Sentiment and Emotion in Social Media COVID–19 Conversations: SAB-LSTM Approach

Ashok Kumar D.[1] and Anandan Chinnalagu[2]
[1,2]*Department of Computer Science, Government Arts College*
(Affiliated to Bharathidasan University, Tiruchirappalli) Kulithalai–639120, Karur, India
E-mail: [1]drashoktrichy22@gmail.com, [2]anandanc@hotmail.com

2020 9th International Conference System Modeling and Advancement in Research Trends (SMART) | 978-1-7281-8908-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/SMART50582.2020.9337098

***Abstract*—Sentiment and Emotion detection in social media conversations remains a challenge and analyzing the people emotion emerged as an important task in this unprecedented time of COVID-19. People sentiment and emotions are affected by lockdowns, social distancing, travel, work-from-home, wearing mask, reading social media posting. Most of them are feeling sad, anger, depressed, and some of them are neutral and happy. The most recent Sentiment Analysis (SA) researches are done using Twitter dataset (short-text) and rule-based (sentiment lexicon) approach, the outcome of these SA models' results is not showing the consistent prediction of people sentiment about COVID-19. To mitigate and overcome limitations of lexicon approach, processing unstructured social media long text posting, getting context based sentiment score, model overfitting, performance problems of sentiment models, authors' proposed and built a novel multi-class SA model using extension of Bidirectional LSTM (SAB-LSTM) with additional layers. In this experiment SAB-LSTM model has been used to process long text of social media posting, news articles text dataset. Experiment result showed, SAB-LSTM model performance is better than traditional LSTM and BLSTM. Compared SAB-LSTM performance metric of Precision, Recall, F1 Score and sentiment score with traditional LSTM and BLSTM. For this experiment collected COVID-19 related dataset from various social media sources such as Twitter, Facebook, YouTube, News articles blogs and collected data from friends and family.**

Keywords: Conversation mining, Sentiment analysis, Text Analytics, Machine Learning, Deep Learning, Sentiment Lexicon, COVID-19 Sentiment.

I. Introduction

People are relying on social media news and also expressing their opinion, emotion and sentiment via social media posting, on-line survey, and blogs about this novel virus. Now this new virus "novel Coronavirus" spreading and infected millions of people globally. The World Health Organization (WHO) had given this virus a name CoronaVirus Disease of 2019 (COVID-19) on 11th February 2020. There are 213 countries and territories affected, more than 20 million (Source: https://www.worldometers.info/). It's a new normal for human life during this pandemic, its' affected people emotion, countries economy, people lifestyle, jobs and social activities. It's important to analyze the current social media data to understand public opinion, sentiment and emotion about current impact of this virus and public concerns about the future impact called 2nd wave of COVID19.

Reviewed the recent COVID19 research works related to public opinion, emotion, and sentiment analysis in section II. Experiment approach and methodology are described in section III. In Section IV, LSTM, BLSTM, authors' proposed SAB-LSTM models literature reviews are presented in detail. Section V has the experimental results and Section VI Concluded experimental findings and future work.

II. Review

In this section, focused on most recent research work related to COVID-19 Sentiment Analysis (SA) and researcher's contributions towards detecting public emotion from social media posting.

A. Most Recent research on SA

Jim Samuel, G.G.Md.Nawaz Ali, Ek Esawi and Yana Samuel are published (11th June 2020) a paper related to COVID19 public sentiment applying machine learning (ML) approach of classifications [14]. Short text and log text of coronavirus tweets were processed *Naïve Bays and Logistics regression classifications algorithms.* Based on the observation, model result showed 91% accuracy for Naïve Bays, 74% accuracy for Logistics regression methods, both classifications performed well for short tweet text. *Both the methods showed weaker performance for long tweet text.* This research showed public sentiment were more negative and fear.

B. SA on Twitter Hastag

Researchers (Karishma Sharma, Sungyong Seo, Chuizheng Meng, Sirisha Rambhatla & Yan Liu) were designed a dashboard to track misinformation and provide Misinformation analysis, Sentiment analysis, Topic and Trend analysis used twitter data [1]. Analyzed the following hashtag conversations #workfromhome, #wfm, #workfromhome, #workingfromhome, #wfhlife, #socialdistance, #socialdistancing and keywords such as COVID-19, Coronavirus, CoronavirusOutbreak, and 2019nCoV. In this analysis, intervention policies of social distancing *show 71.48% positive and 28.52% negative sentiment, work from home shows 77.79% positive and 22.21% negative sentiment.* These researchers suggested the following future work based on their analysis, *(a) Based on this dataset, twitter users are higher in US compare to other countries and half of the tweets posted by US users in English language, if data include for other languages then sentiment score reflects the global sentiment score. (b) Labeling the tweet data is challenging, considering future work towards unsupervised and distance supervising method to alternate method of labeling. (c) Analyze Sentiment in other topics from different domain*.

C. SA Report from UK

Bennett Kleinberg, Isabell evan der Vegt, Maximilian Mozes from UK, reported (14th May 2020). The findings of Real-World Worry (RWWS) dataset showed the emotional response of UK residents about COVID-19 at the point in time this virus affected lives in UK. Data collected from people and request survey participants to express their emotions in text [2]. There were 5000 texts collected *50% of short texts (twitter-sized text) and another 50% long texts*, correlated people's emotional responses to matching categories of the *LIWC2015 lexicon.* The short text matched 92.36% and the long text matched only 90.11%. These researchers suggested the following future research works for research community based on their current analysis and untouched research questions. Here are those five suggested the following future works, (a) mitigating *lexicon-approach limitations,* future work around COVID-19 emotion using different features and a model to provide large scale, real-time measurements of emotional response. *Tweet-sized short texts contained much less information and less suitable for predictive modeling,* experimental setup and study should not mimic a 'natural' twitter experience. (b) Nowadays, many researchers used Twitter data, the future study can be *used non-Twitter data for emotion detection study.* (c) For COVID-19, suggested future research can be focused on manually annotating topics to map what people concerns about this virus. (d) Capturing more samples from long for a longer period to find how emotional response develop over time. Using high-frequency daily sampling for several months that could account for large number of events that may affect emotions. (5) Fifth, utilize other approaches to measuring psychological constructs in text because the rate of out-of-vocabulary terms may be low in the dataset for emotion matching.

D. Corona Tracker SA Article

Corona Tracker Community Research Group track the outbreak and submitted (19th March 2020) the bulletin to WHO. Fairoza Amira et al.[3]. this research group presented a SA news and public emotion report. It shows there were 561 positive and 2548 negative news articles published from Singapore, China, US and Australia. *This sentiment analysis is based on positive and negative words counts in the news article.*

E. Country Wide SA based on Twitter data

Akash D Dubey, (9th April 2020) gathered tweets for the following twelve countries Australia, Belgium, China, France, Germany, India, Italy, Netherland, Spain, Switzerland, UK, and USA, and also country wise sentiment analysis had been done from tweets dataset [4]. In this analysis result shows that overall people were taking a positive and hopeful about COVID-19, there are instances people emotions were fear, sadness and disgust and it had been exhibited worldwide. Researcher used the NRC Word-Emotion Association *Lexicon for analyzing the emotions. NRC contains 10,170 lexical items which not only analyze the polarity and* detected the following emotion of fear, joy, anticipation, anger, disgust, sadness, surprise, and trust. Researcher used COVID19 related keywords to collect 50000 tweets from each country. For the collection, RTweet package in R was used. This researcher suggested the following future studies suggestions, (a) *the tweets which were used for this study were in English language, it's a limitation to the sentiment analysis result, (b) this study done only for eight emotions, and it doesn't count the emotions of sarcasm and irony.* (c) This study result can be used to compare the result of future study, in case if any major shift in people emotions and sentiments from these countries over the period.

III. Methodologies

To overcome limitations of traditional baseline multi-class classifier methods, to solve and mitigate the lexicon approach, the *context based SA, processing long text, model overfitting, accuracy, performance problems, and considering recent researched suggestions for future work on COVID-19* SA problems. Authors developed an extended BLSTM model name called SAB-LSTM. This experiment and methodologies are contributed towards to solve the following challenges.

A. SAB-LSTM Model

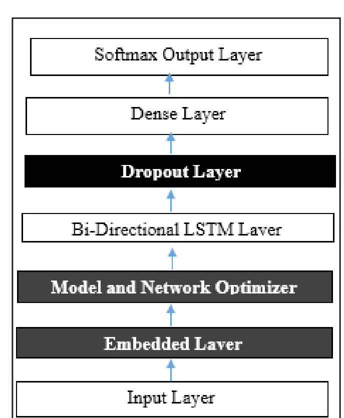

Fig. 1: Authors' SAB-LSTM Model Structure

To overcome lexicon approach limitations, accuracy, performance and overfitting problem, developed SAB-LSTM model which consist of six layers, the following Fig.1 shows the SAB-LSTM model's layer structure. Authors developed their own word embedding, model, network optimizer layers with custom dropout layers to control data in between the BLSTM, Dense and Output layers.

B. *To overcome Limitations of multi-class classifiers*

The traditional text classification algorithms SVM, Naive Bays, and Decision Tree are best fit for binary classification models, and it can be extended for multi-class with tuning. However, neural network provides an extension and flexibility to solve the multi-class problems. Adding N number of neurons with softmax function layer solve the multi-class problems. The N number of neurons in the network can be added based on number of classes

C. *To overcome Limitations of multi-class classifiers*

Traditional sentiment analysis models are not accurate in predicting context-based sentiment, most of the models are rule based. To solve the context based sentiment analysis problem, proposed **Bidirectional LSTM RNN models** which are proven to capture context of the information effectively, proposed SAB-LSTM model processes input data in both forward and backward directions and it memorize the context [22].

D. *Methodology and Process Flow*

The following Fig. 2 flow chart diagram shows the process flow of data pre-processing and model layers.

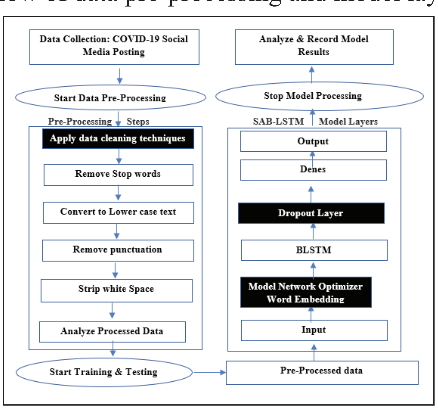

Fig. 2: Methodology and Process Flow Diagram

The social media data collected from various channels are not providing cleaned data and not suitable for training and testing the models. For data pre-processing steps, authors wrote their own Python scripts to clean the dataset, applied techniques to remove various noise data from various social media dataset to improve the data quality.

IV. Literature Review

In this section, reviewed RNN architecture of LSTM, BLSTM and explained about SAB-LSTM model. Discussed about how authors' SAB-LSTM model used to solve overfitting, performance and context-based sentiment problems.

A. *LSTM Review*

The main learning problem of RNN is long term dependency, vanishing gradient and exploding gradient. There are challenges in storing information for long extended time interval using recurrent back-propagation. It took long time Due to insufficient decaying error back flow. Sepp Hochreiter's analysis to solve these RNN problems, introduced a novel gradient-based LSTM method was proposed by Sepp Hochreiter and Jurgen Schmidhuber in 1997 and improved by Felix Gers' team in 2000 [18]. LSTM. In recent years, LSTM become a method used to solve ML problems. For sequential data analysis, the RNN with LSTM memory used and become a more effective and scalable model [17].

LSTM-RNN allows different variants and topologies are derived from traditional LSTM method. LSTM with various extensions and modifications of the algorithms like bidirectional LSTM, Gated Recurrent Unit (GRU), Grid LSTM or N-LSTM are used in practice [19. The Grid LSTM has its variant of Multidimensional LSTM and Stacked LSTM [19].

The following Fig. 3 shows the traditional RNN with full recurrent layer on the left diagram and LSTM with two gates and a memory block showed on the right. The hidden layer in LSTM is a memory block.

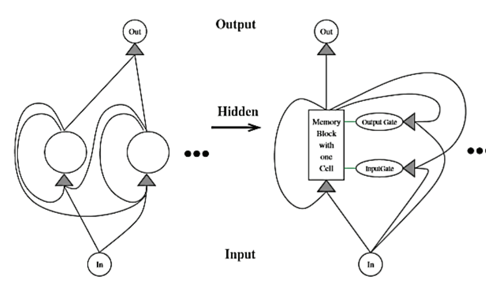

Fig. 3: Full Recurrent RNN on Left and a Memory Block LSTM with One Cell in the Hidden Layer

One or more memory cell constructs a memory block and a pair of adaptive, multiplicative input and output gating units gate the input and output to all cells in the memory block. Constant Error Carousels (CEC) is memory cell which has at its core a self-connected linear unit. The CEC is the central feature of LSTM and it solves the vanishing problem, and its activation called the cell state. CECs back flow is constant when there are no input or error signals to the cell. Input and output gates protect CECs error flow from *forward and backward activation. If the* gates are closed or the gates activation is around 0 then the irrelevant input will not enter the cell. The multiplicative gate units learn, controls the access to the cells and LSTM handles the constant error flow within the cells.

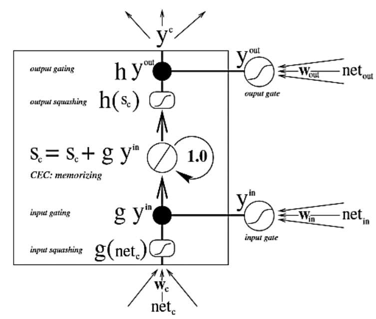

Fig. 4: A Linear Unit with Recurrent Self-connection of Traditional LSTM (One Memory Block With Signal Cell)

The following Fig. 4 shows a memory block with one cell of LSTM [21]. These functions of g squashes input and h squashes output to cell. Here indicates the cell state and 1.0 is the initial CEC weight. The *j* indexed a memory block, *v* indexed memory cell in block j with cell state, indicates the *v* cell of the *j* memory block, is the weight on connection from m to unit *l,f* represents input and output gate logistic sigmoid with range of [0, 1].

B. Forward pass

Forward pass computes the output of the network given the input data. Based on the cell input to cell itself net_c the cell state is updated, while net_{in} to input gate and net_{out} to output gate. To update all unites and compute the error signals for all weights, used a single discrete time steps t = 1, 2, 3, 4…*n*. The following equations are used to compute the input y^{in} and output y^{out} gates activations.

$$net_{in_j}(\mathrm{t}) = \sum_m \omega_{in_j m}\, y^m(t-1)\ ;$$

$$y^{in_j}(t) = f_{in_j}\left(net_{in_j}(t)\right) \tag{1}$$

$$net_{out_j}(\mathrm{t}) = \sum_m \omega_{out_j m}\, y^m(t-1)\ ;$$

$$y^{out_j}(t) = f_{out_j}\left(net_{out_j}(t)\right) \tag{2}$$

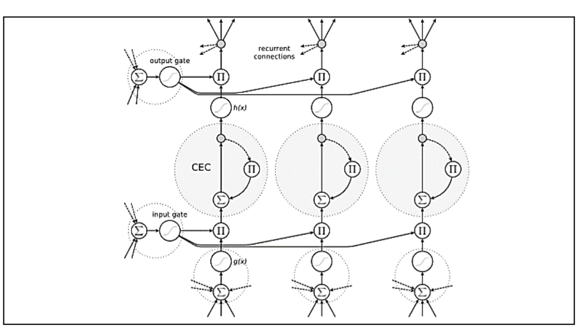

Fig. 5: LSTM Self-connections with Three Cell Memory Blocks

The Fig. 5 shows the recurrent self-connections of a three cell LSTM memory block. The one of the limitations of traditional LSTM is the memorizing efficiency, when cell states start to grow linearly during the progression of a continuous time series input stream. To maintain the cell state over time, the self-connected LSTM network is set the weight to '1' [19]. LSTM function as normal RNN neuron when all memory cell loses it memorizing capacity.

To overcome the limitation of LSTM unit memorization capability, attached the adaptive *Forget gate* to self-connection, so that Forget gate learns to reset the LSTM memory cell's internal state weight to 1.0 if the stored information in the memory is no longer required. The CEC with a progressive, forget gate activation computes weight using a similar logic for the other gates. The following Figure 6 shows extended LSTM with forget gate attached to the memory block [24].

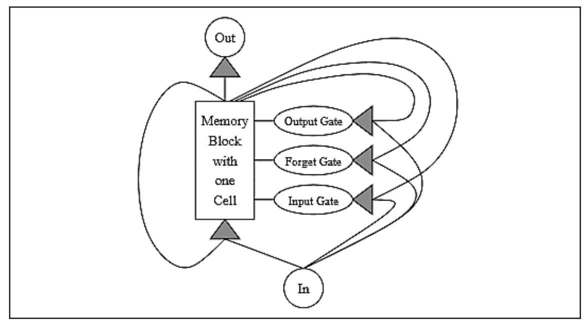

Fig. 6: Extended LSTM with Multiplicative Forget Gate and a Memory Block with One Cell

The Fig. 5 shows the extended LSTM with Input, Output, forget gates, one memory unit with a cell and all the gates are used activation function *f* sigmoid.

$$f(x) = \frac{1}{1+e^{-x}} \tag{3}$$

The following equations are used to compute the extended LSTM forward pass **forget gate activation** $\boldsymbol{y^{\varphi}}$.

$$net_{\varphi_j}(\mathrm{t}) = \sum_m \omega_{\varphi_j m}\, y^m(t-1)\ ;$$

$$y^{\varphi_j}(t) = f_{\varphi_j}\left(net_{\varphi_j}(t)\right) \tag{4}$$

net_{φ_j} is the input to forget gate from network, the logistic sigmoid with range of [0, 1] as squashing function f_{φ_j} Forget gate output becomes the weight to self-recurrent connection with inner state S_c . Fig.7 shows the extended LSTM.

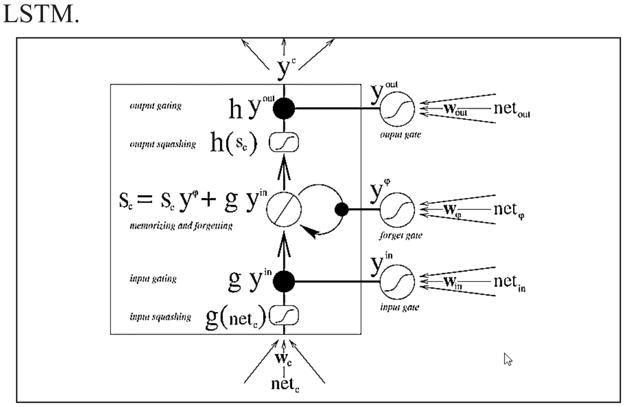

Fig. 7: Extended LSTM with Multiplicative Forget Gate

Here is the extended LSTM's inner state equation.

$$S_{c_j^v}(t) = y^{\varphi_j}(t)\; S_{c_j^v}(t-1) + y^{in_j}(t)\; g\left(netc_j^v(t)\right) \quad (5)$$

With $S_{c_j^v}(0) = 0$ and the bias weights are initialized with negative values for input and output gates, the positive values are initialized for forget gate.

C. Backward pass

Backward pass computes the output error with respect to the expected output and then go backward into the network and update the weights using gradient descent. To compute the network weight for a single input to output network, the back propagation (BP) use the loss function to compute the gradient. The following equation of backward pass for output gate derived [e.g. Felix Ger 2001] [21]. In this example, consider output unites use BP, output gates used truncated Real Time Recurrent Learning (RTRL) method, truncation cut off all errors when memory leaks out happen from cell or gate, so that errors can flow back forever via CECs only. In the following equation, consider function of *squared error objective* on the targets t^k;

$$E(t) = \frac{1}{2}\sum e_k(t)^2 \; ; \; e_k(t) := t^k(t) - y^k(t) \quad (6)$$

Here e_k indicates injected externally error, minimized E using gradient decent by adding weight change $\Delta\omega_{im}$ to the weights ω_{im} from output m to *l unit* using learning rate of α.

$$\Delta\omega_m(t) = \alpha \frac{\partial y^l(t)}{\partial net_l(t)} \left(\sum_k \frac{\partial y^l(t)}{\partial net_l(t)}\, e_k(t)\right) y^m(t-1) \quad (7)$$

For an arbitrary output unit $l=k^l$ the sum in reduces to e_k with $k=k^l$, *the BP* weight changes for the output unites based on the following equation.

$$\frac{\partial y^l(t)}{\partial net_l(t)} = f_k^l\left(net_k(t)\right) => \delta_k(t) = f_k^l\left(net_k(t)\right) e_k(t) \quad (8)$$

To compute the weight changes for the output gates $\Delta\omega_{out_j m}$ and set *l=out* the following equation is derived for output gate,

$$\frac{\partial y^{uj}(t)}{\partial net_k(t)} = f^l_{out_j}\left(net_{out_j}(t)\right),$$
$$\frac{\partial y^k(t)}{\partial y^{out_j}(t)} e_k(t) = h\left(Sc_j^v(t)\right)\; \omega_{kc_j^v}\;\; \delta_k(t) \quad (9)$$

All cells in a memory block contributes to weight changes of the output gate. To compute the weight, sum all *v* cells in a block *j* to get total of δout_j.

$$\delta_{out_j}(t) = f^l_{out_j}\left(net_{out_j}(t)\right)\left(\sum_{v=1}^{s_j} h\left(sc_j^v(t)\right)\sum_k \omega_{kc_j^v}\;\delta_k(t)\right) \quad (10)$$

D. Authors' SAB-LSTM Model

In this section, discussed about authors' sentiment analysis model SAB-LSTM. The following Figure 8 shows the architecture of the model. This model consists of the following six layers, Input, Embedded, BLSTM, Dense, Detection and Output layers.

Input and Word Embedding Layers: In this experiment after data pre-processing, work embedding process run via authors encode embedding methods, it encodes each word with an integer value and pass integers to BLSTM network. Allowed 2000 words in one social media posting or in a sentence with size of 128 embedding vector. The input dimension of (32, 2000, 128) used for training the model. The dataset consists of social media postings, each post has one or more sentences, and each sentence is composed with *n* number of words sequence. Here x representing input $x = x_1, x_2, x_3 \ldots\ldots x_t$ during the word embedding process each input word converted to real-value vector $S = (w_1, w_2, w_3 \ldots\ldots w_n)$ applying the embedding methods Input: $X = (X_1, X_2, X_3 \ldots X_n)$ denotes data matrix of n samples.

SAB-LSTM Topology: Authors model consists of 196 BLSTM memory units, 128 Embedding Layer, 5 Dense Layer and Softmax activation function at output layer.

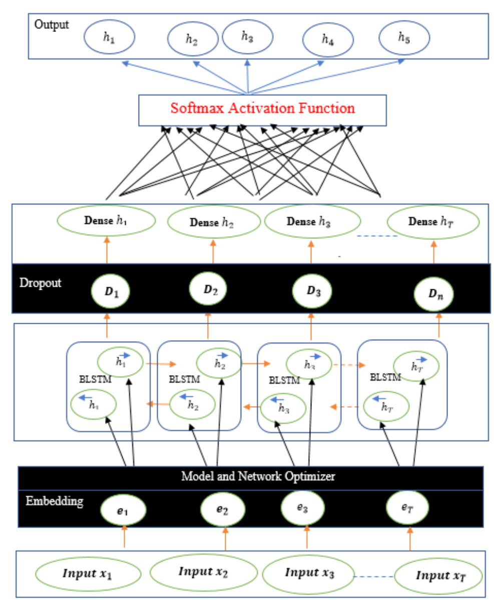

Fig. 8: SAB-LSTM Architecture

Model and Network Optimizer: Model and network optimizer provides the best suitable hidden layer and memory units along with activation function for a given dataset during the training process. It helps to minimize the computational time as well. This optimized function search for the neural network's total number of layers, and it optimizes the hidden layers using customized keras classifier function, add hidden layers to model, for example [196] will be one hidden layer with 196 memory units ,[100,150] two hidden layers with 100 and 150 memory units respectively. This function returns one of the best suitable hidden layer.

Input and Embedder layer pseudo code:

```
# Authors' Embedding Custom Input layer
# Initialize SA_Features (input text) function. Input social media text input data.
For each = Input validate the features.Data Pipeline for validate features.
Batch input for word vectorization.
# Initialize Embedding Layer parameters
For each max # of features and batch processing.
Initialize parameters for embedding
Input Dim, random array initialization.
Output Dim Embeddings initializer, regularize and Constraint and Activity, regularize
#Input data for embedding Input text data to keras Embedding
Encoder Input x=x1. x2, x3 ...xt
Output array = predicted vector(input_array) List the output_array
# Input the embedded input to BLSTM
#Execute BLSTM model code line
```

 # Model Network Optimizer Pseudo Code

```
Define and create layers (layers, activation)
Model = sequential (data)
For i # of nodes in enumerate(layers)
If i==0 then add dense (nodes, input dim=training shape)
and model activation(activation)),
Compile model (optimizer, loss, metrics)
Return model
Repeat model training with various
Provides layers = [[1920,9100,150],[200,250]]
Activations [sigmoid, softmax]
GirdSearch (estimator= model parameters =grid param)
Return result
```

BLSTM Layer: In this section, discussed specifics of bidirectional RNN (BRNN) and bidirectional LSTM (BLSTM), to overcome regular RNN limitations.

Consider time sequence of input vectors X_1^T and correspond sequence of output vectors Y_1^T for BLSTM discussion.

$$X_1^T = \{X_1, X_2, X_3, \ldots, X_{T-1}, X_T\}$$

$$Y_1^T = \{Y_1, Y_2, Y_3, Y_3, \ldots, Y_{T-1}, Y_T\}$$

The following Fig. 9 shows the BRNN structure of Forward State of positive and Backward State of negative time direction [25] with the three-time step. The forward states outputs are not connected to inputs of backward states and backward state outputs are not connected to input of forward states.

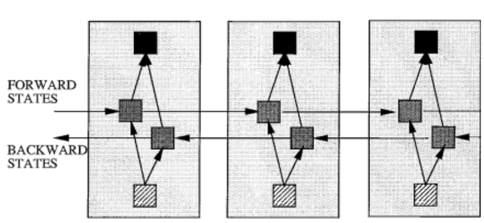

Fig. 9: Bidirectional RNN (BRNN)

The LSTM replace the hidden layer of BRNN will become a BLSTM recurrent neural network. In the previous section, discussed about forward, backward pass.

To summarize, following are the equations for gates and activation functions. LSTM Gates are the activation of sigmoid function, between 0 and 1 is output value of the sigmoid. When the gates are blocked the value is 0 and when the value is 1 then gates allow the input to pass through.

$$\text{sig}(t) = \frac{1}{1+e^{-t}} \tag{11}$$

Here are the equations for all three gates.

Input Gate: $i_t = \sigma(\omega_i [h_{t-1}, x_t] + b_i)$ (12)

Output Gate: $o_t = \sigma(\omega_f [h_{t-1}, x_t] + b_o)$ (13)

Forget Gate: $f_t = \sigma(\omega_o [h_{t-1}, x_t] + b_f)$ (14)

σ Represents sigmoid function

x_t Represents input at current timestamp.

h_{t-1} Represents LSTM block output of previous state at timestamp t-1.

ω_i, ω_f, and ω_o are represents weight for the input, forget and output gates.

b_i, b_o and b_f are represents bias for input, output, forget gates.

The following are equations for cell state for gates.

Cell State of Input gate :

$$\tilde{c}_t = tanh\,(c\omega\,[h_{t-1}, x_t] + b_c) \tag{15}$$

Cell Sate of Output gate:

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tilde{c}_t \tag{16}$$

Cell Sate of Forget gate:

$$h_t = o_t * tanh\,(c^t) \tag{17}$$

$\tilde{c}_t$ Represents input gate cell state at timestamp (*t*).

c_t Represents memory cell state at timestamp (*t*).

h_t Represents cell state of final output at timestamp (*t*)

Here is the pseudo code for bidirectional network.

 # # BLSTM pseudo code:

```
##Forward pass: All the Input data processed for one time slice:
For Time Slice 1 ≤ t ≤ T process through BRNN, determine all predicted output values. Do only the forward pass.
For forward state from t =1 to t = T and backward states
from t = T to t = 1 Do the forward pass for output neurons
##Backward pass: calculate the part of the objective function derivate for the time slice For Time Slice 1 ≤ t ≤ T used in forward pass.
Do backward pass for output neurons. Do only the backward pass.
For forward states from t = T to t = 1 and Backward states from t = 1 to t = T
## Update weights: to output gates in the network
Do Compute weight changes From all cells
For Output gates.
```

Dropout and Dense Layer: Dropout layer regularization method is reducing overfitting and improving model performance when input recurrent connections to BLSTM units are excluded from activation and weight updates during the training the network. In a neural network, a dense layer is the regular layer of neurons, each neuron is densely connected, and each neuron receives input from all the neurons from the previous layer. Every

 ISBN: 978-1-7281-8908-6

input is connected to every output by a weight for the linear operation.

```
##Dropout Layer pseudo code
# Import keras backend layers
Define method My Custom Dropout Layer:
Initialize self-method, output, and arguments:
Construct Super method to call dropout layer:
Build input dropout rate:
Call train to input.
Return input:
If training input
Return ft. dropout (input, rate)
Return input
# dense pseudo code
Initialize model
Initialize Dense layer
for each in I
Initialize activation
```

Output Layer: To calculate the distribution probability, the softmax function is used in the output layer for the event over *n* different events. It takes a class of values and converts them to probabilities with sum 1. It is effectively squashing a k-dimensional **a**rbitrary real vector values to k-dimensional real vector values within the range of 0 to 1. Here is the softmax function's equation.

$$\text{softmax}(z) = \frac{\exp(z)}{\sum_{k=1}^{K} \exp(z)} \quad (18)$$

```
# Sample prediction of the model output
Social media post = ["Most of this pandemic time society getting safe just because of Corona Warriors, even though they have family friends and society but still they are fighting with Covid, they all deserve our thanks and respect. Along with all 'Corona Warriors' happy to stand our Teaching community wonderful work"]
#vectorizing the social media post
Social Post = tokenizer.texts_to_sequences (post)
#Embedding layer process
#BLSTM layer processing
Activation Softmax output Multi-class
#print (post)
Sentiment = model. Predict (input)
If (np.argmax (sentiment) == 0):
Print ("neutral")
If (np.argmax (sentiment) == 1):
Print ("Happy")
If (np.argmax (sentiment) == 2):
Print ("Sad")
If (np.argmax (sentiment) == 3):
Print ("Depressed")
If (np.argmax (sentiment) == 4):
```

V. Experiment and Result

In this section, presented LSTM, BLSTM and authors' SAB-LSTM model experiment results and explain about the environment setting, model training and testing dataset.

A. Experiment Setting

To train and test the models, a standard server environment setup done, and developed models for experiment based on the following model structure shown in the following Fig. 10.

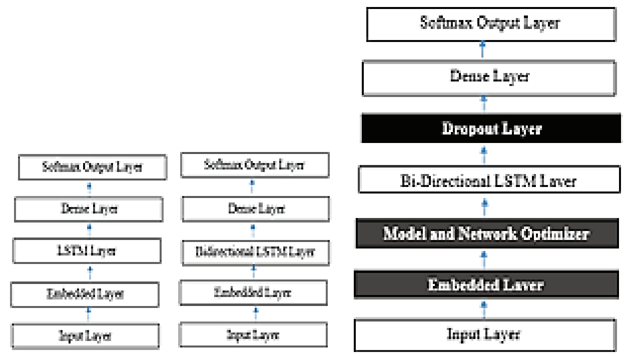

Fig.10: Left: LSTM, Middle: BLSTM & Right: SAB-LSTM

B. Data Collection, Model Training and Testing

For this experiment, collected COVID-19 related conversations posting in social media such as Twitter, Facebook, YouTube, Blogs, news articles and collected opinion from friends and family members. Total 80689 dataset collected and 72620 used for Training and 8069 posting used for testing. In this experiment, allowed 2000 words in one social media posting or in a sentence with size of 128 embedding vector. The input dimension of (32, 2000, 128) used for training the model at one forward and backward pass. Trained the models with various parameter setting to come up with optimized hidden layers and memory units (196) setting. All three LSTM, BLSTM and SAB-LSTM models are trained and tested multiple iterations and recorded model performance for different measures.

C. Model Performance Measures

Performance measures of recall, precision, F1 score are computed and prepared the confusion matrix for all the three models. Following Tables 3, 4 and 5 shows the multi class confusion matrix based on actual value and model predicated values. Calculated True Positive (TP), False Negative (TN), True Negative (TN), False Positive (FP) values from confusion matrix. In the below confusion matrix, the Ture positive (TP) values for classes are marked in bold numbers.

Table III: LSTM Model Confusion Matrix

Actual values ----------→

Class	Neutral	Happy	Sad	Depressed	Fear
Neutral	1263	234	153	15	1
Happy	341	1924	207	16	2
Sad	264	302	2459	39	2
Depressed	28	35	49	515	3
Fear	4	10	15	2	186

Table IV: BLSTM Model Confusion Matrix

Actual values ----------→

Class	Neutral	Happy	Sad	Depressed	Fear
Neutral	1273	244	128	19	2
Happy	374	1891	209	13	3
Sad	297	293	2438	31	7
Depressed	30	29	51	515	5
Fear	5	9	11	2	190

Table V: SAB-LSTM Model Confusion Matrix

Actual values ----------→

Class	Neutral	Happy	Sad	Depressed	Fear
Neutral	**1356**	175	120	13	2
Happy	220	**2119**	145	5	1
Sad	114	151	**2775**	23	3
Depressed	17	17	28	**566**	2
Fear	4	7	10	0	**196**

D. Model Results

Compared model performance measures, based on the result set. The authors' SAB-LSTM model performed better than traditional LSTM, and BLSTM models. The authors' word embedding layer, model and network optimizer layer helped to avoid the overfitting and improved the performance (during the training) and accuracy of the model. The authors' dropout and dense layers improved the accuracy of the prediction.

Table VI: Model Representation: L – LSTM, BL – BLSTM, SBL - SAB-LSTM.

	Precision			Recall			F1 Score		
Class	L	BL	SBL	L	BL	SBL	L	BL	SBL
Neutral	0.75	0.76	0.81	0.66	0.64	0.79	0.71	0.69	0.79
Happy	0.77	0.75	0.85	0.76	0.76	0.85	0.76	0.75	0.84
Sad	0.8	0.79	0.9	0.85	0.85	0.9	0.82	0.81	0.9
Depress	0.81	0.81	0.89	0.87	0.88	0.93	0.83	0.88	0.9
Fear	0.85	0.87	0.9	0.95	0.91	0.96	0.89	0.88	0.92

The following Table 6 shows all model performance measures. Based on these performance measures, authors' SAB-LSTM model scored higher F1 Score for all five classes and BLSTM F1 score is better than the traditional LSTM score. The following Table 7 shows the all three models' emotion % score and Table 8 shows the overall Sentiment score in % for COVID-19 dataset.

Table VII: Models Prediction in %

	LSTM %	BLSTM %	SAB-LSTM %
Sad (-ve)	36	35	37
Happy (+ve)	31	30	29
Neutral	24	25	21
Depressed (-ve)	7	7	7
Fear (-ve)	2	3	6

Table VIII: The Following Table 8 Shows Models Prediction % of the Sentiment About COVID-19.

Sentiment	LSTM %	BLSTM %	SAB-LSTM %
Positive	31	30	29
Negative	45	44	50
Neutral	24	25	21

The overall results show people sentiment is more negative than positive about COVID-19.

VI. Conclusion

This experiment result shows the authors' SAB-LSTM model performance is better than the traditional LSTM and BLSTM models. Adding authors' additional layers helped the models to avoid the overfitting problems and optimized the model parameters dynamically for the given dataset. The validation of the SAB-LSTM result shows more **context-based sentiment** prediction than the traditional LSTM models. The results show COVID19 public sentiment is more negative. To get the worldwide pandemic sentiment analysis, need multi language social media dataset. In this experiment, used only English social media posting from various social channels.

A. Future Work

The future research work needed for exploring a framework to process the Multi-language social media posting and data pipe line for continues mode training. Developing a multi-class classifier for various features of pandemic will help to analyze nature of the pandemic. The SAB-LSTM model can be improved and used for context-based sentiment analysis problems for various domain. Acknowledgement

References

[1] Karishma Sharma, Sungyong Seo, Chuizheng Meng, Sirisha Rambhatla, Yan Liu, " COVID-19 On Social Media: Analyzing Misinformation in Twitter Conversations", Department of Computer Science, University of Southern California, Los Angeles, USA. arXiv:2003.12309v3 [cs.SI] 2020.

[2] Bennett Kleinberg, Isabell evan der Vegt, Maximilian Mozes, "Measuring Emotions in the COVID-19 Real World Worry Dataset" University College London. 2020.

[3] Fairoza Amira Binti Hamzaha , Cher Han Laub , Hafeez Nazric , Dominic Vincent Ligotd , Guanhua Leee , Cheng Liang Tanf , Mohammad Khursani Bin Mohd Shaibg , Ummi Hasanah Binti Zaidonh , Adina Binti Abdullahi , Ming Hong Chungj , Chin Hwee Ongk , Pei Ying Chewl and Roland Emmanuel Salungam. "CoronaTracker: World-wide COVID-19 Outbreak Data Analysis and Prediction" CoronaTracker Community Research Group. 2020.

[4] Akash D Dubey, Assistant Professor, "Twitter Sentiment Analysis during COVID19 Outbreak "Jaipuria Institute of Management, Jaipur. 2020.

[5] Deena Dimple Dsouza, Shalini Quadros, Zainab Juzer Hyderabadwala, and Mohammed A. Mamun, "Aggregated COVID-19 suicide incidences in India: Fear of COVID-19 infection is the prominent causative factor". 2020.

[6] Shelby R. Buckman, Adam Hale Shapiro, Moritz Sudhof, and Daniel J. Wilson, "News Sentiment in the Time of COVID-19", Research from Federal Reserve Bank of San Francisco. [https://www.frbsf.org/economic-research/files/el2020-08.pdf]. 2020.

[7] Amy B Sullivan PsyD, ABPP , Alexa Kane PsyD , Alicia J Roth PhD , Bryan E Davis PsyD , Michelle L Drerup PsyD , and Leslie J Heinberg PhD, "The COVID-19 Crisis: A Mental Health Perspective and Response Using Telemedicine", [https://doi.org/10.1177/2374373520922747], 2020.

[8] C.J. Hutto, Eric Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text", Georgia Institute of Technology, Atlanta, GA 30032. Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved. 2014.

[9] Emily Chen, Kristina Lerman, Emilio Ferrara, "COVID-19: The First Public Coronavirus Twitter Dataset ", University of Southern California, Information Sciences, Marina del Rey, CA 90292. arXiv:2002.0737372v1 [cs.SI]. 2020.

[10] Abhilash Mittal, Sanjay Patidar, "Sentiment Analysis on Twitter Data: A Survey" Delhi Technological University, Ne Delhi, India. ACM. © 2019 Association for Computing Machinery. ACM ISBN 978-1-4503-7195. 2007.

[11] Minqing Hu and Bing Liu, "Mining and Summarizing Customer reviews" University of illinois at Chicago. [International conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA] 2004.

[12] Bo Pang and Lillian Lee, "A Sentimental Education: Sentiment analysis using subjectivity summarization based on minimum cuts' Cornell University, Ithaca, NY. arXiv:cs/0409058v1 [cs.CL]. 2004.

[13] Bo Pang, and Lillian Lee, "Opinion mining and sentiment analysis" [Foundation and Trends in Information Retrieval vol 2] . 2008.

[14] Jim Samuel, G.G.Md.Nawaz Ali, Ek Esawi and Yana Samuel, "COVID-19 Public Sentiment Insights and Machine Learning for Tweets Classification"[doi:10.3390/info11060314 www.mdpi.com/journal/information]. 2020.

[15] Hong Yu and Vasileios Hatzivassiloglou, "Towards Answering Opinion Questions: Separating facts from opinions and Identifying the polarity of opinion sentences" [Conference on Empirical Methods in NLP] January 2003.

[16] Theresa Wilson, Janyce Wiebe and Paul Hoffmann, "Recognizing Contextual polarity in Phrase-Level Sentiment Analysis" . 2005.

[17] Klaus Greff, Rupesh K. Srivastava, Jan Koutn´ık, Bas R. Steunebrink, J¨urgen Schmidhuber, "LSTM: A Search Space Odyssey" arXiv:1503.04069v2 [cs.NE] . 2017.

[18] Sepp Hochreiter, Jurgen Schmidhuber, "Long Short-Term Memory" Neural Computation 9(8)1735-1780, 1997.

[19] Ralf C. Staudemeyer, Eric Rothstein Morris, "Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks"arXiv:1909.09586v1 [cs.NE]. 2019.

[20] Shervin Minaee, , Elham Azimi, AmirAli Abdolrashidi, "Deep-Sentiment: Sentiment Analysis Using Ensemble of CNN and Bi-LSTM Models" arXiv:1904.04206v1 [cs.CL]. 2019.

[21] Felix Gers "Long Short-Term Memory in Recurrent Neural Networks" Lausanne, EPFL. 2001.

[22] Peilu Wang, Yao Qian, Frank K. Soong, Lei He, Hai Zhao "A unified Tagging Solution: Bidirectional LSTM Recurrent Network with Word Embedding. 2015.

[23] Cuixian xu, Yueting Meng, Xiaoyu Qiu, and Xu Wu "Sentiment Analysis of Comment Texts Based on BiLSTM". Publication of IEEE Access. 2019.

[24] Felix A. Gers, Nicol N. Schraudolph, Jurgen Schmidhuber, "Learning Precise Timing with LSTM Recurrent Networks" [Journal of Machine Learning Research 3 115-143]. 2002.

[25] Mike Schuster and Kuldip K. Paliwal, "Bidirectional Recurrent Neural Networks" IEEE Transaction of Signal processing Vol 45. 1997.

 Copyright © IEEE–2020 ISBN: 978-1-7281-8908-6