSTMS	Relu
Activation Function in output Layer	Sigmoid
Epochs	50
Batch Size	200
Learning Rate	0.001

"accuracy" metrics. Adam optimizer achieved the best performance as shown in bold font in Table 4.

Multiple layers

The effect of changing the number of layers of the model has been tested through many experiments, and the results obtained are recorded in Table 5. First, we tested the model with one BiLSTM layer and a single drop out layer of dropout rate 0.5, then using two BiLSTM layers with a drop out layer after each BiLSTM layer, the dropout rate is fixed to 0.5. A noticeable performance increase using two layers (Model 2) rather than single layer (Model 1). Further performance enhancement appeared using three layers as in Model 3 as shown in bold in Table 5, but further layers beyond this third layer in our task with the dataset used increase the noise in data and decrease the accuracy.

Table 6 below summarizes the best model parameters obtained from all previous experiments.

Table 7 Proposed model compared with other state-of-the-art models

Model	Preprocessing	Features	Classification algorithm	Validation accuracy	Test accuracy	Micro F1	Macro F1
Proposed Model	Normalization + a manual emoji lexicon + ARLSTEM	AraVec [18]	Bidirectional LSTM	**0.575**	**0. 498**	0.615	0.440
EMA	Normalization, a manual emoji lexicon + ARLSTEM	AraVec	SVC	0.488	0.489	**0.618**	**0.461**
TW-Star	Emo + Stem + stop	TF-IDF	SVM	NA	0.465	0.597	0.446
UNCC	Tokenization white spaces removal	AraVec + Affective Tweets features	a fully connected neural network	NA	0.446	0.572	0.447
SVM-Unigrams	NA	Unigrams	SVM	NA	0.38	0.516	0.384
Amrita	NA	Doc2Vec	RF	NA	0.254	0.379	0.25

Performance comparison with other machine learning models

A comparison between the proposed model and other models that studied the same task with the same dataset is performed in Table 7. The system achieves about 9% enhancement in validation accuracy compared with the last best model in the same task using Support Vector classifier SVC; it outperforms all the models in terms of accuracy (Jaccard index) and validation accuracy. It outperforms the EMA model with about 0.9%, the other SVM model (TW_star) with 3.3%, and the deep learning model UNCC with 5.2%.

Discussion

Recurrent neural networks (RNNs) are deep learning neural networks that consider sequence and are more suitable for textual data classification. However, RNN suffers from the vanishing gradient problem in classifying long data sequences, hence using Long Short Term Memory LSTM neural networks to solve this problem. While LSTM extracts context in the forward direction, the Bidirectional variation LSTM (BiLSTM) can extract the contextual information by dealing with dependencies in forward and backward directions. The output is a combination of the corresponding states of the forward and backward LSTM. Our proposed approach is the first deep learning model based on BiLSTM for Arabic social media multi label emotion classification. Multiple layers have been tried out; increasing the number of BiLSTM layers more than the three layers proposed did not enhance performance and only increased the computational complexity due to the increase in the number of parameters and weights to be computed through further layers. The Computational complexity of a standard LSTM network per time step with stochastic gradient descent optimization technique is O(W), where W is the total number of parameters [30], whereas in our model using BiLSTM, the computational complexity is O(2w) since the input text is passed twice by forward and backward LSTM cells. Although, BiLSTM has high computational complexity but is effective in size and dimensionality reduction of the feature vector. BiLSTM uses global features from the text. Also in our model, drop out is applied, reducing the number of parameters and preventing model over-fitting.

Also, after experimenting with different epochs for this model, the optimal number of epochs is set to 50; the more epochs only increase the time complexity without any noticeable enhancement of performance. The optimal batch size is 200; increasing the batch size accelerates the run time, but higher batches beyond 200 caused performance degradation. The model performs better than all other models implemented for the same task with the same dataset.

Conclusions and future work

This study proposed a method to build a deep learning model for multilabel emotion classification in Arabic tweets using the SemEval2018 Task1 dataset. Several preprocessing steps are adapted, such as normalization, stemming, replacing the most common emojis with their corresponding meanings using a manually created lexicon of emojis; word embedding proved to be the best technique for feature selection. Aravec pre-trained word embedding model with CBOW builds 300 dimension word vectors for each word in our dataset. Then, the average embedded word vector is calculated for each tweet, and BiLSTM is used for classification. The proposed method achieved the best performance results compared with SVM, RF, and fully connected deep NN. It achieved 9% increase in validation than the previously best obtained by SVM. BiLSTM increases the amount of information through a two-way network and improves the context used by the network. The effect of hyperparameter tuning is investigated using different experiments since the grid automatic search is not supported by Keras libraries in the LSTM model. Future improvements in preprocessing, such as removing ambiguity results from stemming the nouns ending with, ان,ين usually as a مثني, and words ending with ون,ين as جمع plural, and applying more restricted grammatical rules, will enhance model performance. The effect of using different deep learning models, such as CNN, will also be investigated.

Acknowledgements
Not applicable.

Authors' contributions
All authors contributed to the structuring of this paper. All authors read and approved the final manuscript.

Funding
The research has no funding.

Availability of data and materials
The data that support the findings of this study are publicly available, at https://competitions.codalab.org/competitions/17751#learn_the_details-datasets.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Present Address: Systems & Information Department, Engineering Research Division, National Research Centre NRC, Dokki, Giza 12311, Egypt. [2]Faculty of Engineering, Ain Shams University, Cairo, Egypt.

Received: 18 June 2021 Accepted: 4 October 2021
Published online: 19 October 2021

References

1. Mohammad S, et al. Semeval-2018 task 1: affect in tweets. In: Proceedings of the 12th international workshop on semantic evaluation. 2018.
2. Ekman P. An argument for basic emotions. Cogn Emot.1992;6(3–4):169–200
3. Plutchik R. A general psychoevolutionary theory of emotion. Theories of emotion. Amsterdam: Elsevier; 1980:3–33
4. Plutchik R. The psychology and biology of emotion Harper. New York: Collins College Publishers; 1994
5. Trad C et al. Facial action unit and emotion recognition with head pose variations. In: International Conference on Advanced Data Mining and Applications. 2012. Springer.
6. Ruiz-Garcia A. A hybrid deep learning neural approach for emotion recognition from facial expressions for socially assistive robots. Neural Comput Appl. 2018;29(7):359–73
7. Wegrzyn M. Mapping the emotional face. How individual face parts contribute to successful emotion recognition. PloS ONE. 2017;12(5):e0177239
8. Filippini C. Facilitating the child–robot interaction by endowing the robot with the capability of understanding the child engagement: the case of mio amico robot. Int J Soc Robot. 2020;13:1–13
9. Ozcan T, Basturk A. Transfer learning-based convolutional neural networks with heuristic optimization for hand gesture recognition. Neural Comput Appl. 2019;31(12):8955–70
10. Constantine L et al. A framework for emotion recognition from human computer interaction in natural setting. in 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2016), Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM 2016). 2016.
11. Hibbeln MT. How is your user feeling? Inferring emotion through human-computer interaction devices. MIS Q. 2017;41(1):1–21
12. Patwardhan AS, Knapp GM. Multimodal affect analysis for product feedback assessment. arXiv preprint, arXiv:1705.02694, 2017.
13. Karyotis C. A fuzzy computational model of emotion for cloud based sentiment analysis. Inf Sci. 2018;433:448–63
14. Giatsoglou M. Sentiment analysis leveraging emotions and word embeddings. Expert Syst Appl. 2017;69:214–24
15. Abdul-Mageed M, Ungar L. Emonet: Fine-grained emotion detection with gated recurrent neural networks. in Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers). 2017.
16. Badaro G et al. Ema at semeval-2018 task 1: Emotion mining for arabic. In: Proceedings of The 12th International Workshop on Semantic Evaluation. 2018.
17. Badaro G. Arsel: a large scale arabic sentiment and emotion lexicon. OSACT. 2018;3:26
18. Soliman AB, Eissa K, El-Beltagy SR. Aravec: a set of arabic word embedding models for use in arabic nlp. Procedia Comp Sci.2017;117:256–65
19. Sonawane B, SharmaP. Review of automated emotion-based quantification of facial expression in parkinson's patients. Vis Comput. 2020;37(1):17
20. Shoukry A, Rafea A. Preprocessing Egyptian dialect tweets for sentiment mining. 2012.
21. Abainia K, Ouamour S, Sayoud H. A novel robust Arabic light stemmer. J Exp Theor Artif Intell. 2017;29(3):557–73.
22. Mulki H et al. Tw-star at semeval-2018 task 1: Preprocessing impact on multi-label emotion classification. In: Proceedings of The 12th International Workshop on Semantic Evaluation. 2018.
23. Abdullah M, Shaikh S. Teamuncc at semeval-2018 task 1: Emotion detection in english and arabic tweets using deep learning. In: Proceedings of the 12th international workshop on semantic evaluation. 2018.
24. Unnithan NA et al. Amrita_student at SemEval-2018 Task 1: distributed representation of social media text for affects in tweets. In: Proceedings of The 12th International Workshop on Semantic Evaluation. 2018.
25. George A, BG, HB, Soman K. Teamcen at semeval-2018 task 1: global vectors representation in emotion detection. In: Proceedings of the 12th international workshop on semantic evaluation. 2018.
26. Pennington J, Socher R, and Manning CD. Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
27. Rostami M. Review of swarm intelligence-based feature selection methods. Eng Appl Artif Intell. 2021;100:104210
28. Rostami M, Berahmand K, Forouzandeh S. A novel community detection based genetic algorithm for feature selection. J Big Data. 2021;8(1):1–27
29. Rostami M, Berahmand K, Forouzandeh S. A novel method of constrained feature selection by the measurement of pairwise constraints uncertainty. J Big Data. 2020;7(1): 1–21
30. Sak H, Senior A,Beaufays F. Long short-term memorybased recurrent neural network architectures for large vocabulary speechrecognition. 2014. arXiv:1402.1128.

Publisher's Note