Detection of inappropriate anonymous comments using NLP and Sentiment Analysis

N. Sai Nikhita [1], V. Hyndavi [1] , Trupthi M.[1]

[1] Chaitanya Bharathi Institute of Technology, Hyderabad, India.
sainikhitanayani@gmail.com, venkatreddygarihyndavi.3@gmail.com,
mtrupthi_it@cbit.ac.in

Abstract: The world became interactive and socially active now-a-days because of the increase in different types of content sharing applications. These content sharing applications are social media platforms which provide various features so that users can effectively interact and share their thoughts and ideology. One such platform is a discussion forum which promises the anonymous posting of user's views and complaints. The spammers target the forums as the craze of the forums increase. Though these platforms act as medium of knowledge sharing, all of the users don't use these platforms for a positive cause. They are also being used to abuse or bully targeted people taking advantage of their anonymous feature. Spamming and cyber bullying has grown rapidly to a limit that social media is being termed harmful. By reading spam and vulgar comments, readers will be divertedl. Main aim is to detect these bad comments which are vulgar, inappropriate or not related to the specific context. The research is not based on the static contents but it live streams the comments and the entire research is being done. The research is based on NLP, Sentiment calculation and topic detection.

Keywords: PPM Algorithm, TEM Algorithm, SAM Algorithm, Latent Dirichtlet Allocation (LDA), Natural Language Processing (NLP), Machine Learning, Topic extraction

1 Introduction

The social media platforms allow us to interact and share our ideas. These comments are allowed to be posted anonymously to get more genuine views. Though we have the access of reading the comments and coming to a decision but they may be spam and they cause bad impact on the reader's brain. YouTube has a feature of deleting the comments based on the number of dislikes. By this action we can understand the real motive is that to not entertain any spam comments. In our research in this paper our approach is to deal not only with the spam comments but also to look after the bad, vulgar and irrelevant comments which manipulate the readers mind and are out of topic which are of no use. We built a mechanism to identify the spam comments and apply the natural processing techniques and machine learning algorithms.

1.1. Existing System

In the existing system we can see people commenting randomly and we face a number of cases where the comments are Topic irrelevant, Either offensive or vulgar, Spam comments etc. We can generally observe that the forum pages wont make an effort to remove the spam until someone reports them to be vulgar. But by the time people will be seeing the comments and it is waste of time to remove them after getting reports.

1.2. Related Work

The initial research in this area was with the recognition of the email spam by Sahami etc. al. [1].He used probabilistic learning methods to produce filters which are useful in this task. Domain specific features are considered for detecting spam. Carreras etc. al. [2] proved that AdaBoost is more effective than Naïve Bayes and decision trees. As increased growth in World Wide Web, the need for spam detection also has grown simultaneously. So they started to detect spam comments by analytics, the decision trees algorithm was used in [3] by Davison to identify link based web-spam. Drost etc. al. [4] used SVM to classify web spam with content based features. Their method is based on the link spam detection and the solution is based on training data by revealing the effectiveness of classes of intrinsic and relational attributes.

In [5] Mishne etc. al. used machine learning techniques the detection of spam comments. Comments in the posts are checked with the comments of linked pages. In [6] Bhattari etc. al. used the data corpus from the same set to detect spam comments. In [7] the authors made study on the comments and commenting behavior of the videos that have more than 6M views and comparison in sentiment analysis is made on finding the influence of the comments by other people's comments. In [8] Ruihai Dong etc. al. worked on topic detection using nlp.

2 Proposed Method

2.1. System Architecture

In our research we have dealt with detecting the spam comments based on NLP and Machine learning. The first is detecting the profanity of the comment then it goes for preprocessing where we tokenize, lemmatize, stemmetize and it continues with other preprocessing finally we reach to use TEM ,SAM algorithms. The results are found out using sentiment analysis and then displaying the sentiment of the comments using word cloud and barplot.

The dataset we used for this research is live streaming posts of a forum which are obtained by web scraping the forum content. The dataset is dynamic and gets updated as the posts are added.

Our approach consists of four modules. The first module is about finding the vulgarity. The preprocessing is done in the second module. The third module comes with the algorithms where we find out the topic similarity and topic detection by forming dictionary and corpus formation. Finally comes our sentiment analysis. The results are visualized using wordcloud. Wordcloud allows us to visualize the most frequently talked issues whose representation or the weight of the word is more.

Fig. 1. Proposed Architecture

2.2. System Components

A. Profanity Check Module
B. Preprocessing Module
C. Topic Extraction Module
D. Sentiment Analysis Module

The first step we deal here is about profanity check of the comments which can be done in different ways. Profanity is the module we used where we can check our comments based on the list of bad words. If the word consisting profanity is found out then we have to stop further preprocessing and as this comment is not for any use in the forum we try not displaying them.

This research then deals with the preprocessing. The comments are split and tokens are formed. Then we see for lemmatization and stemmatization. Then there comes our POS tagging which is a crucial step in the preprocessing as we need the parts of speech of words for topic extraction. After POS tagging we arrive at a stage of topic detection and topic similarity detection.

So the third module deals with topic extraction using TEM algorithm which depends on the values which are normalized. We understand by topic the main theme which is discussed in the comment and it is given by a set of unigrams and bigrams with a predominant number of occurrences in the comment. Here we applied Latent dirchilet algorithm to find out the topic similarity in the TEM module. Based on the results of the LDA we can find out the topic which the comments are based on.

Next modules deal with sentiment analysis using SAM algorithm we get the results displaying the positivity and negativity of comments.

Profanity Check Module: As growth in number of web users the presence of inappropriate content from users becomes more problematic. Social networking sites, forums, and any online community must take care of the content which is not accepted by the society norms. If this task is not being done the its like posting such content is acceptable

The research on previous works shows that the current systems are not up to the mark. The general practice is they use a static list each time they check for profanity. This dont work if the vulgarity is in form of misspelled words, different languages and other reasons. These drawbacks make the current systems to detect profanity ,obsolete some even depend on outsiders so that they are assigned with the detection of spam comments for the posts. This is suitable and doable upto a particular stage but when the task becomes huge this is not applicable. So all the comments are profanity checked based on vulgarity, abusive words and irrelevant topic discussion.

List Based Approach: This is the most standard approach where That is, in order to determine if a comment contains profanity in a particular forum, these systems simply examine each word in the document.If a match occurs the it is profane. Basically we introduced a system where as soon as the comment is introduced in the forum the comment is being checked for the profanity and the profanity module runs in the background and if it is found to be profane we stop further pre-processing. The profanity module is from Google where they update the list on periodic basis and we make sure that the list is updated in our profanity module which takes care of all the spellings, partially censored and other issues taken care of.

Pre-Processing Module: Preprocessing is an important stage in natural language processing because the words, tokens and other sentences identified in this stage are used for further preprocessing to find ngrams and applying algorithms.

Stop Word Removal: Many words in a sentence are used as joining words but they themselves do not make any sense unless combined and framed grammatically to form a sentence. So we can say that their presence do not contribute to the content or context of the document. Removal of these stop words is necessary because of their high frequency causes obstacles in understanding the actual content of document.

Tokenization: Tokenization is the process where the sentence or a word is broken into tokens. The main aim behind tokenization is to explore the meaning of the tokens formed and how they are preprocessed further to make meaningful outcomes after performing nlp. Though it is readable it still is left out with many punctuation words and expressions which are of no use for us and should be removed. Tokenizing is based on the delimiter which further depends on the language as different languages have different delimiters. Space is a delimiter in English.

Stemmatization: The word is the token is reduced to its root word. The root form is not necessarily a word by itself, it can be formed even by concatenating the right suffix. Stemming is a type of normalization. The sequence is returned with its root word

Lemmatization: The lemma of the word is found. So we can see that the suffixes are removed in lemmatization. The word which is returned is called lemma. These two terms are not same, Stemming is just finding the root word but most times it's not preferable but lemming is a technique where morphological analysis of the words.

Fig. 2. Flow of Pre-processing module

```
Algorithm: PPM

Input : Comments entered in the forum
Output : Filtered Tokens

a) Get profanity list from profanity module
b) If comment contains profanity then
        (i)Delete the comment and don’t display in the forum
c) Else
        (i)Get the stopwords of English language present in nltk module
        (ii)Remove stopwords present in the comments
        (iii)Tokenize the comment using  tokenizer (eg PunktTokenizer) present in nltk
        (iv)Stemmatize the tokens
        (v)Lemmatize the tokens using WordnetLemmatizer
```

Fig. 3. Pseudo code Pre-processing module

Topic Extraction Module: The tokenized comments now need to be preprocessed in such a way that we can extract the topic of discussion and categorize the comments based on the topic being extracted.

POS Tagging: In this processed tokens are assigned to their respective pos tags. The words are classified based on their part of speech, tense, number, case etc. Basic tagsets are like N for noun ,A for adjective etc. Similarly we will have a list of tagsets but each one of them is not useful for topic extraction because as we know that mostly to obtain the topic being discussed in a given sentence or paragraph or document will rely on the nouns being discussed. Not only the nouns but we are also deciding it based on the adjectives and verbs being discussed as they describe the nouns and the situation which is being talked in a sentence.

Unigrams and bigrams: Based on our pos tagged words , grouping of the words based on their part of speech can be done and then we can extract the topic being discussed. N-grams are sequence of items in a sequence can take values 1, 2 etc. but not a large value for N.Here generation of bigrams is taking place. Bi grams are formed by considering adjacent tokens and grouping them together. Before forming them to bi grams they are in uni grams stage and then we form the bi grams. Unigrams and bigrams are generated as they are essential to proceed towards LDA

Topic Extraction: As we have our bi grams there is a need to apply an algorithm to extract topic. LDA is a topic extraction model which is vastly used for this purpose. Here we used LDA(Latent Dirichlet Allocation) to extract the topic .The input to this algorithm needs to be in form of dictionary and corpus. This model is used to identify the topic of the document by classifying text which is present in a document.

First we try to analyze the frequency of terms by a document term matrix. After this has been done we generate a LDA model upon the document. When we apply it, each token is identified by a unique id, which is transformed to a bag of words called corpus then the LDA is applied. Using the frequency of the words topic is extracted. Depending on the extracted topic our forum main discussion will be obtained daily basis and weekly basis.

Fig. 4. Flow of Topic extraction module

```
Algorithm: TEM

Input: Filtered Tokens
Output: Topic of the comment

a) POS Tagging using nltk module
b) Find the candidate nouns (i.e. tokens which are singular nouns (NN), plural nouns
(NNS),    singular proper nouns (NNP), plural proper nouns (NNPS))
c) Generate Bi-grams using nltk module
d) Dictionary formation using gensim module
e) Corpus formation using gensim module
f) Apply LDA algorithm using gensim module
g) If topic of the comment is not relevant to the topic of the forum then
        (i)Delete the comment and don't display in the forum
h) Create a wordcloud of most discussed topics using wordcloud module
```

Fig. 5. Pseudocode Topic extraction module

Sentiment Analysis Module: Sentiment analysis shows the sentiment of the people based on the topic being discussed and how the people's opinions are. This is a classification where the inserted phrase is decided based on the negative, positive and neutral sentiment.

In our research we used sentiwordnet. SENTI WORDNET is a document containing all the synsets of WORDNET along with their "positivity", "negativity", and "neutrality". Each synset has three scores Positive score, Negative score, and Objective score. These scores may range from 0.0 and go upto 1.0, the sum of all the three scores being 1.0. Each score for a synset term has a non-zero value.

So on the result obtained which shows the sentiment of the phrase describes how the opinion of the people is and also the opinion on the topic being discussed which helps a lot in case of our forum where students will be discussing all their issues which paves a way for the management and the teachers to look after the issues which are needed to be taken care and how they need to be handled are also discussed as we provided their suggestions section also so they can reach the staff and be resolved. This system helps not only the faculty and institution but also the students who want their issues to be solved.

```
Algorithm: SAM

Input: Filtered tokens from preprocessing module, SentiWordnet module which
contains synset terms along with their positive and negative sentiment scores

a) For each token in the filtered token list:
        (i)If token= 'not':
                positivescore= 0
                negativescore= thresholdvalue
        (ii)Else:
                positivescore= positive score of synset term in sentiwordnet
                negativescore= negative score of synset term from sentiwordnet
b) Create a plot visualizing the positive and negative sentiments using matplotlib
module
```

Fig. 6. Pseudocode of Sentiment Analysis Module

6 Results

The results are shown in the form of plots like word clouds and barplots

Fig. 7. Barplot visualizing the sentiment

Fig. 8. Wordcloud depicting the most discussed topics

Fig. 7. depicts a barplot representing the positivity and negativity of opinions people have towards the topic. Fig. 8. is a wordcloud depicting the topics discussed on the forum.Frequently discussed topics are depicted in larger size which enables us to know about the most discussed topic. These results are visualized using the result of SAM algorithm for barplot and TEP algorithm for wordcloud on our dataset. Data set is obtained by scraping our website which consists of posts and comments made by users.

7 Conclusion

Our research has overcome the problem with spam comments and all the disadvantages which were in the existing system. In the proposed the system the spam comments will be detected based on finding out its features and also the problem where topic irrelevant comments which lead to misconception are also dealt with. Future enhancements can be made to this research as we are streaming the comments not just taking the static content which provides a great scope not only to remove the spam comments but to make this evaluation of topic to be applicable in other areas of interest.

8 References

1. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A Bayesian approach to filtering junk e-mail”. In AAAI-98 Workshop on Learning for Text Categorization, July 1998, Madison, Wisconsin, pp. 98-105
2. Carreras, X. and Marquez, L., “Boosting trees for anti-spam email filtering”. In Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing, 2001, pp. 58-64
3. Davison B. D., “Recognizing Nepotistic Links on the Web”. In AAAI 2000 Workshop on Artificial Intelligence for Web Search, 2000, pp.23- 28.
4. I. Drost and T. Scheffer., “Thwarting the nigritude ultramarine: Learning to identify link spam”. In ECML’05 Proceedings of the 16th European conference on Machine Learning, 2005, Berlin, Germany, pp.96-107.
5. Gilad Mishne, David Carmel, and Ronny Lempel, “Blocking blog spam with language model disagreement”. In Proceedings of the First International Workshop on Adversarial Information Web (AIRWeb), Chiba, Japan, May 2005, pp. 1-6
6. A. Bhattari and D. Dasgupta, “A Self-supervised Approach to Comment Spam Detection based on Content Analysis”. In International Journal of Information Security and Privacy (IJISP), Volume 5, Issue 1, 2011, pp. 14-32M.
7. Stefan Siersdorfer and Sergiu Chelaru, “How useful are your comments? analyzing and predicting YouTube comments and comment ratings”. In Proceedings of the 19th international conference on World wide web, 2010, pp. 891-900.
8. Ruihai Dong, Markus Schaal and Barry Smyth, “Topic extraction from online reviews for classification and recommendation”