Sentiment Analysis on Bengali News Comment Dataset

Syed Mahbubul Huq
ID 220033725
MSc Data Science
syed.huq@city.ac.uk

1 Problem statement and Motivation

Bengali is one of the most popular languages in the world having 300 million native speakers. It is also considered as the sixth most spoken language in the world. Besides its popularly, it is also considered a very difficult language because of its difficult and complex grammar. Having complex grammatical structure, alphabets, pronunciation and vocabulary makes Bengali a divergent and difficult language. Although it is a popular and complex language, enough study and use of Bengali language in the field of NLP has not been done.

NLP (Natural Language Processing), has a wide use and its popularity is increasing every single day. Sentiment analysis is a popular technique of NLP used to understand sentiments from text. It is used by businesses to monitor customer sentiments which they can use to understand and classify their audience well. One major place to get variety of peoples comments and sentiments attached to it is public comments that can be gathered from online news portal which can be studied to understand people's behavior and opinion.

In order to contribute in the field of NLP on Bengali language, the task of doing sentiment analysis on public comments in news portals are chosen. This would give us a scope to study variety of positive, negative and neutral comments made by several individuals which would make our analysis more valid. Besides, by conducting sentiment analysis on news portal comments we can have an understanding of people's opinion and views on different types of news which can later be used in different sectors for customer segmentation, easy understanding of people's opinion etc.

2 Research hypothesis

In order to get optimal result, different word embedding algorithms should be tested and experimented along with tweaking of selected parameters and word features incorporated with various machine learning algorithms for classification.

For our language Bengali, word features like stemming and stop words addition or deduction can have an impact on the overall result. This factor should also be considered by us while carrying out the experiment. Along with all of these features and working steps, a correct approach in choosing parameters for our algorithms should be done. With out proper tweaking, optimal solution cannot be reached. Lastly, our experiment should not be limited with one type of algorithms, different algorithms with completely different structures should be chosen in order to understand the inner working procedures and efficiency for machine learning models for Bengali language.

3 Related work and background

For our experiment we decided to do background studies and take motivation from the work done on sentiment analysis for Bengali language.

In 2014, S. Chowdhury and W. Chowdhury worked on a paper in conducting sentiment analysis on Bengali Microblog comments. They got maximum accuracy conducting the experiment with SVM.
Later on, in the year 2016, A. Hassan along with his co-authors made an interesting and noteworthy contribution for sentiment analysis using Bengali language, he used deep learning algorithms to find experiment on Bengali and Romanised Bengali language. He achieved the best accuracy using LSTM classification algorithm.

Following the improvement steps, in 2017, M. H. Alam, along with his co-author proposed a sentiment classifier system which performed with almost 98% accuracy rate. They used CNN for their proposed framework.

In 2018, with the growing popularity of different word embedding techniques, S. Hossain along with his co-authors explored different word embedding techniques for Bengali language. This analysis was an important analysis in context to Bengali language in the field of NLP.

While most of the mentioned papers till now dealt with positive and negative sentiment analysis, in 2019 R. A. Tuhin, along with his co-author proposed a hybrid model along with Naïve Bayes to analyse emotion sentiments like happy, sad etc from Bengali test.

In the following year in 2020, S. Haque along with his co-authors used RF, SVM, LR and NB and compared the algorithms in conducting sentiment analysis on Bengali texts.

In the same year, S. Sharmin along with co-author for the first time for sentiment analysis in Bengali introduced attention-based CNN. They achieved the best accuracy for their model combining attention-based CNN along with word embedding technique w2v.

Relating to the previous mentioned paper, sentiment analysis in Chinese language using attention-based mechanism was also studied, as it has similarity with Bengali language in terms of complexity.

Lastly, for our experiment, two important papers where same datasets which we would use was used were the works of Md. Asik and co-authors, and the paper of U. Saha and co-authors published in 2019 and 2022 respectively. The first paper in fact is the dataset evaluation paper where the evaluated the created dataset. For our analysis we would use this dataset. They performed evaluation using SVM, RNN, LSTM and CNN. They achieved maximum accuracy using LSTM. In the later paper, hybrid approach using Bi-LSTM-CNN along with glove was done and a better accuracy was achieved. In the papers which uses our dataset does not deal with linear models like NB, DT or even neural model like MLP. We would incorporate this and compare the result with the work already done with the dataset.

4 Accomplishments

1. Data pre-processing Completed
2. Feature extraction: Completed
3. EDA: Completed
4. Word feature extraction- Completed
5. Using pretrained word embedding techniques like Glove- Not completed. Because of available time and limited pretrained Bengali resources, pretrained word embedding technique was not successfully implemented. For future works, this technique can be implemented.
6. Training-Testing-Error Analysis- Completed
7. Compare models- Completed

5 Approach and Methodology

For completing the task and relate it to our research hypothesis, a planned approach and methodology needs to be followed. Our approach steps is mentioned below:

1. **Dataset selection:** This is the most challenging part for our experiment. As there are limited resources available in Bengali for sentiment analysis, a suitable dataset had to be chosen. After extensive research a suitable dataset was collected More details of the dataset are mentioned in the Dataset section of this report
2. **Data pre-processing:** In this step we clean our data and make it ready for further analysis. More description of this step is discussed further in the report.
3. **Tokenization:** We tokenize our sentence using word_tokenize function of NLTK library.
4. **Stopwords removal:** Using NLTK library, we call Bengali stop words list and remove the stop words from our tokenized words. We keep two copies of our tokenized text, one with stop words removed and the other including stop words, we would use it for comparison later on.
5. **Bengali Stemmer:** We instal a library called bangla-stemmer which we use to find the stem of Bengali words. This is an important process as this would help us to make the words less complicated and this would be useful when applying our model.
6. **EDA and Basic Dataset Statistics:** We perform some EDA and do basic statistics

in our dataset; more would be discussed in the Dataset section of the report.

7. **Word Embedding:** We perform word embedding and convert our tokenised words to vector using Word2Vec and TF-IDF
8. **Split dataset:** We split our dataset into 20% testing and 80% training ratio and export the dataset using pickle.
9. **Train-Test-Evaluate:** Lastly, we train our model, test it and evaluate it for error analysis. More details about our chosen algorithms are described in later part of this section of the report

For following the above methodology, we use word embedding algorithms W2V and TF-IDF and for our classification, we used Naïve Bayes+TF-IDF as our baseline and we run our experiment with Decision Tree and MLP incorporated with our word embedding algorithms.

Word2Vec: Word embedding technique W2V is considered one of the most efficient ways to convert words to numbers (vectors). Gensim library is to create W2V model very efficiently. In this technique, vectors are chosen based on cosine similarity that exists between them which is able to carry semantic meaning of the words.
Pros: - Able to capture semantic meaning
- Size of the vector is very small.
Cons: - Inefficient in handling new words
- No shared representation at sub words

TF-IDF: The TF-IDF strategy is a sort of bag words approach where it assigns value to each word based on the frequency it appeared.
Pros: - Improved version of simple bag of words approach
Cons: - Requires huge sparse matrix and vector size is big
- Computational cost is high

Decision Tree: Machine learning algorithm that uses tree structure to mainly solve classification task is decision tree. In a decision tree, each leaf node represents a class label, whereas each internal node represents an attribute.
Pros: -Requires less effort in pre-processing stage
- Does not require data to be normalised or scaled.
Cons: -Higher time to train
- A small change in parameter can give overall drastic change in result

MLP: A class of feedforward artificial neural network is called a multilayer perceptron (MLP). A MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer.
Pros: -Has a lot of parameters which can be useful for a lot of experiments
- After training testing time is very fast
Cons: -Sometimes because of too many parameters a small change can have drastic effect
- Sensitive to feature scaling.

Choice of Methodology:
Training Methodology: Training models and algorithms are chosen by us very carefully. With respect to the works done previously on our dataset, we decided to choose our training algorithms which has not been used yet. This would give us an option to compare the dataset and study the nature of NLP algorithms
In previous works done for sentiment analysis on our dataset, the authors used RNN, SVM, LSTM, CNN and a hybrid approach (Bi-LSTM-CNN). Overall, they performed well on Recurrent Networks LSTM and Bi-LSTM-CNN. Besides a pretrained word embedding technique was also used, but it did not give higher accuracy.
This gave us inspiration to choose models which has not been used yet to study how the algorithms differ from each other. With this in mind we decided to choose a simple linear machine learning model, Decision Tree and a neural model, MLP. Because of the difference in nature for both of the algorithms and different in structure of the algorithms, it would give us more diversity to study the algorithms and how they differ from each other when experiments are carried out with the same dataset. Besides, both of the algorithms have a good number of parameters, it is very important to tweak and experiment with parameter while doing sentiment analysis on a complex language like Bengali.
The decision for choosing TF-IDF and W2V for word embedding was done solely because of the complete difference in nature on how they perform. As one relies on document retrieval, the other relies in catching semantic information of the text. This would give us diversity in our experiment. Besides, as deep learning models were already being used in previous works on the same dataset, a variation in choosing algorithm with different structure can give us more diversity for algorithm understanding.

Evaluation Methodology: We would use confusion matrix on our model and see how it performs on our test data. As our experiment is dealing with classification problem. Using confusion matrix for our experiment would be an ideal choice. Besides, we would also do manual classification to see where and why our model misclassified to be a different sentiment.

6 Dataset

Our dataset was downloaded from Mendeley Data. A paper in IEEE is also published on the making, structure and baseline evaluation of the dataset. Three individuals annotated the dataset which increases the reliability of the annotations. The data collected was from a widely popular online news portal Prothom-Alo's user comments.

The dataset contains 13,802 data in total along with the annotated sentiment. In total there are 5 different kinds of sentiments Positive, Negative, Slightly Positive, Slightly Negative and Neutral sentiments.

As the dataset is made from public comments from an online news portal, emotions in words for a wide range of people can be captured. This makes our selected dataset unique and challenging. Besides, while commenting, individuals don not carry a formal tone, which makes our data unbiased.

Basic Statistics: We conducted an experiment on our dataset to get basic statistics of textual data:

-Total number of words: 273089

-Total number of unique words: 44501

-Maximum number of words used in each comment: 125

-Avg number of words used in each comment: 20

-Standard deviation words in comment: 17

-Total number of characters 1613301

-Total number of unique characters: 138

-Maximum number of characters used in each comment: 599

-Average number of characters used in each comment: 117

-Standard deviation characters in comment: 102

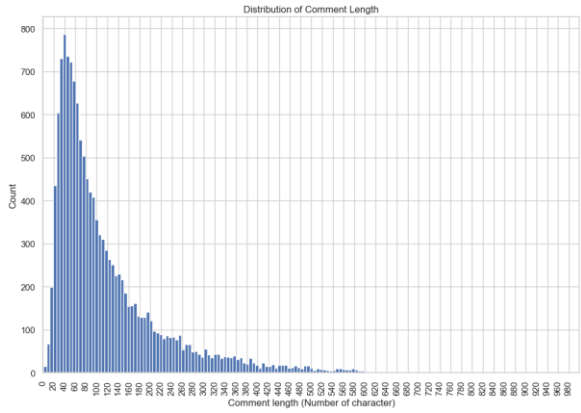

Fig 1: Distribution of Comment Length

In the above figure distribution of comment length in our data is presented.

6.1 Dataset preprocessing

Our data pre-processing involves some particular steps. At first unnecessary columns from our dataset are removed and the dataset columns are renamed to meaningful names.

For the ease and understanding of our analysis, our target sentiments, Positive, Negative, Neutral, Slightly Positive and Slightly Negative are renamed from Bengali to English and to get better understanding of the analysis the classification problem is made to 3 sentiments from 5. We consider Slightly Positive, and Slightly Negative sentiments as Positive and Negative respectively. So, finally we have 3 different sentiments, Positive, Negative and Neutral for our analysis. Afterwards, we check if our analysis has any missing value and we deal with it. We then remove punctuations, numbers and special characters from our Bengali text. Then we conduct tokenization, stemming and stop word removal. It is important to follow this step as this would give us the characters and inner features of the words from the text.

7 Baselines

For the analysis a machine learning model, Naïve Bayes along with word embedding technique TF-IDF is chosen by us. We would use this as a base model to compare the sentiment analysis result with our other chosen algorithms, MLP and Decision Tree. Because of the simplicity of the algorithm, Naïve Bayes is chosen. Besides it does not have too many parameters to tune, this is one of the unique characteristics and shows the simplicity of the algorithm. For our analysis it was

necessary to chose a simple baseline model. As the language we are dealing with, historically showed variation in performance with different models, choosing a simple model for the experiment can be favorable for us. Word embedding technique TF-IDF is chosen to incorporate with Naïve Bayes. Compared to our other word embedding algorithm, TF-IDF.

Although we are performing our analysis with completely different approach which has not been yet done with our dataset. But with the relevant works that already has been done, it is evident that complex, hybrid and deep learning models along with complex word embedding technique tends to perform well. From this, in comparation with our baseline and the model that we would use for our analysis, we can make the following hypothesis statement.

Hypothesis statement: Our base model is expected to perform faster because of the simplicity in nature of it. But in terms of predictive performance for sentiment analysis, complex models is expected to perform better compared to simple models, which in our case would make MLP incorporated with W2V a winner for correctly classifying sentiments.

8 Results, error analysis

For choosing optimal parameters for our model, we decided to run grid search with 5-fold stratified cross validation with our models, Decision Tree and MLP. We set the predefined hyper parameter set on which our grid search will run. For time limitation and to make our experiment less complex we decided to do grid search with our models incorporated with W2V only. For future works we would incorporate grid for our models along with TF-IDF too.

The results on table for grid search conducted on both the algorithms are presented in the Appendices (Fig 2 and Fig 3)

For Decision Tree, 'max_depth': 2, 'min_samples_leaf': 2 gave us the best validation accuracy of 51.4%. From the table it can be observed for sentiment analysis in case of Decision Tree hyperparameter tuning did not give any drastic change of result. We can also observe that when the value of Maximum Depth and Minimum Sample Leaf is less, our model tends to perform comparatively better. This gives us an indication of for Bengali language, minimum hyper parameter values can give us good result for Decision Tree.

In case of MLP, 'activation': 'logistic', 'hidden_layer_sizes': (5, 5), 'learning_rate_init': 0.03 gave us the best validation accuracy of around 51%. For both our chosen algorithms, almost similar result can be seen in case of validation accuracy. An interesting characteristic of MLP can be observed that the model tends to perform well when the activation is set to logistic. With almost every combination activation function of logistic performed better.

Below table gives us summary of the results of our experiment and we can use it to evaluate our results:

Models	Testing Acc	Training Time
NB + TF-IDF (Base)	52.30%	0.02 sec
DT + TF-IDF	54.00%	0.10 sec
DT + W2V	52.10%	0.36 sec
MLP + TF-IDF	43.00%	12.5 sec
MLP + W2V	53.38%	2.11 sec

It can be observed that in terms of time complexity, our hypothesis statement was proven right. Naïve Bayes performed excellently well in case of time complexity. But overall, in terms of predictive performance, Decision Tree incorporated with TF-IDF gave us the best testing accuracy of 54%. Relating to the hypothesis statement, it was assumed MLP and W2V would perform the best, though it is not far behind with 53.38% accuracy. Among all the models, MLP incorporated with TF-IDF gave us the worst result and in terms of both testing accuracy and training time. The training time taken were a lot compared to our other models. This proves about the complexity of MLP and shows the black box nature which it has. If proper tuning is not done, and if proper word embedding techniques are not incorporated, MLP would not perform well, which is also the same case for complex language Bengali which we are dealing with.

To analyze more in depth, we would discuss more about our models result and try to critically evaluate the result. Due to word and time limitations, we would only discuss about our baseline model, Naïve Bayes+TF-IDF along with MLP+TF-IDF to understand more about the inner structure. Rest of the confusion matrix would be presented in the appendices. To compare with our base model, we choose MLP+TF-IDF, because after looking at all the model's classification report,

I thought MLP would be worthy of mentioning despite of it having the least test accuracy only this model showed a little unbiasedness in classification In the below figure, we represent confusion matrix for these two models and try to understand and find our how our base model differs from the best performing model.

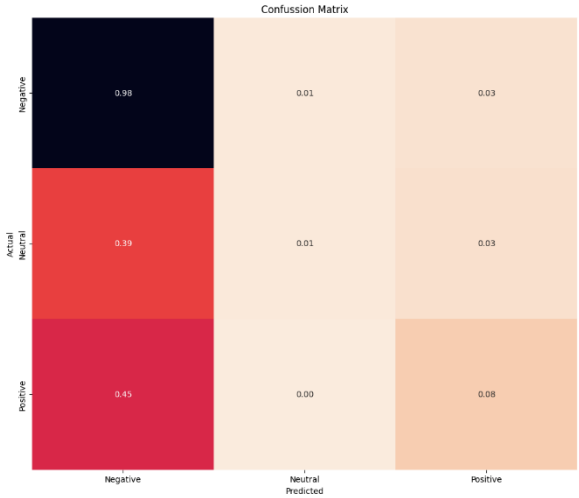

Fig 4: Naïve Bayes+TF-IDF

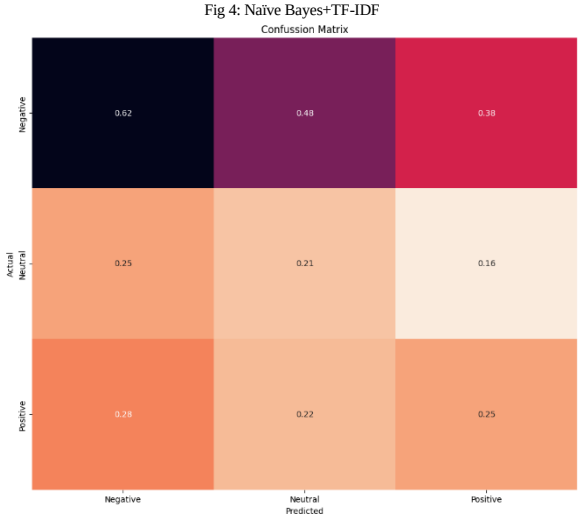

Fig 5: MLP+TF-IDF

Here we can see that our base model correctly identified negative sentiments almost perfectly compared to other sentiments. This shows about the biasness of our model. But on the other hand MLP along with TF-IDF was able to handle the biasness, it was able to identify Neutral and Positive sentiments in a slight better way compared to our other algorithms whose confusion matrix are also presented in the appendices.

Another interesting fact, worth mentioning is that among all the models, MLP+TF-IDF took the most time in training. And compared to other models, more time in training for this model was worth waiting as it was able to perform better in being unbiased in its classification result.

Although the reference paper we followed performed well with 5 sentiments, but most of our model was not able to show distinguish and handle biased data. In the reference paper, of U. Saha and co-authors they were able to achieve an accuracy of 89% using hybrid deep learning method, Bi-LSTM-CNN even without sampling or balancing their dataset. This also proves about their hybrid proposed structure and its ability to handle biased dataset. Besides, our approach was also different to them as we were analysing sentiments for 3 different categories instead of 5.

Referring to the hypothesis statement, that we made that in case of Bengali, complex models were able to handle sentiments better compared to simple linear models. Our models Naïve Bayes, and DT are simple compared to neural structure of MLP. I think this is because of the complex structure that MLP processes, our model MLP was able to correctly identify Neutral and Positive sentiments being comparatively less biased although the overall accuracy of this model was not that good.

Seeing the confusion matrix of MLP+TF-IDF, we can also state that our algorithm sometimes misclassifies Neutral sentiment as Negative. This shows about one of the characteristics of our model of wrongly identifying Neutral sentiments. To understand a bit more of the misclassification, I would manually gather some misclassified by our model MLP+TF-IDF and present about the misclassification and the language features involved in a manual way. In the project notebook, a separate dataset of the misclassified data is created by me. On this report I present first few samples of it where our model misclassified Negative sentiments to be neutral. I present it in the appendices over here. Another worth mentioning character of our model misclassifying is maybe because of length of words, it was observed some of the misclassified sentences were of less words, which is why our model did not correctly classify it.

After also carefully observing the data, I saw some negation words being removed in stop word removal process, which can be taken into consideration in future works

9 Lessons learned and conclusions

From our experiment, we came to understand that because of complexity of Bengali, linear models cannot perform well for doing sentiment analysis. Simple models were not able to handle baseness of the dataset. Although, compared to all the simple models that we used, MLP+TF-IDF was a bit complex and it was less biased in classification but the overall accuracy of 43% in predicting sentiments is not good. Besides, MLP+TF-IDF being less biased also had highest training time.

Complex algorithms like deep learning and neural networking have shown good performance in past works on our dataset, so for future works, hybrid and complex models should be incorporated to get better accuracy. Pre-trained word embedding techniques like GloVe can be used to see and experiment with the performance. Besides, we can extend our experiment by oversampling/undersampling the dataset and compare with the result that we have here.

References

Chowdhury, S., & Chowdhury, W. (2014b). *Performing sentiment analysis in Bangla microblog posts*. https://doi.org/10.1109/iciev.2014.6850712

Hassan, A., Amin, M. M., Azad, A. K. A., & Mohammed, N. (2016c). *Sentiment analysis on bangla and romanized bangla text using deep recurrent models*. https://doi.org/10.1109/iwci.2016.7860338

Hassan, A., Amin, M. M., Azad, A. K. A., & Mohammed, N. (2016b). *Sentiment analysis on bangla and romanized bangla text using deep recurrent models*. https://doi.org/10.1109/iwci.2016.7860338

Hassan, A., Amin, M. M., Azad, A. K. A., & Mohammed, N. (2016d). *Sentiment analysis on bangla and romanized bangla text using deep recurrent models*. https://doi.org/10.1109/iwci.2016.7860338

Alam, M. H., Rahoman, M., & Azad, A. K. (2017). *Sentiment analysis for Bangla sentences using convolutional neural network*. https://doi.org/10.1109/iccitechn.2017.8281840

Sumit, S. H., Hossan, M. Z., Muntasir, T. A., & Sourov, T. (2018). *Exploring Word Embedding for Bangla Sentiment Analysis*. https://doi.org/10.1109/icbslp.2018.8554443

Tuhin, R. A., Paul, B. K., Nawrine, F., Akter, M., & Das, A. (2019). An Automated System of Sentiment Analysis from Bangla Text using Supervised Learning Techniques. In *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*. https://doi.org/10.1109/ccoms.2019.8821658

Sharmin, S., & Chakma, D. (2021). Attention-based convolutional neural network for Bangla sentiment analysis. *AI & Society*, *36*(1), 381–396. https://doi.org/10.1007/s00146-020-01011-0

Chinese Text Sentiment Analysis Based on BI-GRU and Self-attention. (2020, June 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9084784

Data Set For Sentiment Analysis On Bengali News Comments And Its Baseline Evaluation. (2019, September 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9084054

Malik, U. (2022). Implementing Word2Vec with Gensim Library in Python. *Stack Abuse*. https://stackabuse.com/implementing-word2vec-with-gensim-library-in-python/

K, D. (2023, February 7). Top 5 advantages and disadvantages of Decision Tree Algorithm. *Medium*. https://dhirajkumarblog.medium.com/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a

Uniqtech. (2021, December 7). Multilayer Perceptron (MLP) vs Convolutional Neural Network in Deep Learning. *Medium*. https://medium.com/data-science-bootcamp

A Appendices

max_depth	min_samples_leaf	val_acc
2	2	0.514174
2	8	0.514174
2	16	0.514174
4	8	0.508016
4	2	0.507653
4	16	0.5072

Fig 2: Grid Search Table for Decision Tree

activation	hidden_layer_sizes	learning_rate_init	val_acc
logistic	(5, 5)	0.03	0.515805
logistic	(5, 5, 5)	0.003	0.514899
logistic	(5, 5, 5)	0.03	0.513993
relu	(5, 5)	0.03	0.513993
relu	(5, 5, 5)	0.003	0.513993
relu	(5, 5, 5)	0.03	0.513993

Fig 3: Grid Search Table for MLP

annotation	clean_sentence	tokenized_text	tokenized_text_no_stop	tokenized_text_no_stop_stem	tokenized_text_stem
Negative	লিখার সময় পারলে সত্তা লিখার অভ্যাস শিখুম	[লিখার, সময়, পারলে, সত্তা, লিখার, অভ...	[লিখার, সময়, পারলে, সত্তা, লিখার, অভ...	[লিখ, সময়, পারলে, সত্তা, লিখ, অভ্যাস...	[লিখ, সময়, পারলে, সত্তা, লিখ, অভ্যাস...
Negative	সরকার ঘাদের এই ব্যাংকে নিয়ণ দিয়েছে তারা ব্যাংক...	[সরকার, ঘাদের, এই, ব্যাংকে, নিয়ণ, স...	[সরকার, ব্যাংকে, নিয়ণ, দিয়েছে, ব্যাংক...	[সরক, ব্যাং, নিয়ণ, দিয়, ব্যাংকিং, ব...	[সরক, ঘাদের, এই, ব্যাং, নিয়ণ, দিয়,...
Negative	ইসলামি ব্যাংক প্রারম্ভ থেকেই গ্রাহকদের পছুশের...	[ইসলামি, ব্যাংক, প্রারম্ভ, থেকেই, গ্র...	[ইসলামি, ব্যাংক, প্রারম্ভ, গ্রাহকদের,...	[ইসলামি, ব্যাংক, প্রারম্ভ, গ্রাহক, পছু...	[ইসলামি, ব্যাংক, প্রারম্ভ, থাক, গ্রাহ...
Negative	এরা যেখানেই যাবে সেখানেই চুরি যাবে	[এরা, যেখানেই, যাবে, সেখানেই, চুরি, ...	[যেখানেই, সেখানেই, চুরি]	[যেখানে, সেখানে, চুরি]	[এরা, যেখানে, যাবে, সেখানে, চুরি, যাবে]

Fig 6: Misclassified Dataset

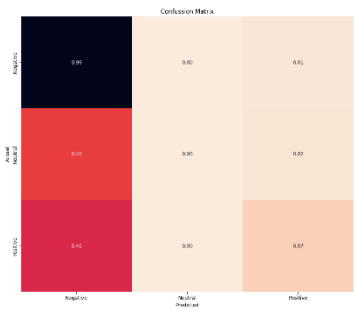

Fig 7: DT+W2V

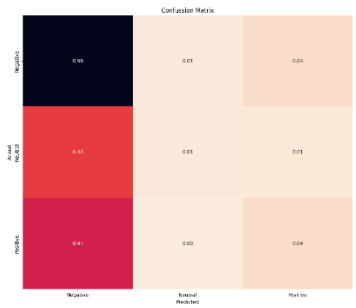

Fig 8: DT+TF-IDF

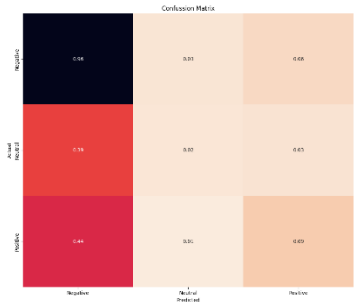

Fig 9: MLP+W2V

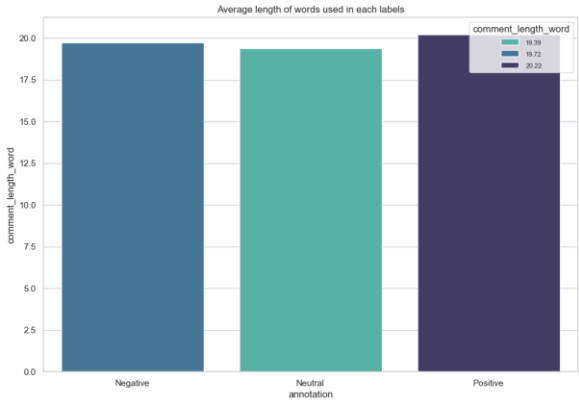

Fig 10: Distribution of comment length

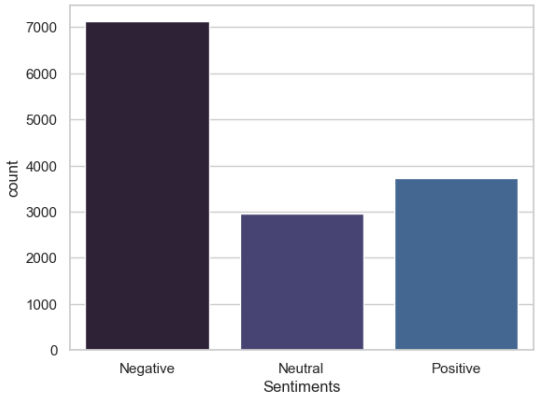

Fig 11: Dataset Statistics