Sentiment Extraction From Bangla Text : A Character Level Supervised Recurrent Neural Network Approach

Conference Paper · February 2018

DOI: 10.1109/IC4ME2.2018.8465606

CITATIONS
5

READS
230

3 authors:

Mohammad Salman Haydar
DataShall Analytics Ltd
2 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Mustakim Al Helal
University of Regina
7 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Syed Akhter Hossain
Daffodil International University
99 PUBLICATIONS **476** CITATIONS
SEE PROFILE

Sentiment Extraction From Bangla Text : A Character Level Supervised Recurrent Neural Network Approach

Mohammad Salman Haydar
Computer Science and Engineering
Daffodil International University
Dhaka,Bangladesh
Email: salman3045@diu.edu.bd

Mustakim Al Helal
Computer Science
University of Regina
Regina, SK, Canada
Email: mhx049@uregina.ca

Syed Akhter Hossain
Computer Science and Engineering
Daffodil International University
Dhaka,Bangladesh
Email: aktarhossain@daffodilvarsity.edu.bd

***Abstract*—Over the recent years, people are heavily getting involved in the virtual world to express their opinions and feelings. Each second, hundreds of thousands of data are being gathered in the social media sites. Extraction of information from these data and finding their sentiments is known as a sentiment analysis. Sentiment analysis (SA) is an autonomous text summarization and analysis system. It is one of the most active research areas in the field of NLP and also widely studied in data mining, web mining and text mining. The significance of sentiment analysis is picking up day by day due to its direct impact on various businesses. However, it is not so straightforward to extract the sentiments when it comes to the Bangla language because of its complex grammatical structure. In this paper, a deep learning model was developed to train with Bangla language and mine the underlying sentiments. A critical analysis was performed to compare with a different deep learning model across different representation of words. The main idea is to represent Bangla sentence based on characters and extract information from the characters using a Recurrent Neural Network (RNN). These extracted information are decoded as positive, negative and neutral sentiment.**

***Index Terms*—Bangla, Sentiment Analysis, RNN, Deep Learning, Character level RNN, NLP in Bengali**

I. Introduction

One challenge in understanding the user opinions from social media is to extract the information from the large amount of opinionated text. It become more complicated when opinions are not made explicitly. However, it is a difficult and a time consuming task for human beings to classify different data and extract the opinion. Sentiment analysis has become vital to data science since the online review is becoming more popular every day. There are many conventional methods for sentiment analysis. Deep learning techniques have also been used in sentiment analysis. For instance, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) etc have been used in practice to solve sentiment analysis problems.

However, sentiment analysis for the reviews or short texts in Bangla has not been much of an addressed research in a large scale till date. Bangla has now an increasing amount of texts used in social media e.g Facebook, blogs etc. Therefore analyzing Bangla text will open a new horizon towards the real life intelligence operations on online sectors. Judging others opinions has a better demand to various businesses. In order to have a better analytics to generate a more accurate information, we need to be able to analyze people reviews. The main contributions of this paper are as follows:

- Showing the effect of character-level representation in Bangla language.
- Making a comparison on traditional representation of words with our approach.

The paper is organized in different sections. After a brief summary of related work in section II we discussed about the data collection, preprocessing and character encoding in section III. Then the methodology, the model and experimental setup has been discussed in section IV. The following section after that, demonstrates the results of our experiment and discussed about the experimental process. Finally, future work and conclusion were drawn.

II. Related Work

Due to the complex grammatical structure and less resources in Bangla, the language went through a very few research work so far and most of the researches on SA have been carried out in English language. Researchers proposed different methods to get the state-of-the-art results. However, some past works related to this topic were studied for this paper.

In [1] a sentiment analysis was performed on Romanized Bangla and Bangla text collected from different social medias. They applied deep recurrent neural network (LSTM) to train their model and they got accuracy of 78% with categorical crossentropy loss.

In [2], the authors have used a semi-supervised method to identify the sentiment of twitter posts. They first annotated the post into positive and negative polarity using a rule based classifier to make training data and then used this data to train their sentiment classifier. They used support vector machine (SVM) and Maximum Entropy (MaxEnt) algorithm and they achieved a result of 93% accuracy on SVM using emoticons as features.

A hybrid method was proposed in [3] to identify the sentiment from the sentence. The authors first determined whether a sentence is subjective or not and they designed a model from the mixture of various parts-of-speech-tagging (POS) features collected from phrase level similarity and then they used syntactic model to perform sentiment analysis. By doing so they achieved overall 63% recall rate using SVM on the news data.

Some other researchers have also worked on social media data to identify the sentiment in [4]. Here they used sentiment analysis on a specific domain. They collected the data (post) from a Facebook group and then applied two different methods to identify the polarity of a post. One of the approaches are Naive Bayes and another one is by using lexical resources. After the experiment, they found that in specific domains lexicon based approach performs better than the other ones.

III. Data Collection and Preprocessing

A. Data Collection

The Data have been collected from Facebook page using Facebook graph api. The data are mostly comments of the users in the posts on the Facebook and we have also collected the reviews from the pages, specifically from the e-commerce and restaurant Facebook pages. As reviews contains direct opinions of the users. We collected 45 thousand plus data from Facebook.

B. Data preprocessing

We remove all the unnecessary data tuples except those containing Bangla. Then we tagged those data manually into Positive, Negative and Neutral class. Figure 1 is showing how the data is looks like after cleaning noisy data.

Text	Class
ভাই আমি একটা নতুন সিম ক্রয় করেছি কিন্তু করতে গেলে টাকা এ আসে না এখন কি করতে পারি	Negative
যে যাই বলুক না কেন রবি নেট ই সেরা ইন্টারনেটের ভাল অপার শুদু রবিই দেই নাইট পেক ৮৮ টাকা দিয়ে ৫জিবি কি মজা	Positive
এ অল্প দামের কি কোন মোবাইল অফার আছে কি কি মোবাইল আছে প্লিজ জানাবেন	Neutral
স্পিড নাই কেউ নিয়েন না ১৯ টাকায় পানিতে যাবে	Negative
ধন্যবাদ	Positive
বন্ধ সিমে ডাটার কি অফার আছে	Neutral

Figure 1. Dataset Sample

And Table I showing the data statistics of our data set after performing cleaning operation. Noisy data are considered those containing english words or only emoticons or only randomized bangla words which are not necessary for our classification.

C. Character Encoding

For using this data to our model, we first represented this dataset in a vector space. There are different methods to represent text data. Tf-Idf, Bag of Words and distributed representation of words (e.g word2vec, Glove) etc are some

Table I
Data Statistics

Class	Number
Positive	8271
Negative	14000
Neutral	12000
total	34271

examples. The major drawback of these representations is that they strictly rely on the words of the documents and if any word is found that was not observed during the training period then the model would not understand this and this word will have no effect on the model. In most of the research, words are considered as the unit of the sentence but it can also be a character. In our research we have taken characters as a unit of the sentence.

In [5], Xiang Zhang et al. performed an empirical study on character level text classification using Convolutional Net on English dataset and found that this method works well on the real life data or on data that is generated by the users. The accuracy depends on some other factors, for instance choice of alphabets, size of the dataset etc. In our work we chose 67 alphabets of Bangla language including space and some special characters. Figure 2 is showing the characters that we have included. We didnt include any numeric characters in

Figure 2. Characters

Bangla. The characters one, two and three in Bangla is used instead of three other Bengali letters for the representation purpose due to pythons limitation to recognize them. We then encoded each character in a sentence using a unique id from the list of characters. This process is illustrated in Figure 3. Here the length of the sequence is l=1024 and we

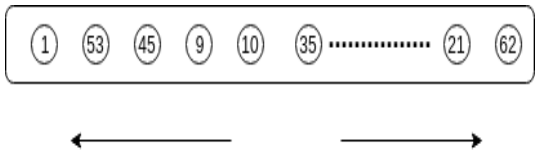

Figure 3. Illustration of Encoding

believe within this length we can take most part of a sentence. Sentences less than the length of 1024 characters were padded using zero and sentences more than 1024 characters truncated to 1024. The characters other than the selected 67 are removed before the encoding phase using regular expression.

IV. Methodology

Deep Learning method has been applied successfully to the Natural Language Processing problems and achieved the state-of-the-art results in this field. Recurrent Neural Net [6] is a kind of Neural Network which is used for processing

sequential data. But later on, the researchers found some mathematical problems to model long sequences using RNN [7][8].

A clever idea was proposed by Hochreiter and Schmidhuber to solve this problem. The idea is to create a path and let the gradient flow over the time steps dynamically [9]. It is know as Long Short Term Memory (LSTM). It is a very popular and successful technique to handle long-term dependency problem. There are some variants of LSTM. One of them is Gated Recurrent Unit (GRU) proposed by Cho et al. [10]. The difference between LSTM and GRU is that it merged forget and input gates into a update gate which means it can control the flow of information but without the use of memory unit and it combines cell state and hidden state along with some other changes. The rest of the thing is the same as LSTM. In [11] Junyoung Chung et al. conducted an empirical study on three types of RNN and found that Gated Recurrent Unit is superior then other two. GRU is also computationally more efficient than LSTM.

$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \tag{1}$$

$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \tag{2}$$

$$\tilde{h_t} = \tanh(W.[r_t * h_{t-1}, x_t]) \tag{3}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \tag{4}$$

Here are the equations that demonstrates how the hidden state h_t is calculated in GRU. It has two gates, one is the *update* gate z, another one is the *reset* gate r. equation (1) and (2) are showing how these two are calculated. The reset gate determines how to combine the new input with the previous memory, and the update gate defines how much of the previous memory to keep around. And finally hidden state h_t is calculated as equation (4).

However, the classification task of the sentiment is a step by step process. For example, if we want to classify the first sentence from the Figure 1 then at first it will go through the preprocessing step. Here all the characters except the defined ones above will be filtered out from the sentence and the remaining sentence will be represented in a vector space. Every character will be given a numeric id and then it will be padded by zero to 1024 characters (any sentence with more than 1024 character will be compressed down to 1024). This vector will be fed through the model and eventually the model maps the input sentence to a sentiment class. In each hidden layer of the model, more lower level and meaningful features are extracted from the previous layer and the output layer calculates the softmax probability of each of the class. The class which has the highest probability is the predicted result. For simplicity and better understanding of the reader we tried not to put all the mathematical details about how the model learned through backpropagation.

A. Model

The baseline model that we compared with consists of one embedding layer with 80 units and 3 hidden layers where two with 128 LSTM unit and another one is the vanilla layer with 1024 unit and output layer with 3 units. We have used a dropout [12] layer between output layer and the last hidden layer with the probability of 0.3.

In our model we used an embedding layer with 67 units, 3 hidden layers where two layers are with 128 GRU units each and one vanilla layer with 1024 units stacked up serially and at last the output layer. Here we have also used a dropout of 0.3 between the output layer and the last hidden layer. Our model is illustrated in Figure 4.

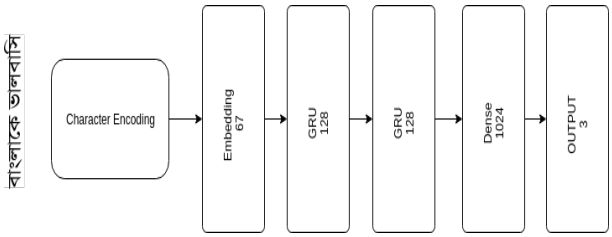

Figure 4. Model Structure

B. Experimental Setup

We ran our model in 6 epoch with batch size of 512 and we used Adam [13] as our optimizer. We also used categorical cross entropy as our loss function. We set the learning rate at 0.01 to train our model. Many different hyperparameters (learning rate, number of layers, layer size, optimizer) were used and this gave us an optimal result. The embedding size was kept 67 as we have 67 characters and the dropout was set to 0.3 between the output layer and the dense layer of the two models. Early stopping was used to avoid overfitting. All our experiments were done in python library named keras [14] which is a high-level neural networks API.

C. Results and Discussion

The result that we achieved from the character level model over word level model is pretty good. We came up with 80% accuracy on character level mode and 77% accuracy from our baseline model with word level representation. Over the recent time, sentiment analysis achieved a highest of 78% accuracy in [1] using LSTM in Bangla with two class classification. Figure 5 showing the training and testing loss of our model. Here we can see that after a certain epoch the training loss started decreasing more than the testing loss. Training keeps decreasing. The testing loss on the other hand decreases at a slower rate compared to the rate of the training loss. So we stopped training at epoch 6 resulting in the saving the model from overfitting. Figure 6 is showing the training and testing accuracy of our character level model. and Figure 7 is showing the comparison between the two models.

The most important observation from our experiments is that the character-level RNN could work for text classification without the need for words semantic meanings. It may also extract the information even if the word is not correct as we are now going through each of the characters individually.

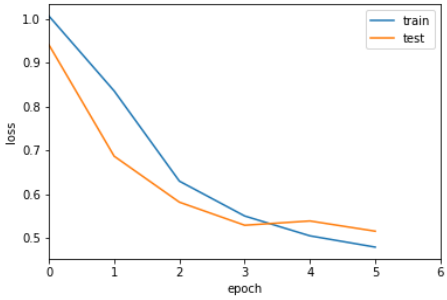

Figure 5. Training and Testing loss

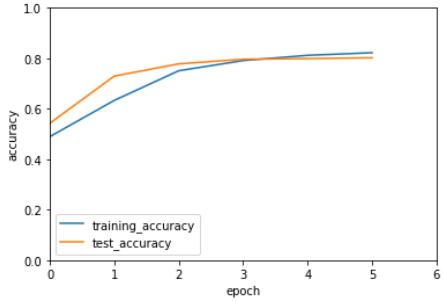

Figure 6. Training and Testing accuracy

However, we need to undergo more study to prove this for Bangla. So we can use this representation to handle real life data from social media. However, to observe the performance of this model across different datasets, more research is needed. Nevertheless, the result depends on various factors including the size of the dataset, alphabet choice, data quality etc. But our dataset is focused on a specific telecommunication campaign domain. So this model can be helpful on some specific application.

We calculated the accuracy as a ratio of correctly classified data and a total number of data from the test set. The equation is as follows:

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \tag{5}$$

V. Future work and Conclusion

To conclude, this paper offers a research based study on character-level RNN for sentiment analysis in Bangla. We compared it with a deep learning model with word level representation and found a good result. However, the model is not a generic of it kind since it worked well with data from a specific domain. Also, we did not address sarcastic sentence analysis in this model. So, if a positive word is used in the sentence with a negative sarcastic perspective the model will not be able to detect this. Hence, this needs to be addressed which is a challenge due to the level of abstraction an user can create through one sentence. So, intensive research is needed in this regard. Our analysis shows that character-level RNN is an effective method to extract the sentiment from Bangla. The model however is still immature and yet to be applied to Romanized Bangla. So, making the model more reliable

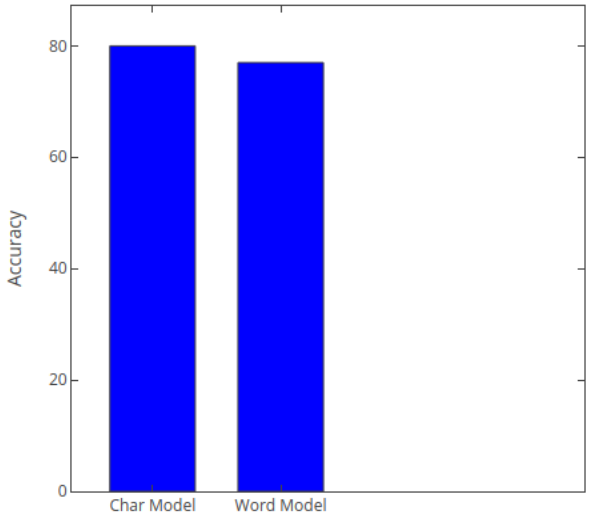

Figure 7. Comparison of the two models

across different data is one future goal of this project that will make it useable in the industry level to extract sentiment from the social media reviews and comments.

References

[1] Hassan, Asif, et al. "Sentiment analysis on bangla and romanized bangla text using deep recurrent models." Computational Intelligence (IWCI), International Workshop on. IEEE, 2016.

[2] Chowdhury, Shaika, and Wasifa Chowdhury. "Performing sentiment analysis in Bangla microblog posts." Informatics, Electronics & Vision (ICIEV), 2014 International Conference on. IEEE, 2014.

[3] Das, Amitava, and Sivaji Bandyopadhyay. "Phrase-level polarity identification for Bangla." Int. J. Comput. Linguist. Appl.(IJCLA) 1.1-2 (2010): 169-182.

[4] Akter, Sanjida, and Muhammad Tareq Aziz. "Sentiment analysis on facebook group using lexicon based approach." Electrical Engineering and Information Communication Technology (ICEEICT), 2016 3rd International Conference on. IEEE, 2016.

[5] Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." Advances in neural information processing systems. 2015.

[6] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533.

[7] Hochreiter, Sepp. "Untersuchungen zu dynamischen neuronalen Netzen." Diploma, Technische Universitt Mnchen 91 (1991).

[8] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks 5.2 (1994): 157-166.

[9] Hochreiter, Sepp, and Jrgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[10] Cho, Kyunghyun, et al. "On the properties of neural machine translation: Encoder-decoder approaches." arXiv preprint arXiv:1409.1259 (2014).

[11] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).

[12] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of machine learning research 15.1 (2014): 1929-1958.

[13] Kingma, Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[14] C. François and others, "Keras", Keras.io, 2015. [Online]. Available: http://keras.io. [Accessed: 16- Nov- 2017].