Pertanika J. Sci. & Technol. 32 (3): 1071 - 1086 (2024)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

An Empirical Evaluation of Adapting Hybrid Parameters for CNN-based Sentiment Analysis

Mohammed Maree*, Mujahed Eleyat and Shatha Rabayah

Faculty of Engineering and Information Technology, Arab American University, P.O Box 240, Jenin, Palestine

ABSTRACT

Sentiment analysis aims to understand human emotions and perceptions through various machine-learning pipelines. However, feature engineering and inherent semantic gap constraints often hinder conventional machine learning techniques and limit their accuracy. Newer neural network models have been proposed to automate the feature learning process and enrich learned features with word contextual embeddings to identify their semantic orientations to address these challenges. This article aims to analyze the influence of different factors on the accuracy of sentiment classification predictions by employing Feedforward and Convolutional Neural Networks. To assess the performance of these neural network models, we utilize four diverse real-world datasets, namely 50,000 movie reviews from IMDB, 10,662 sentences from LightSide Movie_Reviews, 300 public movie reviews, and 1,600,000 tweets extracted from Sentiment140. We experimentally investigate the impact of exploiting GloVe word embeddings on enriching feature vectors extracted from sentiment sentences. Findings indicate that using larger dimensions of GloVe word embeddings increases the sentiment classification accuracy. In particular, results demonstrate that the accuracy of the CNN with a larger feature map, a smaller filter window, and the ReLU activation function in the convolutional layer was 90.56% using the IMDB dataset. In comparison, it was 80.73% and 77.64% using the sentiment140 and the 300 sentiment sentences dataset, respectively. However, it is worth mentioning that, with large-size sentiment sentences (LightSide's Movie Reviews) and using the same parameters, only a 64.44% level of accuracy was achieved.

Keywords: CNN, deep learning, GloVe word embedding, machine learning, sentiment classification

ARTICLE INFO

Article history:
Received: 27 June 2023
Accepted: 05 December 2023
Published: 01 April 2024

DOI: https://doi.org/10.47836/pjst.32.3.05

E-mail addresses:
mohammed.maree@aaup.edu (Mohammed Maree)
mujahed.eleyat@aaup.edu (Mujahed Eleyat)
shatha.rabaia@student.aaup.edu (Shatha Rabaia)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

INTRODUCTION

In the Natural Language Processing (NLP) domain, Sentiment Analysis (SA) has gained significant attention (Hussein, 2018; Maree & Eleyat, 2020; Wang et al., 2018). SA is a classification task that analyzes textual sentences to classify their orientations (i.e., positive, negative or neutral). For individuals and organizations alike, SA provides an important source of valuable information. On the one hand, individuals can use SA to comprehend consumer opinions about products they want to purchase. On the other hand, an organization can use SA to visualize consumer opinions and make informed future decisions (Maree & Eleyat, 2020). Generally, two main SA approaches vary in strengths, weaknesses, and accuracy: (1) Lexicon-based and (2) Machine Learning-based SA models. Using sentiment lexicons comprising word sets paired with their respective sentiment polarities has gained prominence in lexicon-based approaches. Examples of such lexicons include SentiWordNet, SenticNet, and HowNet (Maree & Eleyat, 2020; Shaukat et al., 2020). In contrast, machine learning-based approaches rely on leveraging training samples to make predictions regarding the polarity of opinions. We recommend referring to the following references to delve deeper into these approaches and gain a comprehensive understanding (Yang & Chen, 2017).

Traditional methods, including Support Vector Machine (SVM), Maximum Entropy (ME), and Naïve Bayes (NB) classifiers, are commonly employed for sentiment analysis. These classifiers are often combined with intricate feature extraction techniques to achieve accurate predictions. Among the main limitations of these approaches are the incompleteness of the training data, lack of semantic information about the processed text, domain dependence, and huge computational cost. Newer deep-learning models have been proposed to address these limitations (Stojanovski et al., 2015; Vielma et al., 2020; Yenter & Verma, 2017). Among the main goals in this context are automating the feature learning process and incorporating semantic dimensions through word embedding techniques. We recommend reading a recent survey on the topic for more details on the main advantages of deep learning models (Zhang et al., 2018). An important aspect that we would like to highlight in the context of utilizing deep learning is the ability to couple discrete representations of text using One-hot vectors and distributional representations of words using Global Vectors (GloVe) or Word2Vec.

Starting from this position, we aim to experimentally investigate the impact of exploiting GloVe word embeddings on enriching feature vectors extracted from sentiment sentences and, subsequently, its impact on the quality of the utilized sentiment classifiers, namely FNN and CNN.

In particular, we summarize our contributions to this work as follows:

- Exploring and identifying the influence of various factors on the accuracy of sentiment classification predictions by employing ANN and CNN architectures, utilizing four diverse real-world datasets.

- We experimentally investigate the impact of exploiting GloVe word embeddings on enriching feature vectors extracted from sentiment sentences. Findings demonstrate that using larger dimensions of GloVe word embeddings increases the sentiment classification accuracy.

MATERIALS AND METHOD

Sentiment analysis is a classification problem that aims to identify, extract, and analyze the sentiment orientation of sentences (Hussein, 2018; Qaisar, 2020; Zhang et al., 2018). Among the commonly quoted definitions of the term Sentiment Analysis (SA) is highlighted by Alam and Yao (2019), where the authors define SA as "*a computational process which identifies and categorizes an opinion in a piece of text that expresses the positive, negative, or neutral attitude of a writer towards a particular product, event or personality*" (p. 321). Several approaches have been developed over the past years for analyzing sentiments. Among these approaches is the lexicon-based approach, which uses a dictionary that includes words with their polarities. This approach remains inaccurate as it depends on the word's polarity to determine the text's polarity (Maree & Eleyat, 2020). It can be attributed to the fact that the prior polarity of a word does not necessarily reflect its contextual polarity (Wilson et al., 2009). Recently, researchers witnessed a growing interest in developing machine-learning techniques for SA purposes. Some research works focused on using traditional machine learning techniques such as support vector machines SVMs, Maximum Entropy ME and Naïve Bayes NB models.

Horakova (2015) analyzed the sentiments of the text in the Czech language. The developed application collects data according to several criteria and then classifies user-generated reviews using machine learning. This application consists of three modules, each performing a specific task. The first module collects the data, and the second module pre-processes the raw data collected by the first module, including stop word removal and lemmatization. The third module uses machine learning to classify cleaned texts from the second module. The authors used Selenium Web Driver technology to collect the data, and they used the MorphoDiTa tool to perform morphological analysis to prepare the data for the classification stage. R programming language, namely RTextTools, was employed for the text classification task. Specifically, the authors employed the NB, SVM, Maximum Entropy, Decision Trees and Random Forest to classify sentiment sentences. The results showed the superiority of the Maximum Entropy and Random Forest classifiers over the rest of the classifiers when the lemmatization technique was used as part of the pre-processing pipeline. Despite this achievement, it is important to point out that traditional machine-learning techniques cannot learn features independently. In addition, only relying on the textual content of sentiment sentences will suffer from two main inherent problems.

First, latent semantic dimensions in texts will remain undiscovered and hidden under synonymy, polysemy and other semantically related dimensions, such as hypernyms and meronyms. Second, the dependence of traditional machine learning models on the training data and their domain will make them impractical for capturing the sentiment orientations of sentences in other domains of interest. In an attempt to address these limitations, researchers have recently shifted their focus to analyzing sentiments using deep learning techniques. For instance, Yang and Chen (2017) compared a variety of popular machine learning techniques for sentiment analysis: SVM, NB, ME and Artificial Neural Network method. Researchers discussed these methods in detail, provided an approximate comparison between them, and presented a set of challenges faced by that researchers in the field of sentiment analysis. Specifically, the researchers demonstrated that the NB method and neural networks are highly accurate, whereas the SVM and ME have lower accuracy.

However, despite the improvements introduced by newer deep learning methods, the authors still believe that one of the biggest challenges in this field is to study the various neural networks in depth and determine which features are most effective in sentiment analysis. In the same line of research, many researchers are focusing their efforts on the use of different neural networks, such as feedforward neural networks (FNN), convolutional neural networks (CNN), and recurrent neural networks (RNN) for SA purposes (Rusandi et al., 2021). In general, deep learning approaches follow two main phases: (1) the first phase focuses on word embedding (feature vectorization), and (2) the second phase is used for learning and classification of sentiment orientation of sentences. Stojanovski et al. (2015) attempted to study the importance of pre-trained word vectors to extract sentiments from a dataset of sentences obtained from Twitter. They proposed a deep convolutional neural network with one convolutional layer and two fully connected layers with dropout. A sigmoid activation function was used for the first layer, and a tangent activation function was used for the second layer. Lastly, the softmax activation function was applied to the output layer. The dropout rate that was used ranged between 0.7 and 0.5, respectively. The researchers generated word vectors using three pre-trained word embedding models that are word2vec, global vectors for word representation GloVe, and semantic-specific word embedding (SSWE).

The authors have used three sets from the SemEval Task 10 challenge for training and testing. A set of parameters was used to determine their effect on sentiment analysis accuracy. These parameters are filter window size, number of hidden units, feature maps size, patch size and activation function in the convolutional layer. Experiments showed that using hyperbolic activated tangent units in the convolutional layer, 500 hidden units in the first hidden layer, and 300 hidden units in the second hidden layer, and increasing the size of the feature maps to 300 improved the performance. Further, results indicated that the GloVe word embedding outperformed all other word embedding methods.

A traditional feedforward neural network extracts only the current time information and discards the useful information transmitted in the spatial and time arrangement of the data. To tackle this problem, researchers developed recurrent neural networks (RNNs). However, this type of network faces problems such as gradient explosion and gradient vanishing. As a result, researchers developed LSTM and GRU networks. LSTM can remember long-term information, and its sequential structure is more sophisticated and intelligent than RNN, while GRU is characterized by its high efficiency. Ni and Cao (2020) suggested a model combining LSTM and GRU to extract sentiment polarities. This study used pre-trained word embedding models to create word vectors. Researchers trained and tested their model using both IMDB and Review_Polarity datasets.

Three basic stages make up the model: (1) embedding words layer, during which text is converted into vectors in space using the GloVe model; (2) the output of the first stage is passed to the neural network layer consisting of 64 LSTM units and 64 GRU units in the second stage, and (3) the final stage is the output layer. The results showed that the proposed model performed better than the RNN model in accuracy. Cao et al. (2020) proposed a text sentiment classification based on the Attention Mechanism and Decomposition Convolutional Neural Network model. Parallel Decomposition Convolutional Neural Network (DCNN) was utilized to obtain comprehensive text features. An attention mechanism was integrated to extract important feature information and improve text sentiment classification. Experimental findings indicated that the proposed model performs better than the single-channel model, and the use of the Decomposed Convolutional Neural Network is better than the traditional Convolutional Neural Network. However, it may still be argued that the utilized combined approach may be hindered by high computational costs, especially when handling large-scale datasets. Another model used Bidirectional Encoder Representations from Transformers (BERT) to transform the words in the input sequence into a vector representation (He, 2023). A CNN was used to extract features where the output vector of BERT was convolved along the dimension of sequence length to extract the features in the sequence.

Bidirectional Long Short-Term Memory (BiLSTM) was used to encode the features and capture the long-term dependencies in the sequence. Accordingly, the results of the BiLSTM output were fed into a fully connected layer to make classification predictions. As discussed by the author, the proposed model has produced more precise sentiment classification results than conventional BERT, BiLSTM, CNN, and BERT-BiLSTM models. However, as we argued earlier, the complexity of such SA models can be at a very high computational cost, namely with large-scale datasets. In addition, comparable SA quality can be obtained using conventional models, which can be less complex on the one hand and more efficient on the other.

Identifying sentiment orientation in a given sentence involves several phases, beginning with data collection and pre-processing. It is followed by word vectorization, neural network training, and performance evaluation.

Dataset Collection and Pre-processing

Researchers often rely on Twitter and social media sites to gather datasets and sentiment sentences to test their approaches (Krouska et al., 2016). In our experiments, we used four datasets: the IMDB[1] dataset, which contains 10,662 movie reviews, and the Sentiment140[2] dataset, which contains 1,600,000 tweets collected from Twitter API. Both IMDB and Sentiment140 datasets are publicly available on Kaggle. Additionally, we collected the LightSide dataset consisting of 10,662 samples and 300 generic movie reviews from Twitter. Data collected is raw data that is not ready for use because it may contain unwanted characters, symbols and spelling mistakes (Haddi et al., 2013). In order to make raw data ready for use, it must be prepared by removing unwanted characters, such as numbers, white spaces, hashtags, punctuations and URLs.

Word Embedding

At this stage, the words are converted into vectors in space. The mathematical representations of the texts contribute greatly to the accuracy of the neural network results. Two main approaches can be used to represent word embeddings: discrete representations using One-hot vectors and distributional representations using Global Vectors (GloVe) or Word2Vec (Ni & Cao, 2020; Yang & Chen, 2017). There are many drawbacks to using One-hot vectors, including their extreme sparsity and huge feature vector size. It consumes enormous memory space requirements and makes algorithms more complex, in addition to the inability to show contextual connections among words. Researchers proposed an approach that uses dense vectors to represent features to overcome these limitations. Words are represented as n-dimensional dense vectors using the distributed approach. Where similar vectors represent similar words, relying on this approach, researchers used different approaches to represent words using Zhang and Wallace (2015), where they used random initialization for the word vectors. Then, they allowed the model to learn the most accurate representation of the words. However, this approach was ineffective for handling large-scale sentiment sentence datasets. Researchers recently utilized unsupervised learning to learn word representations from large text corpora (Dos-Santos & Gatti, 2014). This approach provides pre-trained vectors that can be used to perform various NLP tasks. Word2Vec and GloVe pre-trained embeddings are the most efficient and effective ways to convert

[1] https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews/
[2] https://www.kaggle.com/kazanova/sentiment140/

words into corresponding dense vectors. In our experiments, we used GloVe[3] because of its high scalability in speeding up parameter training (Ni & Cao, 2020). Once the words are converted to vectors, they are ready to be passed to the neural network.

Artificial Neural Networks

An artificial neural network can be defined as a mathematical model for the simulation of a network of biological neurons (e.g., human nervous system). It simulates different aspects related to the behavior and capacity of the human brain. Neural networks consist of basic units of computation called nodes or neurons. A neural network consists of a set of layers, each layer containing a set of nodes. Terminal nodes receive data (x). Each entry has a weight (w) determined according to the importance of the entry compared to other inputs. Internal nodes implement an activation function f like (sigmoid, ReLU & tanh) on the weighted sum of its inputs.

$$y = f(x_1 . w_1 + x_2 . w_2 + \mathrm{b}) \tag{1}$$

Equation 1 illustrates how value Y is computed as the neuron's output. In the hidden layers, each neuron receives weighted inputs plus bias from each neuron in the previous layer, as shown by Equation 2.

$$O_i = \sum_{\mathrm{k}=1}^{N_{j-1}} X_k^{j-1} W_{k,i} - b_k \tag{2}$$

where X_k^{j-1} denotes the input from k - th node in the j – th layer, $W_{k,i}$ is the weight of the link between node k and all the nodes in the previous layers, and b is the bias.

The activation function f plays a crucial role as a non-linear function, introducing non-linearity into the output of neurons. It is significant because real-world data often exhibits non-linear characteristics, necessitating neurons to learn and represent non-linear patterns. Many activation functions are used in neural networks, and perhaps the most commonly used are sigmoid, hyperbolic tangent, and Rectified Linear Unit RelU activation functions. A sigmoid activation function adjusts input value into a 1 to 0 range. At the same time, the tangent function adjusts the input values into the range [1, -1]. The ReLU activation function replaces negative values with zero. b a.k.a. the bias represents a constant value that allows the shift of the activation function to better match the prediction with the data.

Based on Equation 2 above, O_i is passed along to an activation function to produce the node output, calculated as $Y_i = f(O_i)$. The sigmoidal function is the most commonly used activation function, defined as Equation 3.

[3] https://nlp.stanford.edu/projects/glove/

$$f(O_i) = \frac{1}{1 + e^{-z_i}} \tag{3}$$

Feedforward Neural Network (FNN)

The feedforward neural network is the simplest type of artificial neural network. It consists of several neurons arranged in layers connected by connections with weights attached. A feedforward neural network has three types of layers: input, hidden, and output. In the input layer, input nodes receive input data and pass it on to the next layer without performing arithmetic operations. Hidden layers are behind input layers, so there is no direct connection to the data source. This layer forms a link between the input layer and the output layer. It performs calculations on the data and passes it to the output layer. A neural network may or may not have hidden layers. FFNs with hidden layers are called Multi-Layer Perceptron (MLP), while a network without any hidden layers is known as a Single Layer Perceptron. In a feedforward neural network containing hidden layers, data passes through them in one direction (forward). The output nodes in the output layer perform arithmetic operations on the data they receive from the network and then pass output to the outside world.

Convolutional Neural Networks

The Convolutional Neural Network (CNN) is among sentiment analysis's most widely used neural networks. Using CNNs automates feature generation, saving training time required by other conventional machine learning. CNNs consist of several layers: convolution, pooling and fully connected Multi-Layer Perceptron (MLP). The first layer in this architecture is a convolutional layer. The output of this layer is called a feature map. This layer uses a kernel, which acts as a sliding window over the feature map, where each piece of data in the convolutional layer is represented as one unit in the feature map. This layer acts as a feature extractor. By applying the max feature or averaging adjacent features on the feature map, the pooling layer reduces them to a single unit. The output of this layer is passed into a feedforward neural network. The general principle of neural networks is learning from errors, so the principle of neural networks can be summarized as follows: first, receiving data, making predictions, comparing the predictions with the real values, then adjusting the weights to predict with greater accuracy next time.

These steps lead to the neural network being trained. The next and final step is testing the model. To further explain the application of CNN in the context of our work, we first map tokens in sentences to their corresponding word vectors from a lookup table obtained based on GloVe term vector representations $L \in R^{kx|V|}$, where k is the dimension of the word vectors, and V is a vocabulary of the words. Each token is projected to a vector $w_i \in R^k$. Consequently, a sentence concatenates the word embeddings $x = \{w_1, w_2, \ldots w_n\}$. After that, the convolution operation $\grave{x_i} = h(W_c . x_{i:i+h-1} + b_c)$ using windows of h sizes is performed on each sentence to produce feature maps. Where $h(.)$ is the hyperbolic tangent function and $x_{i:i+h-1}$ is the concatenation of word vectors from position i to position $i + h - 1$.

RESULTS AND DISCUSSION

For conducting the experiments, we employed the Python programming language to implement the proposed Neural Networks and utilized them to process the four publicly available datasets (Table 1). These datasets include LightSide's movie reviews dataset, a sentiment sentence dataset provided by Maree and Eleyat (2020), sentiment140, and the IMDB dataset. Below, we provide essential information about each of the datasets used.

Table 1
Statistics about the used sentiment review datasets

Dataset	Sentiment Sentences	Positive	Negative
IMDB	50,000	25,000	25,000
Sentiment Sentences from Reference(Maree & Eleyat, 2020)	10,662	5,331	5,3331
LightSide's Movie Reviews	3000	150	150
Sentiment 140	1,600,000	800,000	800,000

After cleaning the data, the HTML tags, punctuation, numbers, and all unnecessary characters and white spaces were removed. It becomes ready for analysis. We have created a word-to-index dictionary using the tokenizer class in the Keras library. Each word in the corpus is a key, while a corresponding unique index is the value. We then loaded the GloVe word embeddings and created a dictionary to contain words as keys and their corresponding embedding lists as values. Creating the embedding matrix is the next step. Each row number corresponds to an index of words in the corpus. In addition, the matrix columns contain GloVe word embeddings for words in our corpus. Such word embeddings support four different vector representations, represented by four-dimension classes: 50, 100, 200 and 300. Thus, by using GloVe pertained model in our experiments, we have generated vectors using these four different representations. We classified the reviews using two types of neural networks. These are FNN and CNN.

Then, we have the training and testing phase using neural networks. Each dataset is broken down into a training set and a testing set. The training set makes up 70% of the total dataset. The testing set makes up the remaining 30%. Our experiments used FFNs with zero hidden and one, two, and three hidden layers. For CNNs, we have used it with different feature map sizes, filter window sizes, and activation functions in the convolution layer. After the convolution layer, we have a max pooling layer, followed by FFNs with a zero hidden layer, one hidden layer, two hidden layers, or three hidden layers. Finally, we have used the accuracy metric to compare these different cases. Tables 2 and 3 illustrate the FNN and CNN network parameters, respectively. In our experiments, we studied the effect of each of the parameters listed below on FNN and CNN network classification accuracy.

- Using multidimensional vector representations of a word.
- Using different numbers of hidden layers.
- Utilizing different activation functions in hidden layers.
- Utilizing multiple activation functions in hidden layers.
- Using different activation functions in the convolution layer.
- Changing feature map sizes used in the convolution layer.
- Changing filter window sizes used in the convolution layer.

Tables 4 to 8 illustrate the variations in the accuracy results when utilizing each of the abovementioned parameters.

Table 2
Used FFN parameters

FFN parameters	
First hidden layer	300 unit
Dropout	0.3
Second hidden layer	50 unit
Dropout	0.2
Third hidden layer	10 unit
Dropout	0.2
Optimizer	Adam
Loss function	Binary-cross entropy
Activation function for the output layer	Sigmoid
Batch size	128
Epochs	6

Table 3
Used convolutional layer parameters

Convolutional layer parameters	
Filters	128, 384
Window size	3, 5
Activation function	Sigmoid, ReLU

Table 4
Experimental results using FNN and CNN without hidden layer

No hidden layers / epochs=6			
Input vector dimensions	**Dataset**	**FNN**	**CNN 128/ ReLU**
300	IMDB	76.33%	89.48%
	sentiment_sentences	70.60%	76.73%
	MovieReviews	51.11%4	52.22%
	Sentiment 140	72.75%	79.63%
200	IMDB	76.04%	89.10%
	sentiment_sentences	70.07%	75.45%7
	MovieReviews	53.33%6	51.11%4
	Sentiment 140	72.03%5	79.31%
100	IMDB	71.60%	88.07%
	sentiment_sentences	68.26%	74.92%
	MovieReviews	53.33%	51.11%
	Sentiment 140	70.49%	79.27%
50	IMDB	69.20%	85.94%
	sentiment_sentences	66.51	71.85%
	MovieReviews	57.77%	58.88%
	Sentiment 140	67.69%	77.21%

Table 5
Experimental results using CNN without hidden layer, using different feature maps, different filter sizes and different activation functions

Input vector dimensions	Dataset	CNN 128/ ReLU	CNN 384/ sigmoid/5	CNN 384/ ReLU/5	CNN 384/ ReLU/3
300	IMDB	89.48%	87.53%	90.20%	**90.56%**
	sentiment_sentences	76.73%	76.51%	76.76%	**77.64%**
	MovieReviews	52.22%	61.11%	63.33%	**63.33%**
	Sentiment 140	79.63%	79.24%	79.09%	**79.98%**

Table 6
Experimental results using FNN and CNN with one hidden layer

One hidden layer / epochs=6 /Dimensions = 300					
Activation function	**Dataset**	**FNN**	**CNN 384/ sig/5**	**CNN 384/ ReLU/5**	**CNN 384/ ReLU/3**
ReLU	IMDB	76.66%	89.50%	89.266%	88.48%
	sentiment_sentences	69.07%	77.11%	77.20%	77.11%6
	MovieReviews	46.66%	46.66%7	53.33%	53.33%
	Sentiment 140	75.68%	79.62%	79.72%	80.73%4
Sig	IMDB	77.14%	89.25%	89.72%	89.96%
	sentiment_sentences	68.63%	75.95%	76.860%	75.10%
	MovieReviews	46.66%	53.33%	46.66%	46.66%
	Sentiment 140	75.97%	79.60%	79.75%	80.21%
Tanh	IMDB	76.53%	89.26%	89.11%	89.5%
	sentiment_sentences	68.48%	76.54%	76.39%	76.67292
	MovieReviews	48.88%	64.44%	46.66%	54.44%
	Sentiment 140	75.70%	78.96%	79.69%	80.37%

Table 7
Experimental results using FNN and CNN with two hidden layers

Two hidden layers / epochs=6 /Dimensions = 300					
Activation function	**Dataset**	**FNN**	**CNN 384/ sig/5**	**CNN 384/ ReLU/5**	**CNN 384/ ReLU/3**
ReLU	IMDB	76.04%	89.10%	86.92%	86.26%
	sentiment_sentences	68.60%	76.61%	75.10%	76.01%
	MovieReviews	47.77%	46.66%	46.66%	52.22%
	Sentiment 140	75.66%	79.56%	79.99%	80.21%
Sig	IMDB	76.36%	89.07%	89.08%	87.34%
	sentiment_sentences	68.07%	77.01%	76.89%	77.45%
	MovieReviews	51.11%	53.33%	53.33%	46.66%
	Sentiment 140	76.08%	79.60%	79.52%	80.36%

Table 7 *(continue)*

Two hidden layers / epochs=6 /Dimensions = 300					
Activation function	**Dataset**	**FNN**	**CNN 384/ sig/5**	**CNN 384/ ReLU/5**	**CNN 384/ ReLU/3**
Tanh	IMDB	74.06%	89.36%	87.20%	89.26%
	sentiment_sentences	67.44%	76.64%	76.26%	76.70%
	MovieReviews	50.00%	53.33%	62.22%	53.33%
	Sentiment 140	75.07%	79.38%	80.10%	80.53%
sig, ReLU	IMDB	76.32%	88.51%	88.71%	89.026%
	sentiment_sentences	68.29%	75.51%	75.39%	76.61%
	MovieReviews	53.33%	46.66%	53.33%	53.33%
	Sentiment 140	75.82%	78.89%	80.00%	80.53%
ReLU,sig	IMDB	75.40%	89.36%	89.00%	89.20%
	sentiment_sentences	69.85%	76.57%	76.17%	77.14%
	MovieReviews	53.33%	53.33%	60.00%	53.33%
	Sentiment 140	75.71%	79.59%	80.10%	80.46%

Table 8
Experimental results using FNN and CNN with three hidden layers

Three hidden layers with / epochs=6 /Dimensions = 300					
Activation function	**Dataset**	**FNN**	**CNN 384/ sig/5**	**Cnn 384/ ReLU/5**	**CNN 384/ ReLU/3**
ReLU	IMDB	75.28%	89.16%	88.56%	85.14%
	sentiment_sentences	67.54%	76.51%	75.54%	76.23%5
	MovieReviews	53.33%	53.33%	50.00%	47.77%
	Sentiment 140	75.87%	79.49%	79.54%	80.57%
Sig	IMDB	76.35%	89.11%	88.94%	88.43%
	sentiment_sentences	69.04%	76.11%	76.86%	74.92%
	MovieReviews	52.22%	46.66%	46.66%	46.66%
	Sentiment 140	76.04%	79.36%6	79.74%	80.69%
Tanh	IMDB	75.93%	89.46%	88.96%	88.15%
	sentiment_sentences	67.54%	76.14134	74.67%	76.07%6
	MovieReviews	48.88%	46.66%	53.33%	46.66%
	Sentiment 140	75.06%	79.38%	79.75%	80.20%

Based on the results in Table 4, we notice that the accuracy increased when we used vectors of larger dimensions to represent the words. The highest classification accuracy was achieved when representing words using GloVe word embeddings with 300 dimensions. As a result, in our experiments, the rest of the runs were performed using GloVe word embeddings with 300 dimensions. As shown in Table 5, using the ReLU activation function in the convolution layer gives better results than using the sigmoid activation function. In addition, the sentiment classification accuracy improved when bigger feature map sizes were used in the convolution layer. We obtained more accurate results in the convolutional layer when using feature maps with size 384 compared to those obtained using feature maps with size 128. Therefore, we used feature maps with size 384 in the convolutional layer in the rest of the experimental runs. Furthermore, we found that using a filter window with a smaller value improves the classification's accuracy, especially when using CNNs with zero hidden layers and one hidden layer.

As shown in Table 5, we obtained more accurate results when using a filter window with size 3 in the convolutional layer compared to using a filter window with size 5. Based on the results in Tables 6, 7 and 8, we notice that the FNN's classification accuracy was best when using sigmoid activation functions in the hidden layers. Nevertheless, the classification accuracy of the convolutional neural networks was better in most cases with the RelU activation function in the hidden layers. As depicted in Table 7, using multiple activation functions in the hidden layers improves the classification accuracy. We obtained the best results using the RelU activation function in the first hidden layer, followed by the Sigmoid activation function in the second hidden layer. Finally, according to the results, using FNN and CNN networks without hidden layers and one hidden layer produced more accurate results than using two and three hidden layers.

We obtained the best result using the FNN with the IMDB dataset (77.14%). In particular, we used one hidden layer with a sigmoid activation function and GloVe word embeddings with 300 dimensions. Considering the Sentiment Sentences dataset used by Maree and Eleyat (2020), we obtained the best result (70.60%) when using FNN without a hidden layer and Glove word embeddings with 300 dimensions. We obtained the best result for the LightSide's Movie_Reviews dataset (57.77%) when we used FNN without a hidden layer and GloVe word embeddings with 50 dimensions. For Sentiment140, we obtained the best result (76.08%) when we used FNN with two hidden layers and a sigmoid activation function. Moreover, when we used CNN, all highly accurate results were obtained using a 384 feature filter with ReLU activation function and filter window with size 3 in the convolutional layer.

For the IMDB dataset, we obtained the best result (90.56%) when using CNN without a hidden layer and GloVe word embeddings with 300 dimensions. Considering the Sentiment Sentences dataset, we obtained the best result (77.64%) when we used CNN without a hidden layer and GloVe word embeddings with 300 dimensions. We obtained the best result for the LightSide's Movie_Reviews dataset (64.44%) using CNN flowed by FNN with one hidden layer with the tanh activation function. For Sentiment140, we obtained the best result (80.73%) when we used a convolutional layer flowed by FNN with one hidden layer with a ReLU activation function.

Comparison with Other SA Models

The results obtained using the IMDB dataset were compared with previous works conducted by Shaukat et al. (2020), Qaisar (2020), Vielma et al. (2020), and Yenter and Verma (2017). These researchers utilized different types of neural networks, including Long Short-Term Memory (LSTM), Single and Multi-branch CNN-Bidirectional LSTM, and CNN-LSTM. The respective results achieved by these researchers are presented in Table 9. As observed in Table 9, the LSTM network demonstrated superior performance to other neural network models, achieving an accuracy of 89.9%. Furthermore, our model surpasses the performance of similar models, with an accuracy of 90.56% (Table 9).

Table 9
Comparison with existing SA models

System	Employed Classifier	Accuracy
Our Result	CNN	90.56%
Sentiment analysis on IMDB using lexicon and neural networks (Shaukat et al., 2020)	lexicon and neural networks	86.67%
Sentiment Analysis of IMDB Movie Reviews Using Long Short-Term Memory (Qaisar, 2020)	Long Short-Term Memory	89.90%
Single and Multi-branch CNN-Bidirectional LSTM for IMDB Sentiment Analysis (Vielma et al., 2020)	Single and Multi-branch CNN-Bidirectional LSTM	89.54%
Deep CNN-LSTM with combined kernels from multiple branches for IMDB review sentiment analysis (Yenter & Verma, 2017)	CNN-LSTM	89.50%

CONCLUSION

Social networking sites and websites have become important platforms for individuals to express their opinions about products and services. Analyzing sentiments is one of the most important techniques to help analyze this large volume of comments. So that individuals and institutions can make informed decisions based on them. Thus, we see that researchers are interested in developing the various techniques used in sentiment analysis. It includes machine learning techniques based on neural networks. This paper employed two types of neural networks for sentiment analysis: the Convolutional neural network CNN and the feedforward neural network (FNN). We studied a set of variables in the neural network to determine how they affect sentiment classification accuracy. These variables include the number of hidden layers used in the network, the activation function used in these layers, the size of the feature maps, the size of the filter window, and the activation function used in the convolutional neural network. In addition, we used glove embedding for word vectorization, whereby we used the different representations supported by the glove.

To test our model, we used four data sets, which included 50,000 movie reviews, 10,662 sentences, 300 public movie reviews, and 1,600,000 tweets. Results show that GloVe word embedding increases accuracy with a large word dimension. Moreover, we found that the convolutional neural network's accuracy improved with a larger feature map, a smaller filter window, and using ReLU activation functions. The neural network's classification accuracy was improved using multiple activation functions in the hidden layers. It is important to point out, however, that among the limitations of our current work are the incompleteness of the training data, lack of semantic information about the processed text, and the domain dependence of the training data utilized for training the SA model. As a future extension of our current work, we will consider comparing the current model with dynamic embeddings using transformer architectures such as the BERT model.

ACKNOWLEDGEMENT

The authors thank the Arab American University of Palestine for funding and support for the research work published based on this research project.

REFERENCES

Alam, S., & Yao, N. (2019). The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. *Computational and Mathematical Organization Theory*, *25,* 319-335. https://doi.org/10.1007/s10588-018-9266-8

Cao, D., Huang, Y., Li, H., Zhao, X., Chen, H., & Fu, Y. (2020, August 25-27). T*ext sentiment classification based on attention mechanism and decomposition convolutional neural network model.* [Paper presentation]. IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China. https://doi.org/10.1109/AEECA49918.2020.9213672

Dos-Santos, C., & Gatti, M. (2014, August 23-29). *Deep convolutional neural networks for sentiment analysis of short texts*. [Paper presentation]. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland.

Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, *17*, 26-32. https://doi.org/10.1016/j.procs.2013.05.005

He, Y. (2023, July 14-16). *BERT-CNN-BiLSTM: A Hybrid Deep Learning Model for Accurate Sentiment Analysis*. [Paper presentation]. IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China. https://doi.org/10.1109/ICPICS58376.2023.10235335

Horakova, M. (2015). Sentiment analysis tool using machine learning. *Global Journal on Technology, 2015*(5), 195-204.

Hussein, D. M. E. D. M. (2018). A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, *30*(4), 330-338. https://doi.org/10.1016/j.jksues.2016.04.002

Krouska, A., Troussas, C., & Virvou, M. (2016, July 13-15). *The effect of preprocessing techniques on Twitter sentiment analysis*. [Paper presentation]. 7th International Conference on Information, Intelligence, Systems & Applications (IISA), Chalkidiki, Greece. https://doi.org/10.1109/IISA.2016.7785373

Maree, M., & Eleyat, M. (2020). Semantic graph based term expansion for sentence-level sentiment analysis. *International Journal of Computing 19*(4), 647-655. https://doi.org/10.47839/ijc.19.4.2000

Ni, R., & Cao, H. (2020, July 27-29). S*entiment analysis based on GloVe and LSTM-GRU*. [Paper presentation]. 39th Chinese Control Conference (CCC), Shenyang, China. https://doi.org/10.23919/CCC50068.2020.9188578

Qaisar, S. M. (2020, October 13-15). *Sentiment analysis of IMDb movie reviews using long short-term memory*. [Paper presentation]. 2nd International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia. https://doi.org/10.1109/ICCIS49240.2020.9257657

Rusandi, M. R., Sutoyo, E., & Widartha, V. P. (2021, November 3-4). *Convolutional neural network for predicting sentiment: Case study in tourism*. [Paper presentation]. Sixth International Conference on Informatics and Computing (ICIC), Jakarta, Indonesia.

Shaukat, Z., Zulfiqar, A. A., Xiao, C., Azeem, M., & Mahmood, T. (2020). Sentiment analysis on IMDB using lexicon and neural networks. *SN Applied Sciences*, *2*(2), Article 148. https://doi.org/10.1007/s42452-019-1926-x

Stojanovski, D., Strezoski, G., Madjarov, G., & Dimitrovski, I. (2015). Twitter sentiment analysis using deep convolutional neural network. In E. Onieva, I. Santos, E. Osaba, H. Quintian & E. Carchado (Eds.), *Hybrid Artificial Intelligence Systems* (pp. 726-737). Springer. https://doi.org/10.1007/978-3-319-19644-2_60

Vielma, C., Verma, A., & Bein, D. (2020). Single and multibranch CNN-bidirectional LSTM for IMDb sentiment analysis. In S. Latifi. (Ed.), *17th International Conference on Information Technology–New Generations* (pp. 401-406). Springer. https://doi.org/10.1007/978-3-030-43020-7_53

Wang, M., Chen, S., & He, L. (2018). Sentiment classification using neural networks with sentiment centroids. In D. Phung, V. S. Tseng, G. O. Webb, B. Ho, M. Ganji & L. Rashidi (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 56-67). Springer. https://doi.org/10.1007/978-3-319-93034-3_5

Wilson, T., Wiebe, J., & Hoffmann, P. (2009). Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, *35*(3), 399-433. https://doi.org/10.1162/coli.08-012-R1-06-90

Yang, P., & Chen, Y. (2017, December 15-17). *A survey on sentiment analysis by using machine learning methods*. [Paper presentation]. IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China. https://doi.org/10.1109/ITNEC.2017.8284920

Yenter, A., & Verma, A. (2017, October 19-21). *Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis.* [Paper presentation]. IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York. USA. https://doi.org/10.1109/UEMCON.2017.8249013

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), Article e1253. https://doi.org/https://doi.org/10.1002/widm.1253

Zhang, Y., & Wallace, B. (2015). *A Sensitivity Analysis of (and practitioners' guide to) Convolutional Neural Networks for Sentence Classification*. arXiv preprint arXiv:1510.03820. https://doi.org/10.48550/arXiv.1510.03820