International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Identification of Gender based on Blogs using Attention-based Recurrent Neural Network

Sachin Minocha, Tarun Kumar, Prem Prakash Agrawal

Abstract*: With the digitization, the importance of content writing is being increased. This is due to the huge improvement in accessibility and the major impact of digital content on human beings. Due to veracity and huge demand for digital content, author profiling becomes a necessity to identify the correct person for particular content writing. This paper works on deep neural network models to identify the gender of author for any particular content. The analysis has been done on the corpus dataset by using artificial neural networks with different number of layers, long short term memory based Recurrent Neural Network (RNN), bidirectional long short term memory based RNN and attention-based RNN models using mean absolute error, root mean square error, accuracy, and loss as analysis parameters. The results of different epochs show the significance of each model.

Keywords*: Deep Learning, LSTM, Bidirectional LSTM, Attention-based LSTM, Embedding, RNN, Activation Functions.

I. INTRODUCTION

The popularity of publicly accessible blogs are increasing day by day and provides an opportunity to collect information from content written by different authors [1]. A blog is a discussion or information from a website that is a very popular means of communication. Blogs contain a proper layout of information given by the bloggers. Thus, anyone can use this vast data to answer the questions like how content presentation varies between bloggers of different genders and different ages, how much can we know about someone by only analyzing their written content [1]. The questions discussed here are of great practical consequence in different domains e.g., from marketing point of view to help companies to have a clear idea of their target groups for attaining a better marketing strategy, from a forensic point of view to know the details of a person who has written a "suspicious text" which plays valuable role in investigation [2]. In this regard, wordsmith peculiarity tries to determine the gender, age, native language or personality type of authors by analyzing their published texts. Here, the main focus is to build the system to identify only gender of the authors. Other authorship details will be a part of future work in this area.

Wordsmith peculiarity work can be done by using artificial intelligence based techniques like neural networks,

evolutionary computing. This work focuses on the deep learning techniques to determine the gender of the blogger. These techniques are adopted due to high accuracy demonstrated for other applications [3]. Moreover, the blog's data is vast and to extract the information from such data deep learning techniques are useful due to their huge advantages. The neural network along with deep learning techniques including Recurrent Neural Network, Long Short Term Memory and Attention-based models has been described in the next section.

II. DEEP LEARNING AND NEURAL NETWORK

For solving day to day problems or tasks users create computer programs but some tasks are so complex that it is impractical for users to code. For such tasks, machine learning is used. Machine learning is a set of algorithms that learns from data and apply it's learning to make intelligent decisions. Deep learning is a subset of machine learning. Unlike machine learning, in deep learning, machines will learn patterns from the given data rather than requiring a human operator to define the patterns that the machine should look for in the data. The deep learning technique works on neural networks (NN). A neural network is a computer program modeled to simulate the processing of the human brain [4]. A neural network consists of input layer, hidden layers and an output layer[5]. Fig.1 shows an example of a Neural Network. A neural network consists of multiple single units called neurons. They are the processing elements of neural networks and convert the weighted inputs into output using activation functions [4].

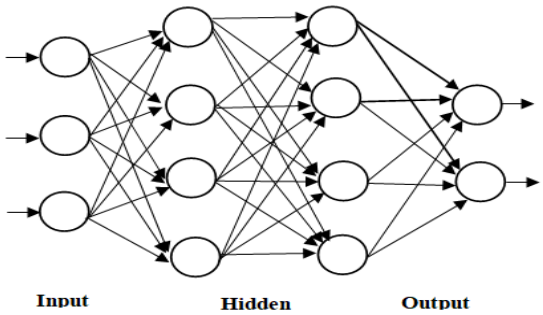

Fig. 1.An example of Neural network

Manuscript published on November 30, 2019.
* Correspondence Author
Sachin Minocha*, School of Computing Science and Engineering, Galgotias University, Greater Noida, India. Email: sachin0111@gmail.com
Tarun Kumar, School of Computing Science and Engineering, Galgotias University, Greater Noida, India.
Email: tarunsharma2910@gmail.com
Prem Prakash Agrawal, School of Computing Science and Engineering, Galgotias University, Greater Noida, India.
Email: premalwar@gmail.com

Lines between neurons represent the information flow in the neural network. Each line has a weight and an activation function(a function to control the information between neurons) [4]. A neural network is trained by weight adjustments between neurons. Neurons receive output signals from the previous layer and process the signal and then pass the output signal to the next layer [4].

In a neural network, we calculate the sum of products of inputs (y) and their corresponding weights (w) and then Activation Function (f) is applied to get the output signal [5] which is then transferred to the next layer. The output is mathematically expressed by (1).

$$O = \sum w_i y_i + b \qquad (1)$$

where w_i are weights of connections, y_i are the inputs and b is the intercept or bias [5].

The activation functions introduce non-linear properties to our neural network. Some popular types of activation function [5] [3] are:

A. Sigmoid

It is an activation function expressed in (2) as:

$$s(y) = \frac{1}{1+e^{-y}} \qquad (2)$$

It ranges from 0 to 1.

B. Tanh (Hyperbolic Tangent)

It is an activation function expressed in (3) as:

$$t(y) = 1 - \frac{e^{-2y}}{1+e^{-2y}} \qquad (3)$$

It ranges from -1 to 1.

C. Relu (Rectified Linear Units)

It is an activation function expressed in (4) as:

$$r(y) = \max(0, y) \qquad (4)$$

It ranges from 0 to ∞.

D. Softmax

It calculates the probability of a particular class over all possible classes. It is expressed in (5) as:

$$g(y) = \frac{e^{y_i}}{\sum_{j=0}^{k} e^{y_j}} \qquad (5)$$

where i = 0, 1, 2…k. It ranges from 0 to 1.

Gradient descent through Backpropagation is used for training the neural network which results in the minimization of cross-entropy loss [6]. Firstly, the gradient of loss w.r.t weights from the previous hidden layer to the output layer is calculated and then chain rule is applied in a backward manner to recursively calculate the gradient of expressions w.r.t the weights [6]. The weights between layers are adjusted by using the calculated gradients. To stop this iterative process certain criteria have to be provided.

III. RNN (Recurrent Neural Network)

Recurrent Neural Network is a powerful type of neural network. It is designed to capture sequential patterns present in data which makes it useful to be applied on text data [5].

RNN has a short term memory by which RNN has an ability to memorize important features of the received input. This makes them precise to predict the upcoming sequence [3] [5]. Unlike Feed Forward neural network RNN has feedback loops such as Back propagation through time (BPTT) to loop information back into the network [6][5]. Compared to the artificial neural network, RNN has a recurrent structure of neuron as shown in Fig. 2(a) and recurrent neuron can be unfolded into a chain like structure shown in Fig. 2(b).

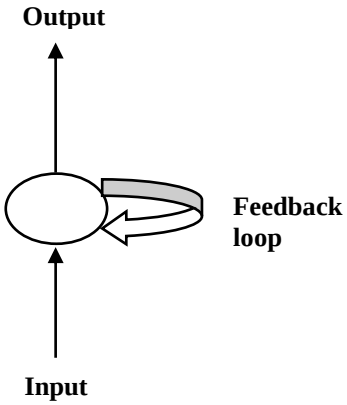

Fig.2 (a). Recurrent Structure of Neuron

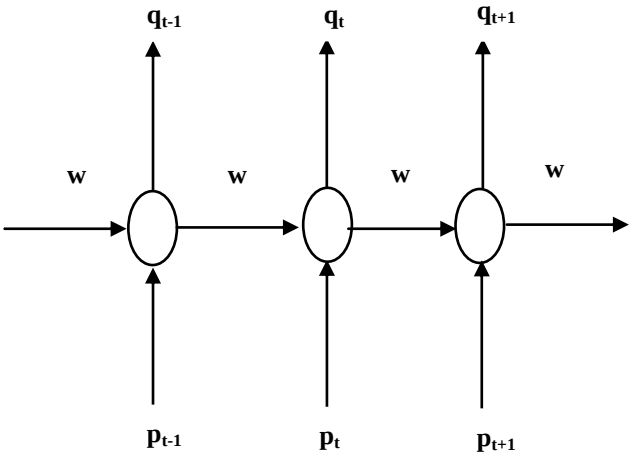

Fig.2 (b). A chain-like structure of Recurrent Neuron

RNN does not require parse trees for the extraction of features from the sentences [6]. Unlike traditionally, it uses word embedding as input which makes it more popular. Word embedding encodes certain semantic and syntactic information [3].

Word embedding is a technique for language modeling and feature learning. It generates a vector of continuous real numbers from the words present in a vocabulary [3]. This technique performs embedding of sparse vector from high dimensional space to a dense vector of lower dimensional space. Embedding vector consists of dimensions representing latent features of a sentence.

Patterns and regularities may be encoded using these vectors [3].

Word2Vec is an example of embedding system which is used as a prediction model to learn word embedding from text in a neural network [3].

Apart from its significance, there are two major issues with RNN i.e. exploding gradient problem and vanishing gradient problem. Gradient measures how much the output of a function changes if we change the input little bit.

A. Vanishing Gradient Problem

It occurs when the gradient of error w.r.t the weights is very-very less than 1 as shown in (7).

$$\frac{\Delta e}{\Delta w} \ll 1 \quad (7)$$

and the new weight is almost equal to the previously assigned weight in (8).

$$w' = w + \Delta w \quad (8)$$

where,

w' is new weight, w is old weight and Δw is change in weight. This problem results in the slow learning of the neural network [5].

B. Exploding Gradient Problem

It occurs when the gradient of error w.r.t the weights is very-very greater than 1 as shown in (9).

$$\frac{\Delta e}{\Delta w} \gg 1 \quad (9)$$

and the new weight is very much changed to the form which is shown in (7).

$$w' = w + \Delta w \quad (10)$$

where w' is new weight, w is old weight and Δw is change in weight. This problem affects the stability and learning of a neural network [5].

IV. LSTM (LONG SHORT TERM MEMORY)

Sequence Prediction problems are considered as one of the most difficult problems in the data science areas [7]. These problems include prediction of the next word in a sentence, finding patterns in the stock market data, etc [7]. Traditional RNN has the limitation of not working effectively with the long term dependencies but they are fine with the short term dependencies. So traditional RNN is not used for long term sequence prediction problems [8].

LSTM is an extension of RNN which offers an effective solution for the sequence prediction problems. LSTM has the ability to memorize the pattern in memory for a long interval of time [9]. In LSTM, memory cells contain a memory unit and a gate mechanism.

In LSTM, we have 3 gates: input gate, forget gate and output gate. The input gate (I_t) is used to know whether to allow new input in or not, forget gate (f_t) is used to know whether the information is useful or not and output gate (O_t) is used to know whether to allow impact of information to output at current time step or not [3]. Input gate (i_t), forget gate (f_t), output gate(o_t) internal memory (C_t) and a LSTM unit output (h_t) at time interval (t) are mathematically expressed in (11), (12), (13), (14) and (15) respectively.

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \quad (11)$$

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \quad (12)$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right) \quad (13)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_{hc}h_{t-1} + W_{xc}x_t + b_c) \quad (14)$$

$$h_t = o_t \odot \tanh(C_t) \quad (15)$$

The LSTM architecture using the input, output, memory is shown in Fig. 3.

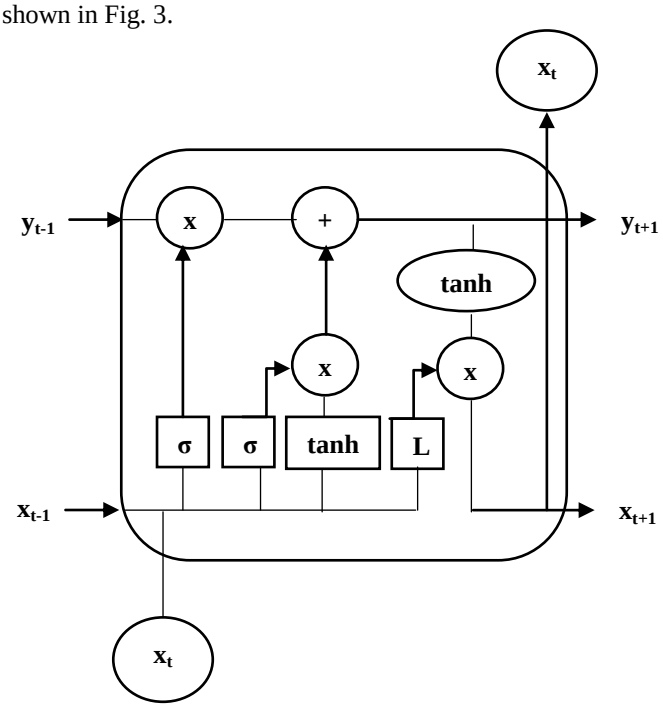

Fig. 3. LSTM architecture

Here, first gate is a forget gate that is regulated by one layer of neural network. Inputs for neural network are h_{t-1} (output of previous LSTM block), x_t (input of current LSTM block), C_{t-1}(memory of previous LSTM block) , bias vector b_f and a sigmoid function σ [10][9].

V. BIDIRECTIONAL RECURRENT NEURAL NETWORKS

Bidirectional RNN is the extension of traditional RNN which improves the model performance on sequence classification problems [6]. Bidirectional RNN is useful when the context of the input is required in the network [10]. Initially, these were introduced to enhance the availability of input information in a network.

Bidirectional RNN connects one hidden layer from backward direction and one Bidirectional LSTM trains two LSTMs instead of one LSTM on the input sequence. The first LSTM trains on the normal input sequence and the second LSTM trains on the reversed copy of the normal input sequence. Bidirectional RNN connects one hidden layer from backward direction and one from the forward direction to the same output signal which results into the simultaneous availability of information in the output layer from past [6] as well as from future .The use of Bidirectional LSTM may not be useful for all sequence prediction problems but can offer some better results to those domains where it is appropriate.

These variants of RNN are popular because they have the ability to decide when to forget and when not to using gates in their architecture. Fig. 4 shows Bidirectional RNN architecture [10].

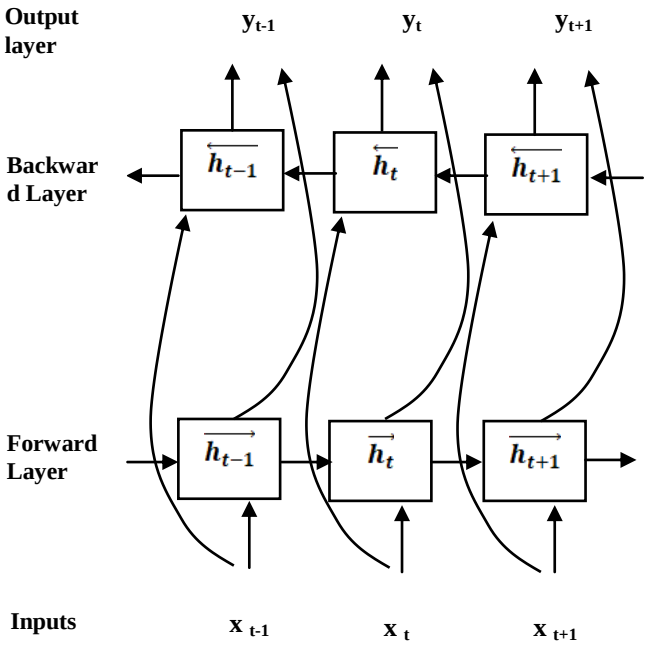

Fig.4. Bidirectional RNN architecture

The A Bidirectional LSTM can be formed using a forward LSTM and a backward LSTM. LSTM functions are represented by LSTM(•) [10].

Mathematically, LSTM can be expressed in (16) as:

$$y_t = f(w_{\vec{h}y}\overleftarrow{h_t} + w_{\overleftarrow{h}y}\overleftarrow{h_t} + b_y) \quad (16)$$

Where,

$\overrightarrow{h_t} = LSTM(x_t, \overrightarrow{h_{t-1}})$

$\overleftarrow{h_t} = LSTM(x_t, \overleftarrow{h_{t+1}})$

and f (•) is an activation function, w_{hy} are the weights from h to y, and b_y is the bias at layer y. Rectified linear unit (RELU) is additionally used here as an activation function.

VI. ATTENTION-BASED RECURRENT NEURAL NETWORK

In previously discussed neural networks, there is a limitation on the length of input sequences that can be learned because of the encoding of input sequences to internal representation of fixed length [3].Due to this for very long input sequences, the performance is very low.

The attention mechanism is proposed to overcome the limitation of Encoder-Decoder Neural network [3]. It allows the network to keep intermediate outputs from the encoder LSTM from each step of inputs and training the model to learn to pay attention on significant sequence of input [3][11].

The working is similar to the encoder-decoder mechanism except for the context vectors which are added between the encoder and decoder.[11] To construct context vector, target and source states are compared by looping over all encoders states and scores are generated for each state. Then softmax is used for normalization of all scores resulting into the probability distribution on target states and to train context vectors weights are introduced. Fig,.5 shows architecture of Attention based RNN.[11]

Mathematically Attention LSTM can be expressed in (17) as:

$$r = H\alpha^T \quad (17)$$

Where,

$$M = \tanh\left(\begin{bmatrix} W_h H \\ W_v v_a \otimes e_N \end{bmatrix}\right)$$

$$\alpha = softmax(w^T M)$$

where, M, α, r, W_h, W_v and w are projection parameters. α consist of attention weights and r is a weighted representation of sentences with a given aspect.[8] The operator used above (a circle with a multiplication sign inside, OP for short here) means: $v_a \otimes e_N = [v; v; \ldots; v]$, that is, the operator repeatedly concatenates v for N times,[8] where, e_N is a column vector with N 1s. $W_v v_a \otimes e_N$ is repeating the linearly transformed v_a as many times as there are words in sentence.[9]

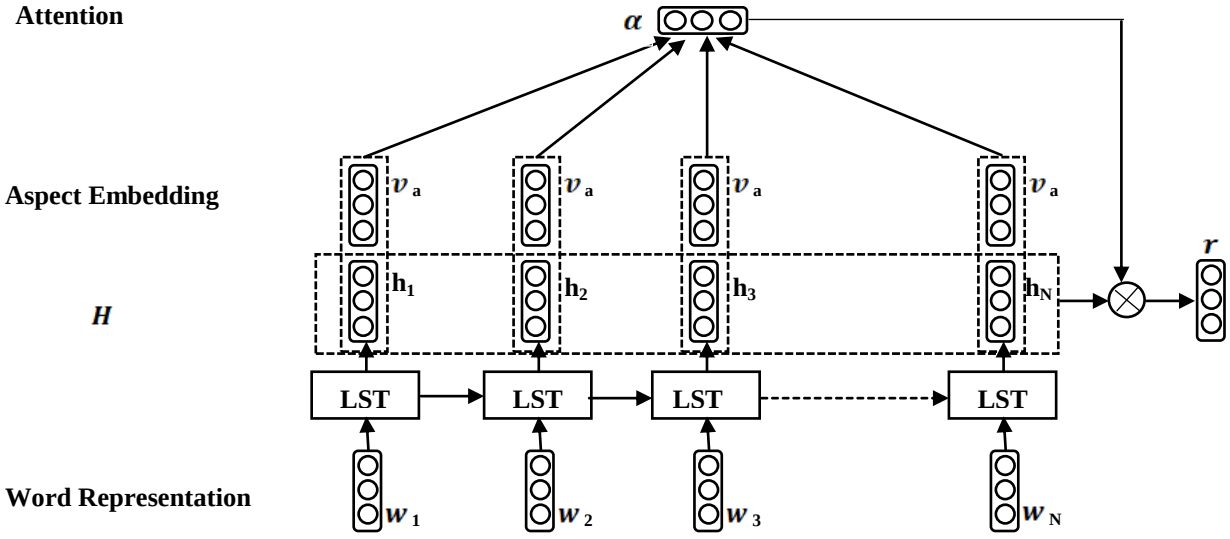

Fig. 5. Attention Based RNN Architecture

VII. RESULTS AND DISCUSSIONS

In this work, 4 deep learning models are implemented which are as follows: ANN (Artificial Neural Network), LSTM (Long Short Term Memory), Bidirectional LSTM and Attention-based LSTM. For implementing these models, the python tool called SPYDER with deep learning libraries is used. The system configurations used for testing the model are an i7 processor with 8 GB RAM and an AMD Graphics Card. The corpus dataset has been used for analysis described in next subsection.

A. The Corpus

For the purpose of this study, we have analysed the blogs and downloaded those blogs which include the gender of the author and at least 400 occurrences of common English words. The full obtained corpus includes over 85,000 blogs [1].

Each sample blog contains all blog entries from start to the end date and formatting of those blog entries were ignored. There is no differentiation done between the quotes and the texts within a blog. Some of the blogs include texts in other languages and those texts are ignored for this study.[1] We have used the following corpus for this study:

i. Blog Authorship Corpus

This Corpus contains blogs of 20,450 bloggers. The corpus combines of 681,000 posts and over 150 million words or approximately 50 posts and 8500 words per person. All included bloggers belong to one of the mentioned age groups (13-17), (23-27), (33-47)[12]. It is observed that the proportion of males and females are almost equal.

B. Analysis parameters

Following performance parameters are calculated by varying the number of epochs:

i. LOSS

Loss is the variation of algorithmic output with respect to the actual or expected output on any particular input.

Loss = ∑ (actual value – observed value)

ii. ACCURACY

Accuracy is the degree to which the calculation result relates to the correct value.

Accuracy = (# of elements correctly classified) ÷ (Total elements)

iii. RMSE (Root Mean Square Error)

RMSE is the measure of the differences between values predicted and the values observed by the model. RMSE represents the standard deviation of the prediction errors (difference between predicted value and observed value).

$$RMSE = \sqrt{\sum (actual - observed)^2 \div (\text{total \# of values})}$$

iv.MAE (Mean Absolute Error)

MAE is the average of the differences between the actual and the observed values.

MAE = ∑ (actual value – observed value) ÷ total # of values

C. Analysis

The comparison of performance for described models on given dataset has been done in this section.

Table I: Comparing Loss between different Deep Learning Models

EPOCHS	ANN 2-layers	ANN 5-layers	LSTM	Bidirectional LSTM	Attention based LSTM
10	39.8	38.7	37.6	30.6	28.3
20	35.7	34.3	30.7	24.9	23.8
30	34.8	31.1	28.7	24.1	23.1
50	33.1	30.6	28.1	23.6	21.9
100	32.7	29.9	27.4	22.2	21.1
150	31.1	29.1	26.3	21.8	20.7
200	27.9	26.4	23.6	21.5	20.2
250	25.1	23.3	21.1	20.7	19.8

Above table I is shown graphically below in Fig.6.

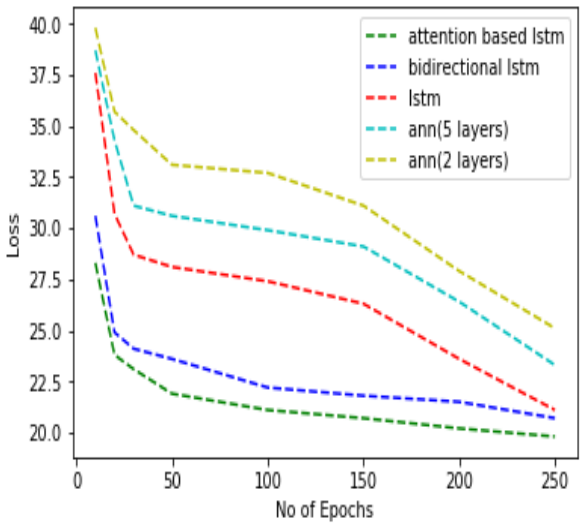

Fig. 6. Comparing Loss between different deep learning models by varying no. of epochs

From above graph, it can be concluded that the loss of Attention-based LSTM is lesser than LSTM and Bidirectional LSTM. Loss of ANN models is higher than LSTM models due to lack of ability to memorize the pattern in the memory for a long period of time.

Table II: Comparing Accuracy between different Deep Learning Models

EPOCHS	ANN(2 layers)	ANN(5 layers)	LSTM	Bidirectional LSTM	Attention based LSTM
10	50.4	52.3	54.6	55.4	58.9
20	51.3	53.2	55.2	55.7	59.8
30	51.9	53.9	56.2	62.2	65.3
50	53.1	54.7	56.9	65	66.9
100	53.8	56.1	57.4	65.2	67.7
150	54.6	56.8	59.6	65.8	68.5
200	56.3	59.7	66.1	67.1	69.8
250	58.7	61.9	67.6	68.2	70.2

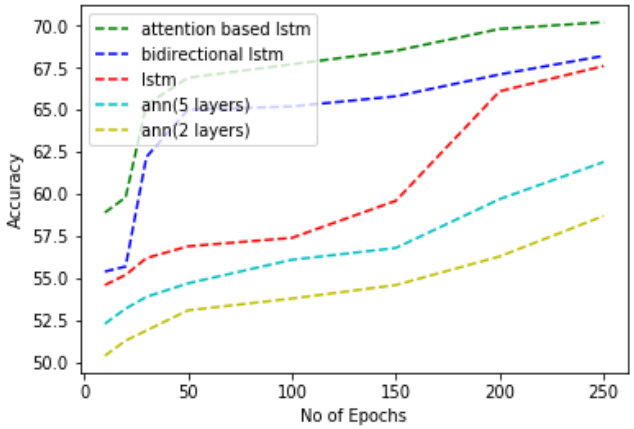

Fig.7: Comparing Accuracy between different deep learning models by varying no. of epochs

From Fig. 7 graph, it can be concluded that the accuracy of Attention-based LSTM is higher than LSTM and Bidirectional LSTM because it tells where exactly to look when the neural network is trying to predict parts of a sequence. Accuracy of LSTM is better than ANN models because it has the ability to memorize the pattern in the memory for a long period of time.

Table III: Comparing RMSE between different Deep Learning Models

EPOCHS	ANN(2 layers)	ANN(5 layers)	LSTM	Bidirectional LSTM	Attention based LSTM
10	59.6	58.8	57.2	53.5	53.1
20	57.3	55.9	54	49	48.7
30	56.8	53.2	50.6	47.8	46.7
50	53.7	51.7	49.6	47.7	46.4
100	52.8	51.1	49.3	47.3	45.7
150	51.9	49.6	48.3	46.6	45.2
200	50.1	47.8	46.7	46	45
250	48.3	46.6	45.9	44.6	43.4

Above table III is shown graphically below in Fig.8.

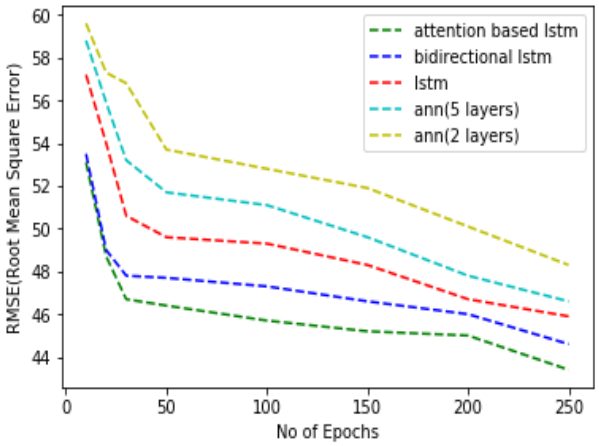

Fig.8: Comparing RMSE for different no. of epochs

From Fig. 8 graph, it can be concluded that the RMSE of Attention based LSTM is lesser than LSTM and Bidirectional LSTM because it pays attention to significant input sequences. RMSE of ANN models is higher than LSTM models due to lack of ability to memorize the pattern in the memory for a long period of time.

Table IV: Comparing MAE between different Deep Learning Models

EPOCHS	ANN(2 layers)	ANN(5 layers)	LSTM	Bidirectional LSTM	Attention based LSTM
10	53.8	53.4	52.6	44.8	43.5
20	53.1	52.8	49.4	41.1	40.7
30	52.7	52.1	48.6	40.2	39.1
50	52.2	51.3	48.2	39.7	38.6
100	51.6	50.2	47.6	38.9	38.1
150	50.9	48.7	45.6	37.8	37.2
200	49.3	47.5	44.7	37.1	36.3
250	48.4	46.6	43.5	36.6	35.4

Above table IV is shown graphically below in Fig.9.

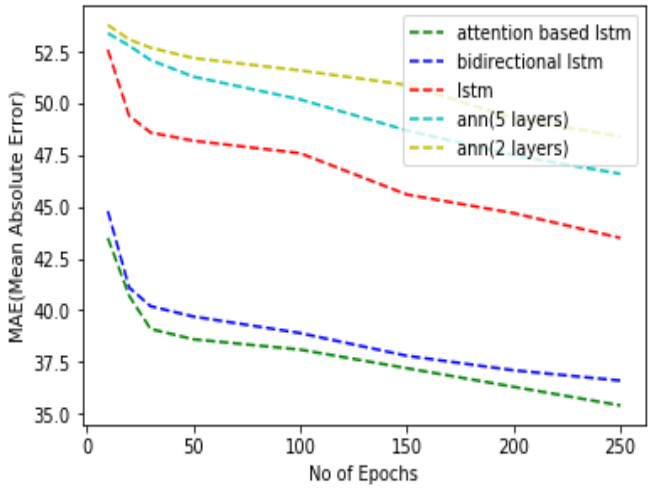

Fig. 9. Comparing MAE between different deep learning models by varying no. of epochs

From Fig. (graph, it can be concluded that the MAE of Attention based LSTM is lesser than LSTM and Bidirectional LSTM due to the ability to tell where exactly to look when the neural network is trying to predict parts of a sequence. MAE of ANN models is higher than LSTM models.

VIII. CONCLUSION

This paper compares the performance of the ANN model, LSTM, Bidirectional LSTM, and the Attention-based LSTM on corpus datasets using accuracy, loss, MAE and RMSE as the parameters. The analysis clearly signifies that the performance of the ANN improves with the increase in number of layers. The LSTM performs better as compared to ANN due to ability of memorization which is further improved by Bidirectional LSTM. The attention-based LSTM outperforms all the other models due to attention on significant sequence. The attention based LSTM can be used to determine the gender of author of any digital content. In future, this technique can be applied for complete author profiling.

REFERENCES

1. J. Schler, "Effects of Age and Gender on Blogging," AAAI spring Symp. Comput. approaches to Anal. weblogs., vol. 6, no. 2006, pp. 199–205, 2005.
2. A. Sboev, T. Litvinova, D. Gudovskikh, R. Rybka, and I. Moloshnikov, "Machine Learning Models of Text Categorization by Author Gender Using Topic-independent Features," in Procedia Computer Science, 2016, vol. 101, pp. 135–142.
3. L. Zhang and L. Corporation, "Deep Learning for Sentiment Analysis : A Survey," Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 8, no. 4, pp. 1–34, 2018.
4. R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research," J. Pharm. Biomed. Anal., vol. 22, no. 5, pp. 717–727, 2000.
5. H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent Advances in Recurrent Neural Networks," arXiv Prepr. arXiv1801.01078, pp. 1–21, 2018.
6. Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning arXiv : 1506 . 00019v4 [cs . LG] 17 Oct 2015," arXiv Prepr. arXiv1506.00019, pp. 1–38, 2015.
7. J. Gonzalez and W. Yu, "Non-linear system modeling using LSTM neural networks," in IFAC-Papers OnLine, 2018, vol. 51, no. 13, pp. 485–489.
8. Y. Wang, M. Huang, L. Zhao, and X. Zhu, "Attention-based LSTM for Aspect-level Sentiment Classification Attention-based LSTM for Aspect-level Sentiment Classification," in Proceedings of the 2016 conference on empirical methods in natural language processing., 2019, no. January 2016, pp. 606–615.
9. G. Rao, W. Huang, Z. Feng, and Q. Cong, "LSTM with sentence representations for document-level sentiment classification," in Neurocomputing, 2018, vol. 308, pp. 49–57.
10. Y. Zhao, R. Yang, G. Chevalier, R. C. Shah, and R. Romijnders, "Optik Applying deep bidirectional LSTM and mixture density network for basketball trajectory prediction," Opt. - Int. J. Light Electron Opt., vol. 158, no. 2018, pp. 266–272, 2018.
11. C. Du and L. Huang, "Text classification research with attention-based recurrent neural networks," Int. J. Comput. Commun. Control, vol. 13, no. 1, pp. 50–61, 2018.
12. I. Markov, G. Sidorov, and A. Gelbukh, "Adapting Cross-Genre Author Profiling to Language and Corpus Notebook for PAN at CLEF 2016," arXiv Prepr. arXiv1801.01078, no. September, 2016.

AUTHORS PROFILE

Sachin Minocha has 5 years of teaching experience. He is pursuing PhD from Department of computer science and engineering, Sant Longowal Institute of Engineering and Technology. He is currently working in School of Computing Science and Engineering, Galgotias University. His research area is Machine Learning.

Tarun Kumar has 15 years of experience in teaching, He is pursuing PhD from Department of Computer Science & Engineering, Uttrakhand Technical University, Dehradun. He is currently working in school of computing science and Engineering, Galgotias University. His research area is Machine Learning and Wireless Sensor Network.

Prem Prakash Agrawal has 13 years of experience in teaching and industry. He has done his M.tech in computer science & Engineering from MNNIT Allahabad. He is currently working in School of Computing Science and Engineering, Galgotias University. His research interest is Machine Learning and Cloud Computing.