

A Method of Deep Learning Tackles Sentiment Analysis Problem in Arabic Texts

Abdulhakeem Q. Albayati¹, Ahmed S. Al-Araji², Saman H. Ameen³

^{1,2,3}Computer Engineering Department, University of Technology-Iraq, Baghdad Iraq
10466@uotechnology.edu.iq, 60166@uotechnology.edu.iq, Saman7011@yahoo.com

Abstract— Sentiment Analysis (SA) is a field of Natural Language Processing (NLP) whose goal is to extract the emotion, sentiment or more general opinion expressed in a human-written text. Opinions and emotions play a central role in human life. Therefore, there are many academic researches in this field for processing many languages like English. However, there is scarce in its implementation with addressing Arabic Sentiment Analysis (ASA). It is a challenging field where Arabic language has a rich morphological structure and there are many other defies more than in other languages. For that, the proposed model tackles ASA by using a Deep Learning approach. In this work, one of word embedding methods, such as a first hidden layer for features extracting from the input dataset and Long Short-Term Memory (LSTM) as a deep neural network, has been used for training. The model combined with Softmax layer is applied to turn numeric outputs from LSTM layer into probabilities to classify the outputs to positive or negative. There are two datasets that are used for training the model separately with each one. The first one is ASTD dataset as a dialectal Arabic type about different tweets from internet, the results with this dataset is compared with another academic work that used the same one. The results from this work outperforms through accuracy about 14.95% and F-score about 15.14% more than what performed in the previous work. The second one is HTL dataset as a modern standard Arabic type about opinions of reviewers on different hotels from several countries. This dataset is bigger in size than the first one to show the size effect on the results of this model. So, the accuracy increased about 11% and F-score about 10.8% more than what performed with the first dataset.

Index Terms— Arabic Sentiment Analysis, Deep Learning, Long Short-Term Memory, Natural Language Processing.

I. INTRODUCTION

Sentiment analysis (SA) has become one of the most important fields in academic researches and in decision making in the world. It is defined as the automatic operation of comprehending an opinion concerning a particular subject obtained from a spoken or written language [1]. Nowadays sentiments and opinions are not limited in articles and advertisements of newspapers, television, and radio. They spread in different social media on the internet [2]. Since the beginning of 2000, the topic of SA has been widely adopted as one of the recognized effective approaches in the realm of Natural Language Processing (NLP) [3]. It has obvious applications in real life; examples include commerce, politics, education, tourism, and health. The majority of these researches, nevertheless, have only adopted the English language and to less extent other Indo-European languages [4]. In spite

of this importance of SA, there is only a moderate amount of the studies regarding Arabic language, and more interestingly, the growth of this research has been quite slow. The Arabic language is usually considered as a rich morphological language and complex, mainly because of its vagueness and wide morphological system. Also, the Arabic language has three main variants: Classical Arabic (CA), Modern Standard Arabic (MSA), and the informal Arabic known as colloquial or Dialectal Arabic (DA). This complex nature in line with resources scarcity and the changing dialects create difficulties restraining the studies in Arabic Sentiment Analysis (ASA) from being moving forward steadily [5]. Therefore, there is a need for more researches and solutions for SA problems especially with a language in Arabic because it has historical and strategic importance. Also, the huge number of Arabic speakers in the internet will add more importance to this language [6]. Several studies have been conducted to study the ASA and enhance its accuracy, ranging from simple linear models to more complicated ones like those adopted in deep neural network models [7]. Recently, Deep Learning (DL) approaches (a subset of Machine Learning (ML) in the Artificial Intelligence (AI) field) have been achieved successfully with SA [8]. All that mentioned above like importance of SA, importance of Arabic language, scarcity of ASA researches and using DL to improve the output of SA models have become a motivation for this work. The current work attempts to investigate one of the DL models used in ASA with the purpose of enhancing the accuracy of ASA. Along with using DL, one of word embedding methods is used in this work. To deal with a text in NLP problem. There are different levels for this purposes like: character-level [9], word-level [10] or document-level [11]. In this work, the word-level is adopted to give way to use an efficient and dense representation in which similar words have a similar encoding. Also, there are many strategies for learning word embeddings such as Word2vec by Google [10], Glove by Stanford University [12], FastText by FAIR [13] and Embedding layer by Keras (DL library) [14] that is used in this work. As described above, the SA is a very important field for many disciplines in various real-life fields. Therefore, this following submits a general review of different approaches proposed to address SA in English and Arabic: First approach used LSTM to predict polarity from tweets in twitter in English language by simulating the interactions of words during the compositional process. Multiplicative operations between word embeddings through gate structures are used to provide more flexibility and to produce better compositional results compared to the additive ones in simple Recurrent Neural Network (RNN) as in [15]. The second one presented a model for carrying out Arabic Sentiment Analysis (ASA) by augmenting an ML (using complement naïve Bayes) approach with a set of features derived from an Arabic sentiment lexicon as well as from the text itself as in [16]. The third approach used an ensemble model, combining Convolution Neural Network (CNN) and LSTM to predict the polarity of Arabic tweets on the Arabic Sentiment Tweets Dataset (ASTD) as in [17]. The fourth one worked to investigate the merit of using deep models for SA in Arabic, focusing on the sentence level sentiment classification. Four different architectures are explored. Basically, four DL models are explored: Deep Neural Network (DNN), Deep Belief Network (DBN), Deep Auto Encoder (DAE), and combined DAE with DBN. Three are based on DBN and DAE, where the input data model is based on the ordinary Bag of Words (BoW), with features based on the recently developed Arabic Sentiment Lexicon in combination with other standard lexicon features as in [18].

The contribution of this research is that one of the DL models has been presented to tackle ASA problem for predicting the sentiment of sentences in two of Arabic datasets. Long Short-Term Memory (LSTM) as a deep neural network is used with word embedding

layer from Keras library and SoftMax as output layer rather than hand-crafted features for ASA model (Hand-crafted features means the properties that are derived from texts manually by experts like applying Bag of words (BoW) and term frequency-inverse document frequency (TF-IDF) methods to give a number that represents the frequency for each word [19]). Also, this work is suitable for MSA as shown later in the section of “Result and Discussion” when applying the model on HTL dataset. In addition to it is suitable for DA as shown in ASTD dataset.

II. THE PROPOSED MODEL

In this work, the method is proposed to address the ASA problems - a field of Arabic NLP (ANLP) - by one of a DL model (LSTM) to predict the sentiment polarity of sentences in the used dataset in this work. There are two datasets used for this work. The first one is called Arabic Sentiment Tweets Dataset (ASTD) and it is taken from [20], whilst the second one is HTL Dataset and is taken from [21]. There are multiple layers in this work as shown

in *Fig. 1*:

A. Preprocessing for Dataset

Datasets are usually noisy and have many non-desirable characters and symbols that may have negative effects on learning quality, therefore it is essential to clean these datasets to make the process of learning by the neural network easier. Nevertheless, it should be clear that the cleaning of datasets will remove some information from the dataset itself, which may be crucial for a specific application or analysis. Data preparation and enhancement on the dataset in this work mainly consist of the following operations like removing any words not in Arabic script, orthographic normalization like (remove diacritics, remove repetition (elongation), normalize some Arabic characters and punctuations), stemming the texts and at the last removing stopwords and re-join remaining words.

B. Word Embedding Layer

To deal with a text in the NLP problem, the first thing that must be applied after preprocessing is converting the text to feature vector from a fixed length of numbers. The Word embedding is a word representation type that lets words with the same meaning to get the same representation for being comprehended by the ML algorithms. In other words, it can be defined as words mapping into vectors with real numbers by utilizing dimension reduction, probabilistic model, or neural network on the matrix of word co-occurrence. Actually, it is a technique of language modeling and feature learning. The resulting embedded vectors are representations of categories where similar categories (relative to the task) are closer to one another. In this work, the word-level is adopted and one of the algorithms of the word embedding layer by Keras is applied as the first hidden layer as shown in *Fig. 1*; to give a way to use an efficient and dense representation in which similar words have a similar encoding and for converting each word in each sentence in the dataset to a vector for preparing texts to be the input values to the next layer of the model (LSTM layer).

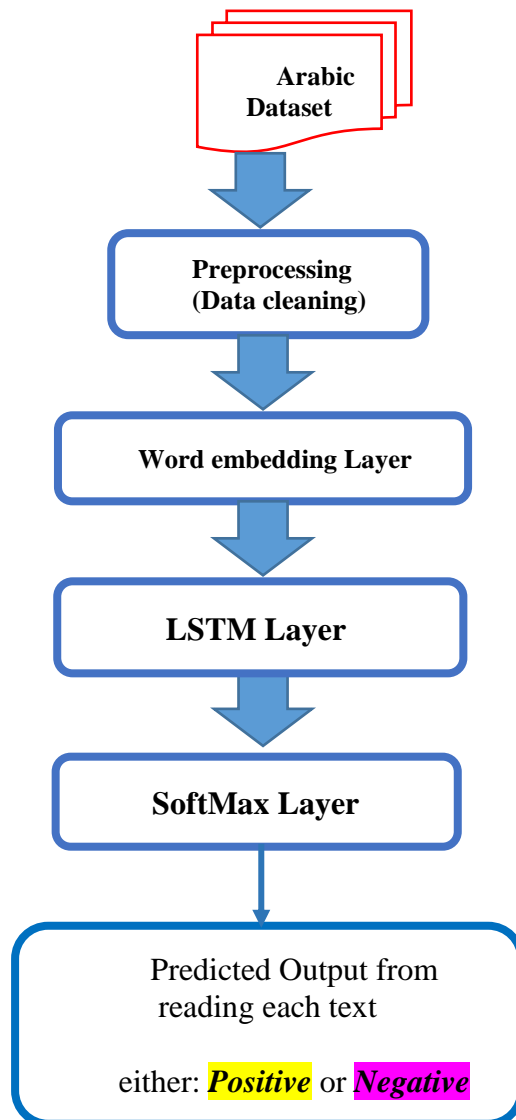


FIGURE 1. Block Diagram for ASA model.

C. LSTM Layer

Lately, the domain of DL has witnessed significant accomplishments in the branch of SA and is seen as the most advanced model in many languages. Nevertheless, the pace of advancements in the ASA still grows slowly and thus requires more progress. Therefore, one of the DL methods (LSTM) is used to address the ASA problem in this paper. LSTM is considered as a type of RNN. RNNs deal with sequences as a feed-forward neural network with a single hyperbolic tangent (tanh) layer by iterating through the sequence elements and preserving a *state* that has data belongs to what it has lastly noticed [22] [23] as shown in Fig. 2 [7]. However, at the same time RNNs suffer from the problem of vanishing gradients, which hampers the learning of long data sequences. The gradients carry information used in the RNN parameter update. When the gradient becomes smaller and smaller, the parameter updates become insignificant which means no real learning is done [23] [24].

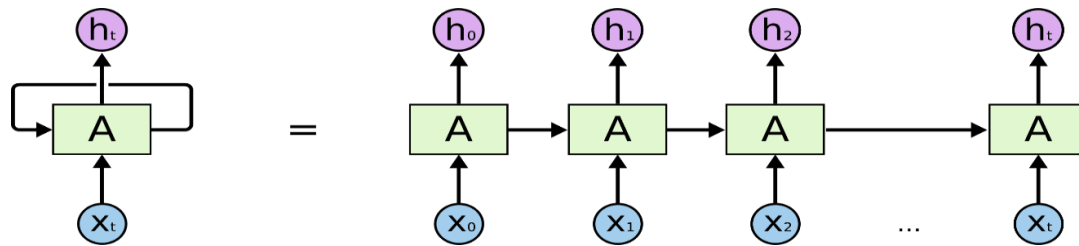


FIGURE 2. RNN folded and unfolded (Where x_t : input at state t , A : hidden (tanh) layer, h_t : output T at state t) [7]

LSTM has capable to resolve this problem of RNN by its ability to learn long term dependencies [25]. LSTMs deep neural networks are categorized by a chain of repeating modules of neural network with four neural network layers that systematically interact instead of one neural layer as in RNN [7] as shown in Fig. 3 [7]:

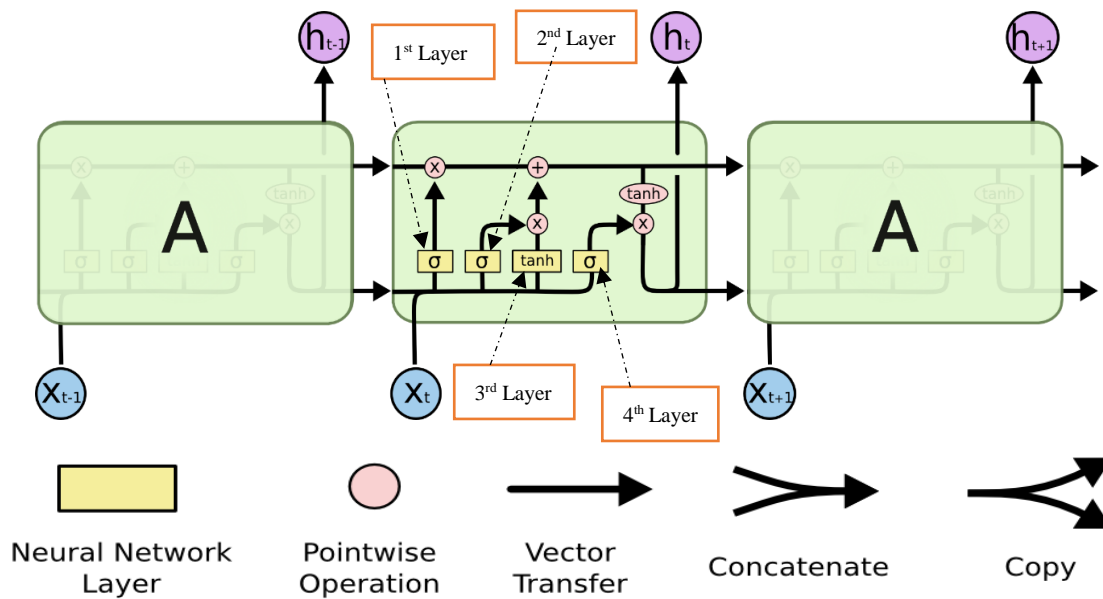


FIGURE 3. LSTM module contains four interacting layers [7]

Because of these characteristics for LSTM, it is used as a deep neural network for the proposed model in this work.

D. SoftMax Layer

SoftMax is frequently noticed in the ANN's output layer side by side with the linear and non-linear activation functions. The classification neural networks are the typical place where the SoftMax function can be applied. The main merit of the SoftMax activation function is its ability to program the network's output to describe the probability that the input will be into which class. Without the abilities of the SoftMax function, the outputs of the neuron are merely values at a numeric scale where the winning class is determined by the largest value [26]. So, SoftMax with this merit is very useful in this work with the output will describe the probability of the input sentence as either positive or negative. SoftMax function has illustrated mathematically in equation (1) below [26]:

$$S_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{1}$$

Where i = the index of the output neuron being calculated, j = the indexes of all neurons in the group/level of layer, Z = designates the array of output neurons.

III. Result and Discussion

To apply ASA model that is proposed in Fig. 1, it has followed the flowchart as shown below in Fig. 4:

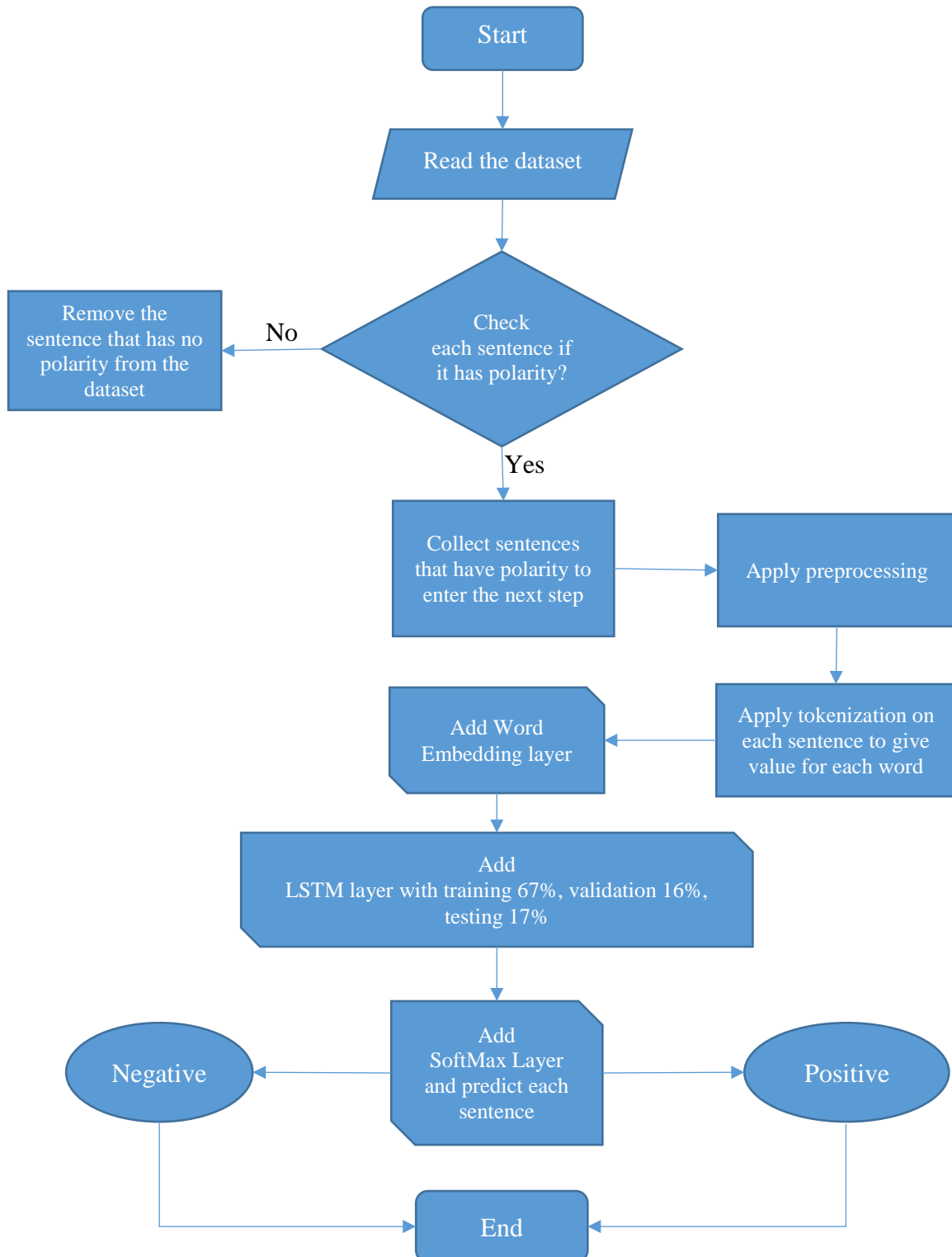


FIGURE 4. Flowchart to Apply ASA model

Step 1: Use python program with Keras library for programming DL model.

Step 2: Read the dataset with pandas library as Comma Separate Value (.csv)

Step 3: Remove any sentence in the datasets that has not positive or negative polarity like neutral polarity or objective sentences.

Step 4: Apply preprocessing function on the sentences of the dataset as mentioned in the subsection A.

Step 5: Apply tokenization on the words after return words from preprocessing and rejoin them in each sentence. This step gives each word a value and return these values as a sequence with each sentence.

Step 6: After that, call model as (sequential) from Keras library to perform word embedding layer and LSTM layer.

Step 7: Add word embedding layer to the model for giving each word an embedding and dense vector for preparing to LSTM layer, every vector for each word represent 100 features, output dimension = 100.

Step 8: Add LSTM layer with number of hidden units = 100. Moreover training = 67%, validation = 16%, and testing = 17%

Step 9: At the end of this model, add SoftMax layer to describe the probability of the two outputs (Positive or Negative).

Step 10: To calculate the loss function for this model, compile this model with “Categorical Cross-Entropy (CCE)”. It is a metric used for single or multiple label categories to measure the differences in probability between them. This function is used with classification problem like the ASA problem in this work. And It is expressed mathematically by sum for all output nodes in the following equation (2) [27]:

$$CE(Y, Y') = - \sum_j \sum_i (Y_{ij} \times \log(Y'_{ij})) \quad (2)$$

Where Y = Actual output, Y' = Prediction output, log = Natural logarithm

Step 11: Finally, apply an appropriate optimization function to decrease the loss function and reach near the correct prediction. This function is called “Adaptive Momentum Estimation (Adam)”.

There are two datasets that are used with this model as shown below:

A. ASTD Dataset

The first dataset is called ASTD, its type is DA and its size is 1.48 MB. It consists of about 10,000 tweets from the internet. The same dataset had been used by the academic researchers in [17] and they used an ensemble model (combining CNN and LSTM models). The current model in this work outperforms in results on the model in [17] as shown in the Table I. This is done by omitting some information that are not important for predicting ASA problem like neutral sentences (neither positive nor negative) and keeping useful information like positive and negative sentences, this procedure will be affected on the results. Table II shows the parameters that are used to configure this model . As mathematically equations express how the accuracy and F-score is measured, see the following equations (3) & (4) [17],[28]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

Where: **TP** = True Positives, **TN** = True Negatives, **FP** = False Positives, & **FN** = False Negatives.

$$F\text{-Score} = 2 * \frac{P * R}{P + R} \quad (4)$$

Where: P = Precision = $\frac{TP}{TP+FP}$, and R = Recall = $\frac{TP}{TP+FN}$

TABLE I. COMPARING RESULTS WITH ANOTHER WORK

Model	Accuracy	F-score
Ensemble model in [17]	65.05%	64.46%
This model (LSTM)	80%	79.6%

TABLE II. LSTM MODEL PARAMETERS ARE USED IN THIS MODEL

Hyper-Parameter	Value
Number of Hidden units in LSTM	100
Learning Rate	0.001
Training dataset	67%
Validation dataset	16%
Testing dataset	17%
No. of Epochs	10
Batch Size	256

The performance for this model to the ASTD dataset and the loss rate has been illustrated as shown in the figures below (Fig. 5 and Fig. 6):

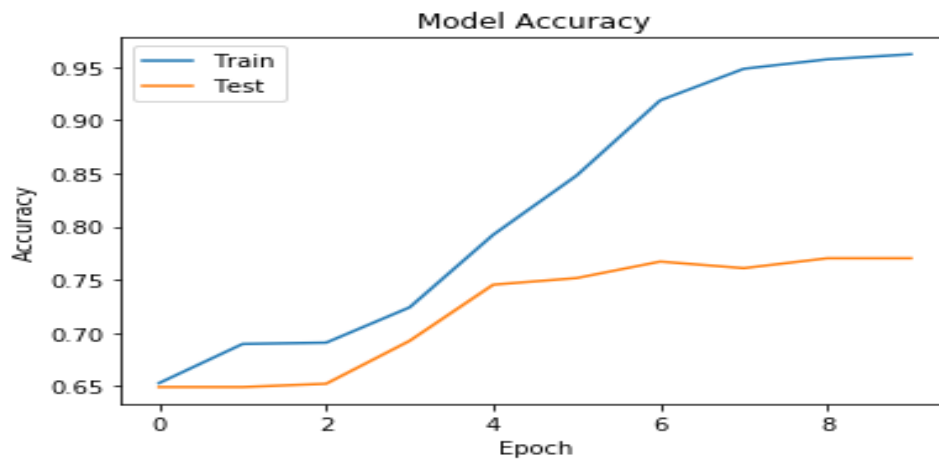


FIGURE 5. Model Accuracy for ASTD

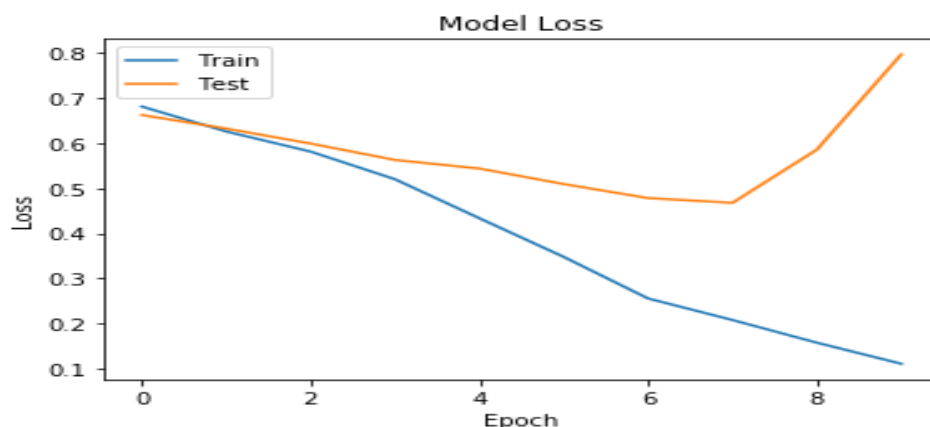


FIGURE 6. Model Loss for ASTD

These differences in the accuracy of the model between the training and testing occurred because the concept of *overfitting* occurred in the model. The overfitting has been occurred when the model learns in the training dataset too well, while it does not perform well with an unseen sample in the testing.

The differences in the loss rate between training and testing will illustrate the overfitting concept as shown in *Fig. 6* above. The loss rate means how well the model is doing. Unlike accuracy, loss is not a percentage. It is a summation of the errors made for each example in the training sets.

Therefore, when the loss rate for the model in the training is decreasing because the model trained well with the dataset (i.e. This dataset becomes biased for the model), while the model does not perform well on unseen data (unbiased data) during the testing. This is led to increase in the model accuracy during training while in testing the accuracy will decrease and this is exactly what the overfitting means and how it occurred in this model.

B. HTL Dataset

Another dataset is called HTL (abbreviation of Hotel) that has size of 14.5 MB about Arabic reviews on several hotels in different countries. After taking this dataset, this model has been applied to it. The results have yielded as following: an accuracy of 91% and an F-score of 90.4% have been achieved. These results proved when the dataset for the model is increased, the performance of the model improved and vice versa. Also, the overfitting problem with bigger dataset in this model is also decreased as shown below in *Fig. 7* and *Fig. 8*:

TABLE III. RESULTS OF HTL DATASET

Name of Dataset	Accuracy	F-score
HTL Dataset	91%	90%

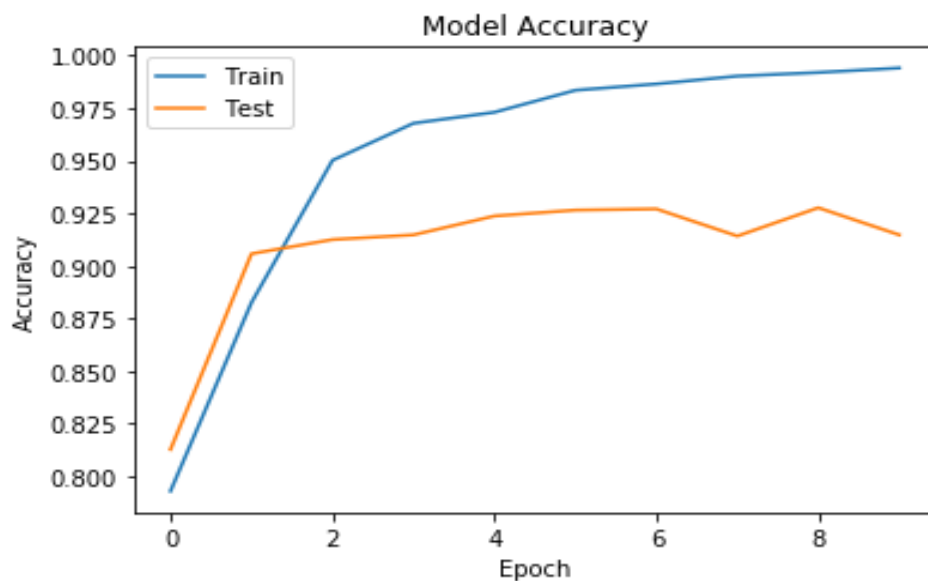


FIGURE 7. Model Accuracy for HTL

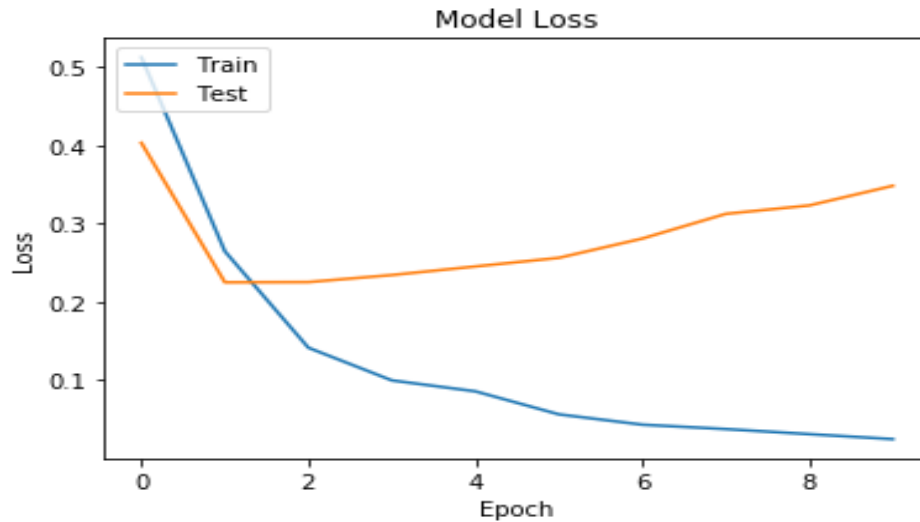


FIGURE 8. Model Loss for HTL

IV. CONCLUSION

Sentiment analysis has become one of the most important fields in academic researches and in decision making in the world. Nowadays sentiments and opinions are not limited in articles and advertisements of newspapers, television, and radio. They spread in different social media on internet. Therefore, there is a need for more researches and solutions for SA problem especially with languages have rich morphology structures like Arabic. Recently, DL approaches have been achieved successfully with SA.

In this paper, one of the DL models has been presented to tackle ASA problem. LSTM neural networks are used with the word embedding layer rather than hand-crafted features for ASA model. Both metrics (accuracy and F-score) are used to measure the performance of the model. Also, this work is compared with another previous academic work [17] that used the same dataset ASTD. The current results in this work outperform the results in [17], where the F-score about 15.14% and the accuracy about 14.95% achieve absolute improvements over them (Accuracy and F-score are metrics that measure the performance of the model). Moreover, there is another dataset (HTL) used in this work which is bigger from the first one to prove when the size of dataset enlarges then the performances of this model will be increased. Therefore, this model can be used in many trending applications in various real-life fields including finance, marketing, political science, health science, communications, and even history.

REFERENCES

- [1] S. Kolkur, G. Dantal, and R. Mahe, "Study of Different Levels for Sentiment Analysis," *Int. J. Curr. Eng. Technol.*, vol. 55, no. 22, pp. 2277–4106, 2015.
- [2] G. S. Kaseb and M. F. Ahmed, "Arabic Sentiment Analysis approaches: An analytical survey," *Int. J. Sci. Eng. Res.*, vol. 7, no. 10, pp. 712–723, 2016.
- [3] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, pp. 1–34, Jul. 2018.
- [4] N. Boudad, R. Faizi, R. Oulad Haj Thami, and R. Chiheb, "Sentiment analysis in Arabic: A review of the literature," *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 2479–2490, Jul. 2017.

- [5] G. Alwakid, T. Osman, and T. Hughes-Roberts, "Challenges in Sentiment Analysis for Arabic Social Networks," in 3rd International Conference on Arabic Computational Linguistics, ACLing 2017, 2017, vol. 117, pp. 89–100.
- [6] A. Shoukry and A. Rafea, "Sentence-level Arabic sentiment analysis," in Proceedings of the 2012 International Conference on Collaboration Technologies and Systems, CTS 2012, 2012, pp. 546–550.
- [7] P. Goyal, S. Pandey, and K. Jain, *Deep learning for natural language processing: Creating neural networks with Python*. Bangalore, Karnataka, India: Apress, 2018.
- [8] S. Al-Azani and E. S. M. El-Alfy, "Hybrid Deep Learning for Sentiment Polarity Determination of Arabic Microblogs," in International Conference on Neural Information Processing, 2017, vol. 10635 LNCS, pp. 491–500.
- [9] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-Aware neural language models," in 30th AAAI Conference on Artificial Intelligence, AAAI 2016, 2016, pp. 2741–2749.
- [10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, 2013, pp. 1–12.
- [11] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in 31st International Conference on Machine Learning, ICML 2014, 2014, vol. 32, pp. 2931–2939.
- [12] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
- [13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Trans. Assoc. Comput. Linguist.*, vol. 5, no. 1, pp. 135–146, Jul. 2017.
- [14] "Home - Keras Documentation." [Online]. Available: <https://keras.io/>. [Accessed: 06-Mar-2019].
- [15] X. Wang, Y. Liu, C. Sun, B. Wang, and X. Wang, "Predicting polarities of tweets by composing word embeddings with long short-Term memory," in ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference, 2015, vol. 1, pp. 1343–1353.
- [16] S. R. El-Beltagy, T. Khalil, A. Halaby, and M. Hammad, "Combining lexical features and a supervised learning approach for arabic sentiment analysis," in International Conference on Intelligent Text Processing and Computational Linguistics, 2018, vol. 9624 LNCS, pp. 307–319.
- [17] M. Heikal, M. Torki, and N. El-Makky, "Sentiment Analysis of Arabic Tweets using Deep Learning," in The 4th International Conference on Arabic Computational Linguistics (ACLing 2018), 2018, vol. 142, pp. 114–122.
- [18] A. Al Sallab, H. Hajj, G. Badaro, R. Baly, W. El Hajj, and K. Bashir Shaban, "Deep Learning Models for Sentiment Analysis in Arabic," in Proceedings of the Second Workshop on Arabic Natural Language Processing, 2015, pp. 9–17.
- [19] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, First. New York: CAMBRIDGE UNIVERSITY PRESS, 2008.

- [20] M. Nabil, M. Aly, and A. F. Atiya, "ASTD: Arabic sentiment tweets dataset," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2515–2519.
- [21] H. ElSahar and S. R. El-Beltagy, "Building large arabic multi-domain resources for sentiment analysis," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9042, pp. 23–34.
- [22] F. CHOLLET, *Deep Learning with Python, First*. Shelter Island in New York: Manning Publications Co., 2018.
- [23] A. S. Al-Araji, "Genetic algorithm and Elman neural network used for tuning the parameters of the PID neural controller based model reference". *Iraqi Journal of Computer, Communication, Control Systems Engineering. University of Technology. Vol. 5, No.1, pp. 111-126, 2005.*
- [24] C. C. Aggarwal, *Neural Networks and Deep Learning*. Yorktown Heights, NY, USA: Springer International Publishing, 2018.
- [25] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [26] J. Heaton, *Artificial Intelligence for Humans, Volume 3: Neural Networks and Deep Learning*, 3rd ed. Washington, USA: Heaton Research, Inc., 2015.
- [27] Z. Zhang and M. R. Sabuncu, "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels," in *Advances in Neural Information Processing Systems 31 (NIPS 2018)*, 2018, pp. 8778–8788.
- [28] W. Pang, "A Machine Learning Approach for Aspect-based Sentiment Analysis on Social Media," *Technical University of Dortmund*, 2018.