Utilizing Gated Recurrent Units to Retain Long Term Dependencies with Recurrent Neural Network in Text Classification

Nidhi Chandra*
Research Scholar, Amity Institute of Information Technology, Amity University Uttar Pradesh, India
Srivastavanidhi8@gmail.com
Dr. Laxmi Ahuja
Dy. Director, Amity Institute of Information Technology, Amity University Uttar Pradesh, India
lahuja@amity.edu
Dr. Sunil K. Khatri
Director, Amity University Tashkent, Uzbekistan.
skkhatri@amity.edu
Dr. Himanshu Monga
Professor, Jawaharlal Lal Nehru Government Engineering College, Sundernagar, Mandi (H.P).
himanshumonga@gmail.com

Received: 11/Jul/2020 Revised: 21/Apr/2021 Accepted: 12/May/2021

Abstract

The classification of text is one of the key areas of research for natural language processing. Most of the organizations get customer reviews and feedbacks for their products for which they want quick reviews to action on them. Manual reviews would take a lot of time and effort and may impact their product sales, so to make it quick these organizations have asked their IT to leverage machine learning algorithms to process such text on a real-time basis. Gated recurrent units (GRUs) algorithms which is an extension of the Recurrent Neural Network and referred to as gating mechanism in the network helps provides such mechanism. Recurrent Neural Networks (RNN) has demonstrated to be the main alternative to deal with sequence classification and have demonstrated satisfactory to keep up the information from past outcomes and influence those outcomes for performance adjustment. The GRU model helps in rectifying gradient problems which can help benefit multiple use cases by making this model learn long-term dependencies in text data structures. A few of the use cases that follow are – sentiment analysis for NLP. GRU with RNN is being used as it would need to retain long-term dependencies. This paper presents a text classification technique using a sequential word embedding processed using gated recurrent unit sigmoid function in a Recurrent neural network. This paper focuses on classifying text using the Gated Recurrent Units method that makes use of the framework for embedding fixed size, matrix text. It helps specifically inform the network of long-term dependencies. We leveraged the GRU model on the movie review dataset with a classification accuracy of 87%.

Keywords: Gated Recurrent Units; Recurrent Neural Network; Word Embedding; Deep Learning; LSTM.

1- Introduction

It has been more than a decade that we are communicating with machines through handwritten codes or programs. It has been always a human dream that machines should understand their language or in other words, they should be able to speak to machines and machines should respond in the same language. Natural language processing helped to make this dream come true. Natural language processing is analyzing and making digital sense to the natural language spoken by humans across the geographies or it is also referred to as machine processing of human languages to channelize interaction between human and machine. There are multiple applications to NLP such as IVR systems integrated with chatbots, standalone chatbots, etc. Where an individual can connect through his phone and speak his query, which is then translated into machine understandable form and processed to address end-user query. There are live robots these days which not only understand language types but also respond in similar language.

NLP uses numerical and statistical techniques to convert textual data into numerical data which machines could understand and maps this data to machine learning and deep learning algorithms to help to bridge the gap of communication between humans and machines.

The text has to be processed in multiple stages or phases before making it machine-readable or understandable also

* Corresponding Author

known as text processing or filtering and helps to serve multiple purposes. This process is corpora dependent and requires text preparation to be performed to make it enable to input into an appropriate machine or deep learning algorithms.

Because of the dependency of machine learning and deep learning algorithms on numerical data to give the best results, word embedding [1][14] plays a critical role to transform preprocessed corpora into numerical types. Word embedding leverages real-value vector representations of words which supports models in predicting and understanding words. The two key algorithms to be used for this purpose are – Word2Vec and Glove [17] [29][40].

Another technique that follows the same approach as machine learning is deep learning which leverages artificial neural networks as computing models. The ANN [27][22] technique is inspired by the network of neurons in the human brain and how they store information in a form of layers. It also tries to depict how the information is retrieved from these neural nets. In computer science implementations these neural nets were implemented as connected nodes forming a network similar to that inside the human brain. These nodes are responsible to learn and store information like text, real-life objects Etc. The neural nets are a collection of layers whose numbers can range from three to hundreds. They are further classified as shallow and deep neural networks depending on the number of layers in them. Shallow networks are confined to three to four layers while deep networks have more than four layers. Because of a greater number of layer processing deep learning models are preferred over shallow networks in complex tasks like facial recognition, text translation, etc.

2- Literature Survey

Lately, there is a functioning pattern towards utilizing different AI strategies for taking care of issues identified with Natural Language Processing (NLP). One of these issues is the programmed recognition of emotion. The investigation of sentiment and emotions has an elaborative history. Sentiment analysis is contextual mining of text which distinguishes and extricates emotional data in the source material and helping a business to comprehend the social feeling of their image, item, or administration while observing on the web discussions.

No such extensive survey exists which should discuss various approaches which researchers are applying to identify the shortfall of an exhaustive report to investigate the collection of different patterns.

This comprehensive survey [12] results depicts the entire, agreed upon, and planned review of views or judgment and emoticon analysis for classification of methods to have comparative analysis for better comprehensions.

Socher et al. [34] has presented a hierarchical structure that centers on the perspective explicit investigation of emotions. To extract labels at the phrase level, they created novel d-dimensional vector portrayals for terms**,** built up a profound learning framework including managing highlight**,** portrayals of sentence parses, which adds to the assurance of a target task Comparison is made of multi-vector RNN and recursive neural tensor system nearby vanilla RNN for this. Using their collaborative multi-aspect feeling layout, these are appended to perspective and feeling labels. They differentiated their sentiment pair recognition model for single and joint viewpoint and differentiated it against multiclass SVMs and Naive Bayes classifiers.

Junyoung et. al. [19] proposed a novel architecture for deep-stacked RNNs. Their study Suggested

RNN, gated-feedback RNN (GF-RNN), expands the existing method of stacking multiple recurring layers by enabling and regulating signals flowing from the upper recurrent layers to lower layers using a global gating unit for each pair of layers. Experiments focused on challenging sequence modeling tasks of character-level language modeling.

Liu et al. [13] presented a hybrid method for bilingual text sentiment classification based on a deep learning approach that combined machine learning with deep learning to provide a stronger result in recognition of feelings.

Chen, Huimin, et al. [5] presented a model that first builds a hierarchical LSTM model to generate sentence and document representations. Afterward, user and product information is considered via attention over different semantic levels due to its ability to capture crucial semantic components. This paper proposes a hierarchical neural network that incorporates user and product information via word and sentence level attentions. With the user and product attention, our model can take account of the global user preference and product characteristics at both word-level and semantic levels.

The consolidated set of words are processed through the process of embedding layer wherein each token is classified as a variable-sized vector with actual meaning, which is many of times referred to as word embedding. There have been experimentations on few specific methods of word embedding initialization such as random Glove [11][36] and SSWE [11]. For the preparation one can improve the model base on the embedding of word learning through word function.

Johnson and Zhang [20] suggested a CNN variant called BoW-CNN which uses bag-of-word conversion in the convolution layer. They also presented a model, called

Seq-CNN, which preserves sequential word knowledge by concatenating the multiple word one-hot vectors.

Tang et al. [32] proposed a neural network to learn the representation of documents, considering the relation between sentences. Next, it learns from word embedding's the sentence representation with CNN or LSTM. A GRU is then used for the adaptive encoding of sentence semantics and their underlying relationships in text representations for the classification of sentiments.

Tang et al. [33] user representations applied, and product representations listed in the analysis. The hope is that such representations can capture essential global clues like individual user expectations and overall product quality which can provide better representations of text.

O Yildirim [28] presented a new model for deep bidirectional LSTM network-based wavelet sequences called DBLSTM-WS was proposed for classifying electrocardiogram (ECG) signals. The new wavelet-based layer is implemented to generate ECG signal sequences. The ECG signals were decomposed into frequency sub-bands at different scales in this layer.

These sub-bands are used as sequences for the input of LSTM networks. New network models that include unidirectional (ULSTM) and bidirectional (BLSTM) structures are designed for performance comparisons. Experimental studies have been performed for five different types of heartbeats obtained from the MIT-BIH arrhythmia database.

B. Athiwaratkun et. al. [4] propose several new malware classification architectures which include a long short-term memory (LSTM) language model and a gated recurrent unit (GRU) language model. He proposed an attention mechanism in addition to temporal max-pooling as an alternative way to construct the file representation from neural features. A new single-stage malware classifier based on a character-level convolutional neural network (CNN) is proposed in this study. Results show that the LSTM with temporal max pooling and logistic regression offers a 31.3% improvement in the true positive rate.

M Zulqarnain et. al. [25] proposed a unified structure to investigate the effects of word embedding and Gated Recurrent Unit (GRU) for text classification on two benchmark datasets included (Google snippets and TREC). GRU is a well-known type of recurrent neural network (RNN), which is the ability to compute sequential data over its recurrent architecture. First, words in posts are changed into vectors via the word embedding technique. Then, the words sequential in sentences are fed to GRU to extract the contextual semantics between words. The experimental results showed that the proposed GRU model can effectively learn the word usage in the context of texts provided training data

Luo, L et. al. [24] improve the performance of internet public sentiment analysis, a text sentiment analysis method combining Latent Dirichlet Allocation (LDA) text representation and convolutional neural network (CNN) is proposed. First, the review texts are collected from the network for preprocessing. Then, using the LDA topic model to train the latent semantic space representation (topic distribution) of the short text, and the short text feature vector representation based on the topic distribution is constructed. Finally, the CNN with the gated recurrent unit (GRU) is used as a classifier. According to the input feature matrix, the GRU-CNN strengthens the relationship between words and words, text and text, to achieve high accurate text classification.

Mostly NER is performed in English rather than Chinese, and the current impact of named entity recognition in Chinese is not very satisfactory due to the complex language characteristics of Chinese. S. Yan et al. suggest a new network framework called BERT-BGRU-MHA-CRF to address the above issues [30]. Experiments show that the model is capable of achieving an F1 value of 94.14.

To detect rumors, Zhou et al. used a combination of convolutional neural networks and gated recurrent unit networks [39]. Their model vectorizes rumor events, then uses CNN to automatically create rumor microblog features, and then uses GRU to mine temporal information between related microblogs under rumor events. The C-GRU model outperforms a sequence of classic models in the experiments.

Y. Pan et al. proposed the Bi-GRU (bidirectional GRU neural network) and attention mechanism model to analyze Chinese text sentiment to solve the problem of high complexity and low efficiency in Chinese LSTM-based Chinese text sentiment analysis [38]. To learn the text features more accurately, the model extracts deep features from the text and combine them with the meaning of the sentence. Multi-Head Self-Attention is implemented in the method, which eliminates dependency on external parameters, assigns weights to word vectors, highlights text attributes, and pays more attention to internal sentence dependencies. Experiments show that the model-based Chinese text sentiment analysis can achieve an accuracy of 87.1 percent.

H. Wang et al. suggested a hybrid learning-based emotion classification model [16]. To measure emotion scores in the entire data collection, the enhanced dictionary classification system is used, and data with high or low scores are explicitly marked; the rest of the method is based on emotion dictionary and Bi-GRU fusion model to calculate emotion score. The COAE2014 (Chinese opinion analysis and evaluation 2014) dataset microblog experiment with emoticons reveals that a single model is not suitable for several forms of diverse contexts, and it is difficult and inaccurate. The multi-model fusion method can effectively boost the single model's error preference and classification effect.

3- Review of Deep Learning

Deep learning helps us extend machine learning algorithms to improve results on text, image, and voice. The algorithms designed and implemented through deep learning draws similarities to stimuli and neurons of the human brain. One finds implementations of deep learning algorithms in electronic vision, computer-based translation, and recognitions.

It is a subfield for machine learning, as both approaches adopt the same primary principle — both machine learning and deep learning algorithms take input and use it to predict the output.

The objectives of Machine learning and deep learning algorithm(s) is to reduce differences between actual and predicted results after being trained on the training dataset. This helps to achieve higher accuracy by establishing the relationship between input and output. Machine learning models despite being self-sufficient need human inputs and feedbacks to confirm whether the prediction is correct or not by itself [23][35]. This, leading to the conclusion that deep learning models are autonomous, thereby making decisions to improve the effectiveness without human intervention. Even though neural networks were lost interest by the research fraternity in the late '90s being computationally very costly, but due to advancement in CPU technology in the past 10 years, several breakthroughs were made in this area and made progress in computer vision, speech, and NLP[10][15]. The key attributes considered for the revival of neural networks are the availability of high-speed computing resources like GPU and vast quantities of training data on the Internet [3].

Recurrent neural networks are used when a sequence of data shifts over time. It is like a conventional feed-forward network, the difference being that it has connections to the same layer of units. RNN can make use of the information in arbitrarily long sequences, but in practice, the standard RNN is limited to looking back only a few steps due to the vanishing gradient or exploding gradient problem [2].

The Long Short-Term Memory Network is a special RNN type capable of learning Dependencies in the long term [7]. LSTM and GRU solve the issue of vanishing gradient too. A slight variation of LSTM is the Gated Recurrent Unit [8].

To learn document representation, Tang et al. [32] suggested a neural network, with the consideration of Relationships in sentences. The sentence representation with CNN or LSTM is first learned from the word Embedded. Then a GRU is used to adaptively encode sentence semantics and their underlying semantics. Links in text representations for classification of emotion.

Chen et. al. [6] proposed to utilize a recurrent attention network to better capture the sentiment of complicated contexts. To achieve that, their proposed model uses a recurrent attention structure and learns a non-linear combination of the attention in GRUs.

Research	Model Used	Dataset	Results
[37] Turian, Joseph et. al.	Tree-LSTM	Sentences Involving Compositional Knowledge (SICK) dataset	Tree-LSTM used for sentiment classification of sentences sampled from movie reviews.
S. Yan et. al. [30]	BERT-BGRU-MHA-CRF	Chinese Language Scripts	Model can achieve F1 value of 94.14.
Zhou et al. [39]	CNN+GRU	Chinese blogs dataset	. The CNN-GRU model outperforms a sequence of classic models in the experiments .
Y. Pan et al. [38]	Bidirectional - GRU	Chinese scripts	Bidirectional - GRU can achieve an accuracy of 87.1 percent.
D. Tang et al. [11]	Conv GRNN and LSTM-GRNN	IMDB and Yelp Dataset	Gated recurrent neural network outperforms standard recurrent neural network in document

			modelling for sentiment classification
M. U. Salur et al. [26]	12 deep learning models and CNN, LSTM, Bi-LSTM, GRU	GSM operator Turkey dataset	CNN+ bi LSTM has higher accuracy i.e. 82.14% as compared to CNN.
] Johnson R et al. [20]	CNN on high-dimensional text data	IMDB dataset	CNN outperforms with, bag-of-n gram approach or word-vector.
Chen P et al. [6]	LSTM, Bi-LSTM, RNN, GRU	SemEval 2014, Tweet Dataset, Chinese news reports	Recurrent Attention on Memory (RAM) with 4 attentions and not trained data set, outperformed compared to rest multi attentions scheme

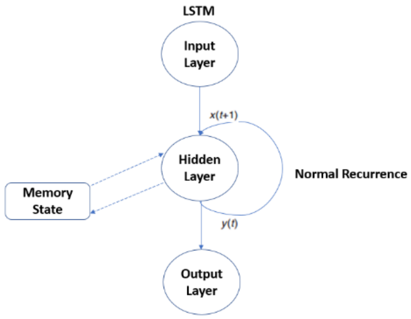
Table 1: Summary of Methods Related to Proposed Approach

The approach presented in this paper is based on [6][11] as depicted in Table 1 to build a GRU-based classifier for a binary polarized dataset [41].

4- Long Short-Term Memory Networks

In 1997, Sepp Hochreiter and Jürgen Schmidhuber first introduced LSTM networks and addressed the problem of retaining information for RNNs for longer periods [37].
RNNs have proven to be the only option to handle sequence classification issues and have proved acceptable to maintain the data from previous results and leverage those results for modification of outputs.
The saver here is an LSTM network which is an RNN architecture that helps to train models over lengthy sequences and to retain memories out of previous input time steps fed into the model.
This would help in handling the problem of gradient disappearance or explosion through the introduction of added gates, inputs, and forgetting gates, enabling better gradient regulations, helping to enable - Information to maintain and forget, thereby controlling the access to Information for the current cell's state helping in the preservation of "long-range dependencies"[2][31].

LSTM comprises memory cells to store information. It computes the input gate, forgets gate, and output gate to manage this memory. LSTM units can broadcast some critical features that inculcate early in the input sequence over a long distance thereby capturing critical long-distance dependencies [7] as depicted in Figure (1).



.

Figure 1. Long-Distance Dependencies

In LSTMs, the rules that govern the state-saved information (memory) are themselves trained neural networks — therein lies the magic. The network can be programmed to know what to recall, while the rest of the recurrent system will at the same time learn to predict the target.
Input influences the memory state (see Figure 1) and influences the layer output as well as a regular recurrent system. But the state of memory continues in all phases of the time series (your sentence or document). So, each input will impact both the memory state and the output of the hidden layer. The memory state mystery which it knows when to recall. Memory state wonders that when learning to reproduce the output by regular backpropagation, it knows what to remember.
Although complex, LSTM is very competitive in various tasks such as the recognition of handwriting, machine translation, and, of course, the analysis of sentiments.
They're typically slower than other standard ones, apart from the LSTM network complexity. Also, an RNN with better initialization and planning, and with less computational complexity, can produce results close to those of LSTM. Besides, where recent information is more important than older information, the LSTM model is often a better choice.

4-1- Gated Recurrent Units:

There are wide ranges of variants of LSTM that are in use today like the Gated recurrent unit (GRU) that creates an update gate through the consumption forget and input gates. It combines the state of the cell using a hidden state and modifies the generated output making target models less complex than regular ones.

GRU can control the flow of information without the need to use a memory unit like LSTM which is proved best to work with large datasets [32]. On the contrary GRU maps to smaller data sets. It is not mandatory though as efficiency would depend to some degree on the complexity of data and model.

4-2- GRU for Text Classifier

Sentiment/ emotion analysis is a common case of use when applying the techniques of natural language processing. Sentiment analysis aims to determine whether to interpret a given piece of text as conveying a 'positive' feeling or a 'negative' feeling.

"The movie was so awesome that I slept peacefully for hours."

To a human reader, it is painfully clear that the book review referred to here transmits a negative feeling. So, how do you build a model of machine learning to recognize sensations? As usual, the use of a supervised learning method requires a text corpus that contains many samples. Each piece of text in this corpus should have a label indicating whether a positive or negative emotion can be mapped to the text.

By looking at the example above, you can already see that such a function might be difficult for a machine learning model to solve. By using a simple tokenization or TFIDF approach, the classifier can quickly misinterpret words like 'wonderful' and 'peacefully' to convey a positive feeling.

To make matters worse, the text contains no word which can be translated explicitly as negative. This observation also brings in the need to link various sections of the text structure so that meaning can be taken from the sentence. The first sentence can be broken up into two parts, for example:

"The movie was so awesome"
"I slept peacefully for hours"

Looking at just the first part of the paragraph, you might infer it's a good remark. It is only when the second sentence is understood that the context of the sentence can be completely interpreted as expressing negative feelings. And there has to be long-term dependency here. And a simple RNN for mission is not good enough.

4-3- Structure of GRU

Gated Recurrent Unit (GRU) ascertains two gates that control data move through each hidden state, called update and reset gates. The existing input and a vector comprising of earlier hidden states should be taken as input. The value of three isolated [28] gates should be calculated by using the below steps –

1. The parameterized input and the hidden state vector need to be calculated for each gate through element-wise multiplication referred to as Hadamard multiplication between vector and weights of each gate.
2. The activation function needs to be applied to respective gates as element by element basis on the parameterized vector.

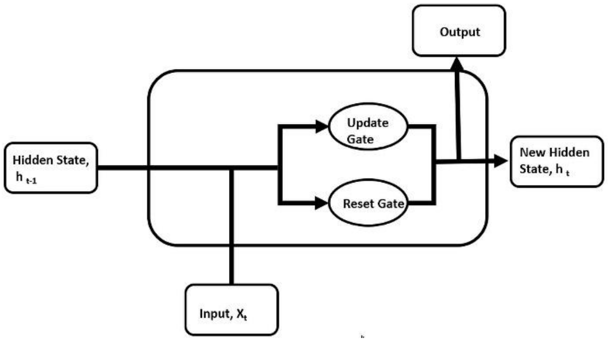


Figure 2: GRU Memory States

In contrast to the LSTM, GRU only has two gates Reset Gate and Update Gate as shown in figure (2). Each hidden state is determined in time-step t utilizing the accompanying conditions.

Reset Gate g_r as explained in Eq. (1) corresponds to the summation of input gate and forget gate in the LSTM network which helps us find the forgotten knowledge of the past. Here X_t represents the input vector and $h_{t-1}$is a hidden state at time t.

$$g_r = \sigma(W_{rx}X_t + W_{rh}h_{t-1} + b) \quad (1)$$

Update gate g_u corresponds to the output gate of the LSTM recurrent unit which helps determine the previous knowledge which should be transit to the future. The sigmoid function is used as an activation function in the update gate as well as in the reset gate [32].

$$g_u = \sigma(W_{ux}X_t + \ W_{uh}h_{t-1} + b) \quad (2)$$

As shown in Eq. 2 h_t represents the hidden state at time t and h_{t-1}depicts the hidden state at time t-1

The new hidden state (proposed) state at time t, i.e. $\breve{h}_t$ is calculated as per the Eq. (3)

$$\breve{h}_t = tanh\,(W_{hx}\,X_t\ + W_{hh}\ .\,(\,g_r.\,h_{t-1}) + b) \qquad (3)$$

First, the Hadamard product of the Reset Gate and the previously hidden state vector as shown in Eq. 3 are calculated. Then this vector is parameterized and then added to the parameterized current input vector.

Hidden state at time t i.e. h_t is often ignored during the evaluation of GRU network and is often included into Reset Gate similar to Input Modulation gate which is a section of input gate and helps to incorporate non-linearity in input and makes it zero-mean as shown in figure (2). It also helps to decrease the effects of prior details on existing Information that are passed into the future.

$$h_t = (1 - g_r).\,h_{t-1} + g_u.\ \breve{h}_t \qquad (4)$$

In this Eq. (4), g_r as Reset gate, g_u as Update gate, and h_t as a hidden state at time t, h_{t-1}= hidden state at time t-1, $\breve{\mathrm{h}}_t$=New Hidden state at time t are taken to calculate the current memory gate.

GRU Gates incorporates initiations of the sigmoid. Like tanh activation, a sigmoid activation Instead of squishing values from -1 to 1 it squishes values from 0 to 1 as shown in figure (3). This is useful for updating or losing data as any number multiplying by 0 is 0, causing values to vanish or be "forgotten. "Any number multiplied by 1 is the same value so that the value remains the same or is "kept." Therefore, the system can realize which information isn't significant and which information ought to be overlooked or kept.

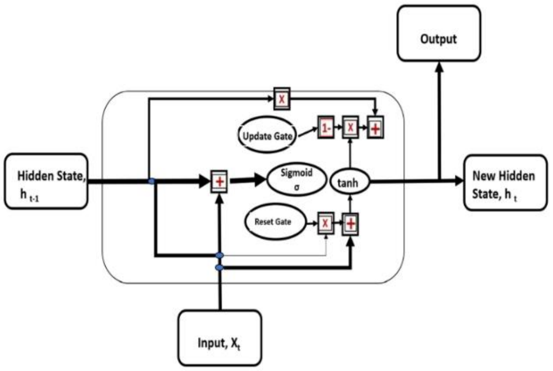


Figure 3: GRU Architecture and Functions

In the first place, we have the forget gate. This gate decides which information to throw away or to keep. Information from the previous hidden state is passed through the sigmoid function and information from the current input. Values range from 0 to 1, the closest to 0 means missing, and the closest to 1 means holding off.
We have the input gate to change cell status. Second, the previous hidden state and current input are passed into a sigmoid function. That determines what values are changed by converting the values between 0 and 1. 0 Does not mean important and 1 does mean importance. In the tanh method, you also transfer the hidden state and current input to squish values between -1 and 1 to help regulate the network. The tanh output is then multiplied by the sigmoid output. The sigmoid output will determine which information to retain from the tanh output is significant.
Gates are merely neural networks that regulate the flow of information through the sequence chain.

In the first place, we have the overlook entryway. This door chooses which data to discard or to keep. Data from the past shrouded state is gone through the sigmoid capacity and data from the current information. Qualities extend from 0 to 1, the nearest to 0 methods missing and the nearest to 1 method holding off.

5- Experimental Setup

The experiments were done on the Anaconda environment which is an open-source package manager, environment manager, and distribution of the Python and R programming languages. This uses the Python 3.6 libraries. The environment was set up on the cloud using Intel Core(TM) i3 @3.4GHz 16GB of RAM and GeForce GTX 1060 GPU on CentOS 7. Keras and TensorFlow libraries were used as supporting libraries and code was written on Jupyter notebook which was set as part of anaconda. To set up the Python environment for running notebook is as h5py-2.9.0, Keras-2.2.4, NumPy-1.16.1, Tensorflow-1.12.0.

5-1- GRU Modelling for Text Classification

A GRU is an extension of a simple RNN which helps to counter the problem of the vanishing gradient by allowing the model to learn long-term dependencies in the structure of the text.

This paper focuses on classifying tweets using the Gated Recurrent Units method that makes use of the framework for embedding fixed size, matrix text. GRUs helps specifically inform the network of long-term dependencies. This is done by the addition of more variables into a basic RNN structure. The GRU layer uses the update gate to decide the amount of preceding information that should be passed on to the next activation while using the reset gate to determine the amount of preceding information that should be forgotten. The GRU update gate behaves similarly to an LSTM's forget gate and input gate. This decides what information you should throw away and what new information you should introduce. The GRU reset gate is another gate that is used to decide how much past information to forget.

GRU's have fewer tensor operations, so they're a little easier than LSTM's to practice. Which one is better, there is no definite winner.

This helps to ease the pressure of RNN in taking on long-term dependencies. For a long time, the GRU model integrates additional elements in the attestation process and becomes more complex while allowing further operations on secret GRU states. With no long-term dependencies, GRU will encode the variable-length series into a fixed-size representation, resulting in a lot of scalability for the model, and without any significant modification, it can be applied directly to longer contents, such as paragraphs, articles, etc.

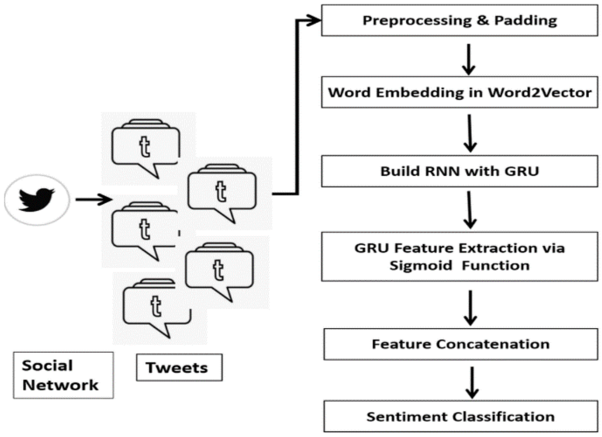


Figure 4: Architecture of the proposed text classifier based on RNN and GRU

To classify sentiments, we leveraged a set of 25000 reviews incorporating positive and negative labels. The encoding of pre-processed tweets is done through a sequence of integers which we sometimes also refer to as word indexes. The ordering of words is decided on their overall frequency in the data set. For example - The token or word is indexed based on frequency, for instance, 2. This indexing of words helps in shortlisting of words depending on their frequencies. Below is the sample code for the training dataset.

For several NLP activities, text representation plays a critical role. Successful word embedding will ease text encoding and boost the efficiency of classification. With that approach, different methods can represent the dataset.

The purpose of this study is to increase the efficiency in classification by combining the strength of different representations of words and different methods of deep learning. Furthermore, RNN and LSTM provide effective performance on various representations of the data. Although RNN is capable of capturing extraction of features in local regions as shown in Figure (4), GRU can extract good features from datasets with long-term dependencies such as natural languages and signals.

This allows the efficient use of RNN in datasets with near semantic relationships such as images. On the other hand, GRU delivers good performance on NLP issues and resolves semantic dependence among the words. Such approaches are useful because they can lead to the problems of classifying feelings. The results of the classification obtained confirmed the contribution. With word embedding methods like Word2Vec and Fast Text, some content disappears in the text [36].

Let's explain this through the example of preprocessing step text like URI data, emoticons, and meaningless grammatical words or stop words are discarded. The removed content still makes the part of the user through the process. To improve the dignity of the emotion one can mix multiple representations of text that helps to convey the honesty of the user's view. Such representations can contribute to various methods of deep learning and their extractions. Figure (5), reflects the deep learning model proposed in this architecture. The proposed model correlates with the embedding of CNN character-level while the second option works with active embedding LSTM Fast Text.

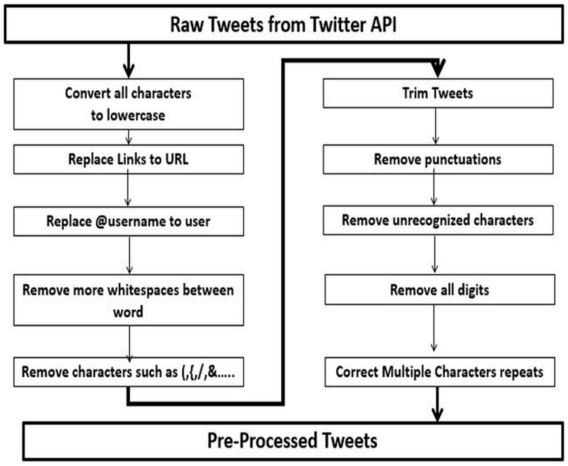

Raw Tweets from Twitter API
Convert all characters to lowercase
Replace Links to URL
Replace @username to user
Remove more whitespaces between word
Remove characters such as (,{,/,&.....
Trim Tweets
Remove punctuations
Remove unrecognized characters
Remove all digits
Correct Multiple Characters repeats
Pre-Processed Tweets

Figure 5: Pre-Processing Steps

The data generated on social media platforms by end-users contains a variety of contents such as slang, special characters, etc. along with standard alphabetic characters but many of these do not contribute to classification as shown in figure 5. For example, @usrname is a kind of neutral word not contributing to positive or negative text while performing sentiment analysis on social media platform posts such as Twitter. These texts contribute to noise in text processing problems [26]. Many available algorithms help increase classification performance by cleansing text data content. End users on social media platforms while posting their thoughts doesn't conform to any standard grammar rules of any spoken language and writes as per their thoughts leading to multilingual texts with many spell issues being posted. To overcome these difficulties in text processing we leveraged the Zemberek framework and performed multiple iterations to evaluate the impact of pre-processing our data set. These iterations helped increase classification performance.

6- Results & Discussion

One of the popular scenarios of Implementing natural language processing technique is Sentiment analysis whose aim is to identify whether a specific text conveys “positive” or “negative” sentiments

"The movie was so awesome”
“I slept peacefully for hours”

As a human reader, the comments in the above excerpt are understood to be of negative sentiment, but when it comes to machine analysis, one would need to create a machine learning model to classify sentiments. When leveraging the supervised learning approach, sampling needs to be done with multiple samples of text corpus with each corpus being labeled as positive or negative sentiment. Once this is done, the next step would be to create a machine learning model using this data.

Based on the example sentence above it is difficult for the machine learning model to classify it as positive or negative. If a tokenization or TFiDF approach is used, words such as 'glory' and 'caliber' could be misunderstood as conveying positive sentiments. Since there are no words interpreted as negative one would need to connect different parts of text structures to infer meaning out of the sentence. For Instance, the first sentence can be broken into two parts:

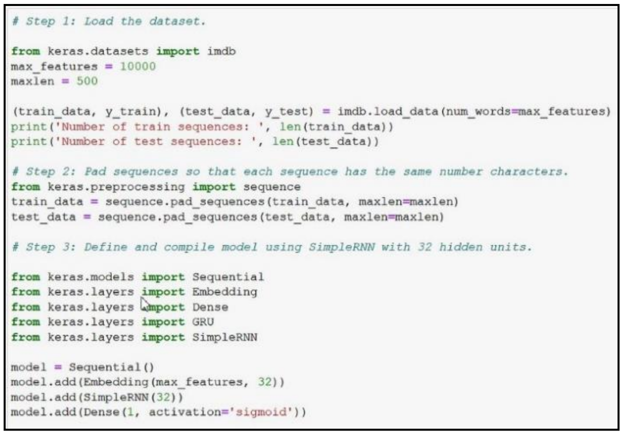
The first part of the sentence concludes the remark as positive. However, when the second sentence is considered its meaning somewhat infers negative sentiment. Hence need arises to retain long-term dependencies. . A simple RNN is, therefore, not good enough for the task. Let's try GRU for the sentiment classification task and see how it performs.

The Implementation is done on environment Implementing TensorFlow-2.3.1 and Keras-2.4.3. The execution is performed on a JupyterHub environment running python-3.8.3 on Conda with standard machine learning packages Installed. The specific

Keras libraries being used for text analysis are models and layers with Sequential, Embedding, Dense, GRU, and RNN.

6-1- Data Set Used

The dataset with 50K movie reviews is being utilized as an input for NLP and Text analytics and binary sentiment classification [41]. It comprises 25,000 highly polar movie reviews for training and testing purposes. For more information on the data set one can refer to this reference: http://ai.stanford.edu/~amaas/data/sentiment/



```
# Step 1: Load the dataset.

from keras.datasets import imdb
max_features = 10000
maxlen = 500

(train_data, y_train), (test_data, y_test) = imdb.load_data(num_words=max_features)
print('Number of train sequences: ', len(train_data))
print('Number of test sequences: ', len(test_data))

# Step 2: Pad sequences so that each sequence has the same number characters.
from keras.preprocessing import sequence
train_data = sequence.pad_sequences(train_data, maxlen=maxlen)
test_data = sequence.pad_sequences(test_data, maxlen=maxlen)

# Step 3: Define and compile model using SimpleRNN with 32 hidden units.

from keras.models import Sequential
from keras.layers import Embedding
from keras.layers import Dense
from keras.layers import GRU
from keras.layers import SimpleRNN

model = Sequential()
model.add(Embedding(max_features, 32))
model.add(SimpleRNN(32))
model.add(Dense(1, activation='sigmoid'))
```

Figure 6: Jupyterhub Notebook Snapshot of RNN Model

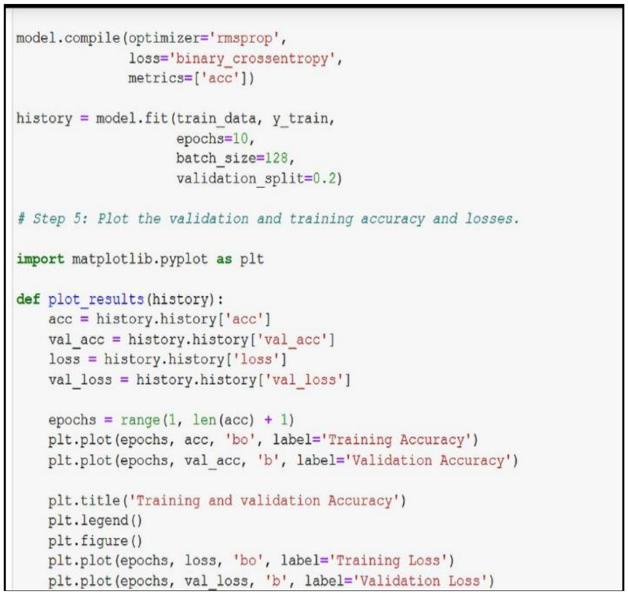
We can define maximum (max_) topmost occurring words while generating the sequence for training as 10,000. Sequence size can be restricted to 500. GRU unit can be used to build an RNN by importing necessary packages such as Sequential from Keras. layers import Embedding, from Keras. Sequential API of Keras is being used to build the model by importing the sequential model API from the Keras model as presented in figure(6). The embedding layer converts the input vector into a fixed-sized vector to be fed into the next layer of the network, if used, it must be added as the first layer to the network. The dense layer can be imported to give distribution over the target variable (0 or the embedding layer takes max_features as input, which is defined by us to be 10,000. The 32 value is set here as the next GRU layer expects 32 inputs from the embedding layer. . GRU unit can be imported to initialize the sequential model API and add the embedding layer, as follows:

```
model = Sequential()
model.add (Embedding (max_features, 32))
```

The embedding layer takes max_features as input, which is defined by us to be 10,000. The 32 value is set here as the next GRU layer expects 32 inputs from the embedding layer. Next, we'll add the GRU and the dense layer, as follows:

```
model.add (GRU (32))
model.add (Dense (1, activation='sigmoid'))
```

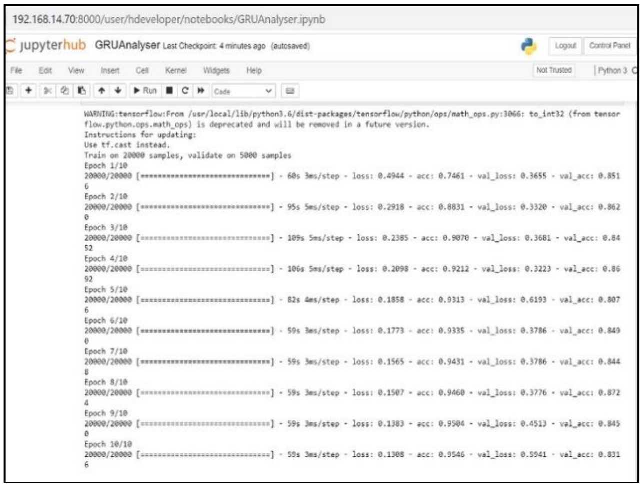
The fixed value integer 32 can be randomly chosen to function as one of the hyperparameters to tune when the network is designed, which also represents the dimensionality of the activation functions. The sigmoid function is used as an activation function as the dense layer only generates a single value which is considered as the probability of review and this is also our target variable. Note that we also assign 20% of the sample from the training data as the validation dataset. We also set the number of epochs to be 10 and the batch size to be 128 – that is, in a single forward-backward pass as shown in Figure (7).



```
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['acc'])

history = model.fit(train_data, y_train,
                    epochs=10,
                    batch_size=128,
                    validation_split=0.2)

# Step 5: Plot the validation and training accuracy and losses.

import matplotlib.pyplot as plt

def plot_results(history):
    acc = history.history['acc']
    val_acc = history.history['val_acc']
    loss = history.history['loss']
    val_loss = history.history['val_loss']

    epochs = range(1, len(acc) + 1)
    plt.plot(epochs, acc, 'bo', label='Training Accuracy')
    plt.plot(epochs, val_acc, 'b', label='Validation Accuracy')

    plt.title('Training and validation Accuracy')
    plt.legend()
    plt.figure()
    plt.plot(epochs, loss, 'bo', label='Training Loss')
    plt.plot(epochs, val_loss, 'b', label='Validation Loss')
```

Figure 7: Jupyterhub Snapshot of Model Validation and Plotting

The model is compiled with the binary cross-entropy loss and the rmsprop optimizer, to track the accuracy (train and validation) as the metric. Next, we fit the model on our sequence data. Note that we also assign 20% of the sample from the training data as the validation dataset.



192.168.14.70:8000/user/hdeveloper/notebooks/GRUAnalyser.ipynb
jupyterhub GRUAnalyser Last Checkpoint: 4 minutes ago (autosaved)
Logout
Control Panel
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Code

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensor
flow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 20000 samples, validate on 5000 samples
Epoch 1/10
20000/20000 [==============================] - 60s 3ms/step - loss: 0.4944 - acc: 0.7461 - val_loss: 0.3655 - val_acc: 0.851
6
Epoch 2/10
20000/20000 [==============================] - 95s 5ms/step - loss: 0.2918 - acc: 0.8831 - val_loss: 0.3320 - val_acc: 0.862
0
Epoch 3/10
20000/20000 [==============================] - 109s 5ms/step - loss: 0.2385 - acc: 0.9070 - val_loss: 0.3681 - val_acc: 0.84
52
Epoch 4/10
20000/20000 [==============================] - 106s 5ms/step - loss: 0.2098 - acc: 0.9212 - val_loss: 0.3223 - val_acc: 0.86
92
Epoch 5/10
20000/20000 [==============================] - 82s 4ms/step - loss: 0.1858 - acc: 0.9313 - val_loss: 0.6193 - val_acc: 0.807
6
Epoch 6/10
20000/20000 [==============================] - 59s 3ms/step - loss: 0.1773 - acc: 0.9335 - val_loss: 0.3786 - val_acc: 0.849
0
Epoch 7/10
20000/20000 [==============================] - 59s 3ms/step - loss: 0.1565 - acc: 0.9431 - val_loss: 0.3786 - val_acc: 0.844
8
Epoch 8/10
20000/20000 [==============================] - 59s 3ms/step - loss: 0.1507 - acc: 0.9460 - val_loss: 0.3776 - val_acc: 0.872
4
Epoch 9/10
20000/20000 [==============================] - 59s 3ms/step - loss: 0.1383 - acc: 0.9504 - val_loss: 0.4513 - val_acc: 0.845
0
Epoch 10/10
20000/20000 [==============================] - 59s 3ms/step - loss: 0.1308 - acc: 0.9546 - val_loss: 0.5941 - val_acc: 0.831
6
```

Figure 8: Jupyterhub Snapshot of epochs Classification on IMDB, [41] dataset.

When we mention validation split as a fit boundary while fitting deep learning model, its parts information into two sections for each epoch, for example, preparing information i.e., Training Data and validation data and since we are suing shuffle also it will rearrange dataset before spitting for that epoch. It prepares the model on training information and validates the model on validation data by checking its loss and precision

At the point when we are training the model in Keras, precision and loss in the Keras model for validating information could be variating with various cases. Ordinarily, with each epoch expanding, loss ought to be going lower and accuracy ought to be going higher as projected in Figure (8).

Epoch	Loss	Accuracy	Val Loss	Val Accuracy
1	0.4944	0.7461	0.3655	0.8510
2	0.2918	0.8831	0.3320	0.8620
3	0.2385	0.9070	0.3681	0.8400
4	0.2098	0.9212	0.3223	0.8600
5	0.1858	0.9313	0.6193	0.8070
6	0.1773	0.9335	0.3786	0.8490
7	0.1565	0.9431	0.3786	0.8440
8	0.1507	0.9460	0.3776	0.8720
9	0.1383	0.9504	0.4513	0.8450
10	0.1308	0.9546	0.5941	0.8310

Table 2: Evaluation metrics accuracy (higher is better) and Loss (lower is better).

GRU was used with word embedding's representation as Word2Vector in the experiment as a sequential input with RNNs models. We achieved the best classification accuracy of 87% in the 8th epoch as shown in Table (2).

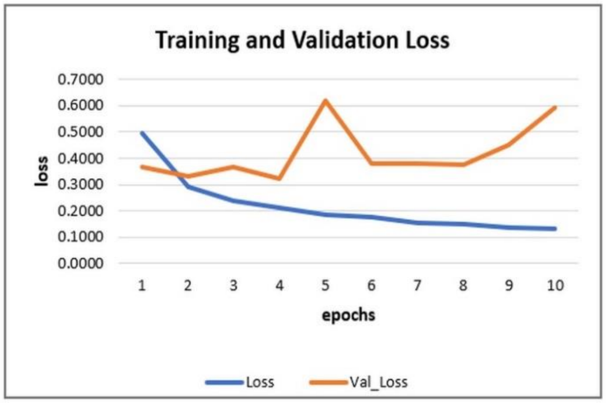


Figure 9: Training and Validation loss

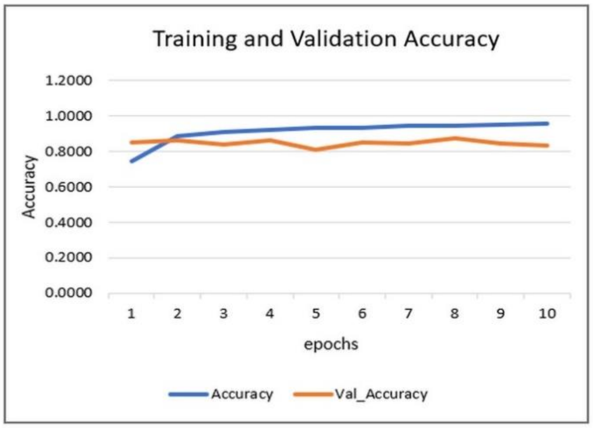


Figure 10: Training and Validation Accuracy

Figure (9) (10) defines the plot of accuracy and loss against epochs where accuracy achieved is directly proportional to values of epochs. Referring to Table (2) one cans infer easily that training set results on the model are comparable to epochs. The accuracy plot indicates that the model is underfitting before the point and overfit after the point. The loss diagram indicates underfitting before the Intersection point and overfitting after that.

7- Challenge

As per the copyright [21] survey by Zhang on sentiment analysis, sentiment analysis has the following granularity levels - document, sentence, and aspect wherein each level of classification add challenges based on granularity. A few of the key challenges involved in sentiment analysis are detection of sarcasm, negation, Word ambiguity, and Multipolarity. sarcasm is not only difficult to understand for a machine but also a human. The repetitive variation in words used in sarcastic sentences makes it hard to successfully train sentiment analysis models. Common topics, interests, and historical information must be shared between two people to make sarcasm available. Multiple approaches can be used for Sarcasm detection like Rule-based, Statistical, Machine Learning, and Deep Learning.

Negation detection can be done by marking as negated all the words from a negation cue to the next punctuation token. GRU and Long Short-Term Memory [9] can store information about a longer data sequence that has been processed.

One of the powerful techniques that have emerged in the recent past is Deep learning [34] which does its learning by using multiple layers of representations or data features and helps to produce state-of-the-art prediction results. There are manyfold challenges associated with traditional neural network techniques like gradient explosion and over-fitting [9], while deep GRU neural network model comes with low update efficiency and poor information processing capability among multiple hidden layers [21]

8- Conclusion and Future Scope

Sentiment analysis is one of the Important areas of natural language processing. Our paper presents the idea of sentiment classification of tweets and reviews using GRU. GRUs can use their update and reset gates to store and filter the information. This removes the issue of the vanishing gradient since the model does not wash out the new input every single time but preserves the relevant data and transfers it down to the network's next steps. Even in complex situations, if carefully trained, they may perform extremely well. We can conclude that gated recurrent units are a suitable model for sentiment analysis. Being a recurrent network, it can effectively capture long sequence data required for natural language understanding.

A GRU is an expansion of a basic RNN, which assists with combatting the vanishing gradient problem issue by permitting the model to learn long haul conditions in the content structure [2][8]. An assortment of utilization cases can profit from this compositional unit. a

State-of-the-art sentiment analysis and other NLP tasks could be created by leveraging GRU networks with 87% accuracy. In the future, we may have another progression over a basic RNN – Long Short-Term Memory (LSTM) network or Bi-GRU and Bi-LSTM Network as recommended in [18]. RNN with LSTM can analyze and predict based on the advantages of LSTM they carry with their new architecture.

References

[1] Andrew L , Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning word vectors for sentiment analysis." In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pp. 142-150. 2011.

[2] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks 5, no. 2, 1994,pp 157-166.

[3] Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35, no. 8, 2013, pp 1798-1828.

[4] B. Athiwaratkun, Ben, and Jack W. Stokes. "Malware classification with LSTM and GRU language models and a character-level CNN." In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2482-2486.

[5] Chen, Huimin, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. "Neural sentiment classification with user and product attention." In Proceedings of the 2016 conference on empirical methods in natural language processing, 2016, pp. 1650-1659.

[6] Chen, Peng, Zhongqian Sun, Lidong Bing, and Wei Yang. "Recurrent attention network on memory for aspect sentiment analysis." In Proceedings of the 2017 conference on empirical methods in natural language processing,2017 pp. 452-461.

[7] Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014 ." arXiv preprint arXiv:1406.1078.

[8] Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014." arXiv preprint arXiv:1412.3555.

[9] C.Zhou, Chunting, Chonglin Sun, Zhiyuan Liu, and Francis Lau. "A C-LSTM neural network for text classification."2015, arXiv preprint arXiv:1511.08630.

[10] Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch." Journal of machine learning research 12, no. ARTICLE 2011, pp 2493-2537.

[11] D. Tang, Duyu, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. "Learning sentiment-specific word embedding for twitter sentiment classification." In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics ,Volume 1, 2014 , pp. 1555-1565.

[12] Hemmatian, Fatemeh, and Mohammad Karim Sohrabi. "A survey on classification techniques for opinion mining and sentiment analysis." Artificial Intelligence Review 52, no. 3, 2019,pp 1495-1545.

[13] Guolong Liu, Xiaofei Xu, Bailong Deng, Siding Chen, and Li. "A hybrid method for bilingual text sentiment classification based on deep learning." In 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD),IEEE, 2016,pp. 93-98.

[14] Guan, Ziyu, Long Chen, Wei Zhao, Yi Zheng, Shulong Tan, and Deng Cai. "Weakly-Supervised Deep Learning for Customer Review Sentiment Classification." In IJCAI,2016, pp. 3719-3725.

[15] Goldberg, Yoav. "A primer on neural network models for natural language processing." Journal of Artificial Intelligence Research 57,2016, pp. 345-420.

[16] H.Wang, and Dequn Zhao. "Emotion analysis of microblog based on emotion dictionary and Bi-GRU." In 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), . IEEE, 2020, pp. 197-200.

[17]J. Pennington, R. Socher, and C. D. Manning. "Glove: Global vectors for word representation". Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532-1543.

[18] Jabreel, Mohammed, Fadi Hassan, and Antonio Moreno. "Target-dependent sentiment analysis of tweets using bidirectional gated recurrent neural networks." In Advances in hybridization of intelligent methods, Springer, Cham, 2018, pp. 39-55.

[19] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, Yoshua Bengio. "Gated feedback recurrent neural networks." In International conference on machine learning, PMLR, 2015, pp. 2067-2075.

[20] Johnson, Rie, and Tong Zhang. "Effective use of word order for text categorization with convolutional neural networks," 2014, arXiv preprint arXiv:1412.1058.

[21] L Zhang, S Wang, B Liu. Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2018 Jul, 8(4):e1253.

[22[Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." In International conference on machine learning, PMLR, 2014, pp. 1188-1196.

[23]Li, Cheng, Xiaoxiao Guo, and Qiaozhu Mei. "Deep memory networks for attitude identification." In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining,2017, pp. 671-680.

[24] Luo LX. “Network text sentiment analysis method combining LDA text representation and GRU-CNN”, Personal and Ubiquitous Computing 23, no. 3 ,2019, pp 405-12.

[25]M Zulqarnain, R Ghazali, MG Ghouse.” Efficient processing of GRU based on word embedding for text classification”, International Journal of Informatics Visualisation, vol 3 No 4, 2019, pp.377-383.

[26] MU Salur, I Aydin. “A novel hybrid deep learning model for sentiment classification”. IEEE Access. 2020 Mar 23;8,pp 58080-93.

[27] Moraes, Rodrigo, João Francisco Valiati, and Wilson P. GaviãO Neto. "Document-level sentiment classification: An empirical comparison between SVM and ANN." Expert Systems with Applications 40, no. 2 , 2013, pp 621-633.

[28] Ö Yildirim. "A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification." Computers in biology and medicine 96 , 2018, pp189-202.

[29] R. Ni and H. Cao, "Sentiment Analysis based on GloVe and LSTM-GRU," In 2020 39th Chinese Control Conference (CCC), IEEE, 2020, pp. 7492-7497.

[30] S Yan, J Chai, L Wu. "Bidirectional GRU with Multi-Head Attention for Chinese NER." In 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), IEEE, 2020, pp. 1160-1164.

[31] S Yang, X Yu, Y Zhou. "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example." In 2020 International Workshop on Electronic Communication and Artificial Intelligence , IWECAI, IEEE, 2020, pp. 98-101.

[32] Tang, Duyu, Bing Qin, and Ting Liu. "Document modeling with gated recurrent neural network for sentiment classification." In Proceedings of the 2015 conference on empirical methods in natural language processing,2015, pp. 1422-1432.

[33]Tang, Duyu, Bing Qin, and Ting Liu. "Learning semantic representations of users and products for document level sentiment classification." In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing ,Volume 1, 2015, pp. 1014-1023.

[34]Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." 2015, arXiv preprint arXiv:1503.00075.

[35]Thet, Tun Thura, Jin-Cheon Na, and Christopher SG Khoo. "Aspect-based sentiment analysis of movie reviews on discussion boards." Journal of information science 36, no. 6 2010, pp 823-848.

[36]T. Mikolov, , Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. "Distributed representations of words and phrases and their compositionality."2013, arXiv preprint arXiv:1310.4546.

[37]Turian, Joseph, Lev Ratinov, and Yoshua Bengio. "Word representations: a simple and general method for semi-supervised learning." In Proceedings of the 48th annual meeting of the association for computational linguistics,2010, pp. 384-394.

[38]Y. Pan and M. Liang, "Chinese text sentiment analysis based on bi-gru and self-attention." In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), . IEEE, 2020, vol. 1, pp. 1983-1988.

[39]Zhou, Zhiyuan, Yuan Qi, Zheng Liu, Chengchao Yu, and Zhihua Wei. "A C-GRU neural network for rumors detection." In 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), IEEE, 2018, pp. 704-708.

[40] Zhang, Wei, Quan Yuan, Jiawei Han, and Jianyong Wang. "Collaborative multi-Level embedding learning from reviews for rating prediction." In IJCAI, vol. 16, pp. 2986-2992. 2016.

[41] Data Set: Large Movie Review Dataset v1.0 by Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Available http://ai.stanford.edu/~amaas/data/sentiment/

Nidhi Chandra is a Research Scholar at Amity Institute of Information Technology and is a faculty at Department of Computer Science & Engineering, Amity University, Uttar Pradesh. She obtained her Master’s Degree in Computer Science from Guru Gobind Singh Indraprastha University, Delhi. With over 17 years of Academic experience and 2 years of Industry experience, her research interest lies in Natural Language Processing, Semantic Analysis and Data mining.

Prof. (Dr.)Laxmi Ahuja Ph.D. (CSE) working as Professor in Amity Institute of Information Technology with the role ranging from Lecturer to Professor to Dy. Director of the Department within a span of 21 years. in academics, her areas of interest include Search Engine, Data Mining and Soft Computing Approaches; published more than 100 research papers in International and National Conferences and Journals in SCOPUS Impact Factor Journals like Springer, Elsevier, Inderscience and several others. She has successfully filed number of patents under domain of Information Technology as an Inventor, which has been published in "International Journal of Patents" by Patent Department, Govt. of India. She has published many Lecture Notes for Springer Book chapter. Organized various sessions in International Conferences and Session chair in various Conferences. She has organized various Guest Lectures, Workshops and Faculty Development Programme. She is Associate Editor of Proceedings of the International Conference on Reliability, Infocom Technology and Optimization (Trends and Future Directions). She is actively involved in research activities. She has currently Eight research scholar under her guidance and supervised three students for Ph.D. She has also published number of Books related to operating system and Information Technology. She is Member of IEEE and Life Member of Computer Society of India, Past Vice chairman of IETE -The Institute of Electronics and Telecommunications Engineers, Senior member of IACSIT, Member of IEEE Society. She is actively involved in various activities of these societies. She is also Technical Chair member in many Conferences. She has Awarded as Guest of Honor in many Events.

Prof. (Dr.) Sunil Kumar Khatri is Director of Campus at Amity University in Tashkent, Uzbekistan. He is a Fellow of IETE, Sr. Life Member of IEEE, CSI and IASCSIT and Member of IAENG. He is Secretary in SREQOM, Convener, EAC, Past IEEE UP Section Executive Council and Past Vice-Chairman of CSI Noida Chapter. Dr. Sunil Kumar Khatri is Editor IJSAEM, Springer Verlag. He is in Editorial Board of several Journals from USA, Egypt, Hong Kong, Singapore, and India. He has twelve edited books, eleven guest edited special issues of international journals, eleven patents filed and more than 225 papers in international and national journals and proceedings. His areas of research are Artificial Intelligence, Software Reliability and Testing, and Data Analytics.

Prof. (Dr.) Himanshu Monga did Ph.D. from Thapar Institute of Engineering and Technology, (TU) Department Of ECE, Patiala, Punjab, INDIA, in the field of Optical networks / coding and Wireless networks. He did bachelor's degree in Electronics & Communication Engineering from Govt. Engineering College, Amravati University, Maharashtra, India and master's degree in Electronics and Telecommunication Engineering & Master's degree in management in Human resource management. Presently working as Professor & Head (ECE) at Jawaharlal Lal Nehru Government Engineering College, Sundernagar, Mandi (H.P). He is also having the charge of Dean Academics at JNGEC & OIC (Computer Centre). he has more than 21 years of work experience in academics and industry/research. He has published more than 140 Research papers in international/national conferences. Have more than 30 Thomson Reuter SCI publications to his credit. He is the life member of IEEE, ISTE and reviewer of many renowned journals. He has authored 08 Books/ chapters in renowned books. He has successfully completed 06 projects worth Rs 70 Lacks and successfully completed many consultancy projects. He has recently published 5 patents.