The Use of Gated Recurrent Unit with First Order Probability
for Sentiment Analysis

Samar Khudair Abbas¹, Loay E. George²

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

Corresponding Author Email: smora.star88@gmail.com

https://doi.org/10.26782/jmcms.2019.08.00035

Abstract

Sentiment analysis is one of the recent important subjects in classification
filed that recently growing using deep learning. With the spread use of internet, many
rising social media, known forums, survey sites, as well as a lot of bloggers produce
massive amount of information in shape of customer sentimental assessments, feelings,
point of view, debate, opinion around various social news, products, trademark, and
protocols, videos etc. Text analysis is an important subject for any system that deals
with strings to extract the useful data. In this paper, the effective methods of deep
learning will been applied for job with sentiment analysis to get rid of the text
analyzing problems and applied some solutions to these problems, by using recurrent
neural network (Gated Recurrent Unit (GRU)). In addition, noisy words will been
removed to reduce the search space. In order to test the system performance, a set of
tests was applied on three datasets. The first and second datasets are collected data
from IMDB that consist of movie reviews expressed through long sentences of English,
and the third dataset is collection of twitter using the Twitter Search API to collect
these tweets by using keyword search, these tweets in English words with short
sentences. The conducted tests on the developed system gave accuracy that range 88%
- 68%, and the time will been reduced with percentage about 89% when compared
with the results of other newly published works. Experimental results on Datasets
demonstrate that our proposed models can learn effective features and obtain superior
performance over the baseline models.

Keywords: Gated recurrent unit, Deep Learning, Recurrent neural network,
Sentiment analysis.

I. Introduction

With the spread use of internet especially pages of social media, an unusual
quantity of information is founded and of attraction to number of study field which
include psychology, entertainment, sociology, business, news, politics, and another
cultural fields of nations. Data mining providing into social media allows producing
enjoyable scene on behavior of human and interaction of human. The system of data
mining applied in connection with the pages of social media give clear realizations
about nation opinions in any subject of life, determine people groups among population masses, analyze changes of study group with the time, locate effective people as well as recommend activity to the peoples.

The process which is used for extracting subjective information dynamically form the text named sentiment analysis. The information extracted include the target identifying (such as measure and a product), the polarity (negative or positive), stance (advantages and disadvantages), or suggestion holder [XIII]. The extracted information also includes indications about effects analyzing as well as calculation of the target aspects which contain what the people dislike or like [VII][IX]. Based on the implementation, it is possible there is grained coarse such as negative and positive. At present, analysis of sentiment is a subject of large benefit and evaluation since it has too many practical implementations. With a assistance of systems used with sentiment analysis, this not structured instructions could be dynamically converted to the structured information of general opinions about brands, services, products, protocols, or every subject that can people talk about it [VII]. 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 [VII].

The previous work [II] introduced a system of recurrent neural network which called quasi recurrent neural network can described as a method to modeling neural order as alternative to classical layers applied through time steps in parallel as well as function of minimalist recurrent through channels in parallel. In spite of missing layers of trainable QRNN showed higher quality of accuracy than long short term memory with similar hidden dimension. [XXII] Proposed a bidirectional network Bi-GRU. This network not just focuses on information of position for aspect terms. It is 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 strengthen of suggested PBAN network. The basic theory of PBAN network is to establish aspect terms position implant to determining the weights of attention. [XVIII] suggested various techniques of LSTM structures for analysis of sentiment with review of movies. The results obtained indicate that method of LSTM RNN gives effective performance than classical RNN and deep neural networks for analysis of sentiment. And used simple models of LSTM and evaluates their performances then layers of LSTM added one after one which gives increment for the accuracy. Finally bidirectional layers of LSTM established to cover information in network with forward and backward. Also [VIII] suggested network named CA-LSTM to combine previous tweets for classification of sentiment. The networks of context Attention based long short-term memory depend on hierarchal framework for simulate the sequence of microblog and determines the tweets and words with various weights utilizing mechanism of attention. [XXV] Proposed a mechanism to creating a model capable of predicting performance of learning, extracting feature of learning, and reasoning of results. Initially common feature of learning verification approach established for convert the raw of data from systems of e-learning to groups of separate features of learning. They submitted developed parallel neural network to display the results of prediction.

In this paper, deep learning neural network is adopted to solve the classification problems that related to social data, by using GRU recurrent neural network which is characterized by specifications of deep learning. deep learning is one of the techniques
in machine learning that determine multiple layers of non-linear information manipulated for extraction of features supervised, as well as classification and pattern analysis [III]. 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 [IV]. In addition, this paper showed the process of elimination of words that appeared slightly to get rid of large size and noisy words. The practical results indicate very well performance comparing to traditional approaches.

II. Materials and Methods

The proposed system in this paper, a method for enhanced text analysis system by using deep learning methods, the overall design of the proposed system shown in figure (1); as shown in the figure, the system consists of two stages: (1) preprocessing Analysis, (2) Training and testing stage. The input of this system is a text files with various sizes and can be taken from many resources. Text preprocessing is the important step within the text mining process and performs a chief role in textual mining strategies [XXIV]. The extraction of important meaningful words is important as well as the removal of irrelevant symbols and words that are not associated with natural language [XVII]. This implies the conversion of original textual data in a structure ready for data-mining [XX].

Figure 1: The General Structure of the Proposed System
In order to minimize the dimensionality of the files' words, special strategies such as cleaning and filtering are applied. Cleaning and filtering methods remove a phrases from the set of all words, which do no longer provide relevant information; stop-word filtering is a standard filtering approach. Words like prepositions, punctuation marks, numbers, and so on are eliminated that contain no informatics as such. [XVII]. Very often, the reviews have such punctuations and words that are useless for classification. The precision of classification is affected due to these elements, thus, pre-processing of data facilitates in removing such unnecessary elements. This process involves the following operations [XI]:

- Lexical Analysis:
- Tokenization
- Convert to Lower Case
- Punctuation Elimination
- Digits Elimination
- Tags Elimination
- Stop-words Removal
- Stemming (Lemmatization)
- Unification of Denial Words
- Finding Words' Frequency

**Lexical Analysis**

It is the process of dividing characters stream right into a listing of phrases or tokens. Lexical Analysis consist of many stages, tokenization is the first stage in any data retrieval systems and query processing includes preprocessing of giving files; it generates respective tokens [XXIII]. The second stage is convert case of the letters to lower case, then the following four particular cases have to be considered with care: digits, hyphens, punctuation marks, tags of HTML. Lexical analysis produced tokens that are parsed and was an internal representation appropriate for assessment with indexes [XXIII] [X].

**Stop-words Removal**

The words that have a very low discrimination value in the all information retrieval structures (e.g. prepositions, interjection and conjunctions, helping verbs) are commonly known as a stop-words, or noise phrases [XXI]. It means that the amount of data by these words isn't valuable. Wherefore, most systems are typically worthwhile to ignore all of these words in an indexing files procedure and processing the queries [XIV]. For this reason, a stop-list is accrued from the phrases within the files that ought to be filtered, since they have no significant meaning in these files [XV].

**Stemming Methods**

Frequently, in indexing and search systems are generally used word stemming aspect to improve the recall by exchanging each phrase with its basic word or root at the time of indexing and searching [Meg13]. This method is used to find out the basis/stem of a word. for example, the words user, users, used, using all can be stemmed to the word “use”. The cause of this technique is to take away various suffixes, to reduce
number of phrases, to have exactly matching stems, to keep reminiscence space and 
time [XX].

The drawback of stemming approach can be summarized as following [XII] [XX]:
1. The stems produced are not always real words.
2. It has at least five steps and sixty rules and hence is time consuming.
3. It is not possible to match something with nothing.
4. Not in all case a suffix should be removed.

Because of this weakness, the adopted stemming operation used can be replaced by 
another one, lemmatization operation, the basic function of both methods 'Stemming' 
and 'Lemmatization' is convert a word to its basic or root [XXIV]. There is the 
difference between both, in stemming the ‘stem’ is obtaining after applying rules used 
of a hard and fast of guidelines but without bothering about the part of speech (POS) 
or the context of the phrase occurrence. In lemmatizing the ‘lemma’ is obtaining a 
word which involves reducing the word forms to its root after understand the POS and 
the context of the phrase within the given sentence [XII].

**Unification of Denial Tools**

Denial words generally refers to a words that have a suffix or prefix turned its 
meaning to a negation words, and frequently consists of the some letter in beginning 
or tail of word. Also the helping verbs that merged with 'not' in tail of it, such as 
('haven't', 'isn't', 'wouldn't', etc.) considered as denial words. Instead of existing more 
than one word they have the same meaning; let it be denial, and instead of appear the 
word slightly and consider it as garbage or noisy word can united in one word. This 
helps to reduce the number of words, so reduce of time consumed. The outcomes are 
unified forms of phrases, usually aiming to cast off denial letters and to convert it to 
unified word.

**Recurrent Neural Networks (RNN)**

The recurrent neural networks structures extend from completely interconnected to 
partly connected networks, containing multiple feedforward networks with 
distinguished layers of input and output [I]. Completely connected networks do not 
have distinguished input nodes layers, and every node has sources from all other 
nodes. As well as it is possible that each node has feedback as shown in figure 2 
[XVI].

![Figure 2: Completely Recurrent Network](image-url)
The recurrent neural network processed the non-fixed length series by having a state of recurrent hidden which its activation at every time is depend on that of the former time. In general, when there is a series \( A = (A_1, A_2, \ldots, A_t) \), the state of recurrent hidden named \( Z_t \) in recurrent neural network make update by (XIX):

\[
Z_t = \begin{cases} 
0 & \text{if } t = 0 \\
\theta(Z_{t-1}, A_t) & \text{otherwise}
\end{cases}
\]

Where \( \theta \) consider function with nonlinearity as example logistic composition sigmoid applying affine transformation. As an option the recurrent neural network could be has output \( B = (B_1, B_2, \ldots, B_t) \) which also could be non-fixed length [V].

**Gated Recurrent Unit (GRU)**

This approach named (GRU) was supposed by Cho in 2014 [V] that its principle depends on making every unit of recurrent capture adaptively the dependencies of various scales of time [VI]. In similar to the unit of long short term memory, the gated recurrent modulate the information flow of interior unit by units of gating without using individual cells of memory. There is linear relation between the nominees activation called \( \tilde{h}_t \) and the prior activation called \( h_{t-1} \) of GRU activation \( h_t \) at time \( t \) [V]:

\[
\tilde{h}_t = (1 - z_t^j)h_{t-1}^j + z_t^j\tilde{h}_t^j
\]

The \( z_t^j \) called gates of update which determine number of times that unit make its updates of activation, or update its contents [VI]. The gate of update is calculated by:

\[
z_t^j = \alpha(W_a a_t + U_a h_{t-1})^j
\]

The steps of getting a linear addition between the available state and novelty calculated state is same to the unit of long short term memory. In the unit of gated recurrent there is no mechanism to monitor the degree which state is exposed. The nominee’s activation \( \tilde{h}_t \) is calculated in the same way that used in the unit of common recurrent [V] [VI]:

\[
\tilde{h}_t = \tanh(W_a a_t + U_a (r_t \tilde{h}_{t-1}))^j
\]

The \( r_t \) gate of reset set, \( \odot \) represent multiplication of element wise. The benefit provided by gate of reset is to forget the prior calculated state when reading the initial character of an input series. The way used to calculate gate of reset similar to the gate of update as follow [V]:

\[
r_t^j = \alpha(W_r a_t + U_r h_{t-1})^j
\]

It can be noticed the graphical summarization of GRU in figures 3 [V]:

---

Copyright reserved © J. Mech. Cont.& Math. Sci.  
Samar Khudair Abbas et al
Remove Garbage Words (Least Redundant Words)

The main target of remove garbage words is to minimize time of running by remove the words that appear few and check if these words effect on accuracy and time or not. The words that appeared a little in the set of data will be removed to reduce the size of data set and implementation time and in same time the accuracy will be monitored to check if it effect or not. The methods used for removing these words are as follow:

1. For the first and second movie reviews datasets (without lemmatization and with lemmatization) the eliminating depend on calculating the number of words based on frequency percentage that states number of sentences that this word appeared divided by number of all sentences in the data set multiplied by 100. After the removal a new file is produced that contain all words except the words have specific percentage.
2. For twitter dataset : The eliminating method of this data is done by selecting specific percentage of words that have lower repetition ordered descending based on frequency. Finally 99% of words will been selected and created new file contained it depend on higher repetition and eliminating 1% of lower words. After the removal a new file is produced that contain all words except the words have specific percentage.

III. Results

In this section, the results of some conducted tests are presented and discussed to evaluate the performance of the established system. The datasets used for training and testing the system that proposed in this paper is three datasets, firstly movie review with English words and long sentences, the file size [31.6 MB] which about 25000 sentences divided into 12500 positive phrases and 12500 negative phrases collected from IMDB, this dataset divided into two datasets the first one without lemmatization and second one with lemmatization. The third data set is Twitter review collected by using the Twitter Search API by using keyword search, these tweets in English words and short sentences the file size [3.94 MB] which about 50,000 sentences. The used training set consists of about 60% samples taken, randomly, and the remaining samples have been treated as testing samples.

Table 1 shows the training and testing accuracy of GRU results for each dataset produced after garbage words removal operation for first and second datasets. The dataset of percentage in table 1 represents the dataset of that ratio of removed words.
### Table 1: GRU Accuracy of 1st and 2nd Dataset

| Dataset of Percentage | GRU Training Acc. 1st Data | GRU Testing Acc. 1st Data | GRU Training Acc. 2nd Data | GRU Testing Acc. 2nd Data |
|-----------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| None                  | 0.8804                     | 0.8644                     | 0.877                      | 0.8561                     |
| 0.1%                  | 0.8549                     | 0.8608                     | 0.8521                     | 0.8606                     |
| 1.1%                  | 0.8425                     | 0.8409                     | 0.8457                     | 0.8544                     |
| 2.1%                  | 0.8287                     | 0.8352                     | 0.8359                     | 0.8409                     |
| 3.1%                  | 0.8201                     | 0.8258                     | 0.8189                     | 0.8242                     |
| 4.1%                  | 0.8097                     | 0.7918                     | 0.8115                     | 0.8031                     |
| 5.1%                  | 0.7943                     | 0.7966                     | 0.8056                     | 0.8149                     |
| 6.1%                  | 0.7807                     | 0.7734                     | 0.7929                     | 0.7974                     |
| 7.1%                  | 0.7683                     | 0.7666                     | 0.7789                     | 0.7887                     |
| 8.1%                  | 0.7397                     | 0.7441                     | 0.7669                     | 0.7676                     |
| 9.1%                  | 0.7447                     | 0.7447                     | 0.7514                     | 0.7522                     |
| 10.1%                 | 0.7347                     | 0.7358                     | 0.748                      | 0.7462                     |
| 11.1%                 | 0.7288                     | 0.7376                     | 0.7351                     | 0.7383                     |
| 12.1%                 | 0.7345                     | 0.7355                     | 0.7289                     | 0.7413                     |
| 13.1%                 | 0.7197                     | 0.7302                     | 0.7351                     | 0.7339                     |
| 14.1%                 | 0.7148                     | 0.7235                     | 0.7261                     | 0.7297                     |
| 15.1%                 | 0.7118                     | 0.7235                     | 0.7275                     | 0.7296                     |
| 16.1%                 | 0.7186                     | 0.7199                     | 0.7203                     | 0.7317                     |
| 17.1%                 | 0.7105                     | 0.7192                     | 0.7182                     | 0.7282                     |
| 18.1%                 | 0.71                     | 0.71                      | 0.7154                     | 0.7174                     |
| 19.1%                 | 0.6945                     | 0.7009                     | 0.7149                     | 0.7192                     |
| 20.1%                 | 0.7002                     | 0.6859                     | 0.7077                     | 0.7068                     |
| 21.1%                 | 0.6845                     | 0.6917                     | 0.7111                     | 0.6986                     |

Table 2 shows the training and testing accuracy of GRU results for each dataset produced after garbage words removal operation for third datasets. The dataset of percentage in table 2 represents the dataset of that remaining words.

### Table 2: GRU Accuracy of 3rd Dataset

| Data Percentage | GRU Training Acc. 3rd Data | GRU Testing Acc. 3rd Data |
|-----------------|----------------------------|---------------------------|
| 100%            | 0.8562                     | 0.78372                   |
| 99%             | 0.8585                     | 0.775962                  |
| 98%             | 0.8525                     | 0.780362                  |
| 97%             | 0.8592                     | 0.781393                  |
| 96%             | 0.8575                     | 0.7844                    |
| 95%             | 0.8579                     | 0.781883                  |
| 94%             | 0.8596                     | 0.77726                   |
| 93%             | 0.8587                     | 0.780758                  |

| Data Percentage | GRU Training Acc. 3rd Data | GRU Testing Acc. 3rd Data |
|-----------------|----------------------------|---------------------------|
| 56%             | 0.8434                     | 0.777933                  |
| 55%             | 0.8462                     | 0.773522                  |
| 54%             | 0.8469                     | 0.771885                  |
| 53%             | 0.8401                     | 0.781785                  |
| 52%             | 0.8476                     | 0.77673                   |
| 51%             | 0.8447                     | 0.77619                   |
| 50%             | 0.8436                     | 0.771616                  |
| 49%             | 0.8428                     | 0.785966                  |

*Copyright reserved © J. Mech. Cont.& Math. Sci.*

*Samar Khudair Abbas et al*
To evaluate the performance of the system, the processing time is another important parameter; tables 3 show the time consumed for each dataset of 1st and 2nd datasets respectively in minutes. Taking into account that the tests have been conducted using a laptop computer (has Processor: Intel ® Core (TM) i7-6500; CPU @ 2.50 GHz; 8 GB RAM, x64-based processor). The dataset of percentage in table 3 represents the dataset of that ratio of removed words.

| Ratio (%) | Time 1st Dataset (Minute) | Time 2nd Dataset (Minute) |
|-----------|---------------------------|---------------------------|
| 92%       | 0.8627                    | 0.776424                  |
| 91%       | 0.8526                    | 0.756619                  |
| 90%       | 0.8594                    | 0.778708                  |
| 89%       | 0.8599                    | 0.782246                  |
| 88%       | 0.8595                    | 0.777193                  |
| 87%       | 0.8552                    | 0.776128                  |
| 86%       | 0.8536                    | 0.777411                  |
| 85%       | 0.8511                    | 0.775905                  |
| 84%       | 0.8511                    | 0.785374                  |
| 83%       | 0.8528                    | 0.778731                  |
| 82%       | 0.8556                    | 0.778419                  |
| 81%       | 0.8524                    | 0.783174                  |
| 80%       | 0.8496                    | 0.780052                  |
| 79%       | 0.8505                    | 0.776992                  |
| 78%       | 0.8554                    | 0.775665                  |
| 77%       | 0.852                     | 0.782377                  |
| 76%       | 0.8479                    | 0.781012                  |
| 75%       | 0.852                     | 0.781653                  |
| 74%       | 0.8463                    | 0.763826                  |
| 73%       | 0.8505                    | 0.761054                  |
| 72%       | 0.8551                    | 0.781146                  |
| 71%       | 0.8488                    | 0.78215                   |
| 70%       | 0.85                      | 0.778977                  |
| 69%       | 0.852                     | 0.775201                  |
| 68%       | 0.853                     | 0.778101                  |
| 67%       | 0.8512                    | 0.711331                  |
| 66%       | 0.8483                    | 0.762928                  |
| 65%       | 0.8501                    | 0.755749                  |
| 64%       | 0.8479                    | 0.781404                  |
| 63%       | 0.8495                    | 0.765376                  |
| 62%       | 0.8475                    | 0.780607                  |
| 61%       | 0.8407                    | 0.766071                  |
| 60%       | 0.8457                    | 0.771333                  |
| 59%       | 0.8385                    | 0.734656                  |
| 58%       | 0.8447                    | 0.782245                  |
| 57%       | 0.8412                    | 0.789248                  |

Copyright reserved © J. Mech. Cont.& Math. Sci.  
Samar Khudair Abbas et al
Table 3: Time Consumed of 1\textsuperscript{st} and 2\textsuperscript{nd} Dataset (in Minute)

| Dataset of Percentage | GRU Time 1\textsuperscript{st} Data | GRU Time 2\textsuperscript{nd} Data |
|----------------------|-------------------------------------|-------------------------------------|
| None                 | 126.5                               | 126.5                               |
| 0.\%                 | 104.1833                            | 113.1                               |
| 1.\%                 | 93.51667                            | 78.86667                            |
| 2.\%                 | 82.15                               | 65.83333                            |
| 3.\%                 | 76.11667                            | 58.45                               |
| 4.\%                 | 67.11667                            | 52.06667                            |
| 5.\%                 | 57.95                               | 49.33333                            |
| 6.\%                 | 48.58333                            | 41.83333                            |
| 7.\%                 | 45.41667                            | 40.41667                            |
| 8.\%                 | 36.91667                            | 38.48333                            |
| 9.\%                 | 32.78333                            | 36.88333                            |
| 10.\%                | 34.11667                            | 31.95                               |
| 11.\%                | 29.65                               | 26.48333                            |
| 12.\%                | 27.38333                            | 26.58333                            |
| 13.\%                | 24.18333                            | 21.81667                            |
| 14.\%                | 21.6                                | 20.16667                            |
| 15.\%                | 21.58333                            | 19.25                               |
| 16.\%                | 20.03333                            | 17.55                               |
| 17.\%                | 20.43333                            | 17.76667                            |
| 18.\%                | 19.1                                | 16.43333                            |
| 19.\%                | 18.66667                            | 16.25                               |
| 20.\%                | 17.98333                            | 16.43333                            |
| 21.\%                | 17.06667                            | 13.83333                            |

Tables 4 show the time consumed for each dataset of 3\textsuperscript{rd} dataset in. The dataset of percentage in table 4 represents the dataset of that ratio of remaining words.

Table 4: Time Consumed of 3\textsuperscript{rd} Dataset (in Minute)

| Dataset of Percentage | GRU Time 3\textsuperscript{rd} Data | Dataset of Percentage | GRU Time 3\textsuperscript{rd} Data |
|----------------------|-------------------------------------|----------------------|-------------------------------------|
| 100\%                | 8.139504                            | 54\%                 | 4.763527                            |
| 99\%                 | 7.981029                            | 53\%                 | 4.583538                            |
| 98\%                 | 8.082238                            | 52\%                 | 4.666637                            |
| 97\%                 | 8.282003                            | 51\%                 | 4.597048                            |
| 96\%                 | 5.930344                            | 50\%                 | 4.430364                            |
| 95\%                 | 5.760152                            | 49\%                 | 4.537246                            |
| 94\%                 | 5.709746                            | 48\%                 | 4.492881                            |
| 93\%                 | 5.667946                            | 47\%                 | 4.374086                            |
| 92\%                 | 5.820699                            | 46\%                 | 4.349739                            |

Copyright reserved © J. Mech. Cont.& Math. Sci.
Samar Khudair Abbas et al

436
| Ratio (%) | Number of Sentences | Removed Ratio (%) | Empty Sentences |
|----------|---------------------|------------------|-----------------|
| 91%      | 5.643767            | 45%              | 4.326173        |
| 90%      | 5.595628            | 44%              | 4.328973        |
| 89%      | 5.112817            | 43%              | 4.31204         |
| 88%      | 5.119146            | 42%              | 4.316952        |
| 87%      | 5.104542            | 41%              | 4.409951        |
| 86%      | 5.079212            | 40%              | 4.36912         |
| 85%      | 5.068459            | 39%              | 4.328908        |
| 84%      | 5.059908            | 38%              | 4.286956        |
| 83%      | 5.109326            | 37%              | 4.295573        |
| 82%      | 5.117836            | 36%              | 4.252188        |
| 81%      | 5.026651            | 35%              | 4.219904        |
| 80%      | 5.031598            | 34%              | 4.175812        |
| 79%      | 4.974771            | 33%              | 4.152402        |
| 78%      | 4.939246            | 32%              | 4.134827        |
| 77%      | 4.949514            | 31%              | 4.569871        |
| 76%      | 5.553344            | 30%              | 4.378455        |
| 75%      | 5.422154            | 29%              | 4.098703        |
| 74%      | 5.343872            | 28%              | 4.057939        |
| 73%      | 5.318763            | 27%              | 4.126434        |
| 72%      | 5.380766            | 26%              | 4.096098        |
| 71%      | 4.865215            | 25%              | 4.143925        |
| 70%      | 4.971248            | 24%              | 4.275932        |
| 69%      | 4.987035            | 23%              | 4.728256        |
| 68%      | 5.117267            | 22%              | 4.788114        |
| 67%      | 5.371408            | 21%              | 3.969441        |
| 66%      | 4.923472            | 20%              | 4.050618        |
| 65%      | 5.215137            | 19%              | 4.281426        |
| 64%      | 5.152163            | 18%              | 4.358477        |
| 63%      | 5.245404            | 17%              | 4.260552        |
| 62%      | 5.468113            | 16%              | 5.421085        |
| 61%      | 4.69148             | 15%              | 5.355551        |
| 60%      | 5.078749            | 14%              | 3.898795        |
| 59%      | 5.665918            | 13%              | 3.850221        |
| 58%      | 4.710444            | 12%              | 3.785317        |
| 57%      | 4.820066            | 11%              | 3.735814        |
| 56%      | 4.938096            | 10%              | 3.674063        |
| 55%      | 4.847031            |                  |                 |

Figure 4, 5 and 6 shows the number of sentences that not contain any word after removing words, therefore it removed from dataset. The x-axis represents ratio of removed words and y-axis represents number of empty sentences.
Figure 4: Empty Sentences of 1\textsuperscript{st} Dataset

Figure 5: Empty Sentences of 2\textsuperscript{nd} Dataset

Figure 6: Empty Sentences of 3\textsuperscript{rd} Dataset
IV. Conclusion and Future Work

In this paper, a system architecture is presented that can be trained on short and long text sentiment analysis sentiment data and we tested using different data sets with different sizes that containing text files (.txt). Accurately identifying the polarity of sentences and discovering analysis of sentiment can highly promote deep learning effect. In this paper, we have proposed two approaches to analyze the sentiment and explain the predicted results. Afterward, we also highlighted the significance of focusing on the key information of an input sequence from the word-feature level beget rid of words that appeared slightly. The test results showed that the proposed system achieved higher accuracy ranged 88% for training and 86% for testing, and when remove garbage words (i.e., least frequency words) the training accuracy reached to about 68% and about 69% for testing. Experimental results on Datasets demonstrate that our proposed models can learn effective features and obtain superior performance over the baseline models.

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 word that have most occurrences in positive sentences or negative. Also for developing the system, can be using the synonyms system (like WordNet in python). In order to tackle the memory space by combining words that have the same meaning. Using the indexing tree for determining the required word, then determine the synonyms words. In addition the proposed system can be developed to be multilingual system, being able to used sentences in any language desired.

References

I. Bengio, Y., Boulanger-Lewandowski, N., and Pascanu, R., “Advances in Optimizing Recurrent Networks”, IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8624-8628, 2013.

II. Bradbury, J., Merity, S., Xiong, C., & Socher, R., “Quasi-Recurrent Neural Networks”, 5th International Conference on Learning Representations, 2016.

III. Bengio, Y., “Learning Deep Architectures for AI”, Foundations and Trends in Machine Learning Volume 2, no 1, pp 1-127, 2009.

IV. Collobert, R., “Deep Learning for Efficient Discriminative Parsing”, 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA. Volume 15, pp. 224-232, 2011.

V. Chung, J., Gulcehre, C., Cho, K., and Bengio, Y., “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”, arXiv preprint arXiv: 1412.3555, 2014.

VI. Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y., “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches”, arXiv preprint arXiv:1409.1259, 2014.
VII. Deng, L., and Wiebe, J., “Sentiment propagation via implicature constraints”, Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 377–385, 2014.

VIII. Feng, S., Wang, Y., Liu, L., Wang, D., and Yu, G., “Attention based hierarchical LSTM network for context-aware microblog sentiment classification”, Springer Science Business Media, LLC, part of Springer Nature, Volume 22, Issue 1, pp 59–81, 2019.

IX. Hridoy, S. A. A., Ekram, M. T., Islam, M. S., Ahmed, F., and Rahman, R. M., “Localized twitter opinion mining using sentiment analysis”, Springer article, Vol. 2, 2015.

X. Hamouda, A., Marei, M., and Rohaim, M., “Building Machine Learning Based Senti-word Lexicon for Sentiment Analysis”, Journal of advances in information technology, vol. 2, no. 4, November 2011.

XI. Janane, S. K., Keerthana, M. S. and Subbulakshmi, B., “Hybrid Classification For Sentiment Analysis Of Movie Reviews”, International journal of engineering sciences & research technology, ISSN: 2277-9655, 2018.

XII. Jivani, A. G., “A Comparative Study of Stemming Algorithms”, IJCTA journal, ISSN: 2229-6093, volume 2, 2011.

XIII. Mohammad, S. M., “Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text”, Emotion Measurement. DOI: Elsevier Ltd 2016.

XIV. Mouthami, K., Devi, K. N., and Bhaskaran, V. M., “Sentiment Analysis and Classification Based on Textual Reviews”, international conference on information communication and embedded systems, pp. 271-276, 2013.

XV. Meyer, D., Hornik, K., and Feinerer, I., “Text Mining Infrastructure in R”, Journal of statistical software, Volume 25, Issue 5, 2008.

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

XVII. Nikita P. Katariya1, M. S. Chaudhari., “Text Prepeocessing For Text Mining Using Side Information”, International Journal of Computer Science and Mobile Applications, Vol.3, Issue. 1, 2015.

XVIII. Pal, S., Ghosh, S., and Nag, A., “Sentiment Analysis in the Light of LSTM Recurrent Neural Networks”, International Journal of Synthetic Emotions, Volume 9, Issue 1, January-June (IJSE), 9(1), 33-39 2018.

XIX. Prabowo, R., & Thelwall, M., “Sentiment analysis: A combined approach”, Journal of Informetrics, vol. 3, no. 2, 143-157, 2009.

XX. Ramasubramanian C., R.Ramya, “Effective Pre-Processing Activities in Text Mining using Improved Porter’s Stemming Algorithm”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 12, 2013.
XXI. Rachel Tsz-Wai Lo, He, B., and Ounis, I., “Automatically building a stopword list for an information retrieval system”, Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR), Volume 5, Pp. 17-24, 2005.

XXII. Shuqin Gu, Lipeng Zhang and Yuexian Hou, “A Position-aware Bidirectional Attention Network for Aspect-level Sentiment Analysis”, 27th International Conference on Computational Linguistics, pages 774–784, Santa Fe, New Mexico, USA, August 20-26, 2018.

XXIII. Singh, V., Saini, B., “An Effective Tokenization Algorithm for Information Retrieval Systems”, in the 1st international Conference on data Mining, DMIN-2014, Bangalore, and DOI: 10.5121, pp. 109–119, 2014.

XXIV. Vijayarani, S., Ilamathi, M. J., and Nithya, M., “Preprocessing Techniques for Text Mining - An Overview”; International Journal of Computer Science & Communication Networks, Volume 5 ,Number 1, Pp. 7-16, ISSN 2249-5789, 2015.

XXV. Wang, X., Wu, P., Liu, G., Huang, Q., Hu, X., and Xu, H., “Learning performance prediction via convolutional GRU and explainable neural networks in e-learning environments”, Springer-Verlag GmbH Austria, part of Springer Nature 2019.