DOI: 10.24996/ijs.2020.61.6.31

ISSN: 0067-2904

The Performance Differences between Using Recurrent Neural Networks and Feedforward Neural Network in Sentiment Analysis Problem

Samar K. Abbas[1], Loay E. George [*]

Computer Science Department, College of Science, University of Baghdad, Baghdad, Iraq

Received: 15/9/2019 Accepted: 31/10/2019

Abstract

With the spread use of internet, especially the web of social media, an unusual quantity of information is found that includes a number of study fields such as psychology, entertainment, sociology, business, news, politics, and other cultural fields of nations. Data mining methodologies that deal with social media allows producing enjoyable scene on the human behaviour and interaction. This paper demonstrates the application and precision of sentiment analysis using traditional feedforward and two of recurrent neural networks (gated recurrent unit (GRU) and long short term memory (LSTM)) to find the differences between them. In order to test the system's performance, a set of tests is applied on two public datasets. The first dataset is collected data from IMDB that contains movie reviews expressed through long sentences of English, whereas the second dataset is a collection of keyword search results of tweets using the Twitter Search API; these tweets are written in English words with short sentences. In this work, a certain pre-processing operation is added to the system and a set of tests is conducted to evaluate the performance enhancement on the whole system due to the addition of these operations. The results of the usage of the traditional feedforward neural networks are poor and do not perform the desired purpose in analysis, because of their inability to save information at a long term and, therefore, their loss of efficiency. While the results of using GRU and LSTM are relatively good and do perform the desired purpose in analysis. A recurrent neural network has been built so that any type of text-related data can be pushed to get the polarity of sentiment by multi deep operations that are dependent on the extracted information.

Keywords: Data mining, Gated recurrent unit, Long short term memory, Feedforward ANN, Recurrent neural network, Sentiment Text analysis.

فروق الأداء بين استخدام الشبكات العصبية المتكررة و الشبكة العصبية للأمام في مشكلة تحليل المعنويات

سمر خضير عباس، لؤي ادور جورج[*]

علوم الحاسوب, كلية العلوم, جامعة بغداد, بغداد, العراق

الخلاصة

مع انتشار استخدام الإنترنت وخاصة شبكة التواصل الاجتماعي ، تم العثور على كمية غير عادية من المعلومات التي تشمل عددًا من مجالات الدراسة مثل علم النفس والترفيه وعلم الاجتماع والأعمال والأخبار والسياسة وغيرها من المجالات الثقافية للأمم. تسمح منهجيات استخراج البيانات التي تتعامل مع وسائل

[*] Email:loayedwar57@yahoo.com

التواصل الاجتماعي بإنتاج مشهد ممتع حول السلوك والتفاعل البشري. توضح هذه الورقة تطبيق ودقة تحليل المشاعر باستخدام feedforward التقليدية واثنين من الشبكات العصبية المتكررة (وحدة بوابات المتكررة (GRU) والذاكرة طويلة المدى (LSTM)) لإيجاد الفرق بينهما. لاختبار أداء النظام ، يتم تطبيق مجموعة من الاختبارات على مجموعتي بيانات عامتين. يتم جمع البيانات الأولى من IMDB التي تحتوي على مراجعات للأفلام معبّرة من خلال الجمل الطويلة للغة الإنجليزية ، ومجموعة البيانات الثانية هي مجموعة من twitter باستخدام واجهة برمجة تطبيقات Search Twitter لجمع هذه التغريدات باستخدام البحث عن الكلمات الرئيسية ؛ هذه التغريدات هي في الكلمات الإنجليزية مع جمل قصيرة. في هذا العمل ، تتم إضافة بعض عمليات المعالجة المسبقة إلى النظام ، ويتم إجراء مجموعة من الاختبارات لتقييم تحسين الأداء على النظام بأكمله بسبب إضافة هذه العمليات. تكون النتائج عند استخدام الشبكات العصبية التقليدية لـ feedforward ضعيفة ولا تؤدي الغرض المنشود في التحليل ، بسبب عدم قدرتها على حفظ المعلومات على المدى الطويل ، وبالتالي تفقد الكفاءة. في حين أن نتائج استخدام GRU و LSTM جيدة نسبيا وتؤدي الغرض المنشود في التحليل. تم بناء شبكة عصبية متكررة بحيث يمكن دفع أي نوع من البيانات المتعلقة بالنص للحصول على قطبية المشاعر من خلال عمليات عميقة متعددة تعتمد على المعلومات المستخرجة.

1. Introduction

Analysis of sentiment is considered as one of the major common implementation of analytics of text and is applied in wide fields of tutorials, applications of mobiles, web sites etc. These fields concentrate on the process of sentiments analysing using various text sources starting from companies surveys such as google opinions and ending by reviews of movies such as international movie database. Analysis of sentiment is currently applied in a wide range in commercial websites and pages of social media such as Twitter, Instagram and Facebook, as well as movie review websites of products, by determining the human opinions [1].

Analysis of sentiment, also commonly called mining of opinions, can be expressed as the operation that implements techniques of NLP, linguistics, dictionary resources, as well as machine learning to bring information such as modality, emotions and mood. After that, the information gained are used to calculate the sense of text documented [2]. From the sense gained, it can be determined if the text document shows negative, positive, or impartial sentiment. Furthermore, there is a much more developed analysis where additional complicated emotions can be applied such as anger, sadness, and sarcasm. Many methods are applied to do this work such as recurrent neural networks, hybrid classification, and deep convolutional neural networks [3].

At present, analysis of sentiment is a subject of large benefit and evaluation since it has too many practical implementations. With the assistance of systems used with sentiment analysis, not structured instructions could be dynamically converted to the structured information of general opinions about brands, services, products, protocols, or every subject that people might talk about [4]. This information could be very beneficial for trading implementations such as analysis of marketing, general relations, reviews of product, net promoter scoring, feedback about some product, and services of customers [4].

A previous work [5] introduced a system of recurrent neural network called quasi recurrent neural network. It can be described as a method to modelling neural orders as alternative to classical layers applied through time steps in parallel as well as functions of minimalist recurrent through channels in parallel. In spite of missing layers of trainable QRNN, it showed higher quality of accuracy than the long-short term memory with similar hidden dimension. Another work [6] proposed a bidirectional network, Bi-GRU, which does not only focus on information of position for aspect terms but alternately embodiment the connection between the sentences and aspect terms using two direction attention techniques. The practical results applied on datasets of SemEval 14 explained the strengthening of the suggested PBAN network. The basic theory of PBAN network is to establish aspect terms of position implant to determining the weights of attention. Authors of a previous study [7] suggested various techniques of LSTM structures for analysis of sentiment with review of movies. Their results indicated that the method of LSTM RNN gives a more effective performance than classical RNN and deep neural networks for analysis of sentiment. They used simple models of LSTM and evaluated their performances, then added layers of LSTM one after one, which provided increment forthe accuracy. Finally, bidirectional layers of LSTM were established to cover information with forward and backward networks. Also, other authors [8] suggested a network named CA-LSTM to combine previous tweets for classification of sentiment. The networks of context and attention-based long short-term memory depend on a hierarchal framework to simulate the sequence of microblog and determine the tweets and words with various weights utilizing mechanism of attention. Another article [9] proposed a mechanism for creating a model capable of predicting performance of learning, extracting feature of learning, and reasoning of results. An initially common feature of learning verification approach was established for converting the raw data from systems of e-learning to groups of separate features of learning. They submitted a developed parallel neural network to display the results of prediction.

In this paper, deep learning neural networks are adopted to solve the classification problems related to social data. Deep learning is one of the techniques in machine learning that determines multiple layers of non-linear information manipulated for extraction of features supervised, as well as classification and pattern analysis [10]. Deep neural networks as well as recurrent neural network have been implemented in fields such as recognition of speech, computer vision, and natural language processing (NLP) [11].

2. Materials and Methods

The workflow of the proposed system consists of three main phases: (a) Preparation phase, (b) Neural network phase, (c) Remove garbage words phase. Each phase consists of many stages.

The general structure of the proposed system is shown in Figure-1.

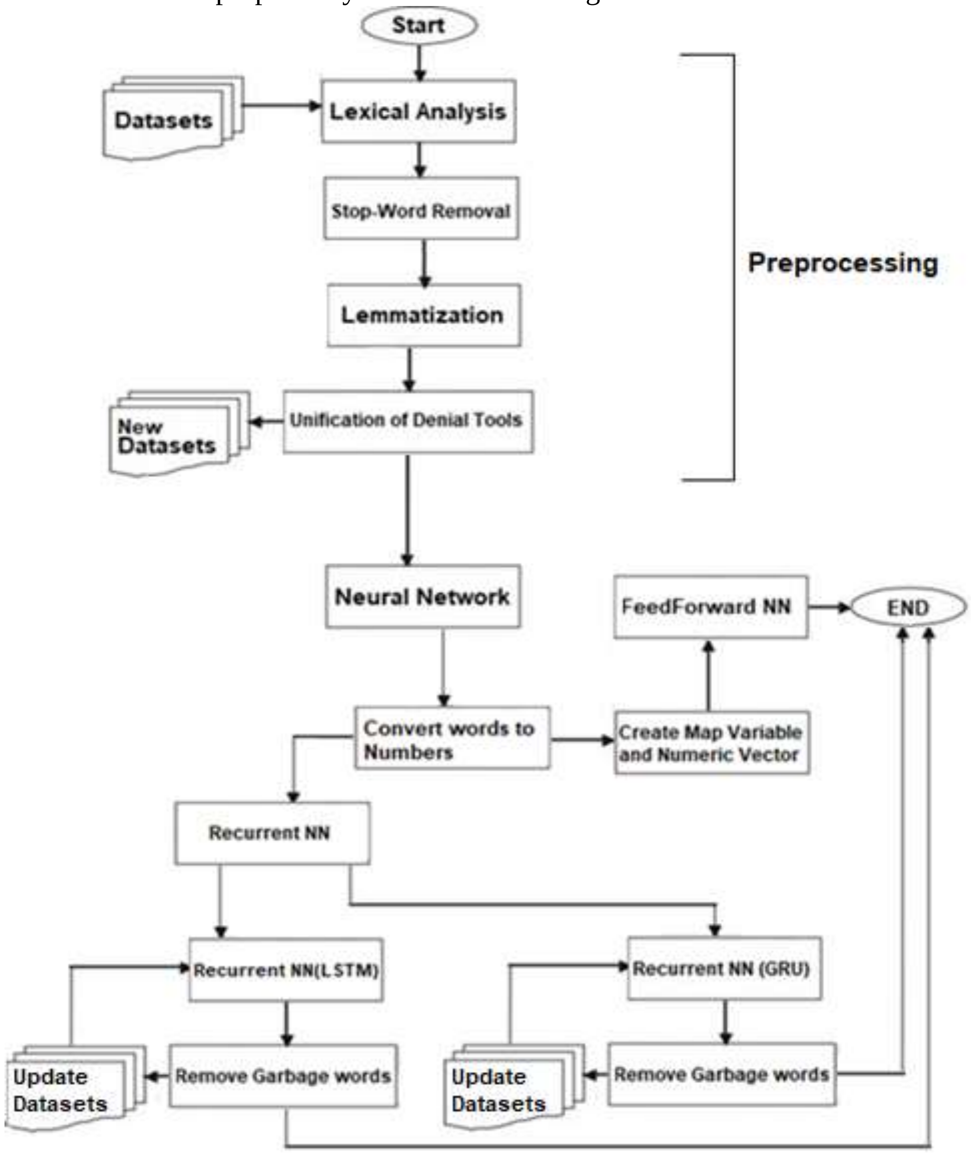


Figure 1- The General Structure of the Proposed System.

2.1. Preparation (Pre-processing) Phase

Operations of preparation are considered as the common essential operations in every data system of mining, in which the system efficiency is based on the files equality applied in forms of separators of word, length of word, noise, etc. The preparation stage includes a series of approaches of manipulation for the text file which are used to prepare the entering data flow to be more suitable for the task of taking out the related information.

2.2. Neural Network Phase

In this section the proposed systems are presented to analyse sentiments by training a set of sentences and testing others to find experimental results by using many types of neural networks. This phase is divided into two section:

2.2.1. Feed-forward Neural Network

One of the general common neural networks that are applied for problems of classification and uniform regression is called feed forward NN. In each layer of this network, the information passes in the forward orientation. The first layer, called the layer of input, accepts the entries; the middle layer, named the hidden layer, is used for performing manipulations and calculations. This determined information is then passed to the layer of output to generate the result.

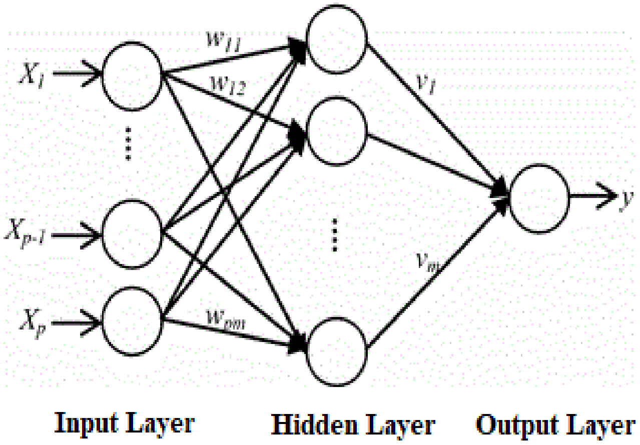


Figure 2-Feed-forward NN.

The activation kind of the equation is applied to differentiate the hyperbolic tangent contiguous for the scale of the entry to hidden points as well to differentiate a function named the logistic sigmoid which is applied to upgrade the output of the hidden signal which transit to points of output [12,13].

$$\text{Logistic sigmoid} = \frac{\mathrm{dF(x)}}{\mathrm{dx}} = \mathrm{F(x)}\big(1 - \mathrm{F(x)}\big) \tag{1}$$

$$\tanh = \frac{\mathrm{dF(x)}}{\mathrm{dx}} = 1 - \mathrm{F(x)}^2 \tag{2}$$

Learning the neural network is the operation of determining a suitable group of weights' amplitudes for points of neural network that enable creating decisions of classification that are very similar to goal values [Sca92]. The wide spread algorithm applied to learn NN of feed-forward is named back-propagation. The oversee training has been applied for purposes of training [14-16].

After the descendant sorting of words that are extracted from dataset , each word is given a unique number or index, then all sentences are converted from a set of words to a set of numbers. After that, some numbers (words) that have a higher occurrence, between 50 to 300 words, are taken and called map variable. The words will be ordered ascendant or descendant to their repetition.

In this experiment, 20% of sentences were used for testing and 80% for training. Feedforward neural networks were created and trained and tested on the training set. The derivative of hyperbolic tangent function was used for weighting the input to hidden nodes, whereas the derivative of logistic sigmoid function (Uni-Polar Sigmoid Function) was used to update the hidden signal output that is sent to output nodes, to train feed-forward neural networks back-propagation used to update and find better weights. To measures error performance, a mean square error (MSE) was used, while visual studio 2010 was used as a platform with C# as programming language.

2.2.2. Recurrent Neural Networks

The recurrent neural network structures extend from completely interconnected to partly connected networks, containing multiple feedforward networks with distinguished layers of input and output [17]. Completely connected networks do not have distinguished input node layers, and every node has sources from all other nodes. Further, it is possible that each node has feedback as shown in Figure-3 [18].

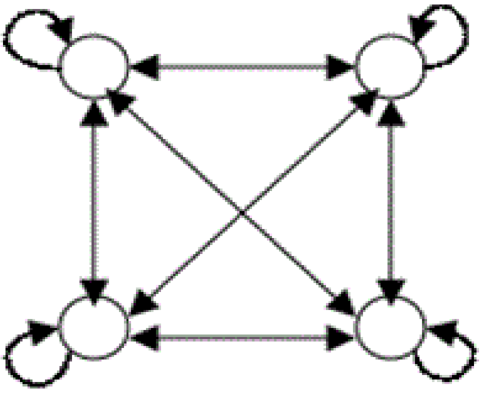


Figure 3- Completely Recurrent Network

The recurrent neural network process a non-fixed length problem of series by having a recurrent hidden layer, the activation of which at every time depends on that of the former time. In general, when there is a series A=(A_1,A_2,…A_t), the state of recurrent hidden named (Z_t) in recurrent neural network makes an update by [19]:

$$Z_t = \begin{cases} 0 & t = 0 \\ \theta(Z_{t-1}, A_t) & otherwise \end{cases} \tag{3}$$

where θ is considered as the function with nonlinearity as an example of logistic composition sigmoid applying affine transformation. As an option, the recurrent neural network could have an output B=(B_1,B_2,…B_t) which also could be non-fixed length [20].

The types of neural networks used in this paper are as follows:

1. **Long Short Term Memory (LSTM)**

LSTM is designed to overcome the difficulties that face traditional neural networks by employing gates. LSTM is characterized by having three gates, namely the input, output, and forget gates. The input gate determines how much the newly computed state for the current input is included, the forget gate determines how much the previous state is included, and finally, the output gate determines how much the internal state should be exposed to the subsequent step. In addition, the LSTM unit has an internal memory that determines how the new enter and the previous memory are combined. Basically, it is a combination of the previous memory multiplied by the forget gate and the newly computed hidden state multiplied by the input gate. The gating mechanism allows the LSTM to learn long-term dependencies. The graphical summarization of LSTM is show in Figure-4.

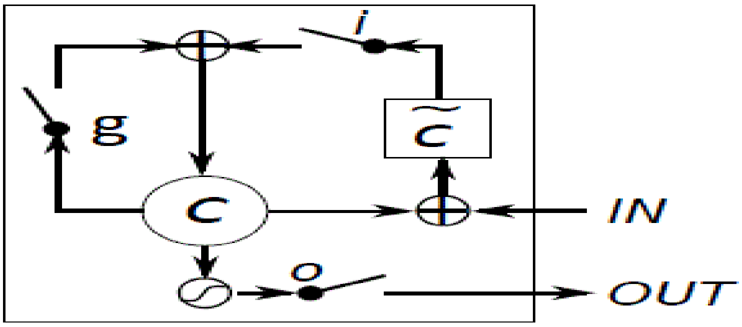


Figure 4- Graphical Summarization of LSTM [20]

Not similar to the unit of recurrent, which easily calculates the sum of weights of the signal that is entered and provides a non-linear function, every unit of jth long short term memory is preserves a memory named C_t^j at a time called t . Then the result Z_t^j or long short term memory activation is calculated by [20-22]:

$$Z_t^j = \alpha_t^j \tanh\left(C_t^j\right) \tag{4}$$

where α_t^jrepresents the gate of output which modulates the memory amount exposure of content. The gate of output is calculated by [20]:

$$\alpha_t^j = \alpha.\left(Y_0 a_t + U_0 z_{t-1} + V_0 c_t\right)^j \tag{5}$$

where α is a function of sigmoid and V_0 is a diagonal matrix. The memory unit C_t^j [21] makes updates by partly removing the available memory and inserting a content of novel memory called $C_t^{\tilde{j}}$.

$$C_t^j = g_t^j c_{t-1}^j + k_t^j c_t^{\tilde{j}} \tag{6}$$

The content of novel memory is:

$$C_t^{\tilde{j}} = \tanh\left(Y_c a_t + U_c z_{t-1}\right)^j \tag{7}$$

The range which can be determined for the forgotten available memory is called the gate of forget, indicated by g_t^j [20]. Also, the degree of content of novel memory that will be inserted to the cell of memory can be modulated with a gate of input called k_t^j [20], as follows:

$$g_t^j = \alpha\left(Y_g a_t + U_g z_{t-1} + V_g c_{t-1}\right)^j \tag{8}$$

$$k_t^j = \alpha\left(Y_k a_t + U_k z_{t-1} + V_k c_{t-1}\right)^j \tag{9}$$

Notice that the two symbols Vg and Vk are considered diagonal matrices. This is not similar to the classical unit of recurrent which replaces its contents every time. The long short term memory has the ability to decide for how long the available memory through the gates remains. In addition, the long short term memory can determine the essential feature of the input series at fast stages. It simply handles that information with a long distance [21].

2. Gated Recurrent Unit (GRU)

Gated recurrent unit is an advanced version of common recurrent networks, depending on the reset gate and update gate. Commonly, there are two directions which determine what data should be delivered to the output. The specific feature of the two vectors is that they can be learned to hold information long time ago, without removing it with time or eliminating information not related to the detection. The update gate defines how much of the previous state is included, and the reset gate applies to the previous hidden state. In short, the reset gate determines how to combine the new input with the previous memory [20]. The graphical summarization of GRU is show in Figure-5.

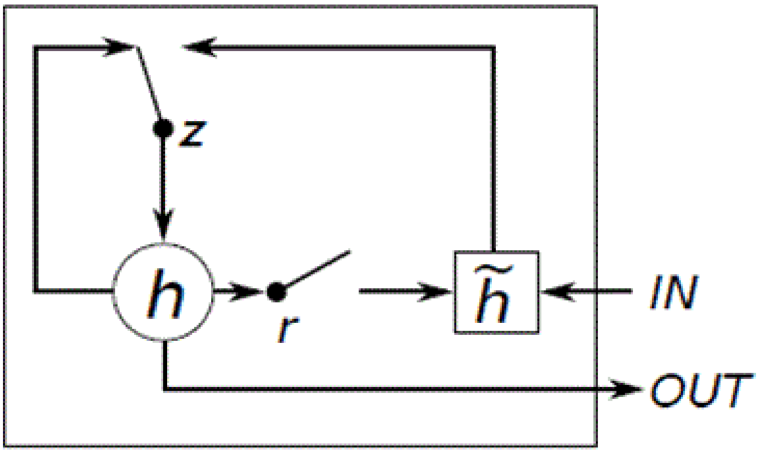


Figure 5- Graphical Summarization of GRU [20]

There is a linear relation between the nominees activation called $\widetilde{h_t^j}$ and the prior activation called $h_{t-1}^j$of GRU activation h_t^j at time t [20].

$$h_t^j = \left(1 - z_t^j\right)h_{t-1}^j + z_t^j \tilde{h}_t^j \tag{10}$$

The z_t^j are called the gates of update which determine the number of times that a unit updates its activation or contents. The gate of update is calculated by [22]:

$$z_t^j = \alpha(W_z a_t + U_z h_{t-1})^j \tag{11}$$

The steps of abtaining a linear addition between the available state and novelty calculated state are the same to the unit of long short term memory. In the unit of gated recurrent, there is no mechanism to monitor the degree to which the state is exposed. The nominee's activation $\tilde{h}_t^j$ is calculated in the same way used in the unit of common recurrent [20, 22], that is:

$$\tilde{h}_t^j = \tanh\big(W a_t + U(r_t \odot h_{t-1})\big)^j \tag{12}$$

where r_t is the gate of reset set and ʘ represents the element-wise multiplication. The benefit provided by the gate of reset is to forget the previously calculated state when reading the initial character of an input series. The way used to calculate the gate of reset is similar to that for the gate of update, as follows [20]:

$$r_j^t = \alpha(W_r a_t + U_r h_{t-1})^j \tag{13}$$

2.3. Remove Garbage Words Phase

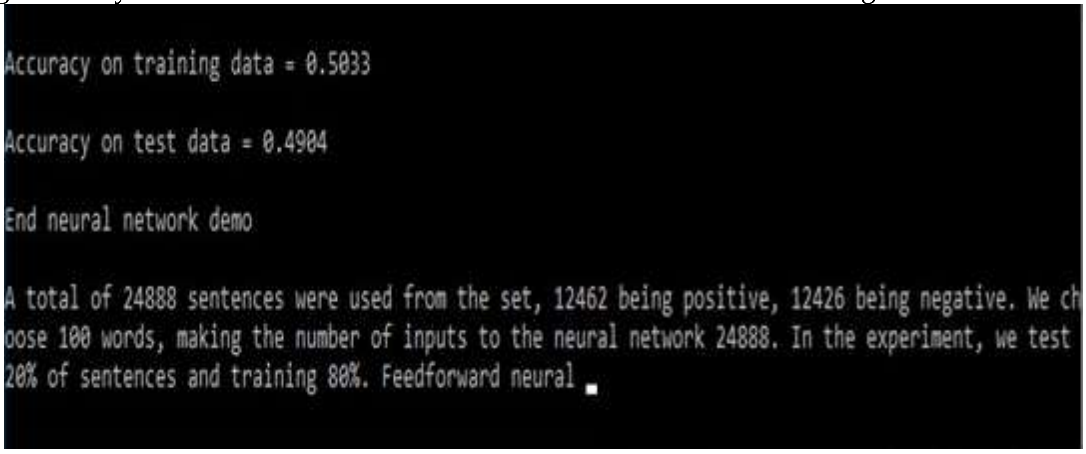
The target of remove garbage words is to minimize the running time by removing the words appearing in low frequency and check if these words affect the accuracy and time or not. The words with low frequency in the set of data will be removed to reduce the size of data set and the implementation time and, at the same time, the accuracy will be monitored to check if it is affected or not.

3. Results

In this section, the results of some conducted tests are presented and discussed to evaluate the performance of the established system. Two datasets are used for training and testing the system proposed in this paper. The first dataset is the movie review with English words and long sentences, which consists of about 25000 sentences divided into 12500 positive and 12500 negative phrases collected from IMDB. The second data set is the Twitter review collected by using keyword search via the Twitter Search API . These tweets are in English words and short sentences, which are about 50,000 sentences.

3.1. Feedforward Results

Movie review dataset can be entered in a feed-forward code. A total of 25000 sentences were used from the set, 12500 being positiveand 12500 being negative. When using 50 words to create the numerical vector, 24842 sentences were produced while the other sentences were empty. When using 100 words, 24888 sentences were produced, whereas using 200 words resulted in 24889 sentences. In the experiment, 50, 150, 100 and 200 neurons, respectively, were created and trained on the training set and tested on the training set, with 0.0001 learning rate and a momentum of 0.0001. In classification accuracy, the accuracy of the training ranged 40% - 53%. Figure-6 shows the results of using 100 words (24888 sentences) and 100 neurons. If a sentence contained only positive or only negative words, then the output neurons would produce results close to 100%, but if a sentence contained a positive and a negative word, the output neurons would be closer to 50%, indicating uncertainty and that the sentence could be classified incorrectly. Limited memory became a problem, because when more sentences were used, the number of words will be increased and there is not enough memory to accommodate the data structures that are needed for training the neural network.



```
Accuracy on training data = 0.5033

Accuracy on test data = 0.4904

End neural network demo

A total of 24888 sentences were used from the set, 12462 being positive, 12426 being negative. We ch
oose 100 words, making the number of inputs to the neural network 24888. In the experiment, we test
20% of sentences and training 80%. Feedforward neural
```

Figure 6- Feed-forward Accuracy of Training and Testing

3.2. GRU and LSTM Results

RNN is one of the most used deep learning techniques to find the sentiment analysis and accuracy of the sentiments. RNN is particularly used in the case of large datasets. Python is used in testing of this work and the output is split using multiple iterations defined under epoch levels for better view of the accuracy at different intervals. The accuracy was found to be around 0.88, which is a good result as a system has been built where any type of text related data can be pushed to get the sentiments and its accuracy.

Table-1 shows the results of the training and testing accuracy of GRU and LSTM for each dataset produced after the garbage words removal operation for the first dataset. The dataset in table-1 represents the dataset of that ratio of the removed words.

Table 1- GRU and LSTM Accuracy of First Dataset

Dataset	GRU Training Accuracy	LSTM Training Accuracy	GRU Testing Accuracy	LSTM Testing Accuracy
None	88%	85%	86%	83%
0%	85%	84%	86%	83%
1%	84%	84%	84%	82%
2%	83%	83%	83%	82%
3%	82%	80%	83%	80%
4%	81%	79%	79%	79%
5%	79%	77%	80%	78%
6%	78%	75%	77%	77%
7%	77%	74%	76%	77%
8%	74%	73%	74%	75%
9%	74%	72%	74%	74%
10%	73%	71%	73%	73%
11%	73%	70%	74%	74%
12%	73%	69%	73%	73%
13%	72%	69%	73%	71%
14%	71%	70%	72%	70%
15%	71%	68%	72%	70%
16%	72%	67%	72%	69%
17%	71%	65%	72%	66%
18%	71%	64%	71%	65%
19%	69%	65%	70%	64%
20%	70%	64%	68%	63%
21%	68%	63%	69%	62%

Table-2 shows the results of training and testing accuracy of GRU and LSTM for each dataset produced after garbage words removal operation for the second dataset.

Table 2- GRU and LSTM Accuracy of Second Dataset

Dataset	GRU Training Accuracy	LSTM Training Accuracy	GRU Testing Accuracy	LSTM Testing Accuracy
100%	86%	85%	78%	81%
99%	86%	85%	77%	81%
98%	85%	84%	78%	81%
97%	86%	84%	78%	81%
96%	86%	83%	78%	80%
95%	86%	83%	78%	81%
94%	86%	83%	77%	81%
93%	86%	83%	78%	80%
92%	86%	83%	77%	79%
91%	85%	83%	75%	79%
90%	85%	83%	77%	79%
89%	86%	82%	77%	78%
88%	86%	82%	78%	78%
87%	85%	81%	77%	78%
86%	85%	82%	77%	78%
85%	85%	81%	77%	78%
84%	85%	82%	78%	77%
83%	85%	81%	77%	76%
82%	85%	80%	77%	75%
81%	85%	81%	78%	76%
80%	85%	80%	78%	75%
79%	85%	80%	77%	75%
78%	85%	80%	77%	75%
77%	85%	80%	78%	74%
76%	85%	80%	78%	74%
75%	85%	79%	78%	72%
74%	84%	80%	76%	73%
73%	85%	78%	76%	74%
72%	85%	78%	78%	76%
71%	84%	78%	78%	75%
70%	85%	77%	77%	73%
69%	85%	78%	77%	74%
68%	85%	76%	77%	73%
67%	85%	76%	71%	74%
66%	84%	76%	76%	73%
65%	85%	75%	75%	73%
64%	84%	76%	78%	73%
63%	85%	76%	76%	72%
62%	84%	75%	78%	72%
61%	84%	76%	76%	72%
60%	84%	74%	77%	74%
59%	83%	74%	73%	73%
58%	84%	74%	78%	74%
57%	84%	75%	78%	73%

56%	84%	72%	78%	74%
55%	84%	73%	77%	73%
54%	85%	73%	78%	73%
53%	84%	74%	78%	72%
52%	84%	73%	78%	72%
51%	84%	73%	78%	72%
50%	84%	72%	77%	71%
49%	84%	73%	78%	73%
48%	84%	73%	77%	72%
47%	85%	74%	75%	70%
42%	85%	74%	77%	72%
41%	84%	74%	77%	72%
40%	84%	74%	78%	70%
39%	83%	75%	77%	71%
38%	83%	75%	77%	70%
37%	83%	75%	77%	71%
36%	84%	75%	78%	71%
35%	84%	73%	77%	71%
34%	83%	74%	77%	72%
33%	83%	72%	78%	70%
32%	83%	73%	78%	71%
31%	83%	73%	77%	70%
30%	83%	73%	77%	70%
29%	83%	73%	78%	69%
28%	83%	73%	78%	69%
27%	83%	73%	78%	69%
26%	83%	73%	76%	68%
25%	82%	73%	76%	67%
24%	83%	73%	78%	68%
23%	83%	73%	78%	67%
22%	82%	72%	77%	66%
21%	82%	72%	77%	67%
20%	83%	72%	77%	66%
19%	83%	72%	71%	65%
18%	83%	73%	76%	65%
17%	82%	72%	75%	64%
16%	82%	72%	78%	64%
15%	82%	72%	76%	64%
14%	82%	72%	78%	63%
13%	81%	73%	76%	63%
12%	82%	72%	77%	63%
11%	82%	71%	73%	63%
10%	81%	71%	78%	62%

4. Result Analysis

a. Differences Between Feed-Forward and Recurrent NN

Results in section 3 showed that the traditional neural network is unable to train on a large dataset like that used in this work because of its inability to save information at a long term, therefore a loss in efficiency occurs. The accuracy of training was about 50% to 53%, which is very low because the polarity is negative or positive, and so, the probability of truth is already 50%. While the results of GRU and LSTM units showed the best effects in the two learning datasets. Deep learning principle of GRU and LSTM units can solve the memory issue because of gates called 'update gates' which determine the number of times that a unit makes its updates of activation, or updates its contents, thus storing information. GRU and LSTM units maintain the present content and add the new content on top of it.

b. Differences Between GRU and LSTM

The differences between accuracies for the two methods (GRU and LSTM) are shown in Figures- 7 and 8 for the first and second datasets, respectively.

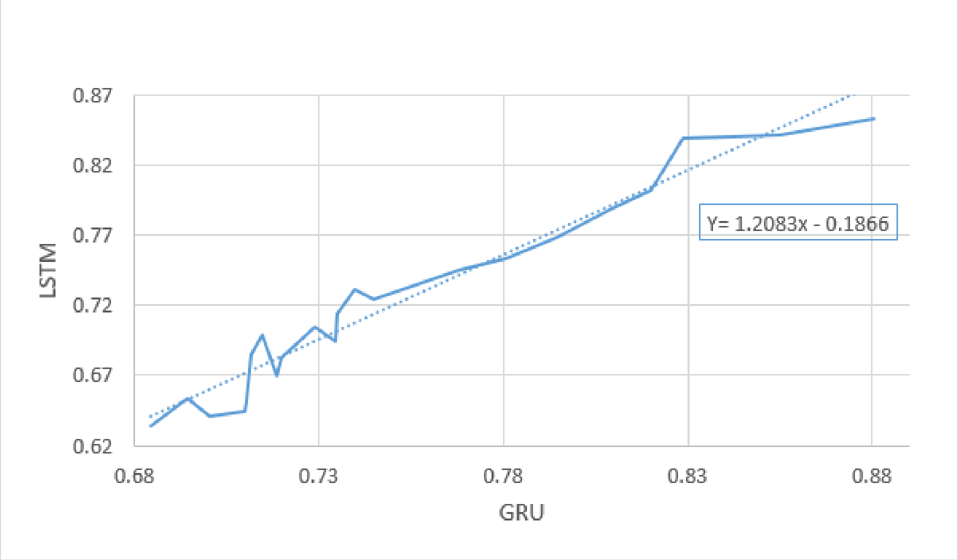


Figure 7- Differences between (GRU & LSTM) for 1st Dataset

For the first dataset, the differences re summarized in equation (14) which explains that the y is stronger than x by 1.2083, where x-axis represents GRU accuracies and y-axis represents LSTM accuracies.

$$Y = 1.2083x - 0.1866 \tag{14}$$

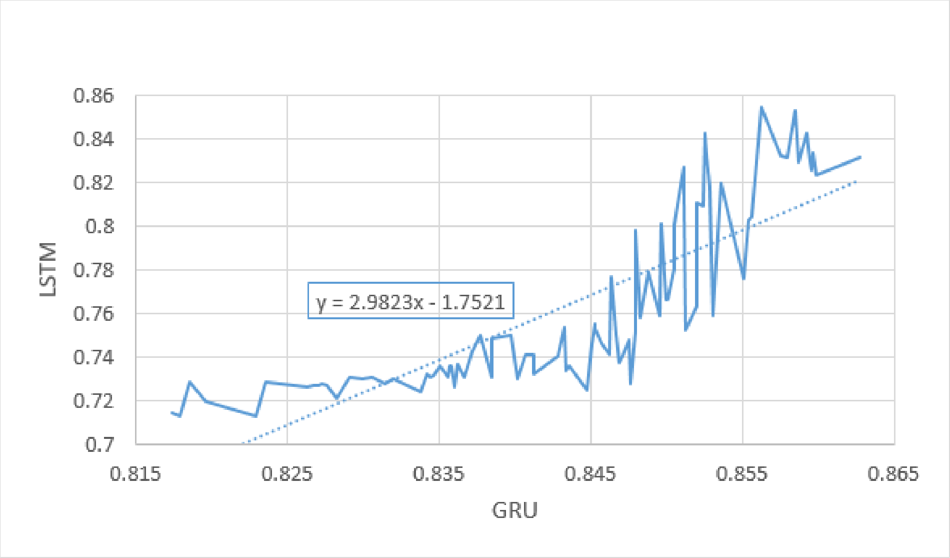


Figure 8- Differences between (GRU & LSTM) for 2nd Dataset

For the third dataset, the differences are summarized in equation (15) hich explains that y is stronger than x by 2.9823, where x-axis represents GRU accuracies and y-axis represents LSTM accuracies.

$$Y = 2.9823x - 1.7521 \tag{15}$$

5. Conclusions and Future Work

In this paper, a system architecture was presented that can be trained on short and long text sentiment analysis sentiment data and tested using different data sets, with different sizes, that contain text files (.txt). Many conclusions have been drawn from the obtained results for the proposed system. In this paper, two approaches have been proposed to analyse the sentiment. Firstly, the performances of the proposed approaches were tested against the traditional classifiers of Feedforward NN. Secondly, the recurrent neural network (GRU, LSTM) models uses recurrent neural network layers to extract effective features. Therefore, a discovering analysis of sentiment can highly promote deep learning effect. The results showed that the traditional neural network is unable to train on large datasets like those used in this work because of their inability to save information at a long term, therefore loss in efficiency occurs. The results showed that GRU and LSTM units have the best effects in the two learning datasets. Deep learning principle of GRU and LSTM units can solve the memory and storing information issues. Afterwards, we highlighted the importance of focusing on the key information of an input sequence from the word-feature level by getting rid of words that appeared slightly (garbage words). In the first dataset that consists of large sentences, removal of garbage words reduced the accuracy by about 18% **-** 22%, whereas in the second dataset that consists of small tweets, the accuracy was reduced by about 4%. This is useful for the social media data because when the garbage words are removed, the size will be reduced, so that the speed of analysing will be increased. This effect takes place because the text data has taken a large size and therefore need more time for analysis.

For future works, the processing of individual words used can be substituted by double or tripartite words that involve many tasks such as understanding the relations between words and determining the decision words that have most occurrences in positive or negative sentences. Also, developing the system can be achieved by using the synonyms system (like WordNet in python). In order to tackle the memory space issue, words that have the same meaning were combined. Using the indexing tree the required words, and then the synonyms words, were determined. In addition, recurrent neural networks could be merged with Convolutional Neural Networks (CNNs) to obtain better results.

References

1. Kiruthika, M., Woonna, S. and Giri, P. **2016**. Sentiment Analysis of Twitter Data, *International Journal of Innovations in Engineering and Technology (IJIET)*, **4**(6): 264-273.
2. Bilgin, M. And Şentürk, İ. F. **2017**. Sentiment Analysis on Twitter data with Semi-Supervised Doc2Vec, *UBMK* International conference on computer science and engineering, pp :661-666.
3. Kaur, J. And Sidhu, B. K. **2018**. Sentiment Analysis Based on Deep Learning Approaches, *Global journal of engineering science and researches, ISSN*, :1496-1500.
4. Deng, L. and Wiebe, J. **2014**. Sentiment propagation via implicature constraints, Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp:377–385.
5. Bradbury, J., Merity, S., Xiong, C. and Socher, R. **2016**. Quasi-Recurrent Neural Networks, 5th International Conference on Learning Representations.
6. Shuqin Gu, Lipeng Zhang and Yuexian Hou, **2018**. A Position-aware Bidirectional Attention Network for Aspect-level Sentiment Analysis, 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA, pp:774–784.
7. Pal, S., Ghosh, S. and Nag, A. **2018**. Sentiment Analysis in the Light of LSTM Recurrent Neural Networks, *International Journal of Synthetic Emotions, January-June (IJSE),* **9**(1): 33-39.
8. Feng, S., Wang, Y., Liu, L., Wang, D. and Yu, G. **2019**. Attention based hierarchical LSTM network for context-aware microblog sentiment classification, *Springer Science Business Media, LLC, part of Springer Nature*, **22**(1): 59–81.
9. Wang, X., Wu, P., Liu, G., Huang, Q., Hu, X. and Xu, H. **2019**. Learning performance prediction via convolutional GRU and explainable neural networks in e-learning environments, *Springer-Verlag GmbH Austria, part of Springer Nature*. **101**(6): 587–604.
10. Bengio, Y. **2009**. Learning Deep Architectures for AI, *Foundations and Trends in Machine Learning*, **2**(1): 1-127.

11. Collobert, R. **2011**. Deep Learning for Efficient Discriminative Parsing, 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, **15**: 224-232.

12. Stansbury, D. **2014**. Derivatives for Common Neural Network Activation Functions, *the clever machine web*.

13. Feng Q. **2015**. Derivatives of tangent function and tangent numbers, *Applied Mathematics and Computation, ELSEVIER journal*, pp: 844-858.

14. Scalero, R. S. and Tepedelenlioglu, N. **1992**. A Fast New Algorithm for Training Feedforward Neural Networks, *IEEE Transactions on Signal Processing*, **40**(1): 202-210.

15. Mohammed, S. N.and George, L. E. **2017**. Emotion Recognition Based on Texture Analysis of Facial Expressions Using Wavelets Transform, *Australian Journal of Basic and Applied Sciences*, **11**(9): 1-11.

16. Knight, K. **1990**. Connectionist ideas and algorithms, *Communications of the ACM*, **33**(11).

17. Bengio, Y., Boulanger-Lewandowski, N. and Pascanu, R. **2012**. Advances in Optimizing Recurrent Networks, IEEE International Conference on Acoustics, Speech and Signal Processing, pp: 8624-8628.

18. Medsker, L. And Jain, L. C. **2001**. *Recurrent Neural Network design and application*, Departments of Physics and Computer Science and Information Systems, American University, Washington, D.C., CRC Press.

19. Prabowo, R. and Thelwall, M. **2009**. Sentiment analysis: A combined approach, *Journal of Informetrics*, **3**(2): 143-157.

20. Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. **2014**. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Universit´e de Montr´eal, CIFAR Senior Fellow.

21. Tai, K. S., Socher, R. and Manning, C. D. **2015**. Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks, Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp: 1556–1566.

22. Graves, A. **2014**. Generating Sequences with Recurrent Neural Networks. Department of Computer ScienceUniversity of Toronto.