Depression Analysis of Bangla Social Media Data using Gated Recurrent Neural Network

Conference Paper · May 2019

DOI: 10.1109/ICASERT.2019.8934455

Abdul Hasib Uddin
Shiram System Limited

Depression Analysis of Bangla Social Media Data using Gated Recurrent Neural Network

Abdul Hasib Uddin[1], Durjoy Bapery[2], Abu Shamim Mohammad Arif[3]
Computer Science and Engineering Discipline, *Khulna University*
Khulna, Bangladesh
Email: [1]abdulhasibuddin@gmail.com, [2]durjoybapery@gmail.com, [3]shamimarif@yahoo.com

***Abstract*—Nowadays, micro-blogging sites like Twitter, Facebook, YouTube, etc., have become much popular for social interactions. People are expressing their depression over social media, which can be analyzed to identify the causes behind their depression. Most of the researches on emotion and depression analysis are based on questionnaires and academic interviews in non-Bengali languages, especially English. These traditional methods are not always suitable for detecting human depression. In this paper, we introduced a Gated Recurrent Neural Network based depression analysis approach on Bangla social media data. We collected Bangla data from Twitter, Facebook and other sources. We selected four hyper-parameters, namely, number of Gated Recurrent Unit (GRU) layers, layer size, batch size and number of epochs, and presented step by step tuning for these Hyper-parameters. The results show the effects of these tuning steps and how the steps can be beneficial in configuring GRU models for gaining high accuracy on a significantly smaller data set. This work will help psychologists and concerned authorities of society detect depression among Bangla speaking social media users. It will also help researchers to implement Natural Language Processing tasks with Deep Learning methods.**

***Index Terms*—Depression, Bangla, Social Media, Gated Recurrent Neural Network, Hyper-parameter Tuning**

I. Introduction

The objective of Artificial Intelligence is to imitate and analyze human behaviors. In this course, detecting human sentiment and emotion is an important part for which Machine and Deep Learning approaches are being widely used. Sentiment and emotion classification can be analyzed further from two different perspectives, specifically, detecting sentiment and emotion from image data, and sentiment and emotion detection from textual data. In both cases, psychological and technical knowledge is essential to analyze the data. In a general sense, sentiment analysis covers the overall area of positive, negative, and neutral sentiment classification tasks. Emotions, such as happiness, sadness, depression, disgust, etc., are rather deep sentiments which are much more difficult to analyze. Some of these emotions are deeper than others which requires high-level psychological knowledge and more sophisticated technical approaches to study.

Depression is one of the deepest human emotions. With a mechanical way of life, more and more people are falling into depression [1]. Depression is a mental disorder which destroys not only the corresponding person but also affects the morality of that person causing him/her to commit unsocial activities, even suicide and murder. According to the World Health Organization (WHO), over 300 million people were suffering from depression in 2017, which indicates an increase of more than 18% between 2005 and 2015 [1].

Most of the research work on depression analysis are survey and one-to-one communication-based. Technical knowledge of depression analysis can help psychologists to detect and analyze depression and causes behind it. Micro-blogging sites like Facebook, Twitter, LinkedIn are now more popular than ever before. On these sites, people frequently share their daily activities and emotional reactions. Wang et al., used Sina, a Chinese micro-blogging website, to collect data and applied psychological knowledge to extract features from it [2]. They demonstrated the use of Naive Bayes, Decision Tree and Rule-based classifiers for their depression detection task.

Bangla is the seventh most spoken language in the world [3]. However, depression analysis in Bangla is still to be done. Riyadh et al., used Naive Bayes classifier with Laplacian Add-1 Smoothing for emotion classification in Bangla [4]. They collected Twitter data from Sentiment140 and used only the core texts for their research work. Their emotion classification covered happiness, sadness, surprise, and disgust. In another research work, Chowdhury et al. performed sentiment analysis in Bangla [5]. They used a hybrid mechanism consisting of both Lexicon based and Machine Learning approaches. They acted upon a semi-supervised bootstrapping approach with Support Vector Machine (SVM) and Maximum Entropy (ME). They collected their training corpus from Twitter.

Neural Network based Deep Learning approaches are becoming more and more popular for semantic analysis, such as emotion classification. Recently, in 2017, Suhara et al., introduced a self-reported history based depression detection procedure using Recurrent Neural Network (RNN) [6]. Hassan et al., used Long Short Term Memory Recurrent Neural Network (LSTM-RNN), a Deep Learning model, for performing sentiment analysis in Bangla [7]. Their work involved both normal and Romanized Bangla.

Can et al., performed multilingual sentiment analysis on limited data using RNN based model [8]. They assembled data from Amazon reviews, Yelp restaurant reviews, and Competition restaurant reviews. Their work covered sentiment analysis in English, Spanish, Turkish, Dutch, and Russian languages. Ayata et al. used both Deep Learning and Machine Learning approach for sentiment analysis [9]. They gathered data from Twitter and trained Support Vector Machine (SVM), Random Forest (RF), Naive Bayes, and LSTM-RNN. Their work concluded with the decision that in most of the cases, Deep Neural Network based approaches perform better than Machine Learning approaches. In their work, only SVM accomplished slightly better performance than LSTM.

In this paper, we introduced the Gated Recurrent Unit (GRU) Neural Network based depression analysis on small Bangla dataset, collected from Twitter, Facebook, and native Bangla speakers using Google form. Labeling human emotion related data requires knowledge of human behaviors. Hence, we hired a Sociology student, who is experienced in dealing with human emotions, to label our dataset. We applied step by step Hyper-parameter tuning while implementing GRU model. Then, we compared each implementation of Hyper-parameter tuning and showed how to tune different Hyper-parameters in an easy way to accomplish better performance on a very small dataset. We have manipulated four different Hyper-parameters, specifically, model size, no. of layers, batch size, and no. of epochs. Apart from applying the GRU model for depression detection from Bangla social media data, our other aim was to focus specifically on GRU model performance over a significantly small dataset. In Bangla, it is reasonably difficult to manage a large dataset due to lack of enough research works. Hence, for Hyper-parameter tuning, we only focused on the aforementioned four Hyper-parameters, without following any specific Hyper-parameter optimization approach.

The rest of this paper is arranged as follows. In Section II, we have discussed some relevant existing research works, our proposed method is described in Section III, and in section IV, the results of our method and relevant discussions are presented. Finally, we conclude our work in Section V along with some future plans.

II. Related Works

In this section, some available research works related to Machine and Deep Learning based sentiment and emotion analyses are described.

A. Machine Learning for Emotion Analysis in Bangla

A Machine Learning based human emotion analysis approach is described by Riyadh et al., in [4]. In this research work, authors selected happiness, sadness, surprise, and disgust for their classification task. They collected tweets from Sentiment140, labeled them manually, eliminated tweets with no emotion, and created a balanced dataset containing 3,750 tweets. 3,500 tweets were selected as their training dataset and 250 tweets as the testing dataset. For feature extraction, Unigram model and Unigram model with POS tagging were used. Authors used the frequency of Bag of Words model as a feature to train their classifier. They used Multinomial Naïve Bayes classifier for the classification task. To avoid zero probability problem of Naïve Bayes classifier, they used Laplacian Add-1 Smoothing by assuming the training dataset to be very large and by adding one to each count. Against 4-way classification, they gained an average of 81% accuracy for Unigram model. For the Unigram model with POS tagging, they gained an average of 79.5% accuracy. For the 5-way classification task, they gained an average of 66% for Uni-gram model. For Uni-gram model with POS tagging, they gained an average of 64.8% accuracy.

B. RNN for Depression Forecasting

A novel method for depression forecasting was established by Suhara et al., using RNN [6]. The authors developed LSTM-RNN based deep learning algorithm. They used their model to produce embedding layers regarding every categorical variable, which also incorporates a day-of-the-week variable to determine the day-of-the-week consequences in their model. They collected depressed data from 2,382 self-declared depressed persons, covering 22 months time span, via a smartphone application. Their model was successfully able to forecast 84.6%, 82.1%, and 80.0% severe depression cases in 1, 3, and 7 days beforehand, respectively. Additionally, the model was successful in detecting overall depression cases with 88.6%, 86.0%, and 84.2% accuracy for forecasting 1, 3, and 7 days beforehand, respectively.

C. Machine Learning for Depression Analysis

Wang et al. conducted an experiment on Sina micro-blog, a Chinese micro-blog, which is one of the most influential social media services in China [2]. They emerged both Psychological and Machine Learning knowledge for their experiment. From Psychological perspective, they used ten features, such as 1st person singular, 1st person plural, positive emoticons, negative emoticons, mentioning, being forwarded, being commented, original blogs, blogs posted between 0:00 - 6:00 o'clock, and one other. From the technical perspective, Machine Learning techniques, such as Decision Tree, Naive Bayes, and Rule-based classifiers were used. Their described method contained mainly three steps, namely, sentence and word segmentation, polarity calculation of sub-sentences, and polarity calculation of sentences. Their model was able to accomplish 80% precision.

D. CNN and RNN for Natural Language Processing

Yin et al. presented a systematic comparison between CNN and RNN in [10]. Their research work covered a wide range of representative NLP tasks, providing basic guidance for Deep Neural Network (DNN) selection. They systematically compared CNN, LSTM, and GRU models. It covered a broad range of NLP tasks, such as sentiment classification, relation classification, textual entailment, answer selection, question relation match, path query answering, part-of-speech tagging, etc. For their experiment, on each step, they trained their model from scratch, used basic setup without complex tricks, like batch normalization, searched for optimal Hyper-parameters, analyzed the basic architecture and utilization of the models. This experiment concluded that, generally, RNNs perform better and robust for NLP tasks, except for some cases, like key-phrase recognition task.

III. Methodology

Our proposed approach for depression detection from Bangla dataset is divided into two steps - *Creating Bangla Dataset* and *Training GRU Model*. We put special efforts for preparing Bangla dataset and shown steps for Hyper-parameter tuning.

A. Creating Bangla Dataset

We used Twitter as our primary data source. We collected 5,000 Bangla data from Twitter and 210 depressed Bangla statements from native Bangla speakers using google form. We prepared our dataset into three steps - *Data Pre-processing*, *Data Labeling*, and *Data Post-processing*.

1) Data Pre-processing: For data pre-processing task, we created a white list containing Bangla alphanumeric characters, punctuations, and space. We scanned each tweet character by character, filtered out all non-white listed characters, and removed multiple consecutive white spaces. An example of twitter data cleaning is given in Fig 2. We only had to pre-process the tweets, as data collected via google form were clean.

Table I: Part of the labeled and stratified dataset

Data no.	Data	Label
34	ওরাল রিহাইড্রেশন সলিউশন ও আর এস প্রস্তুত করার ১২ ঘন্টার মধ্যে বা রেফ্রিজারেটরে রাখলে তা ২৪ ঘন্টার মধ্যে পান করে ফেলা উচিত ।	Non-depressive
33	রাজনাথ সিং সংসদে বললেন এই নাকি ফাইনাল নয় । এই লক্ষ লোক কে আবার নাকি সুযোগ দেওয়া হবে নাগরিত্ব প্রমাণ করার । এটা কি আইওয়াশ নয় ? এরা আর কি প্রমাণ দেবেন ? পাসপোর্ট , আধার ভোটার কার্ড সব ই তো বাতিল করে দিয়েছে । আবার অনেক পুরুষের নাম আছে অথচ তাদের পরিবারের নাম নেই । এরা কি করবে ?	Depressive
118	আহা ! গভীর গহী নের নিঃশব্দ সুর সা থে আঁধা রের ঘ্রাণ !	Non-depressive
117	প্রায় ৪০ লক্ষ বাঙালি দেশ ছাড়া হতে চলেছে সামলাতে পারবে তো পৃথিবী ?	Depressive

2) Data Labeling: Deep human emotions, like depression, are extremely difficult to analyze. In order to make our depression detection dataset more reliable, we hired a Bangla speaking Sociology student, experienced in dealing with human emotions, to manually label our Bangla dataset. A total of 5,000 Bangla tweets were labeled. After labeling, we eliminated incomplete and ambiguous tweets. Our initial tweet set consisted of 984 depressive tweets and 2,930 non-depressive tweets.

3) Data Post-processing: After pre-processing and labeling our dataset, we eliminated redundant data from both tweet set and depressed statement set collected via a google form. After removing redundancy, our dataset was imbalanced with 1,289 non-depressive and only 588 depressive data. An imbalanced dataset may lead to wrong accuracy and over-fitting problems, where accuracy for data within the training dataset would be high, while accuracy for data outside of the training dataset would be low. Hence, we randomly selected only 588 non-depressive data from the labeled dataset to balance with 588 depressive data. In this step, our balanced dataset consisted of only 1,176 data of which 588 were depressive and 588 were non-depressive.
Stratifying Dataset. The balanced dataset was very small to train with a neural network. Therefore, we stratified our dataset to reduce the effect of its small size while training the GRU model. In this step, data were rearranged into a one-to-one approach, that is, one depressive data followed by one non-depressive data, and continuously repeated this process all over the balanced dataset. After stratifying our dataset, we considered this dataset as final and used it to train our GRU model.
Part of an example of labeled and stratified data set is given in Table I. The table shows some Bangla data along with their original labels. The 'Data no.' signifies the positions of the data in the final dataset, and how the final dataset was stratified with one non-depressive data followed by one depressive data.

B. Training GRU Model

For training our Gated Recurrent Neural Network Model, we again divided the training phase into two substeps - *Dataset Splitting*, and *Hyper-parameter Tuning and Training*.

1) Dataset Splitting: Each time before training our model, we split our dataset into 80% training, 10% validation and 10% testing set. Validation dataset was only used to avoid over-fitting. We compared different trained models according to their corresponding testing accuracy.

2) Hyper-parameter Tuning and Training: For implementing our model, we utilized tensorflow implementation described in [11]. To get an accurate result using our small dataset, we fixed learning rate of our model to a very low value of 0.0001. Low learning rate helps the model to avoid over-fitting problem so that our model detects depression accurately from any given data. We tuned 4 GRU model

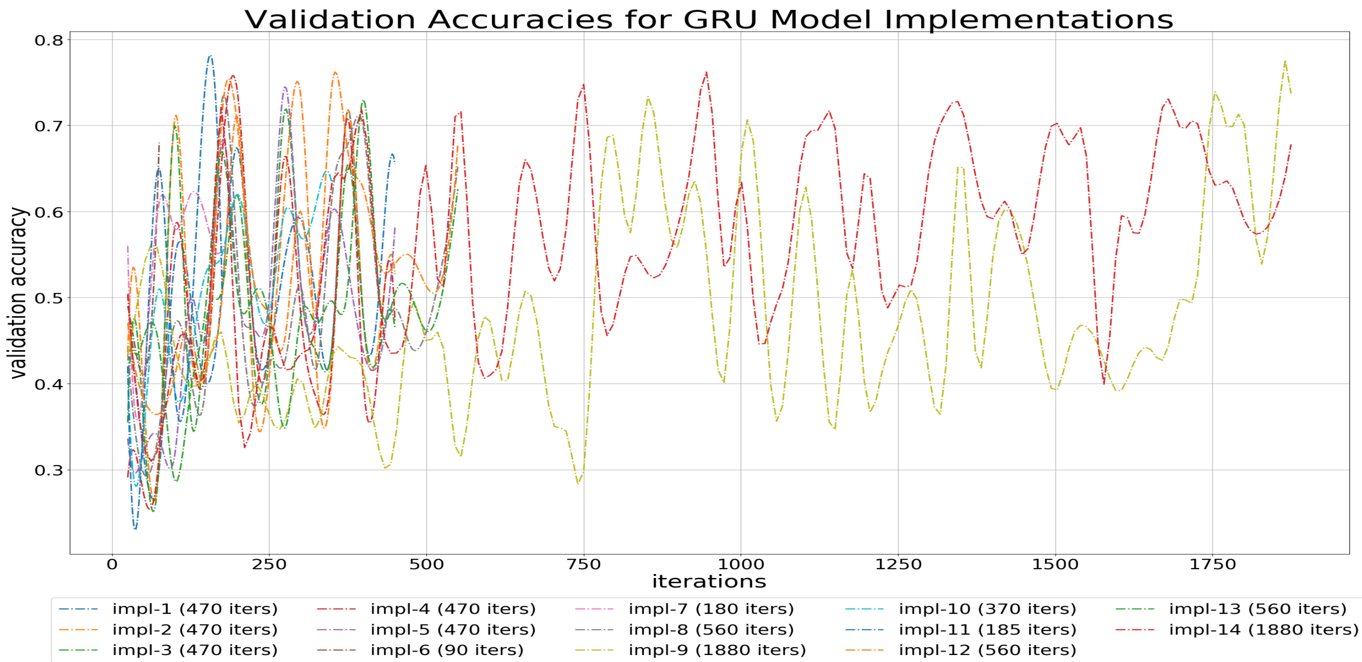

Figure 1: Comparing GRU implementation performances in terms of validation accuracies.

Original data:
নদীর নীচে দেশের প্রথম রেল সুড়ঙ্গ তৈরি হয়েছে কলকাতায়। এবার মেট্রো রাইলের সৌজন্যে দেশের সবথেকে বড় ভূগর্ভস্থ রেল ইয়ার্ড পেতে চলেছে কলকাতায়। যা তৈরি হচ্ছে কলকাতা বিমানবন্দরের ঠিক পাশেই।https://ebela.in/state/kolkata-metro-railway-is-constructing-the-country-s-biggest-underground-rail-yard-near-airport-dgtl-1.839389?ref=state-new-stry
...

After pre-processing:
নদীর নীচে দেশের প্রথম রেল সুড়ঙ্গ তৈরি হয়েছে কলকাতায়।এবার মেট্রো রাইলের সৌজন্যে দেশের সবথেকে বড় ভূগর্ভস্থ রেল ইয়ার্ড পেতে চলেছে কলকাতায়।যা তৈরি হচ্ছে কলকাতা বিমানবন্দরের ঠিক পাশেই।
: ?

Figure 2: Example of data pre-processing.

Hyper-parameters into three steps - *Tuning for GRU Size*, *Tuning Batch Size with No. of Epochs*, and *Tuning No. of GRU layers with No. of Epochs*. Fig 1 represents the effects of Hyper-parameter tuning in each implementation (impl) in terms of validation accuracies.

a) Tuning for GRU Size: We initially fixed no. of GRU layers to 5, batch size to 10 and, no. of epochs to 5 (total 470 iterations) and tuned GRU size over these parameters. We trained our GRU model with size 64, 128, 256, 512, and 1024. Impacts for tuning GRU size are shown in Fig 1 (impl 1 to 5). Our model gained the highest accuracy for GRU size 512 (impl 4).

b) Tuning Batch Size with No. of Epochs: After tuning GRU size, we fixed its size to 512, no. of layers to 5, and tuned batch size along with no. of epochs. We trained our model with batch size and no. of epochs set to 50 and 5 (90 iters), 50 and 10 (180 iters), 5 and 3 (560 iters), 1 and 2 (1880 iters), 25 and 10 (370 iters), 25 and 5 (185 iters), respectively. As time complexity increases while batch size decreases, we had to decrease no. of epochs corresponding to small batch size. With batch size 5 over 3 epochs, our model accomplished the highest accuracy for tuning batch size with no. of epochs. Effects for tuning batch size with corresponding no. of epochs are shown in Fig 1 (impl 6 to 11). It indicates how accuracy changes with different batch size and no. of epochs.

c) Tuning No. of GRU layers with No. of Epochs: In the final step of Hyper-parameter tuning, we fixed GRU size to 512, batch size to 5, and tuned no. of GRU layers of our model. To reduce memory complexity, we decreased no. of epochs while increasing no. of GRU layers. We trained our model with GRU layers and no. of epochs set to 3 and 3 (560 iters), 10 and 3 (560 iters), and 5 and 10 (1880 iters), respectively. For tuning GRU layers with no. of epochs, our model attained the highest result for 3 layers over 3 epochs. The influences of tuning no. of GRU layers on learning curve are shown in Fig 1 (impl 12 to 14).

IV. Result and Discussion

All the steps of Hyper-parameter tuning along with their corresponding test accuracies are given in Table II. On the first step of Hyper-parameter tuning, we started with a low GRU size of only 64 and gradually increased the size to 1024. Our GRU model achieved 59.1%, 70.0%, 67.3%, 74.5%, and 69.1% accuracy for GRU size 64, 128, 256, 512, and 1024, respectively. According to these results, with

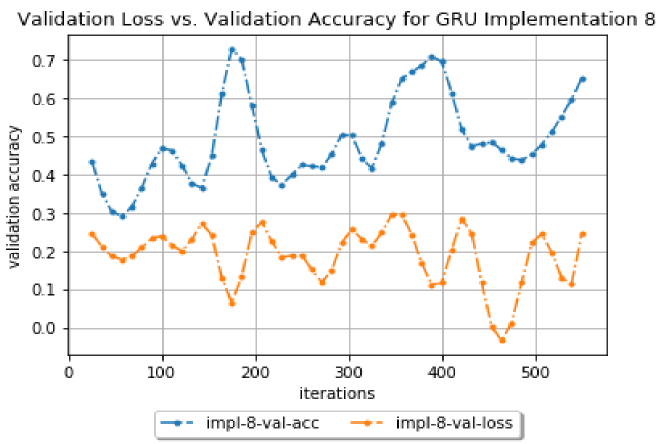

Figure 3: Validation loss vs. validation accuracy for GRU best implementation.

Table II: GRU Hyper-parameter tuning results

GRU implementation no.	**GRU size**	**No. of GRU layers**	**Batch size**	**No. of epochs**	**Test accuracy**
1	64	5	10	5	59.1%
2	128	5	10	5	70.0%
3	256	5	10	5	67.3%
4	512	5	10	5	74.5%
5	1024	5	10	5	69.1%
6	512	5	50	5	52.0%
7	512	5	50	10	61.0%
8	512	5	5	3	75.7%
9	512	5	1	2	70.3%
10	512	5	25	10	57.0%
11	512	5	25	5	61.0%
12	512	3	5	3	74.8%
13	512	10	5	3	69.6%
14	512	5	5	10	56.5%

size 512, the accuracy was the highest for this step [impl 4]. These results show that, in most of the cases, accuracy and GRU size acts proportionally.
As on the first step of Hyper-parameter tuning, the highest accuracy was generated for GRU size 512, we fixed this size on the second step, and tuned batch size along with no. of epochs. Our GRU model gained 52.0% accuracy for batch size 50 over 5 epochs, 61.0% accuracy for batch size 50 over 10 epochs, 75.7% accuracy for batch size 5 over 3 epochs, 70.3% accuracy for batch size 1 (that is, online learning) over 2 epochs, 57.0% accuracy for batch size 25 over 10 epochs, and 61.0% accuracy for batch size 25 over 5 epochs. Here, the results signify that accuracy falls for a very large batch size. Also, it is shown that the required no. of epochs is highly dependent on the batch size. However, for fixed batch size, accuracy increases with increasing no. of epochs. Hence, in case of applying GRU, a set of balanced batch size with reasonably large no. of epochs is required for high accuracy. The highest accuracy for this step was 75.7% for batch size 5 over 3 epochs [impl 8]. Accuracy for training with the online learning approach is 70.3% for over 2 epochs.
On the last step of Hyper-parameter tuning, we fixed GRU size to 512, batch size to 5, and tuned no. of GRU layers along with no. of epochs. Our GRU model accomplished 74.8% accuracy with 5 layers over 3 epochs, 69.6% accuracy with 10 layers over 3 epochs, and 56.5% accuracy for 5 layers over 10 epochs. These results led us to the decision that large no. of GRU layers does not necessarily help to perform better on a small dataset. The highest accuracy for this step was 74.8% for 3 layers over 3 epochs [impl 12].
Comparison among all GRU model implementation test accuracies is presented in Fig 4. The validation loss and validation accuracy of the best implementation [impl 8] for Hyper-parameter tuning are shown in Figure 3. The best accuracy we obtained was 75.7% for 5 layered GRU with size 512, batch size 5, and the learning rate 0.0001 over 3 epochs.
This work is the first attempt to utilize a Deep Learning approach for depression analysis in Bangla. Therefore, we are unable to directly compare our results with any relevant works. However, the method described by Suhara et al., is quite similar to ours [6]. They also used a Deep Learning approach for depression analysis. Nonetheless, they were able to create one of the largest depression datasets with 345,158 records over 22 months time span. Hence, they were able to apply LSTM-RNN with their huge dataset. On the contrary, our small dataset contains only 1,176 data, which is crucially insufficient to train LSTM model. This led us to utilize GRU model. Also, their work was to forecast depression in English. On the other hand, instead of forecasting, our model detects depression, which is in Bangla. Embracing all these constraints, our depression detection model still is able to gain maximum 75.7% accuracy, comparing to their LSTM model, which was able to gain maximum 88.6% accuracy. If we were able to create such kind of large dataset, we might be able to accomplish higher performances.

V. Conclusion and Future Works

In this research, we established a GRU model based depression detection technique by analyzing Bangla text data collected from social media (Twitter, Facebook, and Google form). We applied a sequence of Hyper-parameter tuning and showed the corresponding effects of these operations on a small dataset. The results indicate that

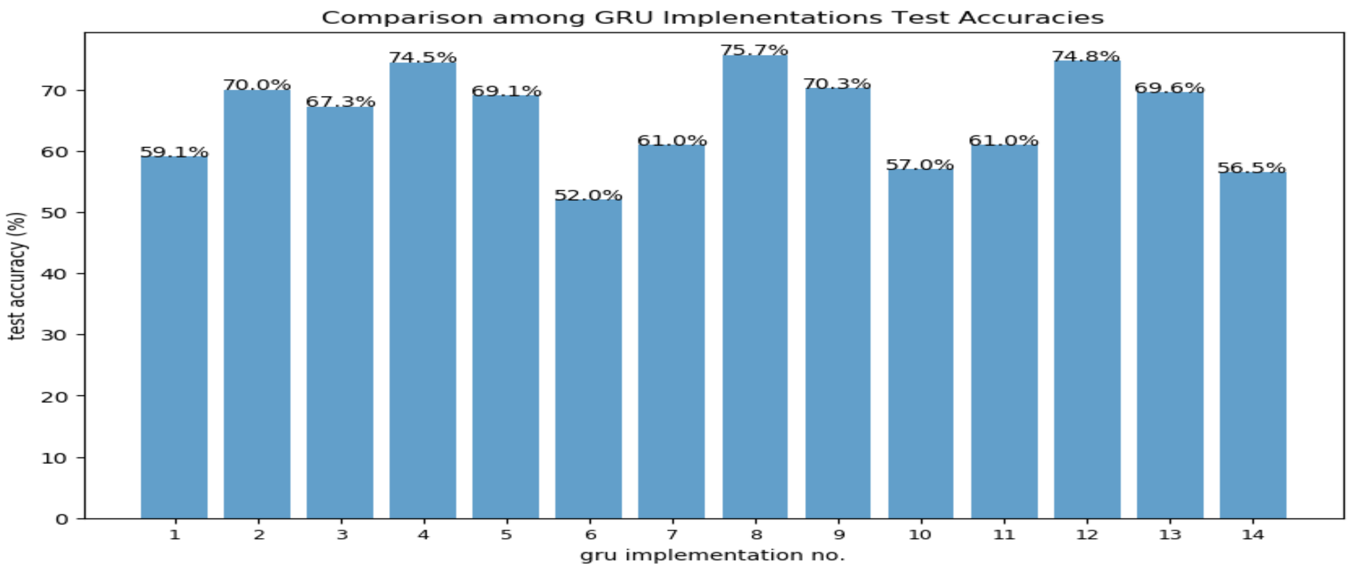

Figure 4: Comparing GRU implementation test accuracies.

in the case of a small dataset, accuracy depends on the variations of batch size and the number of epochs. We have also shown that the required number of epochs is highly dependent on the batch size. However, for fixed batch size, accuracy increases with increasing number of epochs. The results conclude that in case of applying the GRU model on a significantly small dataset, a set of balanced batch size with a reasonably large number of epochs is required to obtain adequate high performance.

This research will guide further investigations to analyze Bangla social media data for detecting depression from both small and large dataset using Deep Learning models. Additionally, multiple experienced persons can be used for labeling the dataset to make it more reliable. Furthermore, we have tuned only four Hyper-parameters without following any specific Hyper-parameter optimization approach. However, such kind of approaches, like Evolutionary optimization, Random search, Gradient-based optimization, and others can be applied for more in-depth analysis.

References

[1] "Depression," World Health Organization, 04-Jul-2017. [Online]. Available: https://www.who.int/mental_health/management/depression/en/. [Accessed: 21-Dec-2018].

[2] X. Wang, C. Zhang, Y. Ji, L. Sun, L. Wu, and Z. Bao, "A depression detection model based on sentiment analysis in micro-blog social network," in Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 201–213, Springer, 2013.

[3] "Languages of the World," Ethnologue. [Online]. Available: https://www.ethnologue.com/. [Accessed: 02-Feb-2019].

[4] A. Z. Riyadh, N. Alvi, and K. H. Talukder, "Exploring human emotion via twitter," in 2017 20th International Conference of Computer and Information Technology (ICCIT), pp. 1–5, IEEE, 2017.

[5] S. Chowdhury and W. Chowdhury, "Performing sentiment analysis in bangla microblog posts," in 2014 International Conference on Informatics, Electronics & Vision (ICIEV), pp. 1–6, IEEE, 2014.

[6] Y. Suhara, Y. Xu, and A. Pentland, "Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks," in Proceedings of the 26th International Conference on World Wide Web, pp. 715–724, International World Wide Web Conferences Steering Committee, 2017.

[7] A. Hassan, M. R. Amin, A. K. Al Azad, and N. Mo- hammed, "Sentiment analysis on bangla and roman-ized bangla text using deep recurrent models," in 2016 International Workshop on Computational Intelligence (IWCI), pp. 51–56, IEEE, 2016.

[8] E. F. Can, A. Ezen-Can, and F. Can, "Multilingual sentiment analysis: An rnn-based framework for limited data," arXiv preprint arXiv:1806.04511, 2018.

[9] D. Ayata, M. Saraclar, and A. Ozgur, "Busem at semeval-2017 task 4a sentiment analysis with word em- bedding and long short term memory rnn approaches," in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 777–783, 2017.

[10] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," arXiv preprint arXiv:1702.01923, 2017.

[11] Mchablani, "mchablani/deep-learning," GitHub. [Online]. Available: https://github.com/mchablani/deep-learning/blob/master/sentiment-rnn/Sentiment_RNN.ipynb. [Accessed: 30-Jan-2019].