Sentiment Analysis of Bangla Microblogs Using Adaptive Neuro Fuzzy System

Md. Asimuzzaman, Pinku Deb Nath, Farah Hossain, Asif Hossain, Rashedur M. Rahman

Department of Electrical and Computer Engineering, North South University

Plot-15, Block B, Bashundhara R/A, Dhaka-1229, Bangladesh

asimuzzaman.md@northsouth.edu, pinku.nath@northsouth.edu, hossain.farah@northsouth.edu, asifhossain38@yahoo.com, rashedur.rahman@northsouth.edu

Abstract—**Sentiment Analysis—also called Opinion Mining is the process that collects opinions through text forms to determine if the opinion being expressed is positive, negative, neutral etc. Our research has been done on Bangla Sentiment Analysis. There are few achievements in this field for Bangla. We put together our paper in the context of Fuzzy Sentiment Analysis. The semantic relations and various grammatical structures of these text forms increased the difficulty of determining the polarity of sentences. In this paper, we have used Adaptive Neuro-Fuzzy Inference System to predict the polarity of Bangla tweets and used fuzzy rules to represent semantic rules that are simple but greatly influence the actual polarity of the sentences.**

Keywords—Bangla; Sentiment Analysis; Opinion Mining; Semantics; Adaptive Neuro-Fuzzy Inference System; Fuzzy Rules.

I. Introduction

Sentiment Analysis is a promising field of study because it helps to determine the opinions of sentences and documents, which is one of the first steps of understanding texts and inferring implied meanings of human communications. It is important because it helps us to understand trends and opinions of the general population. It has commercial applications such as determining the acceptance and effectiveness of a product or service among the consumers etc. Sentiment analysis is relevant more than ever because colossal amounts of user-generated contents are produced every second by social networking sites in the form of Internet forums, blogs, discussion groups, tweets, comments etc. and it is impossible for human workforce to collect all these data, preprocess them into structured form and determine the opinions expressed in those data. We need to make sense of these data automatically and instantaneously understand the overall sentiments of the population on specific topics.

Sentiment analysis or opinion mining is a challenging task of Natural Language Processing (NLP). There have been several researches on Sentiment Analysis using various techniques such as using word frequencies and TF-IDF [1] method etc. without considering the semantic structure of the sentences. However, there are grammatical patterns that influence the semantics of a sentence and simple frequency-based methods fail to incorporate the effects of such rules in their analysis i.e., “This food was delicious, however it was expensive”. Semantic analysis attempts to address the problem by introducing various semantic rules to determine the polarity of sentences. Nonetheless, sentiment analysis is not an exact science and the polarity of sentence is subjective to the opinion holder. Hence, fuzzy sets with its non-deterministic nature, is a good candidate to be used for sentiment analysis.

There have been extensive researches conducted for the sentiment analysis of English texts which showed promising results. This was possible after the advent of World Wide Web which made a lot of textual data instantly available in electronic media. Before this period, it was hard to develop training data to test theories and models. However, sentiment analysis of Bangla texts is still a new area and there is scope of improvement. There are more than 160 million native Bangla speakers and huge amounts of Bangla texts are generated online. Most researches on Bangla texts are performed using news corpus and blogs which are extracted by scraping the news websites using bots. Another source of data is social media where the opinionated texts are shorter in length but they are informal and full of grammatical and spelling errors and in mixed languages and characters. In this research, we attempted to incorporate semantic analysis using supervised Adaptive Neuro-Fuzzy System to predict the polarity of Bangla texts. We have used Bangla tweets for our training and testing data. We have collected the Bangla tweets using Twitter APIs, preprocessed it to remove neutral characters and words, and modified it into the form accepted by the Adaptive Neuro-Fuzzy Inference System (ANFIS).

In the following section, we discussed related work in sentiment analysis done both in English and Bangla. After that we discussed some fundamental aspects of sentiment analysis and researches done in using fuzzy sets to address semantic rules for English texts. We also discussed researches done for Bangla texts. Following that we describe our methodology. The methodology section is subdivided into data-collection, preprocessing and ANFIS design. The following sections are Results and analysis, Conclusion, Future Work.

II. Previous Work

According to the famous review paper by Bing Liu [2], sentiment analysis can be divided the following areas:

sentiment and subjectivity classification which includes identifying opinionated texts and determine their polarities, feature-based sentiment analysis which identifies the feature of an object discussed and its polarity, sentiment analysis of comparative sentences, opinion search and retrieval and finally opinion spam and utility of opinions. Most of the work is done on sentiment and subjectivity classification, however incorporating semantic analysis is difficult and there is scope of research. Accurate sentiment analysis is only possible for simple sentences containing at most one opinion holder and the opinion is expressed about at most one object. In paper [3], the authors used fuzzy sets to model semantic polarity by incorporating different semantic rules as rules of the fuzzy algorithm. The authors used a hybrid approach, using standard unsupervised machine learning classifications, such as Naïve Bayes and Maximum Entropy, and fuzzy sets and used Senti-WordNet for sentiment lexicons. Wordnet is an English lexical database that stores related nouns, verbs, adjectives and adverbs as similar groups of synonyms representing a similar concept [4]. SentiWordNet is used to assign 3 scores on positivity, negativity and objectivity of words using WordNet [5]. However, such resources are not developed for Bangla texts. In paper [3], the authors assigned Parts of Speech (POS) tagging and polarity from SentiWordNet to each word in their opinion lexicon. The authors used the augmented lexicon to design semantic rules for their hybrid model.

One of the precursors of this paper is by Chowdhury, S. & Chowdhury, W. [6]. Their work paved the way in understanding how sentiment analysis should be carried out in Bangla using semi-supervised approach to train their microblog corpus. The dataset has been collected through the Twitter API, filtered as such that it only extracted Bangla tweets. Since the extracted data was noisy it was preprocessed through tokenization, normalization and POS tagging. Tokens such as username, hashtags and URL links are cleared out of the datasets. The constructed lexicon is an association of positive / negative sentiments with words. We further take this approach and include sentiments like neutrality. The semi-supervised approach is used here to train the classifier into labeling the tweets as positive or negative. Contrary to the amazing work done by this paper it has some limitations since it does not consider a neutral sentiment and hence assumed all tweets are subjective; which is something we improved on based on their work. The paper [6] is based on sentiment analysis on Bangla texts using SVM and Maximum Entropy. The authors collected from Bangla tweets using Twitter API v1.1, preprocessed and tokenized the data and then built the Bangla sentiment lexicon by first manually labeling the words and then checked the accuracy by translating them back to English and compared the results of the translated words using SentiWordNet.

To develop the fuzzy part of our paper, we got insights Dalal & Zaveri [7]. Linguistic hedges like modifiers (e.g., “not”) concentrators (e.g., “very,” “extremely”), and dilators (e.g., “quite,” “almost,” and “nearly”) can improve the efficiency of sentiment classification. The authors incorporated fine-grained classification into multiple output classes like “very positive,” “positive,” “neutral,” “negative,” and “very negative.” The feature-set generation phase following the preprocessing phase extracts POS tags from sentences along with linguistic hedges where each descriptor is given sentiment values of positive, negative or neutral from SentiWordNet. Based on the hedge preceding the descriptor, it is used in different equations as per [7]. Following that the average fuzzy score is transformed to normalized fuzzy biased value by mapping [-1,1] to [0,1]. In this paper [7], the authors manually developed a sentiment treebank, converted words to representational vectors and trained a supervised recursive neural network to represent sentences into a tree-type data structure and determine polarity. This accuracy of this model is high; however the performance is dependent on whether the sentiment treebank includes abstract structure of the query sentences. This model is included in the Stanford CoreNLP package.

Natural language sentences contain syntactic rules and complex structure that define the syntactic dependence of words in the sentence in relation to other words in the sentence, depending on the part of speech the word belongs to and the position of the word relative to the sentence and other words belonging to the sentence. The parser thus enlightens the structural and thematic relations of a sentence.

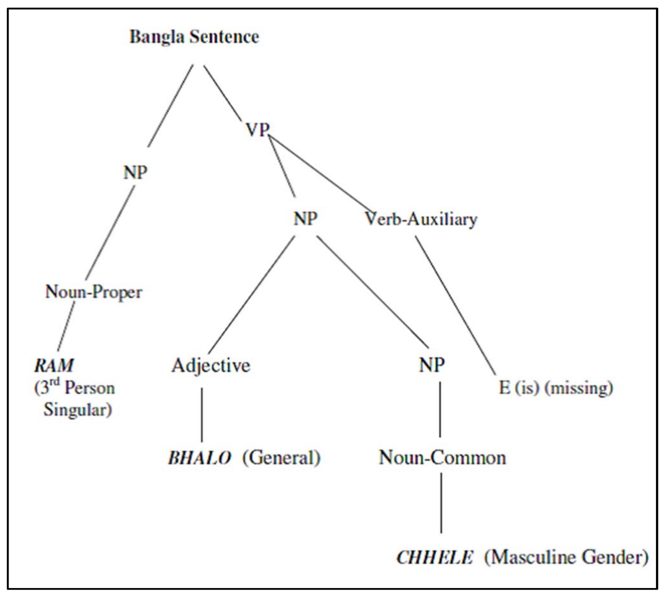

Figure 1. LFG generated grammar tree

This paper [8] tackles Bangla language parsing using LFG (Lexical Functional Grammar) formalism. Extensive computational, linguistic, and psycholinguistic research has led to the evolution of Lexical Functional Grammar that provides a fundamental tool set of constructs for describing the common properties of all natural languages and the certain properties of individual languages. The work done in the paper reflects the scarcity of resources, which would also prove to be a major concern pointed out in later sections. But the LFG parser shows considerable efficacy in parsing simple Bangla sentences. Though Bengali text in the internet is far from simple, parsers are useful for presenting thematic relations as discussed in later sections. A parser, given the structural rules and lexical resources of an individual language, produces a grammar/dependence tree when worked on a sentence. The structure, shape and size of the parse tree are important points to consider as shown in Fig 1. Techniques for parsing differ which results in different parse trees for each technique on the same sentence.

As pointed out by papers [9], [10] and [11] the effectiveness of such a tree in the classification process is reduced by the nature of the parse trees generated by such approaches. The parse trees lack complexity and information is rather lost in the process than what was initially targeted for. Thus strategy of delivering parse trees as input to the classification process has to be weighed out with the current performance and standards of Bangla parsers. Fig.1 shows a LFG generated grammar tree to help us visualize the discussion.

III. METHODOLOGY

In our research, we have used Bangla tweets, preprocessed to remove unwanted characters, performed POS tagging and then organized the data into separate training and testing datasets accepted by our model. For designing our model, we have ANFIS from Matlab's Fuzzy Toolbox.

A. Datasets

We have collected our data from two sources. First, we collaborated with the authors of [6] to use their datasets of Bangla texts. Second, we extracted our Bangla tweets using Twitter APIs. The nature to tweets is that the individual texts of users are limited to 140 characters. Hence, the probability of complex compound sentences is low. The raw tweets are noisy because Tweeter's domain of users is very diverse and often they use wrong spelling and grammar in their tweets.

B. Preprocessing

The steps of preprocessing are given below:

Step 1: The Bangla tweets that we extracted were noisy, had many grammatical and spelling mistakes and additional expressions such as emoticons that sometimes express opinions. In contrast, texts extracted from news and blogs are simpler since the writers maintain strict grammar and write correct spellings. The first step of preprocessing was to clean the tweets for which we wrote a C++ script to take the whole raw Bangla tweets files and removed special characters, leading spaces etc. The raw Bangla tweets also contain special tokens such as username (for example, @lolita, @অনি etc), URL link, hashtag (for example #আশা (English: 'hope'), #2017 etc. and retweets (indicated by RT) and other tags that might not convey any strong or relevant sentiments. We removed such tokens from the data.

Step 2: We removed special characters such as "#" and "@" and extra leading spaces after such characters with only a single space.

Step 3: We removed the substring "RT" and replaced it with a space.

Step 4: We removed all characters in the following set {' \' ',' \' ', ' ! ', ' $ ',' % ',' ^ ',' * ',' (',') ',' - ',' } ',' { ',' | ',' ; ',' / ',' ~ ',' , ',' = '} and replaced such characters with a space. We only considered single special characters and the set of single characters covers most of the characters that are usually used. Emojis or emoticons play a vital role in the final sentiment of a sentence or tweet. However, in this step, all characters that are part of emoticons are also removed. We overcome this problem by using textblob seperately to include polarties of emojis which is described below.

Step 5: We used TextBlob for part-of-speech tagging and determination of polarity of emoticons. TextBlob is a Python library which has many built-in Natural Language Processing (NLP) tools. It has a community of users and its github repository is well maintained. TextBlob uses data integrated from WordNet for its various tools. It also has stemming facilities for English texts. However, for stemming Bengali sentences we have used this [12] stemmer. Stemming is the process of reducing a word to its simplest form. For example, given the Bangla word 'করছে' will be 'করা' after stemming where both words meaning is good. Here, 'করছে' is the continuous present tense form of the verb 'করা'. It is observed that sentences in the tweets, can be very complex which makes it difficult for determining the sentiment of texts. However, the sentiments derived from emojis in the texts are highly correlated with actual sentiments of the texts. Hence, considering the polarity of emojis greatly affects the performance of the model.

The primary objective of the pre-processing phase is to identify a set of data features that can be used to construct an accurate model for Bangla sentiment analysis. A small caveat is that features mentioned here refer to the features to be used by machine learning algorithms and not the features used for feature-based sentiment analysis. In order to reduce the time taken for training the model and also to increase the performance of the model by reducing possible noise, only the highly sentimental tweets were considered to be included in the dataset and the neutral tweets were ignored. This selection is done automatically using the sentiments of the individual uni-grams present in the tweets. For example, the sentence "সকালে বৃষ্টি হয়েছিলো", which means "It rained today in the morning", will be ignored since the sentence does convey any sentiment.

There can also be duplicate tweets arising from situations where users copied other users' tweets and posted them from their accounts. Duplications can be avoided when verifying the dataset of tweets manually before assigning polarity labels to the tweets themselves. Duplicates do not pose a big problem but they may offset the output of the model.

The processed texts are bundled together and stored in a seperate file named 'processed_text' with each tweet starting from a newline.

C. Tokenization

Before we can find the polarity of the whole tweets, we tokenized the tweets into individual words or unigrams. For tokenization, we used python's NLTK library. We called the function nltk.tokenize (processed_text) which returned a set of objects represent the unigrams or words of each sentimental tweet. Tokenization is important because the sentiment, meaning and order of the words within the sentences are used as features to represent each sentences and to train the predictive model. For example, "আজকে তারা ভালোই খেলেছে", which means "Today, they played well", will be tokenized as 'আজকে'(*Today*), 'তারা'(*they*), 'ভালোই'(*well*) and 'খেলেছে'(*playing*).

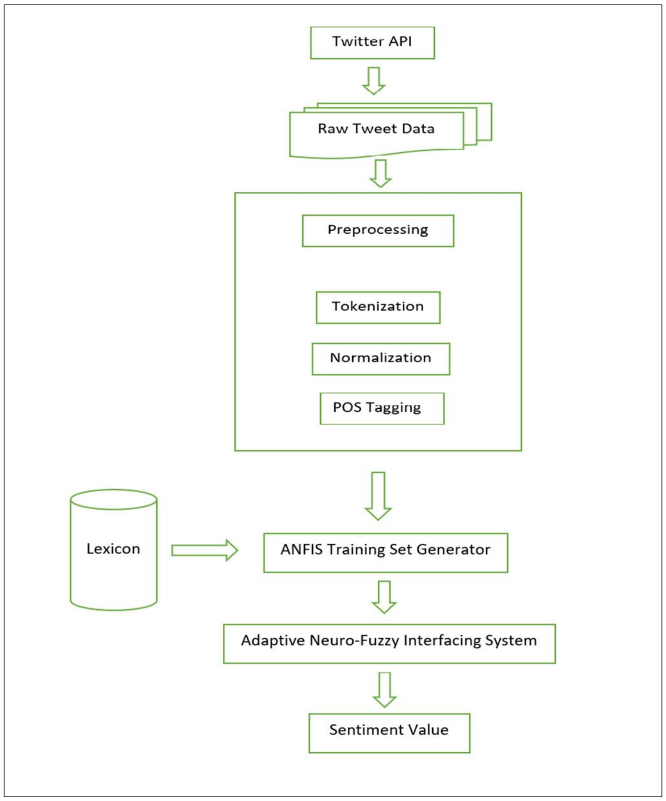

Figure 2. System Architecture

D. Determination of Polarity

In our model, we used Adaptive Neuro-Fuzzy Inference System to predict the polarity of a text. We included one fuzzy output variable to represent the polarity. The output variable has three fuzzy values represented by triangular membership functions. The values are negative, neutral and positive with their maximum peak at -1, 0 and 1 respectively. We manually determined the polarities of the training set and we applied heuristic where we made average each of the polarities of the sentimental nouns, verbs, adjectives, adverbs and emojis and assigned the 5 averages to 5 input variables which represented the nouns, verbs, adjectives, adverbs and emojis. Then we used the training set to train the Sugeno ANFIS model.

E. Training Set Construction

In case of Bangla tweets or blog posts, large labeled data set is nearly unavailable. So that, at first we have worked on smaller dataset to train our fuzzy interfacing system. We have manually labeled our training dataset to implement supervised technique for training the neuro fuzzy system.

In our pre-processed data we have tagged the data according to the words' related POS (Parts of Speech). Firstly, we have tokenized the text using NLTK Python library. Consider this text-

"আজ সবাই অনেক ভাল অভিনয় করছে, আশা করি আগামী দিনও ভাল কাজ হবে।" *English translation: Today everyone is acting nicely, hopefully, they will do well in upcoming days.*

After tokenizing the data becomes-

'আজ', 'সবাই', 'অনেক', 'ভাল', 'অভিনয়', 'করছে', ',' 'আশা', 'করি', 'আগামী', 'দিনও', 'ভাল', 'কাজ', 'হবে', '।'

Each and every token of this stream was then translated using Samsad online Bangla to English dictionary. A PHP script was written to fetch corresponding English translation from the online dictionary's HTTP get method and HTML rendering. So, the stream becomes-

'Today', 'everybody', 'many', 'good', 'acting', 'doing', ',', 'hope', 'do', 'tomorrow', 'day', 'good', 'work', 'is'

The main goal of word by word translation is to get the polarity value of each word or token from SentiWordNet. It requires POS tagged data thus we have used NLTK POS tagger library in Python to tag words according to their POS like the example given below-

"আজ_NN", "সবাই_DT", "অনেক_JJ", "ভাল_JJ", "অভিনয়_VBG", "করছে_VBG", ",_SYM", "আশা_NN", "করি_VBP", "আগামী_NN", "দিনও_NN", "ভাল_JJ", "কাজ_NN", "হবে_VBZ", "।"

We have considered only Noun, Adjective, Adverb and Verbs from the token stream as previous studies suggest that POSs' other than these 4 do not have any effect on the overall sentiment score of any test case. In [13], researchers have analyzed the polarity of blog posts using verb and adjectives but also showed that noun, adverb can also play role in determining polarity of sentences. So that, every English word of those four type POS's are then input into the SentiWordNet by a Java program to get their associated polarity value ranging from -1.00 to 1.00.

F. Modeling ANFIS

To model our fuzzy classifier for classifying positive and negative tweets we have defined five input variables to train the neural network. For determining the value of input variables we have considered the statistical average polarity value of each of the four part-of-speeches and also for the emojis. A single tweet constitutes a single training unit.

First input variable is determined by calculating the average polarity of all Nouns of specific tweet.

$$V_{noun} = \frac{\sum P(x_{noun})}{N_{noun}}$$

where, N_{nouns} represents the occurrence of noun in a certain training unit, x_{noun} is the polarity value for each token from the noun set, V_{noun} is the value of first input variable for a certain training unit. Following the same method, values of rest of the input variables for adjective, adverb, verb are calculated.

Second input variable is determined by calculating the average polarity of all Adjectives of respective training unit.

$$V_{adjective} = \frac{\sum P(x_{adjective})}{N_{adjective}}$$

Third input variable is determined by calculating the average polarity of all Adverbs of respective training unit.

$$V_{adverb} = \frac{\sum P(x_{adverb})}{N_{adverb}}$$

Then, the fourth input variable is determined by calculating the average polarity value of respective training unit.

$$V_{verb} = \frac{\sum P(x_{verb})}{N_{verb}}$$

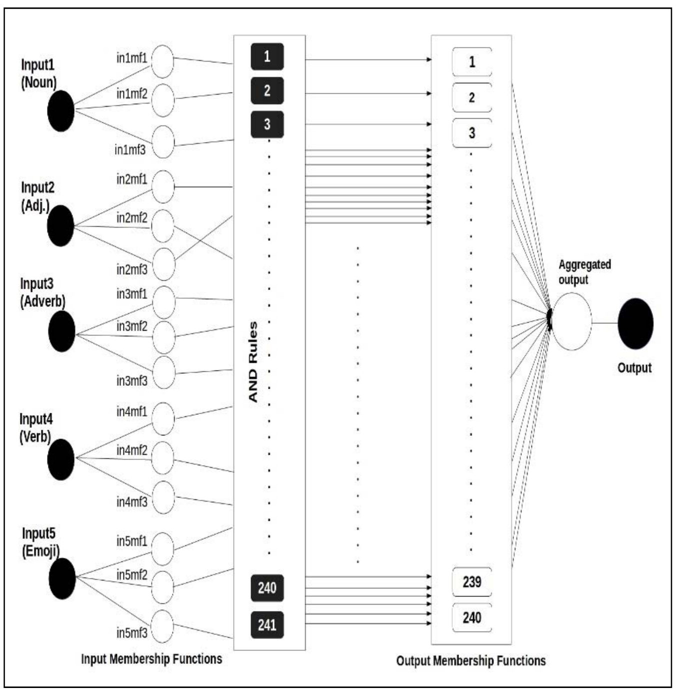

Figure 3. ANFIS structure

In any tweet or blog post, emoticons or emojis play vital role in determining the polarity of that tweet or blog post. So that, in our data pre-preprocessing section we have fetched associated polarity of all the emojis of all training set data and determined the average polarity for of each training unit as the fifth input variable of ANFIS

To generate the Fuzzy Interfacing System (FIS) we have used MATLAB to train the FIS and let it adapt with the changing input pattern and generate membership functions using grid partitioning system. Our classifier membership function is Generalized linear Bell shaped membership function. We have used hybrid optimization method for training the neural network. After generating the FIS (Fuzzy Interfacing System) we have trained the neural network.

Sentences of natural language can have negation which inverts the overall sentiment of corresponding sentences. For example, in English language, "*She didn't perform well in the exam*" is the negation of "*She performed well in the exam*". We have extracted the negation words or word fragments adjacent to the verbs and adjectives from the input sentences. List of the Bengali negation words and word fragments are-

Table 1. Bengali negation words

নয়
নাই
না
নাহ
নেয়
নি
নায়

After encountering negation word, the sentiment classifier negates the calculated polarity. For instance, "সেদিন যে বইটা পড়লাম, সেটা ভালো ছিলো।" *English translation: The book I read that day was good*. For this sentence the classifier shows positive sentiment. Negating the sentence with "না" gives "সেদিন যে বইটা পড়লাম, সেটা ভালো ছিলো না।" *English translation: The book I read that day wasn't good*. By considering the negation word the classifier gives negative sentiment.

IV. RESULT

The adaptive neural network formulated five membership functions for five input variables

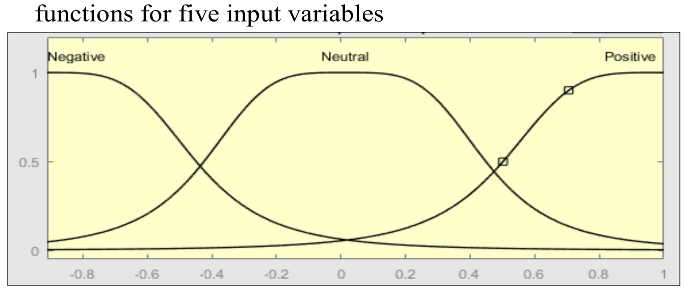

Figure 4. Input MF for noun words

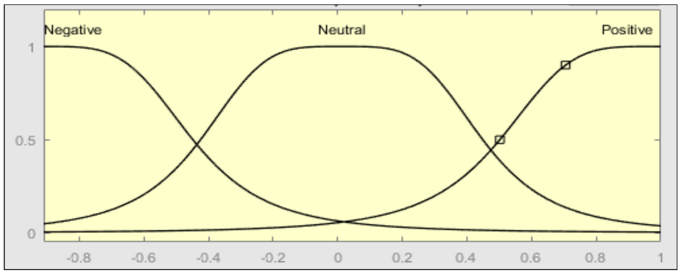

Figure 5. Input MF for adjective words

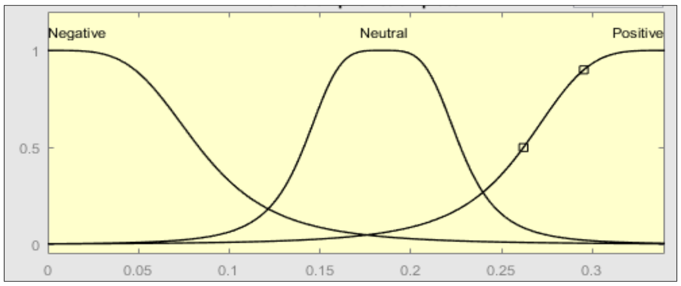

Figure 6. Input MF for adverb words

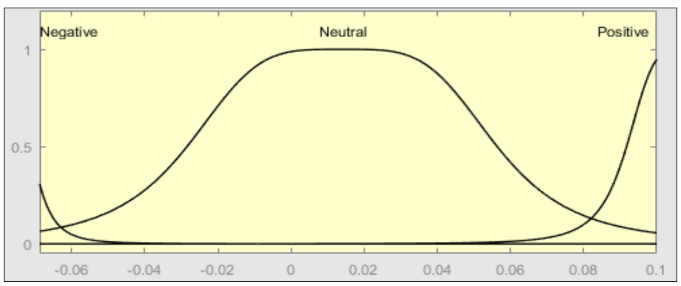

Figure 7. Input MF for verb words

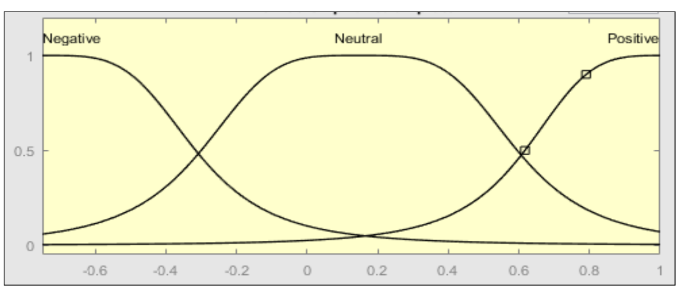

Figure 8. Input MF for emoji

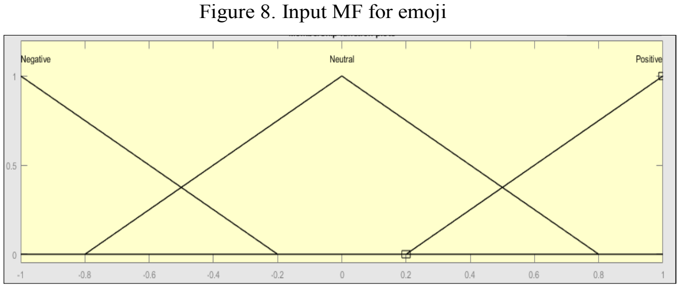

Figure 9. Membership function for mapping output sentiment

From the plots of five membership functions, which adapted according to our training data, we can determine that the effect of adjectives for determining positive or negative polarity is very high when verbs have less effect on determining the positive or negative sentiment.

V. TESTING AND EVALUATION

The trained classifier/predictor is tested against the test set. MSE (mean square error) is chosen as the appropriate testing metric for quantifying the prediction accuracy of the trained classifier. The classifier gives an accuracy of (MSE) 0.0529(approximation) over a test set of approximately 50 predictions. The training set lacked in variety of sentiment bearing tweets. The training set also had fairly small number of training tweets that consisted of a fairly small set of polar lexicons that could influence the polarity of a tweet. Hence, we infer that the classifier will perform better and maintain a low MSE score at the same time if we train it with a large dataset of evenly distributed polarity labels, since such a dataset will have more features to consider for adjusting the model's weights and parameters to predict the polarities.

It must also be made sure that the test data does not contain spam tweets. As mentioned in previous sections, the training data was initially manually sanitized off spam tweets. Thus large errors may occur if the test set contains spam which the classifier was not trained on in the first place.

Absence of an extensive Bangla spell checker also weighs heavy on the performance of the predictor. Thus misspelled Bangla words meant less data for the classifier to learn. Preprocessing hence falls short as stemming could not figure out the correct spelling the lexicon.

Another major source of errors is due to informal nature and highly complex sentence structure that the Bengali language encourages. This complexity of grammar also deepens the need for a very large dataset of Bangla tweets. The presence of English lexicon in the Bangla sentences has further increased the complexity and the informal nature. Thus the presence of considerable noise in the training dataset has to be considered with great care.

Our ANFIS was trained using tweets having discrete polarity value (-1 for negative polarity and +1 for positive polarity) which conveys strong positive or negative value and the ANFIS outputs a real number value. Some examples from our testing are given below-

"আজকে জীবনে ককটেল এর আওয়াজ এত কাছ থেকে শুনলাম। :3 ভয় পাইছি :(" *"English translation: Heard bomb/cocktail bursting from a short distance today, really afraid"* These sentences contain negative sentiment. When tested in the ANFIS it output -1.000 which also means strong negative sentiment.

"বহু বছর পর আবার হলের ল্যান এ. :) :D" *"English translation: After so many years using LAN of hall :) :D"* This sentence contains positive sentiment (+1.00). When tested we got +0.4543. This type of fractional continuous values increase the error value due to the difference between +1 and +0.4543 for instance.

We can avoid it by mapping the ANFIS output through another fuzzy interfacing system which gives negative, positive or neutral as output. Thus, we can use the membership function shown in figure 8 to map the ANFIS output to evaluated sentiment. Figure 10 depicts the error reduction with the increase of number of epochs.

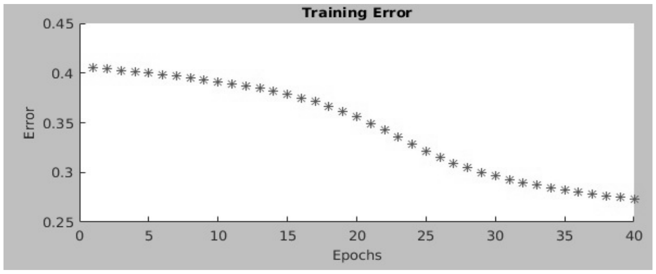

Figure 10. Training Error per epoch

VI. CONCLUSION

The prediction of sentiment analysis of English texts has been well explored in the past. There have been many well-established models which provide good accuracy in predicting the sentiments of a piece of text. However, the accuracy depended heavily on the simplicity and forthrightness of the sentences. Sentiments depend on the semantics for the complex sentences and the incorporation of semantic analysis has been a great hurdle for researchers. Nonetheless, recently there has been promising researches where the semantics are represented in a graphical manner and used to predict the sentiments. In another research [14], the semantic structures are included in a predictive recurrent neural network model in the form of sequence patterns learned by the hidden layers of the neural network. All these researches are done using English texts as datasets. The polarities of sentences are also subjective since a phrase can be either positive or negative depending on the circumstances, the texts preceding the phrases which determine the overall meaning of the phrase and also the actual intention of the person who used the phrases. In this research, we aimed to create a predictive model for Bangla texts using Adaptive Neuro-Fuzzy Inference System that incorporates imperceptible patterns in the presence of different words. Our model shows a promising accuray in predicting the polarities of the testing datasets. We used Neuro-Fuzzy model to incorporate the subjective nature of the polarities, i.e., a phrase can be somewhat positive or negative but cannot be assigned an absolute polarity. Since the MSE score after 40 epochs of training ANFIS is low, the outcome of our research is encouraging.

VII. FUTURE WORK

Our dataset of twitter tweets may have spam tweets – tweets that are duplicates to a large extent and/or tweets that are entirely composed of objective constructs such as links to external resources or a picture. Spam detection can be formulated as a classification problem on its own with two classes, spam and non-spam. There can be different categories of spams and each category needs its specific pattern of classification. The objective of spam classification is fairly simple since any classification models can be used with training data of accurately labeled data. Adding spam detection and classification can improve the quality of training of the classifier.

The accuracy of our model can be further improved by increasing the number of input nodes to represent highly influential words and words that signify complex grammatical structures, such as “কিন্তু (But)”, “যদিও (However)”, “তবু (Nonetheless, Although)”, “উপরন্তু (Additionally)” etc. We can also experiment with both the number of hidden layers and the number of nodes in each of the hidden layers to see whether the model gives the correct polarities for complex and compound sentences. Even with extensive efforts, it will still be difficult to predict the polarities of sarcastic or ironic phrases. There has been a number of noteworthy and independent researches going on where Bangla texts are used as dataset. We play o ncollaborate with the authors of such researches and build a common structured Bangla dataset with accurate meanings and polarities of Bangla words. We can also crowd-source the process of determining the polarities of Bangla words through online portals and the polarity of a Bangla word will be the average polarity of all the polarity values received for that word. Such efforts will reduce the hassles for future researchers and also expedite the hypothesis testing process and increase the effectiveness of future researches.

REFERENCES

[1] D. Okanohara and J. Tsujii, “Assigning Polarity Scores to Reviews Using Machine Learning Techniques,” *Journal of Natural Language Processing*, vol. 14, no. 3, pp. 273–295, 2007.

[2] B. Liu, “Sentiment Analysis and Subjectivity”, in *Handbook of Natural Language Processing,* 2nd ed., N. Indurkhya, F. J. Damerau, Ed. Florida: Chapman and Hall/CRC, 2010, pp. 627-666.

[3] O. Appel, F. Chiclana, J. Carter and H. Fujita, "A hybrid approach to the sentiment analysis problem at the sentence level", *Knowledge-Based Systems*, vol. 108, pp. 110-124, 2016.

[4] “SentiWordNet,” *SentiWordNet*. [Online]. Available: http://sentiwordnet.isti.cnr.it/. [Accessed: 07-Jan-2017].

[5] Wordent, “What is WordNet?,” *Princeton University*, 17-Mar-2015. [Online]. Available: http://wordnet.princeton.edu/. [Accessed: 07-Jan-2017].

[6] S. Chowdhury and W. Chowdhury, “Performing sentiment analysis in Bangla microblog posts,” *2014 International Conference on Informatics, Electronics & Vision (ICIEV)*, 2014.

[7] M. Dalal and M. Zaveri, "Opinion Mining from Online User Reviews Using Fuzzy Linguistic Hedges", *Applied Computational Intelligence and Soft Computing*, vol. 2014, pp. 1-9, 2014.

[8] M. Haque and M. Khan, "Parsing Bangla Using LFG: An Introduction", *BRAC University Journal*, vol. 2, no. 1, pp. 105 - 110, 2005.

[9] K. A. Hasan, A. Mahmud, A. Mondal, and A. Saha, “Recognizing Bangla Grammar Using Predictive Parser,” *International Journal of Computer Science and Information Technology*, vol. 3, no. 6, pp. 61–73, 2011.

[10] G. K. Saha, “Parsing Bengali text,” *Ubiquity*, vol. 2006, no. April, Jan. 2006.

[11] "Deeply Moving: Deep Learning for Sentiment Analysis", *Deeply Moving: Deep Learning for Sentiment Analysis*, 2017. [Online]. Available: http://nlp.stanford.edu/sentiment/index.html. [Accessed: 07- Jan-2017].

[12] "rafi-kamal/Bangla-Stemmer", *GitHub*, 2014. [Online]. Available: https://github.com/rafi-kamal/Bangla-Stemmer. [Accessed: 07- Jan-2017].

[13] P. Chesley, B. Xu and R. Srihari, "Using Verbs and Adjectives to Automatically Classify Blog Sentiment", in *Proceedings of AAAI-CAAW-06, the Spring Symposia on Computational Approaches to Analyzing Weblogs*, 2006, pp. 27-29.

[14] Z. Yangsen, J. Yuru, T. Yixuan, "Study of Sentiment Classification for Chinese Microblog Based on Recurrent Neural Network", in *Chinese Journal of Electronics,* Vol.25, No.4, July 2016