I.J. Intelligent Systems and Applications, **2020, 4, 25-36**
Published Online August 2020 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2020.04.03

Arabic Opinion Mining Using Combined CNN - LSTM Models

Hossam Elzayady
Military technical college; /Department of computer engineering, Cairo, Egypt
E-mail: hossamelzaiade@gmail.com

Khaled M. Badran
Military technical college; /Department of computer engineering, Cairo, Egypt
E-mail: khaledbadran@mtc.edu.eg

Gouda I. Salama
Military technical college; /Department of computer engineering, Cairo, Egypt
E-mail: gisalama@mtc.edu.eg

Received: 08 March 2020; Revised: 01 May 2020; Accepted: 14 June 2020; Published: 08 August 2020

Abstract: In the last few years, Sentiment Analysis regarding customers' reviews in order to comprehend the opinion polarity on social media has received considerable attention. However, the improvement of deep learning for sentiment analysis relating to customer reviews in Arabic language has received less attention. In fact, many users post and jot down their reviews in Arabic daily, so we ought to shed more light on Arabic sentiment analysis. Most likely all previous work depends on conventional classification techniques, such as KNN, Naïve Bayes (NB), etc. But in this work, we implement two deep learning models: Long Short Term Memory (LSTM) and Convolution Neural Networks (CNN), in addition to three traditional techniques: Naïve Bayes, K-Nearest Neighbor (KNN), Decision trees for sentiment analysis and compared the experimental results. Also, we offer a combined model from CNN and Recurrent Neural Network (RNN) architecture where this model collects local features through CNN as the input for RNN for Arabic sentiment analysis of short texts. An appropriate data preparation has been conducted for each utilized dataset. Our Conducted experiments for each dataset against traditional machine learning classifier; KNN, NB, and decision trees and regular deep learning models; CNN and LSTM, has resulted in impressive performance using our proposed combined (CNN-LSTM) model with an average accuracy of 85,83%, 86,88% for HTL and LABR datasets respectively.

Index Terms: Sentiment Analysis, Deep Learning, Recurrent Neural Network, LSTM, Convolutional Neural Network.

1. Introduction

Social media is considered a crucial tool in communication and widely used since the existence of the internet. People nowadays communicate most of the time using different applications of social media. It is the easiest way to exchange recent news and points of view. Because social media is being used daily, so there is a huge volume of reviews, feedbacks, and articles. Today social media is used by many firms and agencies to reach out to their clients. Companies' main concern is to get feedback about whatever services they provide whether it is positive or negative; this comes under the umbrella that is called "sentiment analysis" [1, 10, 24]. But most of the sentiment analysis work usually made for English written data, while the number of researches carried out for Arabic data is considered to be fairly few. Arabic resources, which emphasize on mining and analyzing views and sentiments, are so hard to find [24]. The complexity of Arabic language is probably the answer for not using it in a wide range as one word can have various forms utilizing various suffixes, affixes, and prefixes. Several words with whole various meanings can be created using the same three-letter root [2, 3, 11]. But still, Arabic posts are growing rapidly in various fields and extremely increasing daily, so Arabic analysis is a must [5]. In general, many of approaches suggested for sentiment analysis, utilize conventional machine learning methods that depend mainly on feature engineering [4], that's why these features have a vital role in classification performance [6]. As a result, most techniques aim for the right features to gain the best performance [6]. Apparently, most machine learning algorithms utilize fixed-length feature vectors, documents ought to be described as fixed-length feature vectors [7]. Bag-of-words is considered one of the famous methods which utilize to represent each document, that is mainly because of its effectiveness and simplicity but, no consideration to word order [7]. Methods like this may cause misunderstanding in sentiment classification, this is mainly due to the possibility for same-words phrases to refer to various opinions [9]. The identical meaning of different words is not well-clarified by

the representation of Bag-of-words [9]. Another common method is N-gram to represent a sentence. This manner is considered better than the others [13]. Initially, Words are projected to a high-dimensional space. Then the classifier input can be represented by fixed-size of input sentence represention. Actually, it can be said that the word ordering in short sentences is taken into consideration by n-gram models, but they still face the problem of data sparsity. Recently, Deep Learning is the best machine learning algorithms to analyze extraordinary data of all types [12]. Deep learning is superior to traditional machine learning because Deep learning does not need "Feature Engineering" as feature extraction is embedded in the deep learning algorithms, where features are extracted in a fully automated manner and without any intervention of a human expert. Deep learning can fix complex issues and process several layers in order to perform features extraction operations from raw data. Additionally, it also reveals the hierarchical representations required through different tasks. Deep learning has obtained a remarkable improvement in different Artificial Intelligence subjects, The most famous of these subjects were speech recognition and image recognition and lately, It has been extensively used in natural language processing tasks like sentiment analysis [5]. In this research, we aim to combine two major deep learning models (CNN, LSTM) for setting sentiment analysis methodology on customers' reviews in Arabic text. We investigate the models utilized in [44]. We have taken the same suggested approach in [44], in addition to the following:

- Due to the complexity of the Arabic language, some steps of data preparation are conducted as the models in [44] are implemented for English data.
- Different hyper-parameters are utilized for getting high accuracy, also some training steps are obviously discussed.
- Three traditional machine learning techniques based on N-gram method for feature extraction are utilized and compared results with deep learning models. Results indicated that combined CNN-LSTM achieves impressive accuracy than all used models.

The rest of this research is organized as follows, Section 2 describes the related works, while in Section 3&4 deep learning and machine learning models used in our suggested system are discussed. Section 5 shows the steps of training. Section 6 shows the experimental results and discussion. Finally, conclusion and plan of futute work are offered in Section 7.

2. Related Work

Sentiment Analysis attracted many people to set researches about it. That was mainly because of the gradual increasing of Social Networks' data of people sharing their ideas thoughts, point of view, comments and daily life [41]. Walaa Medhat et al. [10] detailed explained various applications of sentiment analysis, several algorithms and techniques of SA and their originating references were clarified and categorized.

Yang et al. [12] introduced the most popular methods of sentiment analysis that are used from the perspective of machine learning technologies, including Artificial Neural Network (ANN) method, NB technique, Support Vector Machine (SVM) technique, Maximum Entropy method and performance assessment and obstacles. Pang et al. [8] were considered the first to extract sentiment from movie reviews by applying machine learning, For getting features, a bag of words and unigram are employed when several classifications algorithms are carried out. according to the applied classifiers, a various ratio of accuracy is obtained, for example, SVM achieved 82.9% of accuracy, while accuracy was 78.7% by utilizing NB classifier. El-Beltagy et al. [39] inspected an ML-based sentiment analysis model combining several features, most of them were extracted utilizing an Arabic sentiment lexicon. Various steps were kept into consideration; text's length, the number of segments and emoticons. Accuracy increasing was highly-noticed along with six of seven datasets. This system was applied to Saudi, Egyptian, Levantine, and MSA social media datasets. They used these datasets to implement a new model that excels all current models for Arabic sentiment analysis.

One of the machine learning models derived from human brain is neural network, which is several neurons forming a wide network. Bengio et al. in 2003 utilized their proposed approach for language modeling purposes, which relying on neural network and achieved outstanding performance than the state-of-the-art n-grams model [27]. One of the major merits of neural networks is their elasticity during its design, it has many layers, each layer may have different numbers of the nodes. The network can learn more complicated models, the more layers it has. When there are several hidden layers, it is known as Deep Learning. However, simple feed forward neural network still cannot capture any advantage by gathering more layers as its training process is helpless [28]. Bengio et al. offered in 2007 an unsupervised pre-training process holding the name of auto-encoders, as it explains a process of encoding large features to smaller features. It was easily proved that the model without pre-training weights has less performance than the model with unsupervised pre-training weights [29].

Recurrent neural network (RNN) is one of the basic models of deep learning, Mikolov et al. tried in 2010 to use RNN technique on speech recognition [30]. They prove that RNN excels n-gram method. RNN has many qualities in language modeling by utilizing the prior state to calculate its present state that shows the exact context in almost a lot of natural languages. Still, simple RNN faced some difficulties in the cases of delivering the information through a long

sequence. In 1977, it was suggested to solve this problem by using LSTM; which is an RNN with an additional memory [31]. Wang et al. in 2015 introduced LSTM with Trainable Lookup-Table (LSTM-TLT). They used a substitution of stable look-up table by trainable lookup-table. The substituted look-up table also pre-trained by word2vec (Mikolov et al., 2013 [32]). LSTM-TLT reveals excellent performance compared to the latest technology in Twitter sentiment analysis.

Convolution Neural Network (CNN) is another kind of deep learning technique that was proposed by LeCun et al. in 1998 for the purpose of identifying documents categorization [34]. CNN is formed by different layers to execute various functions. The major one is called convolution layer. It is utilized to extract features from group of neighbor inputs. In 2012 the utilizing of CNN in the issues relating to images recognition became popular and achieved better performance than other methods [33]. CNN models for NLP showed outstanding effects in sentence modeling [36], semantic parsing [35], search query retrieval [37], and other NLP issues [38]. Lately, the DNN-based model has offered perfect scores for many tasks in NLP [42, 43]. Although these models conducted in a good way, they are still late at testing and training, that ought to prevent these models from using a massive amount of data, needing stacking a lot of convolutional layers for capturing long-term dependencies.

Joint CNN-LSTM is introduced in [44] to implement english sentiment analysis on Twitter data and compare their accuracy against common LSTM and CNN deep learning models.

There are a few interesting research papers about Arabic sentiment analysis. In 2015, Al Sallab et al. suggested a deep learning approach for Arabic text classification and utilized different structures inspired by DNN, DBN and Deep Auto-Encoders [46]. In 2017, Alomari et al. proposed various supervised machine learning for Arabic sentiment analysis using different features and preprocessing strategies [11]. In 2018, Abdelhade el al. investigated deep learning models and traditional machine learning models for Arabic tweets [5].

3. Deep Learning Models

3.1. Long Short Term Memory

LSTM is defined as an RNN with an additional cell of internal memory. It allows searching for both long and short patterns in data as well as the issue of vanishing gradient is discarded by training RNN [15]. The main feature from using LSTM is to "remember" prior values for any amount of time. The architecture of LSTM is portrayed in Fig.1, a common LSTM unit is composed of cells, each cell includes the following components, an input gate, a forget gate and an output gate. The flow of information is controlled by each gate mentioned above. In the Initially stage, the information which ought to be ignored from the cell state is detected by the forget gates. The second stage, the new information which is being kept within the cell state is defined by input gates. The third stage, old cell state is updated by utilizing the prior input gates and forget gates information to compute the new value of the cell state. Ultimately, output value is specified by the output gates, which depend on the state of the cell [14,16]. The gates are computed as:

$$G_i^t = \sigma\,(w_i\ x^t + U_i h^{t-1} + b_i) \tag{1}$$

$$G_f^t = \sigma\,(w_f\ x^t + U_f h^{t-1} + b_f) \tag{2}$$

$$G_o^t = \sigma\,(w_o\ x^t + U_o h^{t-1} + b_o) \tag{3}$$

Where the weight matrix of each gate is represented by both *U* and *W* while bias is described by *b*. As the following Subscript *i, f,* and *o* refer to the variables for the three gates, input, forget, and output. Sigmoid function is represented by σ. G^t refers to the gate at time *t*, the input at time *t* is specified by x^t , whereas hidden activation is referred by h^{t-1} at time *t-1*. The cell state *C* at time t can be computed as follows:

$$C^t = G_f^t \times C^{t-1} + G_i^t \times tanh(W_C\ x^t + U_C\ h^{t-1} + b_C) \tag{4}$$

The variable for cell state is represented by *C*. *tanh* is hyperbolic tangent function. From (4), it could be observed that C^t is an output of including the prior cell state C^{t-1} with the present input x^t by a proportion of gate value. We can compute the hidden activation h^t as follows:

$$h^t = G_o^t \times tanh(C^t) \tag{5}$$

h^t will also illustrate the network output after completing the last input of the sequence, as depicted in Fig.2. LSTM can be enlarged via time. At each time step, the word vector sequence is utilized. The output and cell state of prior LSTM are employed in present LSTM. Eventually, the extracted output from the last cell is inserted into a totally connected layer with softmax classification. Backpropagation is utilized for training this network.

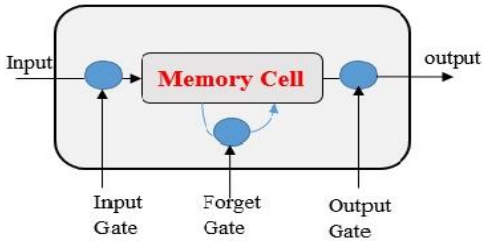

Fig.1. LSTM cell [15].

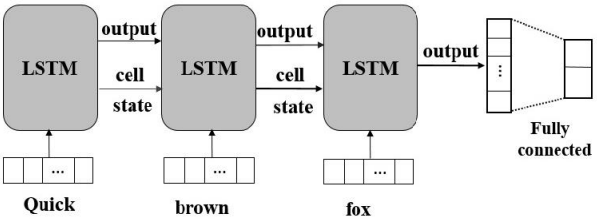

Fig.2. LSTM with a totally connected layer [15].

3.2. Convolutional neural network

CNNs are popularly used in computer vision, but CNNs also have recently been emerged into several NLP tasks, The CNN design which is utilized to texts categorization according to their sentiment is demonstrated in Fig.3. We can notice that input documents can be displayed in a matrix form, every row of this matrix matches to only one token. Commonly, a token may be treated as a word. Thereafter a token can be represented by a vector in every row. taking into account a document containing the maximum number of N tokens (Padding methodology is applied for documents less than N tokens) and each token is clarified with a d dimension vector, the document will be clarified using a matrix $A \in \mathbb{R}^{N \times D}$. Subsequently, a convolution process could be executed on a window of h tokens through linear filters. As a result of padding, all documents matrix rows are equal in length and every row represent a token, it is conceivable to utilize filters in which their widths are similar to the dimensionality of the word vectors [9, 21, 22].

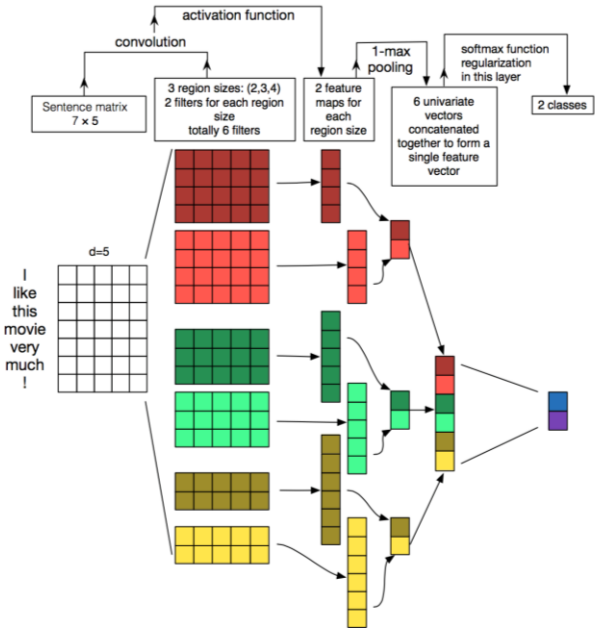

Fig.3. CNN architecture for document classification [9].

Thence, it is necessary to utilize filter $w \in \mathbb{R}^{h \times d}$ in order to execute the convolution process on h words. Permit A [*i*:*j*] is a window of tokens from token *i* to token *j* of the document matrix A [9, 21, 22]. By relying on a token window of size h, it is possible to compute a feature c_i as follows:

$$c_i = f(w \cdot A\,[i{:}j + h - 1] + b) \tag{6}$$

Where $i = 1 \ldots N - h + 1$, $w \in \mathbb{R}^{h \times d}$ indicates the filter weight matrix, f refers to the activation function and · refers

to the dot product, b ∈ R indicates the bias term. A feature map is formed after applying filter w to each possible word window and putting computed features together in a vector as follows:

$$C = [\, c_1 , c_2 , \ldots . , c_{N-h+1}] \tag{7}$$

Where $c \in \mathbb{R}^{N-h+1}$. The size of the documents and the filter applied to it determine the feature map's size. We are in a dire need for pooling process to get a fixed length vector. And in this system, a max pool is being applied to obtain the biggest value from each feature map that conforms to the specific filter applied. Moreover, many filters with different sizes can be used and applied here and in this case, one feature value is going to be estimated for each filter. So we can adhere to this pattern as a solution for the documents with variable length, as each document has a set number of features values commensurate with the number of filters used in convolution step. Then we come to the next step in which the feature values will be moved to fully connected softmax layer whose results are a probability distribution over labels [9,21, 22]. For regularization, the penultimate layer dropout is being used as an efficient manner in which proportion p of the hidden units is set to be zero in training.

3.3. CNN-LSTM Model

The structure of CNN-LSTM model is depicted in Fig. 4, so the combination comprises of a beginning convolution layer which is able to get word embeddings as an input of CNN model in which windows of various length and different weight matrices are used to create a number of feature maps. Its output will then be pooled to a smaller dimension which is then fed into an LSTM layer. The main objective behind this gathering model (CNN-LSTM) is that local features will be extracted with the aid of convolution layer after which LSTM layer has the capability of using the ordering of said features to learn about the input's text ordering [44].

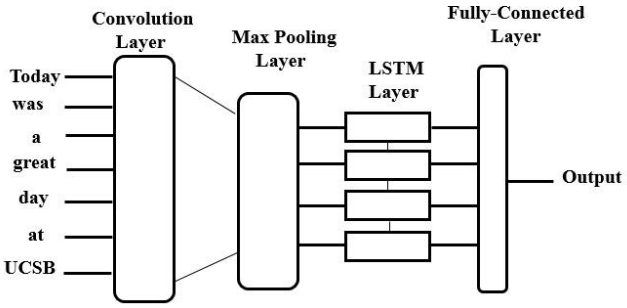

Fig.4. CNN-LSTM Model [44].

4. Machine Learning Models

4.1. Naïve Bayes

Naïve Bayes is considered one of the algorithms which has simplicity, it is generally utilized in text classification with TF-IDF features, among other applications. The main idea that NB relies on Bayes' theorem, where each pair of features is supposed to be classified separately and independently from each other [10]. For instance, there might be a plausibility that the fruit is an apple in case of, it is circular, yellow, and almost 3 inches in diameter. Notwithstanding these features may rely on one another, these characteristics may share in that this fruit is orange and hence the reputation " Naïve" [25]. The theory of Bayes is the basis on which The algorithm was implemented, which is described as follows:

$$P(A/B) = [P(B/A)\ P(A)]/P(B) \tag{8}$$

4.2. Decision Tree

One of the most prevalent techniques for machine learning is decision tree, the attraction of Decision trees return to its simplicity of inspecting models and its ability to deal with both categorical and continuous features. The implementation of a decision tree classifier mainly relies on tree structure. Class labels in this archichture are presented by leaves, while branches refer to the conjunction of features that result in the aforementioned classes. Actually, it executes a recursive binary apportioning of the feature space. Every phase is chosen covetously, pointing at the finest perfect choice particular stage, by a progressive increment in information gain [5,10].

4.3. K-Nearest Neighbor

K nearest neighbor or (KNN) Algorithm is a simple algorithm; KNN is a non-parametric lazy learning algorithm. It

means that it does not make any assumptions on the underlying data distribution. It utilized for classification and regression in both processes, the input, comprises of the k training examples in the feature space. The output is a class membership. The way it works lies in classifying the neighbors' votes, through categorizing the object to the most popular class between its k nearest neighbors. If k = 1, that means the object holds a strong bond relating to that individual nearest neighbor's class [23]. If we got similar scores of each nearest neighbor text, comparing to the test text, we shall use the classes' weight in the neighbor's document. The weighted sum in KNN classification could be calculated from [5]:

$$\text{Score}\ (d,s) = \sum_{d_j \in knn(d)} \text{sim}\ (d, d_j)\delta(d_j, s) \tag{9}$$

Where *knn(d)* is the set of k nearest neighbors of document d. If d_j belongs to sentiment *s*, $d\ (d_j, s)$ equal to 1, or otherwise 0. Document d belongs to the sentiment *s* that has the highest score.

5. Proposed Solution

Fig. 5 illustrates the whole design of the proposed solution, taking into account the aspect discussed earlier.

5.1. Sentiment analysis Data stets

- **Hotel Reviews (HTL):** 15K Arabic reviews were collected for the hotels from [40]. 13K users have carried out those reviews for 8100 Hotels [17].
- **Book Reviews (LABR):** The book reviews were gathered by [20] from a society driven [19], this dataset has more than 16448 book reviews in Arabic reviews [18].

5.2. Data Preprocessing

Data preparation is considered an essential stage during analysis of data and it aims to neglect any needless data, which are supposed unnecessary. Preprocessing includes:

- Eliminating all punctuation mark from the whole context like (., : "; ').
- Eliminating Stop Words, which are words' group that has nothing to do with changing the text's meaning, like prepositions. A list of 179 Arabic stop words is utilized.
- Eliminating non-Arabic numbers, letters, single Arabic letters, particular symbols ($, , &, |, _, -, …).
- Shortening some of the letters that repeated more than once, in single letters (نهياااااااااااااار) to انهيار.

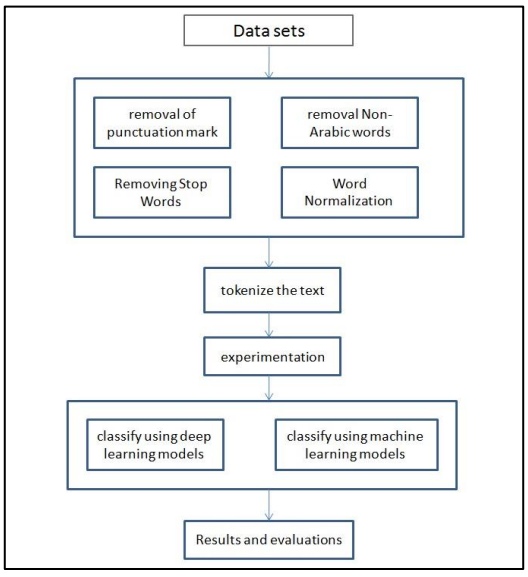

Fig.5. Proposed system architecture.

5.3. Word Embedding

Word representation can be another term to word embedding. The starting point of word embedding could be dated back to 1986, known as distributed representation suggested by Hinton. word Embedding has the ability to convert words into a real vector of low dimensions and gives us permission to explore and find any similarity of words by using cosine method [26]. For instance, the representation for "vehicle" ought to be more likely to "truck," than, say, "Fish". This is the concept that word embedding proves. We decided to represent the words in our dataset as 32-dimensional arrays. We

employed Keras' Word Embedding methods to build these vectors.

5.4. RMSProp Optimizer

Throughout our conducted experiments, we utilize the RMSprop optimizer. It is considered a very popular optimizer for recurrent neural networks' training and is a version of Resilient Propagation (rprop) for mini-batch learning. The main theory that RMSProp mostly depends on, is partitioning a gradient by a running average of its recent magnitude. First we compute the running average r_t given by (10).

$$r_t = (1 - \gamma) f'(\theta_t)^2 + \gamma\, r_{t-1} \tag{10}$$

γ Indicates the decay term and $f'(\theta_t)$ refers to the derivative of the loss function considering parameters θ_t at time step t. Then the root mean square of the running average is used for splitting the learning rate, the updated rule is calculated by the following equation [16]:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{r_t + \varepsilon}} f'(\theta_t) \tag{11}$$

Where $\boldsymbol{\alpha}$ refers to learning rate and $\boldsymbol{\varepsilon}$ is the fuzz factor.

6. Experiment

The architecture of the models is built by Keras library. Numpy, pandas, csv and re libraries are used for text preprocessing as text reading and text tokenization. Keras is high-level neural networks that support Tensorflow and Theno as backend. Furthermore, Keras is a deep learning tool that simple to recognize and permit us to utilize GPU to accelerate the training and testing process.

The training and testing set is selected using Sklearn library to split the dataset into training data and testing data randomly according to the specific training rate. We compute the results according to the accuracy value which is computed as:

$$Accuracy = (TP + TN)/(TP+TN+FP+FN)$$

Where:

- TP (True Positives): the number of positively-labeled test sentences that are precisely categorized as positive.
- TN (True Negatives): the number of negatively-labeled test sentences that are precisely categorized as negative.
- FP (False Positives): the number of negatively-labeled test sentences that are improperly categorized as positive.
- FN (False Negatives): the number of positively-labeled test sentences that are improperly categorized as negative.

We organize two experiments as follow. Experiment 1 is conducted to evaluate the accuracy of three traditional machine learning techniques (NB, KNN and Decision tree). Experiment 2 is a comparison between the standard two deep learning models (CNN, RNN) against our combine model (CNN-LSTM).

6.1 Experiment 1: Three traditional machine learning algorithms

Experiment is implemented using two Arabic datasets as in (V.A), to be prepared for eliminating any needless data and each row of datasets was converted into a vector of unigrams, bigram and trigram using (TF-IDF) technique [45, 47]. 75 % of data is chosen randomly for training sets and the rest is allocated for testing. We perform 5-fold cross-validation for getting precise accuracy, this process is repeated until each fold of the 5 folds has been used as the testing set. We get the accuracy of the model by calculating the average value of the 5 testing sets. The experiment outcome summary shown in Table 1 concluded as follows:

Table 1. Average accuracy of N-gram

Datasets	N-gram	accuracy		
		KNN	***NB***	***Decision tree***
HTL	unigram	75.1%	71.8%	71.6%
	bigram	75.4%	68.3%	**72.7%**
	trigram	**76.7%**	69.8%	71.5%
LABR	unigram	52.4%	79.3%	67.4%
	bigram	51.6%	**81.1%**	67.3%
	trigram	51.4%	80.2%	68.9%

Results show that there is no interrelation between the used classifier and N-gram features representation where the best results for (HTL) dataset are achieved by KNN (76.6%) using trigram features while best results for (LABR) dataset are obtained by NB (81.1%) using bigram features.

6.2. Experiment 2: LSTM, CNN and (CNN-LSTM) Models

Before passing the datasets to word embedding layer, it is vital to use preprocessing steps getting rid of the less useful parts of the text then the sentences are transformed to a sequence of word indices. All sequences have the same length. The identical length is the number of the words in the longest sentence in the dataset that equals max words for each dataset, so the sentences which their length is less than max length are padded by zeros so through this method, we can handle sentences in the same way as dealing with images.

For all models, we trained each dataset using the parameters presented in Table 2, and we recorded the model's accuracy when trying to label the testing set. We attempted to reduce cross entropy loss function between the outcomes of the softmax layer and their corresponding labels.

Table 2. Parameters selected for deep learning models

Embedding Dimension	32
Epoch	10
Batch Size	64
Filters	64
Convolution function	ReLu
Kernel Size	3
Pool Size	2
Dropout	0.7
LSTM state dimension	300
Word Embeddings	Not Pre-trained

Fig.6 shows the validation accuracy of the three models for HTL dataset during training in 10 epochs, validation accuracy for CNN model start with 69.2% in first epoch and reached to 84.5% at final epoch while validation accuracy for LSTM model start with 71.2% in first epoch and reached 87.2% at the end, finally, validation accuracy for combined (CNN- LSTM) start with 70.6 and reached the peak with value 89.2%.

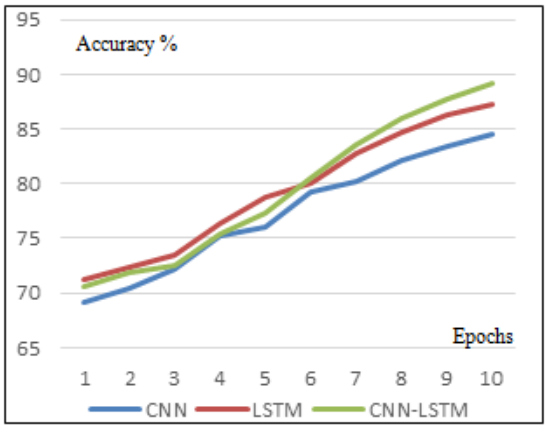

Fig.6. Validation accuracy for HTL dataset

Fig. 7 also indicates the validation accuracy of the three models for LABAR dataset during training in 10 epoch, at

first epoch CNN model is beginning with 50.9% and score is 86.5% at final epoch while accuracy for LSTM model start with 55.3% at first epoch and achieved 89.2% at the end, finally, validation accuracy for combine (CNN- LSTM) start with 52.6 and reached the peak with value 92.2%.

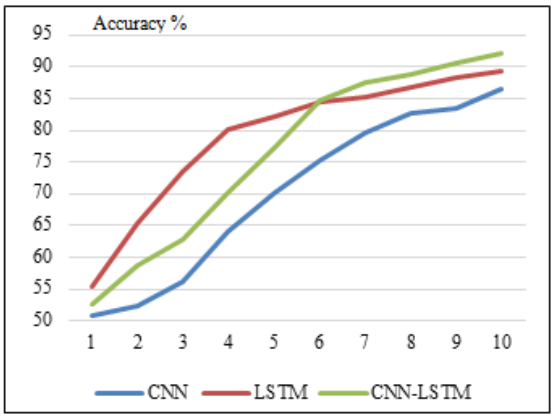

Fig.7. Validation accuracy for LABAR dataset

From Fig. 6 and 7, it is obvious that validation accuracy of (CNN-LSTM) model excel CNN accuracy during 10 epochs, on the other hand, validation accuracy of LSTM model outperformed (CNN-LSTM) accuracy from first epoch till epoch 6 then, we notice the results of our combined model achieved better results than LSTM model from epoch 7 till 10.

The results in Table 3 show the average accuracy of each network taken after 5 tests. For these tests, we generated training sets 75% of data and 25% of data are separated for testing.

From Table 3, it could be noticed that the accuracy of (CNN-LSTM) is 85.38% for (Dataset HTL) and 86.88% for (Dataset LABR) which is a little higher than the single one of CNN which is 80.25%% for (Dataset HTL) and 83.75% for (Dataset LABR) or LSTM which is 84.39% and 85.33%, demonstrating that CNN-LSTM model is more valuable than CNN and LSTM in sentence classification. These outcomes seem to reveal that our initial intuition was exact and that by joining both CNNs and LSTMs, we can take advantage of the capability of CNNs in extracting local patterns and the ability of LSTMs to exploit the long term dependencies.

Table 3. Average accuracy of different deep learning models

Models	Avg. Accuracy (Dataset HTL)	Avg. Accuracy (Dataset LABR)
LSTM	84.39%	85.33%
CNN	80.25%	83.75%
(CNN-LSTM)	85.38%	86.88%

Through the previous experiments of conventional machine learning and deep learning. The results reveal that all models of deep learning were superior to traditional machine learning even we use different feature representation (unigram, bigram, and trigram). On the contrary, deep learning models learn by creating a more abstract representation of data as the network grows deeper, as a result, the model automatically extracts features and yields higher accuracy results.

7. Conclusion

It is noteworthy that we applied various machine and deep learning methods throughout our Study on Arabic reviews data for sentiment analysis purposes. In this paper, we suggest an effective sentiment prediction approach using joint CNN and RNN architecture. Some preprocessing steps have been performed. The two conducted experiments led to many results; both deep learning models CNN, LSTM achieved a significant enhancement in the accuracy compared to traditional machine learning techniques (NB), KNN and Decision Tree also; we prove that our (CNN-LSTM) approach has excellent performance on two Arabic datasets and achieved competitive classification accuracy while exceeding many other different techniques. For upcoming plans; we aim for the following: (1) test different types of RNNs aside from LSTMs for our models. For example, using bidirectional LSTMs might yield an even better result, (2) we believe that accuracy can be improved by using LSTM as an alternative for pooling layers in order to minimize the loss of detailed.

References

[1] Vateekul, P., & Koomsubha, T. (2016, July). A study of sentiment analysis using deep learning techniques on Thai Twitter data. In Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference on (pp. 1-6). IEEE.

[2] El-Makky, N., Nagi, K., El-Ebshihy, A., Apady, E., Hafez, O., Mostafa, S., & Ibrahim, S. (2014, December). Sentiment analysis of colloquial Arabic tweets. In ASE BigData/SocialInformatics/PASSAT/BioMedCom 2014 Conference, Harvard University (pp. 1-9).

[3] Alwakid, G., Osman, T., & Hughes-Roberts, T. (2017). Challenges in Sentiment Analysis for Arabic Social Networks. Procedia Computer Science, 117, 89-100.

[4] Altowayan, A. A., & Tao, L. (2016, December). Word embeddings for Arabic sentiment analysis. In Big Data (Big Data), 2016 IEEE International Conference on (pp. 3820-3825). IEEE.

[5] Abdelhade, N., Soliman, T. H. A., & Ibrahim, H. M. (2017, September). Detecting Twitter Users' Opinions of Arabic Comments During Various Time Episodes via Deep Neural Network. In International Conference on Advanced Intelligent Systems and Informatics (pp. 232-246). Springer, Cham.

[6] Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 1555-1565).

[7] Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International Conference on Machine Learning (pp. 1188-1196).

[8] Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

[9] Roshanfekr, B., Khadivi, S., & Rahmati, M. (2017, May). Sentiment analysis using deep learning on Persian texts. In Electrical Engineering (ICEE), 2017 Iranian Conference on (pp. 1503-1508). IEEE.

[10] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4), 1093-1113.

[11] Alomari, K. M., ElSherif, H. M., & Shaalan, K. (2017, June). Arabic Tweets Sentimental Analysis Using Machine Learning. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 602-610). Springer, Cham.

[12] Yang, P., & Chen, Y. (2017, December). A survey on sentiment analysis by using machine learning methods. In Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2017 IEEE 2nd Information (pp. 117-121). IEEE.

[13] Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg.

[14] Li, D., & Qian, J. (2016, October). Text sentiment analysis based on long short-term memory. In Computer Communication and the Internet (ICCCI), 2016 IEEE International Conference on (pp. 471-475). IEEE.

[15] Vateekul, P., & Koomsubha, T. (2016, July). A study of sentiment analysis using deep learning techniques on Thai Twitter data. In Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference on (pp. 1-6). IEEE.

[16] Baktha, K., & Tripathy, B. K. (2017, April). Investigation of recurrent neural networks in the field of sentiment analysis. In Communication and Signal Processing (ICCSP), 2017 International Conference on (pp. 2047-2050). IEEE.

[17] ElSahar, H., & El-Beltagy, S. R. (2015, April). Building large arabic multi-domain resources for sentiment analysis. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 23-34). Springer, Cham.

[18] Altowayan, A. A., & Tao, L. (2016, December). Word embeddings for Arabic sentiment analysis. In Big Data (Big Data), 2016 IEEE International Conference on (pp. 3820-3825). IEEE.

[19] http://www.goodreads.com

[20] Aly, M., & Atiya, A. (2013). Labr: A large scale arabic book reviews dataset. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Vol. 2, pp. 494-498).

[21] Severyn, A., & Moschitti, A. (2015, August). Twitter sentiment analysis with deep convolutional neural networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 959-962). ACM.

[22] Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

[23] Huq, M. R., Ali, A., & Rahman, A. (2017). Sentiment analysis on Twitter data using KNN and SVM. Int J Adv Comput Sci Appl, 8(6), 19-25.

[24] Hammad, M., & Al-awadi, M. (2016). Sentiment analysis for arabic reviews in social networks using machine learning. In Information Technology: New Generations (pp. 131-139). Springer, Cham.

[25] Desai, M., & Mehta, M. A. (2016, April). Techniques for sentiment analysis of Twitter data: A comprehensive survey. In Computing, Communication and Automation (ICCCA), 2016 International Conference on (pp. 149-154). IEEE.

[26] Day, M. Y., & Lin, Y. D. (2017, August). Deep Learning for Sentiment Analysis on Google Play Consumer Review. In Information Reuse and Integration (IRI), 2017 IEEE International Conference on (pp. 382-388). IEEE.

[27] Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. Journal of machine learning research, 3(Feb), 1137-1155.

[28] Deng, L. (2014). A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing, 3.

[29] Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. In Advances in neural information processing systems (pp. 153-160).

[30] Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In Eleventh Annual Conference of the International Speech Communication Association.
[31] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
[32] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).
[33] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
[34] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
[35] Yih, W. T., He, X., & Meek, C. (2014). Semantic parsing for single-relation question answering. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Vol. 2, pp. 643-648).
[36] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.
[37] Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014, April). Learning semantic representations using convolutional neural networks for web search. In roceedings of the 23rd International Conference on World Wide Web (pp. 373-374). ACM.
[38] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug), 2493-2537.
[39] El-Beltagy, S. R., Khalil, T., Halaby, A., & Hammad, M. (2016, April). Combining lexical features and a supervised learning approach for Arabic sentiment analysis. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 307-319). Springer, Cham.
[40] http://www.tripadvisor.com.
[41] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), 1-135.
[42] Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.
[43] Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2016). Very deep convolutional networks for natural language processing. arXiv preprint.
[44] Sosa, P. M. (2017). Twitter Sentiment Analysis Using Combined LSTM-CNN Models.
[45] Elzayady, H., Badran, K. M., & Salama, G. I. (2018, December). Sentiment Analysis on Twitter Data using Apache Spark Framework. In 2018 13th International Conference on Computer Engineering and Systems (ICCES) (pp. 171-176). IEEE.
[46] Al Sallab, A., Hajj, H., Badaro, G., Baly, R., El Hajj, W., & Shaban, K. B. (2015). Deep learning models for sentiment analysis in Arabic. In Proceedings of the second workshop on Arabic natural language processing (pp. 9-17).
[47] Elhadad, M. K., Badran, K. M., & Salama, G. I. (2017). A novel approach for ontology-based dimensionality reduction for web text document classification. International Journal of Software Innovation (IJSI), 5(4), 44-58.

Authors' Profiles

Hossam Elzayady is a Ph.D. candidate at the Department of computer engineering, received a Bachelor Degree in computer engineering and Masters of Science degree from the MTC, Cairo, Egypt, in 2005 and 2018, respectively. His research interests are in artificial intelligent, data science, machine learning.

Khaled BADRAN received a Bachelor Degree in computer engineering and Masters of Science degree from the MTC, Cairo, Egypt, in 1995 and 2000, respectively. He also received the Ph.D. degree in Electrical and Computer engineering from Sheffield University, UK, in 2009. He is currently a faculty member of the Department of Computer Engineering, MTC. His research interests are in artificial intelligent, data mining, semantic web and database security.

Gouda I. Salama, received the Bachelor engineering and Masters' engineering degrees from MTC, Cairo, Egypt, in 1988 and 1994, respectively. As well, he received the Ph.D. degree in Electrical and computer engineering from Virginia Tech. University, U.S.A., in 1999. He is currently a faculty member with the Department of Computer Engineering, MTC. His research interests are in image and video processing, pattern recognition, and information security.

How to cite this paper: Hossam Elzayady, Khaled M. Badran, Gouda I. Salama, "Arabic Opinion Mining Using Combined CNN - LSTM Models", International Journal of Intelligent Systems and Applications(IJISA), Vol.12, No.4, pp.25-36, 2020. DOI: 10.5815/ijisa.2020.04.03