Grenze International Journal of Engineering and Technology, Jan Issue

A Review on Detection of Offensive Language in Social Media

Mr. Raghuram A.S[1], Bhoomika B R[2], Gokul D[3], Malavika Kuppanda[4] and Mohammed khaleeq[5]
[1-5]ATME College Of Engineering, Mysore, India
Email: rbraghuram958@gmail.com, bhoomikgowda2910@gmail.com, logaiahmurthy@gmail.com, malavikakuppanda8@gmail.com, mdkhaleeqmdk@gmail.com

***Abstract*— Since the textual contents on online social media are highly unstructured, informal, and often misspelled, existing research on message-level offensive language detection cannot accurately detect offensive content. This feature of the social media to express something openly to the world have created the major problems for these online businesses and negatively impacted the well-being of the societal decorum. There are increasing cases of the abuse or offense on the social media like Hate speech, Cyber-bullying, Aggression or general Profanity. It is very much important to understand that this behavior can not only immensely affect the life of an individual or a group but could be suicidal in some cases adversely hampering the mental health of the victims .**

***Index Terms*— Hate speech, Offensive.**

I. INTRODUCTION

This increasing negative situation on the internet has created a huge demand for these social media platforms to undertake the task of detecting the objectionable content and taking the appropriate action which can prevent the situation becoming more worse. This task of detecting the offensive content can be performed my human moderators manually, but it is both practically infeasible as well as time consuming because of the amount of the data generated on these social media platforms, therefore there is a need to fill this gap.

To address concerns on children's access to offensive content over Internet, administrators of social media often manually review online contents to detect and delete offensive materials. However, the manual review tasks of identifying offensive contents are labour intensive, time consuming, and thus not sustainable and scalable in reality. Some automatic content filtering software packages, such as Appen and Internet Security Suite, have been developed to detect and filter online offensive contents. Most of them simply blocked webpages and paragraphs that contained dirty words. These word-based approaches not only affect the readability and usability of web sites, but also fail to identify subtle offensive messages. For example, under these conventional approaches, the sentence "you are such a crying baby" will not be identified as offensive content, because none of its words is included in general offensive lexicons. In addition, the false positive rate of these word-based detection approaches is often high, due to the word ambiguity problem, i.e., the same word can have very different meanings in different contexts. Moreover, existing methods treat each message as an independent instance without tracing the source of offensive contents.

Grenze ID:01.GIJET.9.1.625

II. OBJECTIVE

This increasing negative situation on the internet has created a huge demand for these social media platforms to undertake the task of detecting the objectionable content and taking the appropriate action which can prevent the situation becoming more worse. This task of detecting the offensive content can be performed my human moderators manually, but it is both practically infeasible as well as time consuming because of the amount of the data generated on these social media platforms, therefore there is a need to fill this gap.

III. RELATED WORKS

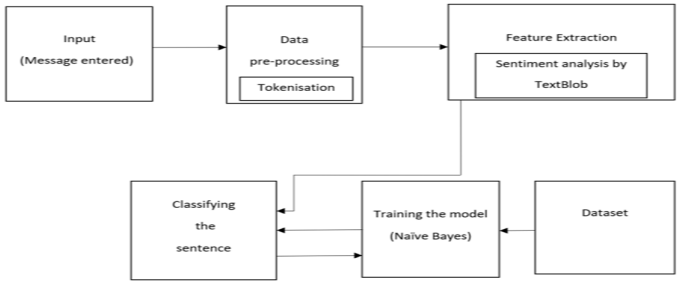
In this section, we review existing methods on Detection of offensive language in social media, and then focus on text mining based offensive detection research.

A. Cyberbullying Detection on Social Networks Using Machine Learning Approaches

Dataset

The training dataset used consisted of 1066 samples contained in a CSV File. Each sample is a sentence followed by the corresponding target label. The target label is 'pos' for a Non Cyberbullying Sentence and 'neg' for a Cyberbullying Sentence. The data is then fed to the model for training.

Building Classification Models:



System Architecture After the dataset is collected, a combined classifier is created. The values from Naïve Bayes Classifier model, Polarity of the phrase (Sentiment Analysis), and Vulgar confidence are then passed to a Combined classifier which combines results from other three classifiers and returns the appropriate Cyberbullying confidence level, which is then displayed by a bot.

This customer classifier model can be further trainer.

The Naive Bayes method is a supervised learning algorithm that solves classification problems and is based on the Bayes theorem. It is mostly used in text classification problems that necessitate a large training dataset. It's a probabilistic classifier, it makes predictions based on an object's probability. After the dataset is collected, a Naive Bayes classifier is created. The created model is then stored in a pickle file so that the file can be restored to update the model. We then use pickle.dump() to dump the data. It classifies text by reading from the pickle file and returns a float value between 0 and 1, where 0 if it is a positive message and 1 if negative. Naive Bayes theorem can be used to calculate the posterior probability P(c|x) using P(c), P(x), and P(x|c).

$$P(c|x)=\frac{P(x|c)P(c)}{P(x)}$$

The probability of the message 'd' being in class 'c' is computed as follows,

Where, P(tk |c) is the conditional probability of term tk occurring in a message of class c. B. Sentiment Analysis TextBlob, a python library that aims to provide access to common text-processing operations, is used for Sentiment Analysis. When a user sends a message, the message is checked for the phrase's polarity. The parameter is the phrase which has to be checked for profanity. The polarity will be any float bounded by -1 and 1 where 1 indicates the sentence is most negative and -1 indicates its most positive. C. Vulgar Confidence We import the mysql.connector because python needs a MySQL driver to access the MySQL database. The database contains a table named badwords containing a list of bad words. The list of badwords is fetched from the database and stored in a list. The function checks if a word is in the swear list and returns the vulgar confidence which is -1 if there are no bad words and 1 if there are bad words.

B. Offensiveness Content Filtering Methods in Social Media

Popular online social networking sites apply several mechanisms to screen offensive contents. For example, Youtube's safety mode, once activated, can hide all comments containing offensive languages from users. But pre-screened content will still appear—the pejoratives replaced by asterisks, if users simply click "Text Comments." On Facebook, users can add comma-separated keywords to the "Moderation Blacklist." When people include blacklisted keywords in a post and/or a comment on a page, the content will be automatically identified as spam and thus be screened. Twitter client, "Tweetie 1.3," was rejected by Apple Company for allowing foul languages to appear in users' tweets. Currently, Twitter does not prescreen users' posted contents, claiming that if users encounter offensive contents, they can simply block and unfollow those people who post offensive contents.

In general, the majority of popular social media use simple lexicon-based approach to filter offensive contents. Their lexicons are either predefined (such as Youtube) or composed by the users themselves (such as Facebook). Furthermore, most sites rely on users to report offensive contents to take actions. Because of their use of simple lexicon-based automatic filtering approach to block the offensive words and sentences, these systems have low accuracy and may generate many false positive alerts. In addition, when these systems depend on users and administrators to detect and report offensive contents, they often fail to take actions in a timely fashion. For adolescents who often lack cognitive awareness of risks, these approaches are hardly effective to prevent them from being exposed to offensive contents. Therefore, parents need more sophisticate software and techniques to efficiently detect offensive contents to protect their adolescents from potential exposure to vulgar, pornographic and hateful language.

C. NLP and Machine Learning Techniques for Detecting Insulting Comments on Social Networking Platforms

They propose a novel methodology employing Natural Language Processing and Machine Learning to analyze texts and predicting abusive behavior. The pipeline involves extraction of a suitable data set from various online sources, preprocessing, ground truth building, feature engineering and selection, classification.

Being a supervised learning problem, the goal is to classify a text from an online user which could be in the form of comments, and status/post updates into two categories – "Bully" & "Non-Bully". The main starting point is to collect relevant data from various online platform. This type of data mainly consist user comment, posts, images, videos and audios.

D. Sarcasm Detection in Politically Motivated Social Media Content

Feature Extraction

They extract the following five groups of features from the text of the tweets and their associated information.

Contextual Features: The detection of sarcasm must consider contextual and factual information. We capture this contextual information in the features extracted from the tweets, these features encode the semantic relationship among the words and sentences numerically in highdimensional vectors. Two types of contextual features are extracted; those that consider the importance of the words and also that consider the meaning of the words in context.

Features measuring the importance of each word are based in statistical analysis, and include n-grams and TermFrequency and Inverse Document Frequency (TF-IDF). In the n-grams method, a sample of text is represented by the most frequent instances of every unique n continuous words as a dimension. The most frequent word grams are selected from the entire corpus. The tweets were represented through the bigram (2-gram) vectors, and the weight for each bigram is its TF-IDF score which is given by:

$$T F - IDF = tf * \log(T df) \quad (1)$$

In Equation (1), T F is the number of times a particular term occurs in a tweet, T is the total number of tweets, and df is the number of tweets containing that particular term. The main advantage of a TF-IDF score over the simple frequency counts of the n-gram method is that it assigns a higher weight to the terms that occur more frequently through the entire data set. Thus, the TF-IDF score should assign a higher weight to those phrases that are the most important in determining whether a tweet is sarcastic or not. We calculated the TF-IDF vector representations of the top 2000 most relevant bigrams. We used the TF-IDF implementations from the NLTK library to extract these features.

Although the TF-IDF score provides a differentiated representation of the words based on their frequency of occurrence, it does not preserve any relationship between the words. Word embeddings are a powerful technique that represent semantically related words as closely related vectors. Words with similar meanings are mapped to lowdimensional, non-sparse vectors that exist near each other in a pre-defined vector space. A good word embedding can preserve the contextual information behind words in a tweet that a n-gram/TF-IDF scheme

cannot. We use Word2Vec, which is a popular technique to create distributed numerical representations of word features using a two-layer neural network with back propagation [25]. Word2vec trains words against other words that neighbor them in the input corpus. Word2Vec allows us to encode the context of a given word by including information about preceding and succeeding words in the vector that represents a given instance of a 1541 word. Therefore, the results obtained from using Word2vec may result in a much better classification.

E. Automating the Detection of Cyberstalking

Based on [12], authors implemented binary classifier; Naïve Bayes, Rule-based JRiP, Tree-based J48 and Support Vector Machine (SVM). Two experiments were set up where first experiment used binary classifier to train three labels dataset (intelligence, culture &race and sexuality). Second experiment was integrated three datasets into a single dataset and trained using multiclass. Result shows that binary classifier with three label were better in terms of accuracy instead of using multiclass classifiers with one dataset.
Accuracy of rule-based JRip was better than other binary classifiers.
In other studied by [13] and [14], they mentioned binary classifier (SVM) used as classification algorithm. By integrated BoW and polarity, result for F-scores was better than using single feature in SVM classification.
On the other hand, [15] implemented linear SVM, logistic regression, decision tree and AdaBoost as classifiers. However, only linear SVM gave a better result instead of decision tree and AdaBoost. Result of precision and recall for both linear SVM and logistic regression were quite similar for cyberaggresion detection.
Linear SVM was also used by [16] to learn featured. EBoW model showed better performance in terms of precision and recall compared to BoW, sBow, LSA and LDA when implemented linear SVM as classifier.
In summary, all of the studies used SVM as classification algorithm. Frequent use SVM by researchers shows that SVM is popular among others classifiers in supervised learning approach. SVM is suitable for high-skew text classification such as to detect cyberbullying using content-based features [18]. Any circumstances such as missing data, type of features and computer performance, SVM still outperform other classifiers [17]. Table 1 shows the summarization of data source, features used and classification in cyberbullying detection for each research works as discussed.

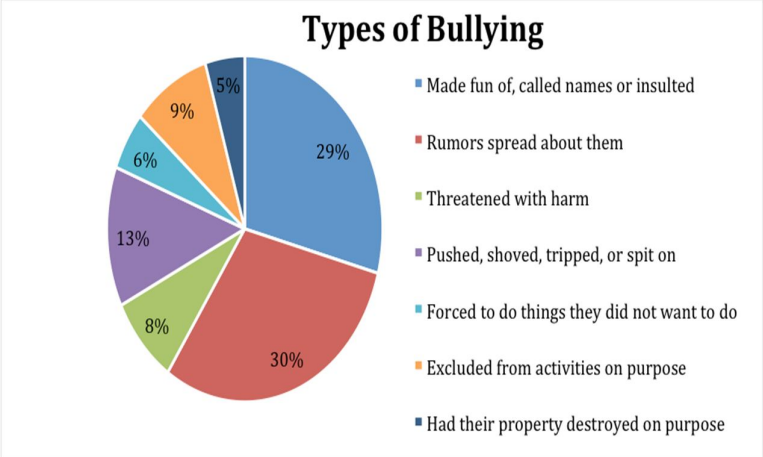
TABLE I. SUMMARIZATION OF STUDY IN CYBERBULLYING DETECTION

Study	Data Source	Feature	Classification
[12]	*YouTube* and *Formspring.me*	Content-based feature (Profane, Negativity, Subtlety)	SVM Naïve Bayes JRip J48
[13]	*Ask.fm*	Content-based feature (BoW) Sentiment-based feature (Polarity)	SVM
[14]	*Ask.fm*	Content-based feature (BoW) Sentiment-based feature (Polarity)	SVM
[15]	*Instagram*	Content-based feature (Profanity, Linguistic, Image, Cyberaggression) Network-based feature (Network graph, Comments)	Linear SVM Logistic regression Decision tree *AdaBoost*
[16]	*Twitter* (http://research. cs.wisc.edu/bullying/data.html)	Content-based feature (BoW, Bullying) Latent semantic feature	Linear SVM

F. Detection of Bad Language Using Social Media

Problem Statement:

Now a days with the increasing usage of social media platforms like Facebook and Twitter, we often see people to misuse this freedom of speech. Some of them try to use this platform to post the offensive or abusive things about a person or a group. This in turn can negatively impact the mental health of a community/ group or an individual. Considering the sensitivity of the topic, we aim to tackle this problem of detecting the offensive language in social media through cutting edge techniques in Machine Learning, Deep Learning and Natural Language Processing.



REFERENCES

[1] [1] IEEE Paper - P.-C. Tsou, J.-M. Chang, H.-C. Chao, and J.-L. Chen, "CBDS: A cooperative bait detection scheme to prevent malicious node for MANET based on hybrid defense architecture," in Proc. 2nd Intl. Conf. Wireless Common., VITAE, Chennai, India, Feb. 28– Mar. 03, 2011, pp. 1–5.

[2] [2] IEEE Paper - Cloud Storage and Retrieval - A user perspective, Abhishek V Student, dept. of MCA R.V. College of Engineering Bangalore, INDIA. Megha S N Student, dept. of MCA R.R. College of Engineering Bangalore, INDIA

[3] [3] S. Salawu, Y. He, and J. Lumsden, "Approaches to Automated Detection of Cyberbullying: A Survey," IEEE Trans. Affect. Comput., pp. 1–25, 2017

[4] [4] Cyberbullying Detection on Social Networks Using Machine Learning Approaches Author: IriniSpyridakis, Madison Holbrook, Brent Gruenke, SrinithiSellakumaranLatha Date of Conference: 16-18 Dec. 2020 Date Added to IEEE Xplore: 28 April 2021

[5] [5] NLP nd Machine Learning Techniques for Detecting Insultin Comments on Social Networking Platforms Author: Hitesh Kumar Sharma, K Kshitiz, Shailendra Date of conference: 22-23 June 2018 Date Added to IEEE Xplore: 27 August 2018.

[6] [6] Sarcasm Detection in Politically Motivated Social Media Content Author : Hieu Nguyen, Jihye Moon, Nijhum Paul, Swapna S. Gokhale Date of Conference: 30 Sept.-3 Oct. 2021 Date Added to IEEE Xplore: 22 December 2021