IJCI v10 -01(2023) 65-77
International Journal of Computers and Information (IJCI)
Available online at *https://ijci.journals.ekb.eg/*

Arabic Language Sentiment Analysis using Bidirectional Long Short Term Memory

Osama M.Elsamadony[a], Arabi Keshk[a] , Amira Abdelatey[b]

[a] *Computer Science Department, Faculty of Computers and Information, Menoufia University, Shebin Elkom 32511, Egypt*
[b]*Information Systems Department, Faculty of Computers and Information, Menoufia University, Shebin Elkom 32511 , Egypt*
osama.mohamed.elsamadony@gmail.com, arabi.keshk@ci.menofia.edu.eg,amira.abdelatey@ci.menofia.edu.eg

Abstract

The amount of data generated in the digital era is huge since the super growth of social networks. Sentiment analysis (SA) seeks to extract opinions from a text and determine the polarity (positive, negative, or neutral). SA is widely used to refer to English. The topic of this study is SA in the Arabic language. There is an amalgamation between Word2Vec and Bidirectional Long-Short Time Memory (BLSTM) used in this paper. Firstly, words in reviews are transferred into their corresponding vectors with word representation models. Secondly, the sequence of words in the sentences passes as input to BLSTM. BLSTM not only captures long-range information and solves the gradient attenuation problem, but it also better represents the future semantics of the word sequence. The polarity was calculated using Word2Vec representation models, which rely on meaning and context. A BLSTM-based deep learning (DL) architecture is proposed. The result shows that the BLSTM Model Architecture surpasses ML, CNN, and LSTM Architectures with a maximum accuracy of 94.88 percent.

Keywords: Type BLSTM; Sentiment Analysis; Word2Vec; CNN; DL.

1. Introduction

SA is a computational activity that extracts opinions and determines polarity from written text [1]. It is one of the NLP tasks [2]. Since it is used to extract the polarity from the written text. Therefore, it is used to track opinions, judgments, and beliefs expressed in reviews, blogs, tweets, comments, and social networking sites toward a product, a service, issues, events, a person, and so on [3]. Nowadays, many customers make use of this feedback for their decision-making [4]. Most inquiries have concentrated on English SA, but few have tended to Arabic SA [5]. Arabic is the most spoken dialect in the Arab nation, with 47,572,891 people speaking this dialect. According to Internet World Stats, it was reported that Arabic was the fourth most popular language on the Internet with 237,418,349 users [6]. DL, a branch of ML, is currently deemed a core technology of today's Fourth Industrial Revolution [7]. State-of-the-art systems related to the AI domain depend on DL techniques and approaches and have also accomplished huge gains in diverse fields [8]. DL, the ML subfield, relies on a lot of algorithms to gain numerous representation levels to find the model for complex patterns in the data [9]. One of the major advantages of DL over diverse traditional ML algorithms is the capacity to perform by itself feature engineering [10]. BLSTM architecture is used. The BLSTM in Arabic SA with word representation is explored. First, a word representation model relied on Word2Vec to convert words to their related vectors. Second, BLSTM is used. As a default behavior, LSTMS retains information for a long time. Since LSTM is fed with the vector embedding which represents each word generated based on the word representation. To generate word embedding, the word2vec model has been used. The major contribution is summarized as:A BLSTM model for effective Arabic SA is identified, which makes greater use of text sequence relationships for learning sentence semantics and storing contextual information. As well as taking future contextual information into account is a core of BLSTM. So, this resulted in a considerable refinement in Arabic SA performance. To achieve overall higher accuracy, Skip Gram (SG) and Continuous Bag of Words (CBOW) word representation, types of Word2vec models; were also used and combined with BLSTM. The paper organization is as follows: Section II summarizes word2vec with DL models as related work. In Section III, the proposed architecture is demonstrated. Section IV displays the evaluation and experimental results. Section V finally, concludes the proposed architecture and addresses upcoming directions.

2. **Related Work**

SA classifies tweets or text as positive, negative, or neutral. A massive amount of data exists on social sites which need to be processed to acquire meanings from data which makes SA a pretty challenging mission. SA is implemented and applied using traditional ML methods. Some challenges already exist in traditional ml such as (A) Words can have dissimilar meanings if they exist in dissimilar domains [11]. (B) Preceding methods fail to deal with long sequences [12]. (c) Most attention is required for the existing data [13]. DL techniques are used in the architecture to meet challenges. DL-based neural networks have fulfilled great enhancement on Arabic SA. Generally, these models are composed of layers for projection which maps words of content to its comparing vectors. These vectors are combined with different NN to create a representation with a fixed length. They may isolate into diverse categories Based on the structure, among them CNN, repetitive neural systems, and other neural systems. We review the related work of SA in both categories (DL, ML). Khasawneh [14] shows that in research, the SA domain is still open in Arabic, and there are numerous procedures still not connected. Also, he described that Twitter is the most used data set as a source of such types of SA research. The highest accuracy achieved for tweets as a data corpus and ML algorithms or hybrid algorithms was 85.95%. Azmi [15] proposed a hybrid technique for Arabic tweet SA that combines ML techniques and semantic procedures. In their method, the lexical-based classifier is used to label the training data and the output is utilized to train the SVM model. Their results presented that the approach achieved an accuracy of 84.01% using the SA approach. El-Masri [16] showed that machine learning techniques are the most common techniques used in SA. These methods use classifiers to identify the labels of text automatically. This technique is used when the dataset is labelled. This study showed a significant increase in research in the Arabic SA field. Duwairi [17] presented different techniques for SA. Several ml techniques were used to classify Arabic data, such as Naïve Bayes (NB), as a probabilistic classifier using the probability of the term to predict class, run the NB classifier, and compare the results with other techniques, such as SVM. The study included MSA and colloquial Arabic, and the accuracy of the study shows the accuracy result: using the NB algorithm 66.20%; using the SVM algorithm 75.25%. Kim [18] accomplished impressive results in sentence classification with Convolutional Neural Systems. Kim set the base for how to prepare and demonstrate content by CNN in English. Bensalah [19] proposed the DL technique in the Arabic language which relied on the 1-D CNN with a maximum accuracy of 87.73 percent using FastText with two hundred dimensions. Zhou [20] proposed a sentence SA classification technique with CNN. Under this technique, it was clear that fine-tuning with a learning process-specific vector presents further benefits in performance in the SA classification process. Yin [21] proposed unsupervised pre-training and multi-channel embedding to improve the SA classification accuracy. Elshakankery [22] displayed RNN furthermore two diverse ML classifiers and the most elevated precision accomplished is break even with 85%. Alayba [23] used many ML classifiers in Arabic and also used CNN on the dataset. but the dataset is very small and this influences the performance of the classification. The used dataset is Arabic tweets related to health services. The mentioned dataset was gathered from Twitter and it contains Six Hundred Twenty-eight positive tweets, and one thousand three hundred and ninety-eighth negative tweets, to give Two Thousand Twenty-six tweets. Alwehaibi [24]presented LSTM with many pre-trained word representations. The used dataset is composed of tweets but with a small size. It is about 15k tweets. Smadi [25] Suggested employing two supervised ML approaches (SVM, RNN) to the dataset of Arabic hotel reviews. The outcomes point to SVM surpassing RNN in the whole task compared to RNN. For all that, the RNN was found to be better and faster by evaluating the time needed during learning training and testing. Tai [26] show that BLSTM has proven good outcomes in NLP for text processing and learning. The used dataset is SICK (Sentences Involving Compositional Knowledge) composed of nine thousand nine hundred twenty-seven sentences. Tai found that updating word embedding throughout the training process ("fine-tuning" word representation) returns a significant raise in performance.

3. The proposed Architecture

In this study for Arabic SA, a DL model which integrates word2vec word embedding with BLSTM architecture is used. BLSTM is the most powerful to deal with SA. LSTM is anticipated by Hoch Reiter and Schmid Huber in 1997. LSTMs are directed to ignore the long-term dependency issue [27]. LSTM stands for Long Short-Term Memory Networks are very valuable when the proposed neural network arrangement has to switch between recollecting later things and things from a long time prior. Since people don't start their consideration from zero each moment. They get it and based on their understanding of point-of-reference words, they treat each word. They don't transfer everything absent and begin considering from zero once more. Traditional neural systems can't perform that and it seems like a primary inadequacy exists. Repetitive neural systems address and face this issue since they are systems but with circles existing in them, letting data continue. Recurrent neural networks address this problem since they are networks with loops actually in them, letting the information continue and persist [27]. Since Recurrent Neural networks performed well on sequential input such as text than convolutional neural networks because they take into account the current input in addition to the previous input. However, it does not generally work well in long-term dependencies. This is due to vanishing and exploiting gradient issues during training [28].

Fig.1 appears the proposed architecture of the Arabic SA system using the BLSTM. The first stage of the proposed architecture is data collection, and the second stage is data preprocessing and then after preprocessing the dataset a word embedding matrix is generated relying on the word2vec (CBOW+SG) models. Word embedding sentences of the generated data by word embedding matrix are then going as an input feature to the BLSTM network. The BLSTM output is sent then to the existing fully-connected "softmax" layer to determine the polarity.

3.1. Data Collection

The LABR dataset which is explained in [29] was utilized. The mentioned dataset contains over 63.000 reviews and 342,199 numbers of sentences and 4134853 numbers of tokens. The dataset contains 3736 tokens per review with sixty-five average tokens per review. For word embedding (word2vec) we used more than one way, the first way is pre-trained word2vec. In the pre-trained word2vec learning model, ArWordVec [30] has been used which was created from 55M tweets. The second way, the word2vec was trained by us from scratch depending on over 4 million Arabic tweets which we crawled from Twitter. We didn't use any data related to the users themselves or their details. Since the most important for us is the public text, therefore we just stored the public text. The dataset obtained from different random geo locations was categorized into positive and negative sentiments considering the different dialects. Arabic users express their opinions using a dialect, Arabizi, while some of them use MSA. In Arabic, there are many different Arabic dialects (AD), such as Levantine, Egyptian, Cairene, Gulf, Moroccan, Tunisian, Algerian, and others. Therefore, most of the AD is covered

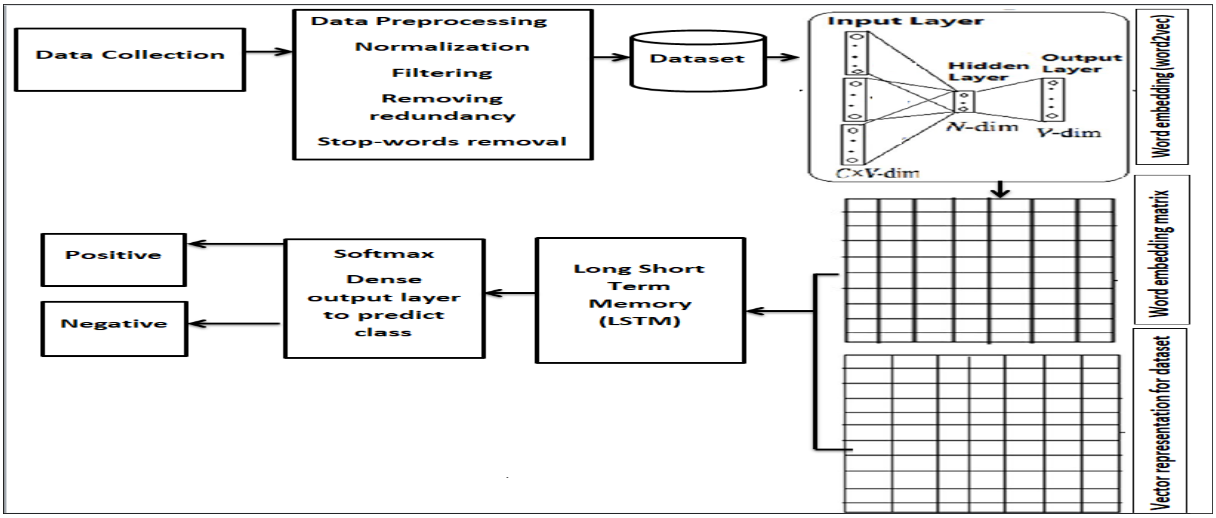


Fig. 1. DL Model with BLSTM Architecture.

3.2. Data Preprocessing

Before determining the polarity of reviews a diverse number of steps have been utilized in the stage of preprocessing: First more Arabic characters and Hamza, which own a lot of shapes, were reduced and normalized to only one shape. Then the additional useless data were filtered in the existing reviews, such as “#". Then the replication of characters was removed this was done to decrease the vocabulary size of the dictionary for the same word existing. For example: “ياااااااقيوم” converted to “يااقيوم”. Afterward, the frequent term was removed. Finally, the preprocessed reviews are split into tokens.

3.3. Word Embedding

Generally, there are two ways of representation: discrete and distributed representations. One-hot is the exemplary discrete representation which is a classical rule-based technique. One-hot represents words as vectors typically each word is expressed as a long vector that is the same in size such as a vocabulary dictionary. So, if there are “100000” words that exist in the dictionary, then the words in the sentence are represented with a size of “100000” dimensional vector. The vector obtained by one-hot contains 1 where the word is within the word reference, and the rest values are zeros. In this manner, the vector gotten by one-hot is binary 0 and 1. Therefore there are two issues, the first one is sparsity which means that most of the elements are zeros and the second one is high dimensionality which increases the space and time for computational processing. Therefore, it fails to provide the semantic relationship between words. To avoid the issue, this paper employs distributed embedding (word representation). In opposition to the other type, word representation obtains dense continuous vectors. As well as its ability to calculate the similarity of the words [31]. For a distributed word representation word2vec has been used. In the NLP mission, after the preprocessed input text was converted into numbers using the diversity of ways the classifiers were trained. The word2vec word embedding algorithm is used in this paper. Word2Vec is developed by Google to extract word representation Mikolov [31]. In word representation, pre-trained word2vec and trained word2vec (which was trained by us) have been used. ArWordVec [30] is applied in the pre-trained model which was created from 55M tweets. In word2vec which was trained by us, a large dataset of over 4 M tweets in Arabic has been used. SG and CBOW methods have been used. SG representation to predict the surrounding words and CBOW representation to predict the center word. The word2vec representation power is that it can train the big and large-scale datasets to yield small-dimensional dense word vectors representation. So, CBOW and SG are used both to represent the words in a small-dimensional space. Fig2. appears the word2vec model. Since the words are usually the tiny significant main element of a sentence, therefore the preprocessed data is tokenized first to split a chain of sentences into small parts like words. Then, SG and CBOW models of word2vec are used. These models for representing the appropriate numerical vector of each word can evaluate word similarity and also be implanted close together in vector space which is more effective for learning word representation computationally than One-hot-encoding vector representation.

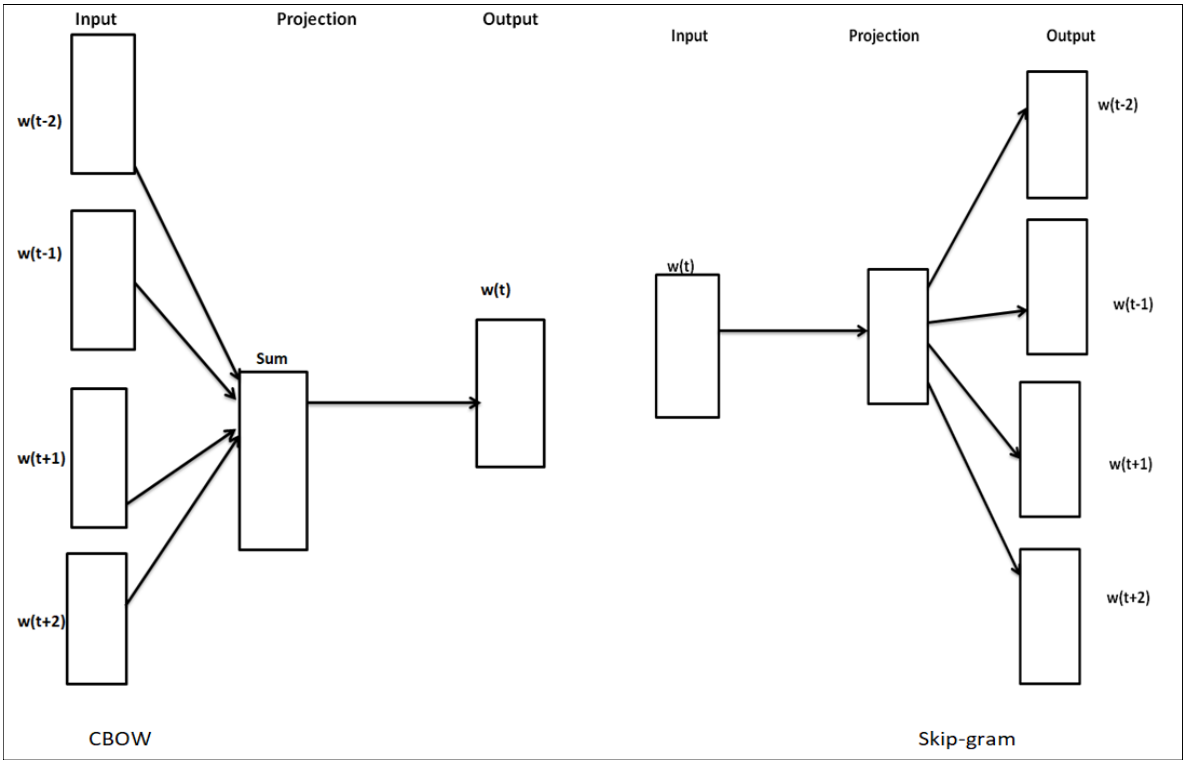


Fig. 2.Word2vec Model

3.4. LSTM and Bidirectional LSTM

LSTM become recently has a powerful among NLP analysts due to their eminent ability to learn and model from sequential data. LSTM seeks to solve the problem of RNN which is called exploding and gradient vanishing [27]. LSTM has a particular architecture to deal with long-term dependencies along with sequential data. RNN and LSTM networks present memory in the model. And the memory in the network is helpful because, when treated with sequential data like text, the common sense of the word relies on the context of the precedent text. A limitation of the RNN is that it is just capable of treating short-term dependencies. LSTM directs this issue by presenting a Long Term Memory in the network. Since the limitations of RNN are that it is just able to use the preceding context, while in natural language, the text predominating has a long-term dependency. For instance when a model is seeking to forecast the succeeding word in the following: “I live in Egypt...I speak fluent (...).”The succeeding word in this text should be "Arabic". The latest information recommends the succeeding word is likely the name of the language, but to realize which language, the earlier text has to be seen. LSTM networks address the mentioned issue of treating long-term dependencies after employing a word embedding layer that represents every word by its related vector which is trained by Word2Vec. The sequence of the words {T1, …, Tn} goes one by one and is fed to LSTM cells every Ti term is turned to its related Xi vector with Word2Vec and input one by one into LSTM. The cells will be trained on embedded words and output corresponding prediction words. Finally, the LSTM output is sent to the softmax layer to output the polarity of a review. Softmax is a function that is usually used in the latest layer of the NN. It takes random results to average into zero, one form. LSTM units are generally made up of the memory cell, input gate, output gate, and forget gate. The mentioned gates decide the information to flow in and out at the present step. The mentioned memory cell is for remembering over time the values. The other gates control the information flow from the cell and out of the cell. LSTM's decision to dump or keep the information using the forget gate. And this is a method of the preceding hidden state as well as the present input. The decision of which information to be kept inside the cell state is facilitated by the input gate. Then, the cell state is updated and relies on both the present-time step candidate value and the old state. A given time step candidate value is determined relies on both the preceding state and present time step input. While the preceding time step state yields an effect on the present state of the cell combines with the forget gate. This candidate value creates an effect on the state of the cell treated with the input gate. And finally, the output gate finds out which state part is related to the cell to be supplied as an output in the state given. When taking into account the gated approach with LSTM networks, gates through themselves can dominate gradients pass. Therefore, this makes a big effect on resolving exploding and vanishing gradient issues in the sequential data treatment. Mathematically, (input, forget, and output gates), represented by Equations. 1, 2, and 3 Where ct”, "ct", "h$t{-}1$", plus "ht" are the cell state candidate existing at the time stamp t, preceding hidden state input, succeeding hidden state input. One issue from the LSTM is related to not posting word information sufficiently adequately as the sentence that exists is read only in the forward direction. Obtaining the polarity highly relies on contextually related information for reviews. There is a lack to a certain degree in the ability to take contextually related information into account in the direct feedforward NN and therefore act more poorly in the Arabic SA.

$$I_t = \sigma([h_{t-1}, x_t] + b_i) \quad (1)$$

$$F_t = \sigma([h_{t-1}, x_t] + b_f) \quad (2)$$

$$O_t = \sigma([h_{t-1}, x_t] + b_o) \quad (3)$$

Such as “bi”, “bf” and “bo” for the biases related to (input, forget, output) gates. Hence, the Arabic SA Architecture here is to apply BLSTM Network with its capability of obtaining contextually related information by treating backward dependencies and forward-related dependencies simultaneously. Since it processes data in two directions. Furthermore, the proposed BLSTM let us see ahead by utilizing the forward LSTM. While forward LSTM operates the sequences in chronological order, while the backward LSTM, works and operates the sequences in reversed order.

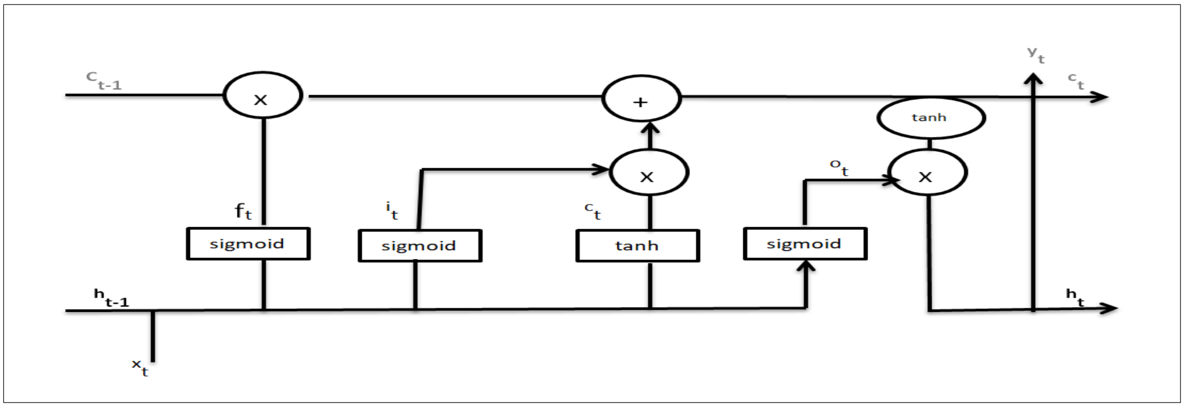


Fig.3. Represents the basic block of LSTM architecture

4. Experimental Results and Evaluation

This current section displays the evaluation of applying the proposed architecture which is BLSTM integrated with word representation (Word2Vec) models to determine the polarity of the presented review. Furthermore, ML, a deep CNN, and LSTM were used. The method of “most similar” in the genism model was applied to give the most _similar existing words which already exist. The mentioned method relies on cosine similarity as well as used to evaluate word2vec representation performance and how the model captures the similarities well. "Cosine similarity” is the similarity measure between two existing vectors which measures the similarity of the related angle among them. The cosine of zero is one, and it is smaller than one for another angle which lies in [0, 0.5π]. Thus, it is not a judgment of magnitude but a judgment of orientation. Therefore, two existing vectors with the same orientation have a similarity cosine of one; two perpendicular vectors have a similarity of zero; while the two opposite vectors have minus one similarity, independent of the existing magnitude. So, the unit is maximally "similar" in case they are already both parallel while maximally "dissimilar" in case they are perpendicular. As illustrated in Table two to Table seven, the Arabic SA using the BLSTM model surpasses the ml, deep CNN model, and LSTM. Therefore, integrating BLSTM with word representation models improves the Arabic SA classification performance. In addition, the outcomes with higher dimensions outperform lower dimensions. For the word embedding, we used word2vec because we achieved the highest score with word2vec, while the rest of the ways and methods does not achieve this score. We started with one hot encoding and the highest score achieved is 73 percent. Then we used Frequency-based embedding TF and TF-IDF and the highest score achieved using Frequency-based embedding is 80 percent. And then we used prediction-based embedding. In the prediction-based embedding, we used word2vec. The highest score achieved using word2vec is 94.88 percent. The results of the pre-trained word2vec using ArWordVec [30] are higher than the one trained by us so the mentioned results in the tables using the pre-trained word2ve.

Table 1. Hyper-Parameters Details

Parameter	Values
Optimizer	Adam
Epoch size	100
Batch size	32
Dropout	0.3
Loss function	Binary cross-entropy

Table 2 shows the deep CNN model accuracy for using CBOW word embedding. We started from the baseline we started firstly with machine learning. Different ML classifiers were used for learning the model. Seven classifiers were used Random Forest, Support Vector Machine, Logistic Regression, k-nearest neighbors, Decision Tree, and Xgboost. The highest score achieved using machine learning was 83.22 percent using Xgboost with CBOW word embedding. The results demonstrate that the highest performance was achieved for Arabic SA with deep CNN plus CBOW using the highest dimensions. The highest accuracy achieved is 87.97 percent with the CNN + CBOW model therefore the outcomes say that deep CNN with CBOW word embedding surpasses the machine learning with CBOW with 4.75 percent.

Table 2. Deep CNN Model with (CBOW)

Dimension	Word2vec(CBOW)
100	85.20
150	86.01
200	86.40
250	86.89
300	87.09
350	87.22
400	87.60
450	87.69
500	87.97

Table 3 displays the deep CNN model accuracy for using SG word embedding. Before working with deep CNN integrated with SG word embedding. Diverse ML classifiers were used for learning the model. The highest score achieved using machine learning was 82.89

percent using Xgboost with SG word embedding. In Table 3, the results reveal that the performance for Arabic SA increases in proportion to the dimension size. The highest accuracy achieved is 88.73 percent using deep CNN with the SG model. Therefore the outcomes say that deep CNN with SG word embedding surpasses machine learning with SG by 5.84 percent. Therefore the outcomes say that both CBOW and SG word embedding with deep CNN surpasses CBOW and SG with machine learning. As well as SG surpasses CBOW in deep CNN while CBOW surpasses SG in machine learning with 0.33 percent.

Table 3. Deep CNN Model with (SG)

Dimension	Word2vec(SG)
100	84.39
150	85.20
200	85.61
250	85.98
300	86.01
350	87.24
400	87.75
450	88.11
500	88.73

Table 4 addresses the LSTM model accuracy for using CBOW word embedding. In Table 4, the results demonstrate that the lowest performance for Arabic SA with LSTM using the CBOW model is 91.50 percent while the highest performance is 93.74 percent. Therefore CBOW word embedding with the LSTM model surpasses both CBOW word embedding in both deep CNN and machine learning. CBOW with the LSTM model surpasses CBOW with machine learning by 10.52 percent. While CBOW with the LSTM model surpasses CBOW with deep CNN by 5.77 percent.

Table 4. LSTM Model with (CBOW)

Dimension	Word2vec(CBOW)
100	91.50
150	91.95
200	92.68
250	93.01
300	93.19
350	93.50
400	93.59
450	93.61
500	93.74

Table 5 displays the LSTM model accuracy using SG word embedding. In Table 5, the results demonstrate that the highest performance for Arabic SA with LSTM using the SG model is 93.86%. Therefore SG word embedding with the LSTM model surpasses SG word embedding in both deep CNN and machine learning. SG with the LSTM model surpasses SG with machine learning by 10.97 percent. While SG with the LSTM model surpasses SG with deep CNN by 5.13 percent. The outcomes say that LSTM surpasses machine learning by 10.64 percent. As well as LSTM surpasses deep CNN with 5.13 percent using CBOW and SG. Therefore LSTM surpasses both deep CNN and machine learning using both CBOW and SG word embedding.

Table 5. LSTM model with (SG).

Dimension	Word2vec(SG)
100	90.15
150	90.17
200	92.88
250	93.64
300	93.18
350	93.52
400	93.70
450	93.82
500	93.86

Table 6 displays the BLSTM model accuracy using CBOW word embedding. In Table 6, the results demonstrate that the lowest performance for Arabic SA with BLSTM using the CBOW model is 91.80% while the highest performance is 94.09%. The outcomes show that CBOW with the BLSTM model surpasses CBOW with machine learning by 10.87 percent. CBOW with the BLSTM model surpasses CBOW with deep CNN by 6.12 percent. CBOW with the BLSTM model surpasses CBOW with LSTM by 0.35 percent. Therefore CBOW word embedding with the BLSTM model surpasses CBOW word embedding in machine learning, deep CNN, and LSTM.

Table 6. BLSTM model with (CBOW).

Dimension	Word2vec(CBOW)
100	91.80
150	91.99
200	92.70
250	92.88
300	93.20
350	93.70
400	93.72
450	93.97
500	94.09

Table 7 shows the BLSTM model accuracy using SG word embedding. In Table 7, the results demonstrate that the highest performance for Arabic SA with BLSTM using the SG model is 94.88%. The outcomes show that SG with the BLSTM model surpasses SG with machine learning by 11.99 percent. SG with the BLSTM model surpasses SG with deep CNN by 6.15 percent. SG with the BLSTM model surpasses SG with LSTM by 1.02 percent. Therefore SG word embedding with the BLSTM model surpasses SG word embedding in machine learning, deep CNN, and LSTM. Therefore BLSTM surpasses machine learning, deep CNN, and LSTM using both CBOW and SG word embedding.

Table 7. BLSTM model with (SG).

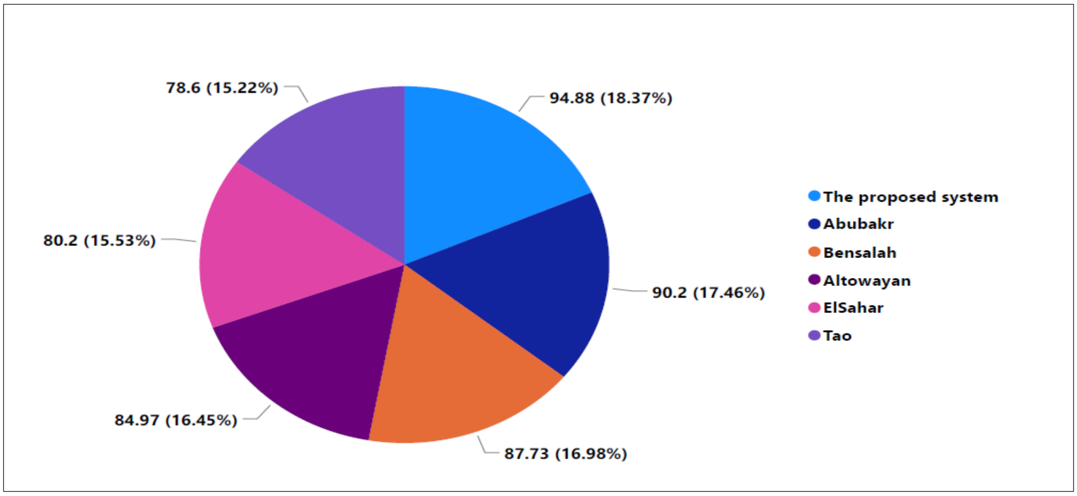
Dimension	Word2vec(SG)
100	91.65
150	91.97
200	92.13
250	92.37
300	92.89
350	93.41
400	93.95
450	94.54
500	94.88

As shown Skip-gram outperforms a continuous bag of words in:

- CNN, LSTM, and BLSTM models.
- Deep CNN with 0.76%
- LSTM with 0.12%
- BLSTM with 0.79%.

Also

- BLSTM outperforms deep CNN with ٦.١٥ %.
- BLSTM outperforms LSTM by 1.02%.



.Fig. 4. Performance comparison between the proposed architecture with other studies

Fig.4 shows that accuracy obtained compared to the obtained accuracies in Bensalah [19], Abubakr [32], ElSahar [33], Altowayan [34], and Tao [35] using LABR data. Therefore the proposed architecture beats different works and refines the accuracy obtained with 7.15%, 4.68%, 14.68%, 9.91%, and 16.28% respectively this is owing to more than one aspect: firstly, using BLSTM for learning sentence semantics and storing contextual information while considering the future contextual information. Secondly, by joining the BLSTM with word representation utilizing word2vec the accuracy for SA classification moved forward with word representation to capture morphological data as well as the syntactic and semantic data related to the existing words within the data.

5. Conclusion and future works

Through this research paper, integration between BLSTM and word representation like Word2Vec has been presented to discover the polarity of the reviews. Besides, ml, a deep CNN model and LSTM have been applied. The outcomes specify that BLSTM Model outperforms CNN and LSTM Architectures. The findings specify that the performance improved concerning integrating the extracted features, like Word2Vec, with the classifiers. Therefore, the performance is improved for classification. In addition, the outcomes specify that the vectors with high dimensions have good performance more than vectors with low dimensions. Farther more, SG outperforms CBOW in BLSTM, LSTM, and CNN. As a future heading, we look forward to utilizing Glove models for Arabic SA classification.

References

[1] M. Taboada, “Sentiment Analysis: An Overview from Linguistics,” Annu. Rev. Linguist., vol. 2, no. 1, pp. 325–347, 2016.

[2] A. Rajput, “ Natural language processing, sentiment analysis, and clinical analytics,” in Innovation in Health Informatics, 2020.

[3] S. T. Bharathi and S. Charles, “Classification Levels , Approaches , Tools , Application and Challenges in Sentimental Analysis- A Survey,” International Journal of Computer Sciences and Engineering., vol. 6, no. 11, pp. 210–215, 2018.

[4] A. Marshan, G. Kansouzidou, and A. Ioannou, “Sentiment Analysis to Support Marketing Decision Making Process: A Hybrid Model,” in Proceedings of the Future Technologies Conference, vol. 1289, no., pp. 614–626, 2020.

[5] O. Oueslati, E. Cambria, M. Ben HajHmida, and H. Ounelli, “A review of sentiment analysis research in Arabic language,” Future Generation Computer Systems, vol. 112, pp. 408–430, 2020.

[6] World statistics. Accessed: June. 28,2021 [online]. Available:https://www.internetworldstats.com/.

[7] I. H. Sarker, “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions,” SN Computer Science, vol. 2, no. 6, pp. 1–20, 2021.

[8] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. E. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: A survey,” Heliyon, vol. 4, no. 11, pp. e00938, 2018.

[9] B. B. Benuwa, Y. Zhan, B. Ghansah, D. K. Wornyo, and F. B. Kataka, “A review of deep machine learning,” International Journal of Engineering Research in Africa, vol. 24, pp. 124–136, 2016.

[10] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” Electronic Markets, vol. 31, no. 3, pp. 685–695, 2021.

[11] S. Rani and P. Kumar, “A Sentiment Analysis System to Improve Teaching and Learning,” Computer (Long. Beach. Calif), vol. 50, no. 5, pp. 36–43, 2017.

[12] J. R. A. Moniz and D. Krueger, “Nested LSTMs,” J. Mach. Learn. Res., vol. 77, pp. 530–544, 2017.

[13] X. Fu, J. Yang, J. Li, M. Fang, and H. Wang, “Lexicon-Enhanced LSTM with Attention for General Sentiment Analysis,” IEEE Access, vol. 6, pp. 71884–71891, 2018.

[14] R. T. Khasawneh, H. A. Wahsheh, I. M. Alsmadi, and M. N. Ai-Kabi, “Arabic sentiment polarity identification using a hybrid approach,” in 2015 6th International Conference on Information and Communication Systems (ICICS), pp. 148–153, 2015.

[15] H. K. Aldayel and A. M. Azmi, “Arabic tweets sentiment analysis - A hybrid scheme,” Journal of Information Science, vol. 42, no. 6,pp. 782–797, 2016.

[16] M. El-Masri, N. Altrabsheh, and H. Mansour, “Successes and challenges of Arabic sentiment analysis research: a literature review,” Social Network Analysis and Mining, vol. 7, no. 1, pp. 1–22, 2017.

[17] R. M. Duwairi and I. Qarqaz, "Arabic sentiment analysis using supervised classification,"in 2014 International Conference on Future
Internet of Things and Cloud, pp. 579–583, 2014.

[18] Y. Kim, "Convolutional neural networks for sentence classification," EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf., pp. 1746–1751, 2014.

[19] B. Nouhaila, A. Habib, A. Abdellah, and I. El Farouk Abdelhamid, "Arabic Sentiment Analysis Based on 1-D Convolutional Neural Network,"in The Proceedings of the Third International Conference on Smart City Applications, vol. 183, pp. 44–55, 2021.

[20] A. Dahou, S. Xiong, J. Zhou, M. H. Haddoud, and P. Duan, "Word embeddings and convolutional neural network for Arabic sentiment classification," COLING 2016 - 26th Int. Conf. Comput. Linguist. Proc. COLING 2016 Tech. Pap., pp. 2418–2427, 2016.

[21] W. Yin," Multichannel variable-size convolution for sentence classification," arXiv preprint arXiv:1603.04513,2016.

[22] K. Elshakankery and M. F. Ahmed, "HILATSA: A hybrid Incremental learning approach for Arabic tweets sentiment analysis," Egyptian Informatics Journal, vol. 20, no. 3, pp. 163–171, 2019.

[23] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Improving Sentiment Analysis in Arabic Using Word Representation," in 2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pp. 13–18, 2018.

[24] A. Alwehaibi and K. Roy," Comparison of Pre-Trained Word Vectors for Arabic Text Classification Using Deep Learning Approach," in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp.1471-1474,2018.

[25] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, and B. Gupta, "Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews," Journal of computational science, vol. 27, pp. 386–393, 2018.

[26] K. S. Tai, R. Socher, and C. D. Manning, "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks," ACL-IJCNLP 2015 - 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. Asian Fed. Nat. Lang. Process. Proc. Conf., vol. 1, pp. 1556–1566, Feb. 2015.

[27] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4470–4474, 2015.

[29] M. Aly and A. Atiya, "LABR: A large scale arabic book reviews dataset," in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, vol. 2, pp. 494–498, 2013.

[30] M. M. Fouad, A. Mahany, N. Aljohani, R. A. Abbasi, and S. U. Hassan, "ArWordVec: efficient word embedding models for Arabic tweets," Soft Computing, vol. 24, no. 11, pp. 8061–8068, 2020.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," 1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc., Jan. 2013.

[32] 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.

[33] H. ElSahar and S. R. El-Beltagy, "Building large arabic multi-domain resources for sentiment analysis," in International conference on intelligent text processing and computational linguistics, vol. 9042, pp. 23–34, 2015.

[34] A. Aziz Altowayan and A. Elnagar, "Improving Arabic sentiment analysis with sentiment-specific embeddings," in 2017 IEEE international conference on big data (big data), pp. 4314–4320, 2017.

[35] A. A. Altowayan and L. Tao, "Word embeddings for Arabic sentiment analysis," in 2016 IEEE International Conference on Big Data (Big Data), pp. 3820–3825, 2016.