US AIRLINES SENTIMENT ANALYSIS USING LSTM

N.SRIRAM,
Assistant Professor,
KG College of Arts and Science, Coimbatore.

Abstract— With the development of Internet and large amount of text data, it has been a very significant research to get vital information from text ocean. This paper promotes a RNN language model based on Long Short Term Memory (LSTM), which can get proper sequence information effectively. LSTM is better in analyzing long sentence emotions effectively compared with RNN model. LSTM is capable of learning long term capabilities. Remembering information for a long period of time is its natural. LSTM is a special kind of RNN. This paper aims at analyzing sentiments of US airlines with only two attribute positive and negative. Experiments show that LSTM can better in accuracy rate and recall rate than RNN.

Keywords-sentiment analysis; RNN, LSTM

I INTRODUCTION

Sentiment analysis gets importance in these days. Every business organization now a days are very interested in knowing their customers emotion or sentiment towards their product or services. There are several ways are there to do sentiment analysis. Lexicon based and rule based methods gets outdated because of the arrival of the machine learning methodologies. Now machine learning methods in turn gets older because of neural networks.

II. RELATED RESEARCH

In the study of text sentiment analysis, the sequential relationship between words is of critical importance. Mikolov [1] proposed a language model known as Recurrent Neural Network (RNN), which is publicly recognized as pretty suitable to process text sequence data. RNN consists of three modules, which are input layer, hidden layer and output layer. In RNN, the input layer at time 't' together with the hidden layer at time 't' are aggregated as a new input layer to calculate the hidden layer at time 't'. With such a loop structure, the hidden layer successfully reserves all information in previous words, which improves the performance of identifying the sequential relationships between words [1]. So RNN is a network that contains loops, and it allows information to be persistent.

In theory, the RNN language model could cover the time order structure of the whole text, and deal with long-term dependence problem. In practice, however, RNN could not learn the knowledge successfully. When the interval between the relative information of texts and the current location to be predicted becomes large, some problems will come out. As there are too many unfold layers in the back propagation through time optimization algorithm(BPTT), which leads to history information loss and gradient attenuation while training. To overcome this difficulty, some researchers put forward a strategy named Long Short-Term Memory (LSTM), which leads to better experimental results in some application scenarios.

LSTM through deliberate design to avoid long-term dependence, in practice, remember the long term information is the default behavior of LSTM. At present, LSTM network is the most widely used one, it replaces RNN node in hidden layer with LSTM cell, which is designed to save the text history information. LSTM uses three gates to control the usage and update of the text history information, which are input gates, forget gates and output gates respectively. The memory cell and three gates are designed to enable LSTM to read, save and update long-distance history information. The structural diagram is shown in Figure 1 [2].

Figure 1 provides an illustration of an LSTM memory block with a single cell. An LSTM network is the same as a standard RNN, except that the summation units in the hidden layer are replaced by memory blocks, as illustrated in Figure 2 [2].

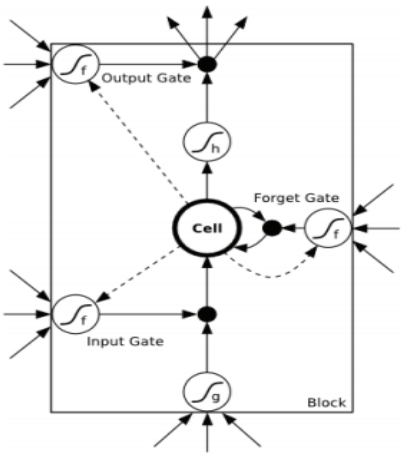

Fig.1.The LSTM cell

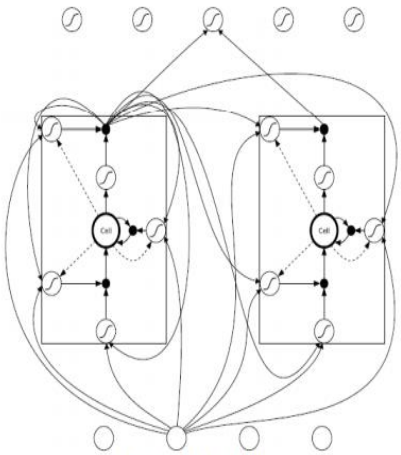

Figure 2. A LSTM network.

Fig. 2. A LSTM network

The cells state of LSTM is key, the vertical line run through in the middle, only a small amount of information interacted. LSTM remove or add ability from information to the state of the cell through the 'gate' structure, and the 'gate' is a way to make information selectively. The first step, forget gates determine which information from the cell state should be discarded. The second step, input gates determine the new information that is stored in the cell state. The third step, update the old cell state, using the above input gates and forget gates information to calculate the updated value of the cell state. Finally, the output gates determine the value of the output, which is based on the state of the cell [8].

The calculation process of LSTM mainly includes 4 steps. 1)Calculate the values of forget gate and input gate 2)Update the state of LSTM cell. 3)Calculate the value of output gates. 4)Update the output of the whole cell [2][3][4][5]. The detailed formula is shown as below. Input Gates:

$$a_l^t = \sum_{i=1}^{I} w_{il} x_i^t + \sum_{h=1}^{H} w_{hl} b_h^{t-1} + \sum_{c=1}^{C} w_{cl} s_c^{t-1}$$
$$b_l^t = f(a_l^t)$$

Forget Gates:

$$a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{h=1}^{H} w_{h\phi} b_h^{t-1} + \sum_{c=1}^{C} w_{c\phi} s_c^{t-1}$$
$$b_\phi^t = f(a_\phi^t)$$

Cells:

$$a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1}$$
$$s_c^t = b_\phi^t s_c^{t-1} + b_l^t g(a_c^t)$$

Output Gates:

$$a_l^t = \sum_{i=1}^{I} w_{iw} x_i^t + \sum_{h=1}^{H} w_{hw} b_h^{t-1} + \sum_{c=1}^{C} w_{cw} s_c^t$$
$$b_w^t = f(a_w^t)$$

Cell Outputs:

$$b_c^t = b_w^t h(s_c^t)$$

Here g(z) is sigmoid function, h(z) is the tanh function.

III. SENTIMENT ANALYSIS PROCESS

The process of recognizing and classifying the opinions, feelings or sentiments expressed in opinioned data, in order to ascertain whether the attitude of the writer towards a particular service, product etc. is negative, positive or neutral is known as sentiment analysis. The SA process mainly contains five stages as shown in figure 3.

A. Collecting and Preprocessing Data

It is the most important stage of sentiment analysis. If the quantity of the data is insufficient or if the quality of the collected data is poor then this might result in bad models and hinder the overall performance of the model. Service consumers or product users nowadays publish their opinions, feelings, sentiments, their experience about the product or service online on social media or public forums like discussion boards, blogs, product reviews, micro-blogs as well as on their personal logs. Opinions and feelings or sentiments are expressed in different ways, with different context of writing, vocabulary, usage of short forms and slang which makes

the data huge and disorganized. Manual analysis of opinionated data is virtually impossible. Consequently, special programming languages and techniques are used to process and analyze the opinionated data.

B. Text Preparation

Next phase in sentiment analysis is text preparation. The process of text preparation filters the extracted opinionated data before analysis. This process also includes identifying and removing non-textual contents and the contents that are not relevant to the field or topic of study from the opinionated data.

C. Sentiment Detection

At sentiment detection phase in sentiment analysis, each sentence extracted from the review and opinion is inspected for subjectivity. The sentences or statements with subjective terminologies are retained and the sentences or statements which carry objective expressions are rejected. Sentiment analysis or opinion mining is done at different levels of language such as at the lexical, morphological, discourse, semantic and pragmatic levels.

VI RESULT

The dataset is divided into testing and training sets. First the training set in trained and a model is created.

```
Train on 7232 samples, validate on 500 samples
Epoch 1/10
7232/7232 [==============================] - 21s 3ms/step - loss: 0.5730 - acc: 0.7688 - val_loss: 0.4338 - val_acc: 0.7960
Epoch 2/10
7232/7232 [==============================] - 20s 3ms/step - loss: 0.4080 - acc: 0.8191 - val_loss: 0.3696 - val_acc: 0.8340
Epoch 3/10
7232/7232 [==============================] - 21s 3ms/step - loss: 0.3306 - acc: 0.8550 - val_loss: 0.2776 - val_acc: 0.8800
Epoch 4/10
7232/7232 [==============================] - 20s 3ms/step - loss: 0.2467 - acc: 0.8993 - val_loss: 0.2147 - val_acc: 0.9140
Epoch 5/10
7232/7232 [==============================] - 20s 3ms/step - loss: 0.1885 - acc: 0.9266 - val_loss: 0.1954 - val_acc: 0.9300
Epoch 6/10
7232/7232 [==============================] - 20s 3ms/step - loss: 0.1549 - acc: 0.9382 - val_loss: 0.1780 - val_acc: 0.9380
Epoch 7/10
7232/7232 [==============================] - 20s 3ms/step - loss: 0.1410 - acc: 0.9484 - val_loss: 0.1842 - val_acc: 0.9380
Epoch 8/10
7232/7232 [==============================] - 20s 3ms/step - loss: 0.1231 - acc: 0.9537 - val_loss: 0.1877 - val_acc: 0.9300
Epoch 9/10
7232/7232 [==============================] - 20s 3ms/step - loss: 0.1121 - acc: 0.9588 - val_loss: 0.1923 - val_acc: 0.9280
Epoch 10/10
7232/7232 [==============================] - 20s 3ms/step - loss: 0.1024 - acc: 0.9599 - val_loss: 0.1984 - val_acc: 0.9280
```

The following diagram shows training validation loss.

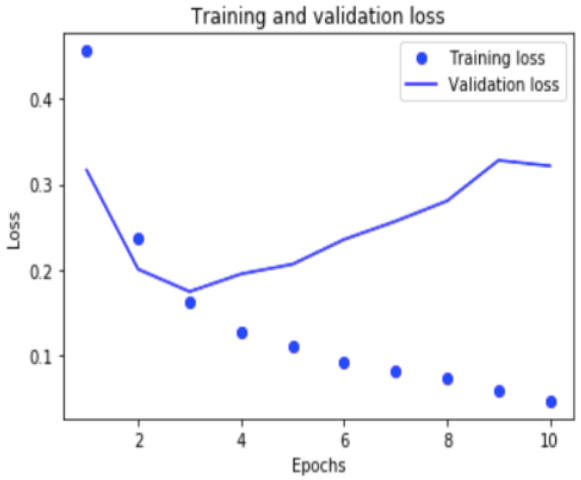

Fig. 4.Training and Validation loss

The following diagram show training accuracy at various epochs. The x axis holds epoch value and the y axis holds the accuracy.

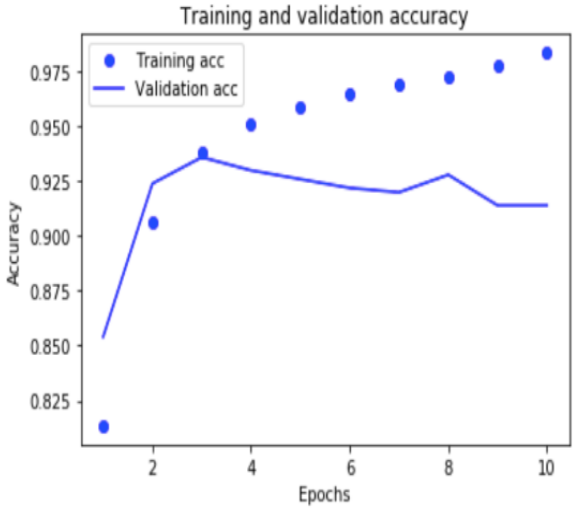

Fig. 5.Training and validation Accuracy

From the above diagram it is clear that the epoch 3 yields the highest accuracy level out of 10 epochs. The accuracy earned is as follows.

pos_acc 78.26086956521739 %
neg_acc 94.44444444444444 %

CONCLUSION

This paper shows the US airlines sentiment analysis with small amount of tweets and with only two classes positive and negative. In future, the same data set will be analyzed with large amount of data set for more accuracy. This paper takes only two classes positive and negative in future the third class neutral can be added with improved sentiment analysis.

REFERENCES

[1] Sepp Hochreiter and Jurgen Schmidhuber, LongShort-Term Memory, Neural Computation, pages 12-91,1997.

[2] Alex Graves, Supervised Sequence Labelling with Recurrent Neural Networks, Studies in Computational Intelligence, pages 385-401, 2011.

[3] Daniel Soutner and Ludxk M¾ller, Application of LSTM Neural Networks in Language Modelling, Lecture Notes in Computer Science, pages 105-112,2013.

[4] Martin Sundermeyer, Zoltan Tuske, Ralf Schluter, Hermann Ney,Lattice Decoding and Rescoring with Long-Span Neural Network Language Models, ICASSP, pages 661-665, 2014.

[5] Martin Sundermeyer, Hermann Neyand Ralf Schluter, From Feedforward toRecurrent LSTM Neural Networks for Language Modeling, IEEE/ACM Transactions on, 23(3):517-529, 2015.