KSII Transactions on
Internet and Information Systems

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

Journal:	*KSII Transactions on Internet and Information Systems*
Manuscript ID	TIIS-AB-2022-Feb-0152
Manuscript Type:	Artificial Intelligence & Big Data
Date Submitted by the Author:	14-Feb-2022
Complete List of Authors:	Maree, Mohammed; Arab American University, Information Technology Eleyat, Mujahed; Arab American University, Computer Systems Engineering Rabaia, Shatha; Arab American University, Dept. of Computer Science
Keywords of your Paper:	Sentiment Classification, Machine Learning, Deep Learning, CNN, GloVe Word Embedding

KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. X, NO. X, December 20XX 134

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

Mohammed Maree*, Mujahed Eleyat, Shatha Rabayah
[1] Faculty of Engineering and Information Technology, Arab American University
Jenin, Palestine
[e-mail: {mohammed.maree, mujahed.eleyat, shatha.rabaia}@aaup.edu]

*Correspondence concerning this paper should be addressed to Mohammed Maree, Department of Information Technology, Faculty of Engineering and Information Technology, Arab American University, P.O Box 240 Jenin, 13 Zababdeh 00970-4-2418888 ext. 1607, Palestine. E-mail: mohammed.maree@aaup.edu

Abstract

Sentiment analysis aims at understanding human emotions and perceptions through the utilization of various machine learning pipelines. However, conventional machine learning techniques are often hindered with feature engineering and inherent semantic gap constraints that limit their accuracy. To address these challenges, newer neural network models have been proposed in an attempt to automate the feature learning process and enrich learned features with word contextual embeddings to identify their semantic orientations. In this article, we quantify the impact of various parameters on the quality of the produced sentiment classification results using Feedforward Neural Networks (FNNs) and Convolutional Neural Networks (CNNs). The quality of each of these neural network models is measured using four real-world datasets consisting of 50,000 movie reviews (IMDB), 10,662 sentences (LightSide Movie_Reviews), 300 public movie reviews, and 1,600,000 tweets (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 achieved accuracy of the CNN with a larger feature map, a smaller filter window, as well as using ReLU activation function in the convolutional layer was 90.56% using the IMDB dataset, while 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 64.44% level of accuracy was achieved.

Keywords: Sentiment Classification, Machine Learning, Deep Learning, CNN, GloVe Word Embedding.

1. Introduction

Sentiment Analysis (SA) has started to become among the most active research fields in the Natural Language Processing (NLP) domain [1-3]. It can be defined as a classification task that aims to analyze textual sentiment sentences to classify their orientations (i.e., positive, negative or neutral). For individuals and organizations alike, SA provides an important source of valuable information. Individuals, on the one hand, can use SA to comprehend consumer opinions about products they want to purchase. An organization, on the other hand, can use SA to visualize consumer opinions about their products, so they can make future decisions regarding them based on this information [3]. In general, there are two main SA approaches that both have their strengths and weaknesses, and vary in their accuracy in terms of the results they produce. These are: i) Lexicon-based SA approaches and ii) Machine Learning based SA models. As far as lexicon-based approaches are concerned, lexicons containing sets of words associated with their sentiment polarities, such as SentiWordNet, SenticNet and HowNet lexicons are employed [3, 4]. On the other hand, machine learning based approaches rely on using training samples to predict the polarity of opinions. For a comprehensive overview of these approaches, we refer the reader to [5]. Conventional methods such as Support Vector Machine SVM, Maximum Entropy ME and Naïve Bayes NB classifiers, which are often combined with complex feature extraction process, are utilized for this purpose. Among the main limitation of these approaches are the incompleteness of the training data, lack of semantic information about the processed text, domain-dependance and huge computational cost when processing verbose sentiment sentences. To address these limitations, newer deep learning models have been proposed [6-8]. Among the main goal in this context is to automate the feature learning process; requiring less training data, and introducing semantic dimensions through the utilization of word embedding techniques. For more details on the main advantages that deep learning models introduce, we refer the reader to a recent survey on this topic [9]. 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. The rest of this paper is structured as follows. In section 2, we introduce the main approaches used to analyze sentiments and highlighting related works in this context. In section 3, we introduce and discuss our proposed methodology in details. In section 4, the experimental setup is presented, along with a description of the employed datasets. The performance of the proposed methods is quantified and further compared to other existing models. Finally, in section 5, we present our conclusions and point to further research directions in the future work.

2. Related Work

Sentiment Analysis is a classification problem that aims at identifying, extracting, and analyzing the sentiment orientation of sentences [1, 3, 9-11]. Among the commonly quoted definitions of the term *Sentiment Analysis (SA)* is one highlighted by Alam and Yao [12], 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*". 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. As it depends on the polarity of the word to determine the polarity of the text, this approach remains inaccurate [3]. In particular, this is due to the fact that a word's prior polarity does not reflect its contextual polarity [13]. 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. In [14], the researchers explain their application that analyzes 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 the machine learning approach. This application consists of three modules, where each one performs a specific task. The first module is responsible for collecting the data, the second module pre-processes the raw data collected by first module, including stop words removal, and lemmatization. The third module uses the machine learning approach 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 to performs 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 both the Maximum Entropy and Random Forest classifiers over the rest of the classifiers when 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 by their own. In addition, only relying on 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 dependance of traditional machine learning models on the training data and their domain will make them impractical for capturing the sentiment orientations of sentence in other domains of interest. In an attempt of addressing these limitations, researchers have recently shifted their focus on analyzing sentiments using deep learning techniques. For instance, the authors of [5] compared a variety of popular machine learning techniques for sentiment analysis. Namely SVM, NB, ME and Artificial Neural Network method. Researchers discussed these methods in detail and provided an approximate comparison between them, as well as presented a set of challenges that researchers in the field of sentiment analysis are facing. Specifically, the researchers demonstrated that the NB method and neural networks are characterized by high accuracy, whereas the SVM and ME are of 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 the field of sentiment analysis. In the same line of research, many researchers are focusing their efforts on the use of different neural networks, such as feed-forward neural networks (FNN), convolutional neural networks (CNN), and recurrent neural networks(RNN) for SA purposes. In general, deep learning approaches follow two main phases: (i) the first phase focuses on the word embedding (feature vectorization), while (ii) the second phase is used for learning and classification of sentiment orientation of sentences. In [8], the authors attempted to study the importance of pre-trained word vectors to extract sentiments from a dataset of sentences obtained from Twitter. The authors proposed a deep convolutional neural network with one convolutional layer followed by 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 on 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 the 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, using 500 hidden units in the first hidden layer, and 300 hidden units in the second hidden layer, as well as increasing the size of the feature maps to 300 has 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 (RNN). However, this type of networks faces problems such as gradient explosion and gradient vanishing. As a result, researchers developed LSTM and GRU network. 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. So in reference [15], the researchers suggest a model that combines both LSTM and GRU to extract sentiment polarities. In this study, researchers 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: First, embedding words layer, during which text is converted into vectors in space using the GloVe model. Afterwards, 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. The final stage is the output layer. The results showed that the proposed model performed better than the RNN model in terms of accuracy.

3. Proposed Methodology and Theoretical Framework

The process of identifying the sentiment orientation of a given sentence passes through several phases, starting from data collection and preparation phase, to word vectorization, training the neural network, and finally testing its quality. In the next sections, we introduce the details of each of these phases.

3.1 Data Acquisition and Cleansing

Researchers often rely on Twitter and social media sites to gather datasets and sentiment sentences that they use to test their approaches [16]. In our experiments, we used four datasets, namely the IMDB[1] dataset, which contains 10,662 movie reviews, the Sentiment140[2] dataset, which contains 1,600,000 tweets collected from twitter API. Both IMDB and Sntiment140 are publicly available at Kaggle. We also collected the 10,662 LightSide dataset, and 300 other generic movie reviews from Twitter. Data collected is raw data that is not ready for use

---

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

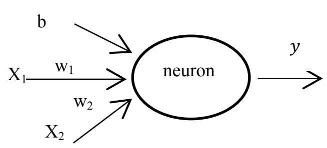
because it may contain unwanted characters, symbols, spelling mistakes, etc. [17]. In order to make raw data ready for use, it must be prepared by removing unwanted characters, such as numbers, white spaces, hashtags, punctuations and URLs.

3.2 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, namely discrete representations using One-hot vectors and distributional representations using Global Vectors (GloVe) or Word2Vec [5, 15]. There are many drawbacks to using One-hot vectors, including their extreme sparsity and huge feature vector size. This consumes enormous memory space requirements and makes algorithms more complex. In addition to the inability to show contextual connections among words. To overcome these limitations, researchers proposed an approach that uses dense vectors to represent features. Using the distributed approach, words are represented as n-dimensional dense vectors. Where similar words are represented by similar vectors. Relying on this approach, researchers used different approaches to represent words, such as in reference [18] where the authors used random initialization for the word vectors, and then allowed the model to learn the most accurate representation of the words. However, this approach was not effective for handling large-scale datasets of sentiment sentences. Recently, researchers utilized unsupervised learning in order to learn word representations from large text corpora [19]. 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 words into corresponding dense vectors. In our experiments, we used the GloVe[3] because of its high scalability in speeding up parameter training [15]. Once the words are converted to vectors, they are ready to be passed to the neural network.

3.3 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.*" [20]. 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, ...etc.) on the weighted sum of its inputs.



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

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

Equation 1 illustrates how value Y is computed as the output of the neuron. Activation Function f is a non-linear function, where it introduces non-linearity into neuron output. The reason why this is important is that most real-world data is non-linear, and we need neurons to learn non-linear representations. 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. While the tangent function adjusts the input values into the range [1, -1]. ReLU activation function replaces negative values with zero. b a.k.a. the bias represents a constant value that allows to shift the activation function in order to better match the prediction with the data.

3.3.1 Feedforward Neural Network (FNN)

The feedforward neural network is the simplest type of artificial neural networks. It consists of several neurons arranged in layers connected by connections with weights attached to them. There are three types of layers in a feedforward neural network: input layer, hidden layer and output layer. In the input layer, input nodes receive input data and pass it on to the next layer without performing any arithmetic operations. Hidden layers are located 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. Where 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.

3.3.2 Convolutional Neural Networks - CNNs

The Convolutional Neural Network is among the most widely used neural networks for sentiment analysis. Using CNNs automates feature generation, saving training time required by other conventional machine learning. CNNs consist of several layers, namely convolution layer, pooling layer 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 model.

4. Experimental Setup and Evaluation

To perform the experiments, we used Python program language for implementing the proposed Neural Networks and utilize them for processing the four publicly-available datasets described in **Table 1.** These are: LightSide's movie reviews dataset, sentiment sentence dataset provided by [3], sentiment140, and the IMDB dataset. Below, we present some basic information about the used datasets.

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[3]	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 by removing HTML tags, punctuation, and numbers, along with all unnecessary characters and white spaces. It becomes ready for analysis. We have created a word-to-index dictionary using the tokenizer class in 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 word 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, which are represented by four dimension classes: 50, 100, 200 and 300. Thus, by using GloVe pertained model in our experiments, we have generate vectors using these four different representations. We classified the reviews using two types of neural networks. These are: FNN, and CNN. Then we have training and testing phase using neural networks. Each dataset is broken down into training set and testing set. Training set makes up 70% of the total dataset. The testing set makes up the remaining 30%. In our experiments we have used FFNs with a zero hidden layer, one hidden layer, two hidden layers, and three hidden layers. For CNNs we have used it with different feature map sizes, different filter window sizes, and different activation functions in the convolution layer. After convolution layer, we have 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 accuracy metric to compare between these different cases. **Tables 2 and 3** illustrate both 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 networks 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.

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 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

Tables 4-8 illustrate the variations on the accuracy results when utilizing each of the abovementioned parameters.

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 Diffrent Feature Maps, Different Filter Sizes and Diffrent 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%
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%

4.1 Discussion of the Results

Based on the results in **Table 4**, we notice that when we have used vectors of larger dimensions to represent the words, the accuracy increased. Whereas the highest classification accuracy was achieved when representing words using GloVe word embeddings with 300 dimensions. As a result, in our experiments, for 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 accuracy of the sentiment classification improved when bigger feature map sizes were used in the convolution layer. Where we obtained more accurate results when using features maps with size 384 in the convolutional layer compared to the results obtained when using features maps with size 128. Therefore, in the rest of the experimental runs, we used features maps with size 384 in the convolutional layer. 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 filter window with size 3 in the convolutional layer compared to using 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 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 when 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 using one hidden layer produced more accurate results than when using two, and three hidden layers.

As we can see in **Tables 4-6**. When we used the FNN with the IMDB dataset, we obtained the best result (77.14%). In particular, when we used one hidden layer with sigmoid activation function and GloVe word embeddings with 300 dimensions. Considering the Sentiment Sentences dataset that is used by the authors of [3], we obtained the best result (70.60%) when using FNN without hidden layer and using Glove word embeddings with 300 dimensions. For the LightSide's Movie_Reviews dataset, we obtained the best result (57.77%) when we used FNN without hidden layer and using GloVe word embeddings with 50 dimensions. And for Sentiment140 we obtained the best result (76.08%) when we used FNN with two hidden layers and sigmoid activation function. Moreover, when we used CNN, all highly-accurate results were obtained when we have used 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 hidden layer, and using GloVe word embeddings with 300 dimensions. Considering the Sentiment Sentences dataset, we obtained the best result (77.64%) when we used CNN without hidden layer, and using GloVe word embeddings with 300 dimensions. For the LightSide's Movie_Reviews dataset, we obtained the best result (64.44%) when using CNN flowed by FNN with one hidden layer with tanh activation function. And for the Sentiment140 we obtained the best result (80.73%) when we used a convolutional layer flowed by FNN with one hidden layer with ReLU activation function.

4.2 Comparison with Other SA Models

In this section, we compare the results that we obtained when using the IMDB dataset with other similar previous works, namely [4], [11], [7] and [6]. The researchers used different types of neural networks such as Long Short-Term Memory, Single and Multibranch CNN-Bidirectional LSTM and CNN-LSTM. The results obtained by the researchers are shown in **Table 9**. As shown in this table, the LSTM network proved to outperform the rest of the neural network as it produced the highest accuracy result which is 89.9%. As shown in the table below, our model outperforms the rest of the similar models, where the accuracy reached 90.56%.

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[4]	lexicon and neural networks	86.67%
Sentiment Analysis of IMDB Movie Reviews Using Long Short-Term Memory[11]	Long Short-Term Memory	89.90%
Single and Multibranch CNN-Bidirectional LSTM for IMDB Sentiment Analysis[7]	Single and Multi branch CNN-Bidirectional LSTM	89.54%
Deep CNN-LSTM with combined kernels from multiple branches for IMDB review sentiment analysis[6]	CNN-LSTM	89.50%

5. Conclusions and Future Work

Social networking sites and websites have become an important platform for individuals to express their opinions about the products and services offered to them. An analysis of sentiments is one of the most important techniques used 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. This includes machine learning techniques based on neural networks. In this paper, we employed two types of neural networks for sentiment analysis, namely, 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 the accuracy of sentiment classification. These variables include the number of hidden layers used in the network, the activation function used in these layers, as well as the size of the feature maps, the size of the filter window, and the activation function used in the convolutional neural network. As well as that, we used the glove embedding for word vectorization, whereby we used the different representations supported by 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 when GloVe word embedding is used with a large word dimension, accuracy increases. 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 also improved by using multiple activation functions in the hidden layers.

References

[1] D. M. E.-D. M. Hussein, "A survey on sentiment analysis challenges," *Journal of King Saud University - Engineering Sciences,* vol. 30, no. 4, pp. 330-338, 2018. Article (CrossRef Link).

[2] M. Wang, S. Chen, and L. He, "Sentiment Classification Using Neural Networks with Sentiment Centroids," in *PAKDD*, 2018. Article (CrossRef Link).

[3] M. Maree and M. Eleyat, "Semantic Graph based Term Expansion for Sentence-Level Sentiment Analysis. ," *International Journal of Computing* vol. 19, no. 4, pp. 647-655, 12/30 2020. Article (CrossRef Link).

[4] Z. Shaukat, A. A. Zulfiqar, C. Xiao, M. Azeem, and T. Mahmood, "Sentiment analysis on IMDB using lexicon and neural networks," *SN Applied Sciences,* vol. 2, no. 2, p. 148, 2020/01/02 2020. Article (CrossRef Link).

[5] P. Yang and Y. Chen, "A survey on sentiment analysis by using machine learning methods," in *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC),* IEEE, pp. 117-121, 2017. Article (CrossRef Link).

[6] A. Yenter and A. Verma, "Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis," in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference*, 19-21, pp. 540-546 2017. Article (CrossRef Link).

[7] C. Vielma, A. Verma, and D. Bein, "Single and Multibranch CNN-Bidirectional LSTM for IMDb Sentiment Analysis," 2020. Article (CrossRef Link).

[8] D. Stojanovski, G. Strezoski, G. Madjarov, and I. Dimitrovski, "Twitter sentiment analysis using deep convolutional neural network," in *International Conference on Hybrid Artificial Intelligence Systems*, Springer, pp. 726-737, 2015. Article (CrossRef Link).

[9] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 8, no. 4, p. e1253, 2018. Article (CrossRef Link).

[10] D. Mohey El-Din, "A Survey on Sentiment Analysis Challenges," *Journal of King Saud University - Engineering Sciences,* 04/01 2016. Article (CrossRef Link).

[11] S. M. Qaisar, "Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory," in *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, 13-15, pp. 1-4, 2020. Article (CrossRef Link).

[12] S. Alam and N. Yao, "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis," *Computational and Mathematical Organization Theory,* vol. 25, 09/01 2019. Article (CrossRef Link).

[13] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis," *Computational linguistics,* vol. 35, no. 3, pp. 399-433, 2009. Article (CrossRef Link).

[14] M. Horakova, "Sentiment analysis tool using machine learning," *Global Journal on Technology,* 2015. Article (Web link).

[15] R. Ni and H. Cao, "Sentiment Analysis based on GloVe and LSTM-GRU," in *2020 39th Chinese Control Conference (CCC)*, IEEE, pp. 7492-749, 2020. Article (CrossRef Link).

[16] A. Krouska, C. Troussas, and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," in *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, IEEE, pp. 1-5, 2016. Article (CrossRef Link).

[17] E. Haddi, X. Liu, and Y. Shi, "The role of text pre-processing in sentiment analysis," *Procedia Computer Science,* vol. 17, pp. 26-32, 2013. Article (CrossRef Link).

[18] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820,* 2015. Article (Web link).

[19] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 69-78, 2014. Article (Web link).

[20] N. K. Kamila, *Handbook of research on emerging perspectives in intelligent pattern recognition, analysis, and image processing*. IGI Global, 2015. Article (CrossRef Link)