Detect Cyberbullying Based on Naive Bayes and Data Mining

Anket Sah
Department of Computer Science
San Jose State University
San Jose, CA
anket.sah@sjsu.edu

Angela Wang
Department of Computer Science
San Jose State University
San Jose, CA
anqi.wang02@sjsu.edu

Dr. Mike Wu
Department of Computer Science
San Jose State University
San Jose, CA
ching-seh.wu@sjsu.edu

***Abstract*—The social networking websites like Twitter, Facebook and Google+ have brought people closer together to share their happy moments, pictures and videos. But it is also used as a medium for people to vent out their frustration, bully or harass others. Cyberbullying is a kind of bullying that uses electronic methods like mobile phones, internet and the social media to harass vulnerable people as well as send harmful messages. It is an abusive social behavior that involves using social media negatively. It is therefore important to detect this cyberbullying to help reduce the harm to people and create a healthy Internet environment. Some of the typical solutions involve classification and text mining to detect cyberbullying. We propose an approach that makes use of machine learning and data analysis to solve the problem of cyberbullying. We have developed a customized web scraper that extracts cyberbullying tweets related to corona virus. This harvested raw data is then pre-processed to remove the outliers/noise in the data. Our machine learning model is then run on this cleaned data to evaluate experiment results. These results are then visualized using the mat-plot library to derive useful insights.**

Keywords – data pre-processing, machine learning, Naive Bayes, web crawler

I. Introduction

Social media is popular among teenagers to communicate with each other and it also influences their connections and relationships with each other. With the increasing use of Facebook, Instagram and Twitter, more users comment on the posts, and these comments are not filtered which causes the problem of Cyberbullying. Cyberbullying is the use of an aggressive, insulting language to bully others on Social media. It is important to detect and prevent this cyberbullying, help people notice these insulting words on the internet and protect them.

Our project detects cyberbullying with the help of machine learning and data analytics. To achieve these goals, we have created our own dataset by creating a customized web crawler. First, we have collected the data by crawling the Twitter website, as well as used openly available datasets from Kaggle and a few newspaper articles such as New york times and the Washington post. Then we pre-process this crawled data which involves data cleaning, editing, reduction, and data wrangling. After pre-processing, we make use of Naive Bayes machine learning method to analyze and model Cyberbullying data. Finally, we visualize this data and derive results.

To help in the ongoing research efforts to protect people from being bullied online or taken advantage of during Covid-19, we have concentrated on extracting cyberbullying tweets related to Corona virus. During these trying times, with the spread of Corona virus, people are getting frustrated and restless due to shelter in place orders. Social media has become a platform for these people to vent out their frustration and attack others. There are also incidents of racism on the rise which has spiked the cases of cyberbullying by 28%. As per statistics, 37% of the total suicides occur because of cyberbullying. We want to stop this and help save lives, which is why we have made use of cyberbullying data available in the times of corona virus. Our proposal is novel as we have made use of data available during Corona virus to detect and prevent direct cyberbullying. We have created a new dataset with the help of our web crawler. We have then applied Naive Bayes machine learning method to train our model and test it to detect cyberbullying, which has not been performed before. For our future work, we plan to improve indirect cyberbullying detection by identifying sentences that involve indirect harassment or indirect cyberbullying sentences and performing text classification on this data.

II. RELATED LITERATURE

The research regarding Cyberbullying on Social networks has been carried out for the past couple of years. For example, Ting, Lou et al. talk about how they use social network mining method to detect cyberbullying and prevent the occurrence of this serious socially abusive behavior [1]. They use three main methods that include keyword matching technique, opinion mining and social network analysis. The traditional approach here is the Keywords matching method, which means matching the cyberbullying categories with experiment text. It is based on a dictionary of different types of cyberbullying. There are many machine learning models and artificial intelligence approaches such as genetic algorithms or neural networks. The problem with using keyword matching is that it is hard to analyze the sentiment of comments and continuous attacks of cyberbullying, especially for those related to social network relationships. The paper comes up with another approach by using the advantages of social network mining methods.

The social network mining method contains several steps, that include feature extraction, feature selection, classification and cyberbullying detection [1]. After extracting the keywords, it will compare them with social network analysis measurement and sentiment characteristics. The most important one is classification as mentioned in the paper [1]. The paper doesn't point out the certain classification model applied to this implementation. However, there are existing machine learning models such as Support Vector Machine that can be used to classify the keywords. The results show that this implementation can detect more than 70 percent of cyberbullying posts by recalling the model multiple times. The contribution of this paper is that it provides a new trend to solve the cyberbullying detection problem and the performance is good. It uses three key features to measure the detection in the implementation, which are keyword, social network analysis measurement, and sentiment. By using these measurements, it can analyze the complex and potential cyberbullying and hidden meanings in the sentences. However, the volume of the dataset used in this paper is not large, it may influence the final result of the implementation. Also, the classification models are not specified in paper and different keyword matching models may have poor performance due to unfiltered noise.

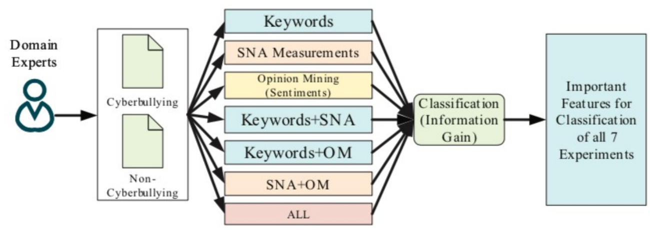


Fig. 1 Design For Cyberbullying Detection [1]

Win et al. make use of supervised learning models like support vector machine to detect Cyberbullying in Indonesian language [2]. The most important part is that the implementation considers the gap in the analysis of cyberbullying based on the language context, which is not straightforward cyberbullying and hard to detect. The language barrier problem is a new field in cyberbullying detection and there is little research related to it. They use around 250 different posts and the result of evaluation shows that SVM has a good performance in cyberbullying detection [2]. Based on the result, it is feasible to use SVM to detect indirect or potential cyberbullying, such as other languages and pictures. Compared with the other research [2], there is no existing dataset for Myanmar language, so they collected datasets from posts and preprocessed the data. It is useful to learn from their method, how to process the data. They crawl data from Facebook and clean this data by removing unnecessary function words. They define the rules by themself based on their demand. In our project, we can also define the rules for data cleaning. The word segmentation part is similar to research by tokenizing the data. The most important part of this implementation is using support vector machine classification and F-score prediction to evaluate the result [2].

The contribution of this paper is that it proves SVM classification is useful for cyberbullying that is not-straightforward, such as using some other language and sending offensive text photos. Their data collection process helps understand how they define the rules for data cleaning. It solves the language barriers in cyberbullying detection. Based on their results, SVM can be a candidate for our implementation. However, it needs a future design to improve quantitative performance and extend the dataset.

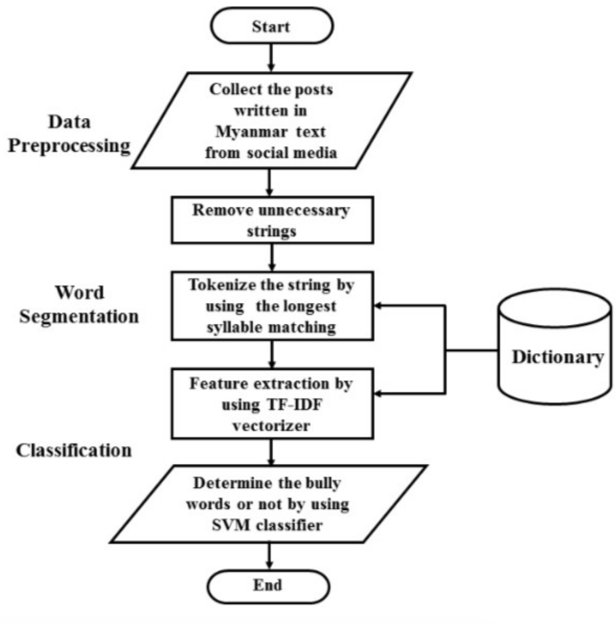


Fig. 2 Process of Cyberbullying detection in Indonesian Language [2]

In [3], the authors K. D. Gorro et al. describe their approach of detection and classification of cyberbullying, based on the use of SVM and Selenium. The authors emphasize that the main motive for almost 21% of the teens to go on social networking platforms like facebook, twitter is to check whether anyone is saying something wrong to them or bullying them [3]. Bullying in any forms such as Facebook posts, tweets, messages or emails affect mental health and should be stopped completely. For this purpose they make use of a customized web crawler that reads through facebook urls and generates the data set that will be used for classification and analysis.

Due to certain website parsing issues, human clipping and extracting data from various websites can be repetitive, difficult, and time consuming. Web Scraping or automatic processing of information from the Web rather than physically extracting it is not a novel concept. It is also referred to as data mining or information retrieval sometimes. Site crawling is achieved by developing software programs that query a web server, seek details in either of HTML, XML, and JSON formats, and then interpret it to obtain the necessary information [4]. In various fields, site crawling was also used to create reasonable solutions particularly in customer operation.

Vikas et al [5] emphasize the use of a supervised machine learning model to detect cyber aggressive comments that are made by peers on a social media network. The authors suggest a two step model to detect these types of comments. The raw comments are taken as input and unwanted strings are identified and removed. The data is then normalized and features of data are extracted. From a proposed set of features, a feature selection algorithm chooses the limited set of features, as the machine learning algorithm cannot handle all the features which are of the order of hundred thousands. Chi square method then selects k best features and data is parsed to derive insights. The algorithm for this method gets an ACC score of 77.65 with 0.58 recall and 0.7 precision which can be improved by increasing the feature set size to attain better accuracy.

On the other hand, authors in [4] pre-process the data obtained after crawling specific Facebook urls. The next step is feature identification. Using TF-IDF test as well as countvectorizer, each of the Facebook posts were identified and labelled to find the most widely used attributes in the Fb post relevant to online harassment. The SVM model was then trained on this labelled data where 70% was used for training, whereas the rest 30% for evaluation. Every word identified widely was used to harvest a new collection of Facebook posts. Specialized query on Facebook has been used to accomplish this phase through the use of Selenium. A total of 2263 articles were collected on Facebook and listed as Cyberbullying articles. This model has many drawbacks since the database fails to cover all phrases that can be used in harassment, even with the high accuracy provided in this research. Facebook graphs have many limitations which prevent harvesting of accurate information.

Owing to a versatile usage in tweets, the above approaches lack generalization, and only use principle-based understanding that is hard to model. All these works do not differentiate between all the negative remarks posed on twitter feed / threads and discussions between people and non-participants such as celebrities, public figures etc. To overcome these problems, we propose an effective detection model that identifies cyberbullying remarks aimed at members engaged in online communications.

III. Proposed method

This section discusses the conceptual framework of the research project in order to detect and prevent cyberbullying related to corona virus. Figure 3 illustrates the workflow diagram for the research.

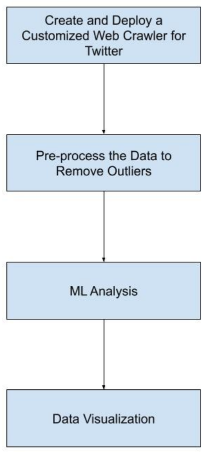


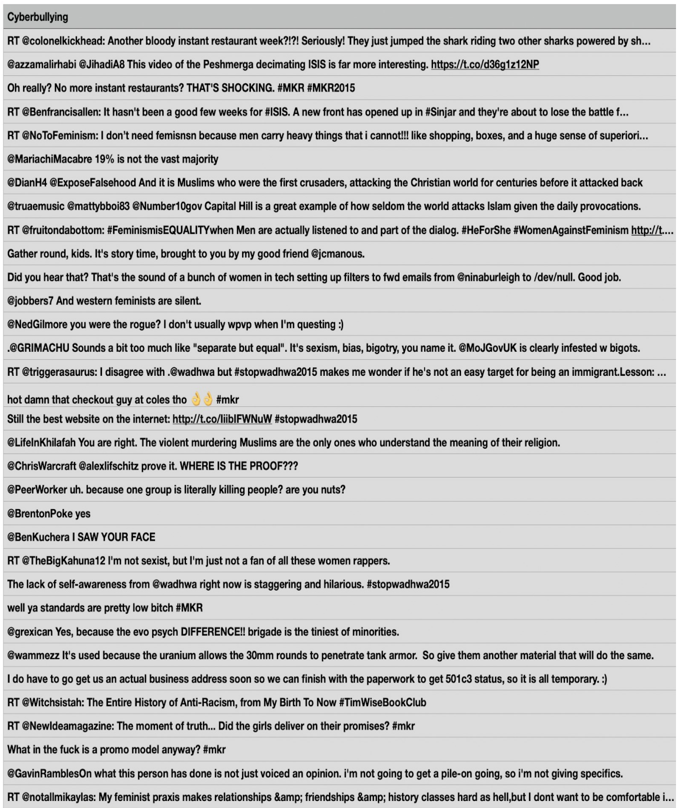
Fig. 3 Proposed Methodology Workflow

A. Web Crawler

We have made use of Twitter Api to create a customized web crawler for Twitter. It makes use of Twitter.com as the default url for crawling data. Twitter does not allow for openly scrapping of any of its data as it is associated with users and could infringe upon user privacy. For the same, we have to sign up for a developer account on Twitter giving a comprehensive explanation as to why we wish to scrape this data, what data we will be scrapping and how that data will be used. Twitter takes into account our description and if the developer account is approved, assigns a consumer key, API secret key, access token and access token secret that can be used to scrape the data we wish for. For crawling the data, we pack the consumer and API secret keys into an authentication handler. The access token and access token secret are packed along with the authentication handler into the tweepy.API to scrape the data. The hashtag that we want to search for is passed as argument to the tweepy.Cursor() that performs a search for that specific hashtag along with a count that specifies the number of tweets to retrieve. CSV writer is used to write these retrieved tweets to an output file.

B. Data Resource

This cyberbullying corpus consists of tweets related to corona virus. Due to the recent news, our bullying content is also focused on Coronavirus issue. We extract data under the Coronavirus tweets and label it into cyberbullying data and non-cyberbullying data. With the help of twitter API, we get this necessary cyberbullying data related to coronavirus from tweets. This data contains user id, text message, and bullying type. We found that under the main field, there are quite a few racist and sexual bullying that is occuring on the internet. For example, "*443574538576478208 RT @DarylMansbridge Just don't find Woman comedians funny in the slightest #coronavirus pandemic*", is data from a bullying dataset and it is of type – sexual bullying. For effective detection, we will preprocess data and detect cyberbullying based on these features as shown in the figure 4.



Cyberbullying
RT @colonelkickhead: Another bloody instant restaurant week?!?! Seriously! They just jumped the shark riding two other sharks powered by sh...
@azzamalirhabi @JihadiA8 This video of the Peshmerga decimating ISIS is far more interesting. https://t.co/d36g1z12NP
Oh really? No more instant restaurants? THAT'S SHOCKING. #MKR #MKR2015
RT @Benfrancisallen: It hasn't been a good few weeks for #ISIS. A new front has opened up in #Sinjar and they're about to lose the battle f...
RT @NoToFeminism: I don't need femisnsn because men carry heavy things that i cannot!!! like shopping, boxes, and a huge sense of superiori...
@MariachiMacabre 19% is not the vast majority
@DianH4 @ExposeFalsehood And it is Muslims who were the first crusaders, attacking the Christian world for centuries before it attacked back
@truaemusic @mattybboi83 @Number10gov Capital Hill is a great example of how seldom the world attacks Islam given the daily provocations.
RT @fruitondabottom: #FeminismisEQUALITYwhen Men are actually listened to and part of the dialog. #HeForShe #WomenAgainstFeminism http://t....
Gather round, kids. It's story time, brought to you by my good friend @jcmanous.
Did you hear that? That's the sound of a bunch of women in tech setting up filters to fwd emails from @ninaburleigh to /dev/null. Good job.
@jobbers7 And western feminists are silent.
@NedGilmore you were the rogue? I don't usually wpvp when I'm questing :)
.@GRIMACHU Sounds a bit too much like "separate but equal". It's sexism, bias, bigotry, you name it. @MoJGovUK is clearly infested w bigots.
RT @triggerasaurus: I disagree with .@wadhwa but #stopwadhwa2015 makes me wonder if he's not an easy target for being an immigrant.Lesson: ...
hot damn that checkout guy at coles tho 👌👌 #mkr
Still the best website on the internet: http://t.co/liiblFWNuW #stopwadhwa2015
@LifeInKhilafah You are right. The violent murdering Muslims are the only ones who understand the meaning of their religion.
@ChrisWarcraft @alexlifschitz prove it. WHERE IS THE PROOF???
@PeerWorker uh. because one group is literally killing people? are you nuts?
@BrentonPoke yes
@BenKuchera I SAW YOUR FACE
RT @TheBigKahuna12 I'm not sexist, but I'm just not a fan of all these women rappers.
The lack of self-awareness from @wadhwa right now is staggering and hilarious. #stopwadhwa2015
well ya standards are pretty low bitch #MKR
@grexican Yes, because the evo psych DIFFERENCE!! brigade is the tiniest of minorities.
@wammezz It's used because the uranium allows the 30mm rounds to penetrate tank armor. So give them another material that will do the same.
I do have to go get us an actual business address soon so we can finish with the paperwork to get 501c3 status, so it is all temporary. :)
RT @Witchsistah: The Entire History of Anti-Racism, from My Birth To Now #TimWiseBookClub
RT @NewIdeamagazine: The moment of truth... Did the girls deliver on their promises? #mkr
What in the fuck is a promo model anyway? #mkr
@GavinRamblesOn what this person has done is not just voiced an opinion. i'm not going to get a pile-on going, so i'm not giving specifics.
RT @notallmikaylas: My feminist praxis makes relationships & friendships & history classes hard as hell,but I dont want to be comfortable i...

Fig. 4 Cyberbullying Dataset

C. Data pre-processing/Data Cleaning

The data written to the output file by the scrapper cannot be used directly for analysis as it is inconsistent data that has unwanted symbols like "@, {}, $", incomplete, inaccurate and missing words. We need to preprocess this data.

Normally, the data pre-processing step contains data cleaning, which filters the useless information, data integration which clusters the similar parts of the data, and data extraction which extracts the useful data and only keeps the meaningful information. In our project, we are mainly concerned about the content. After applying a web crawler on the Twitter urls, we get data that conatins user id, resource url, and random number as well as some invalid characters. We will pre-process and clean these outliers by our customized java program.

First, we define a stopword file that contains some commonly used words but have no meaning in analysis. For example, the stopword file contains words like "me", "you", "ourselves", "@", "," and other punctuations. This stopword file is predefined by us that includes all these words. The stopword files help us filter useless information like symbols, some names and commonly used words. After hitting a stop word, the Java program will read the stopword file and use it as the filter standard to filter out useless words. It will then store the data into a (SET) data structure and use it as reference. Secondly, the java program will read the training data as well as the test data. While loading the data, it will first check whether it is a stop word, if so, it will ignore it and if not it will read it into the data structure for usage. It will also remove duplicate words and unwanted symbols. For example, if the word appears multiple times in the same content, it will be counted only once. After the data cleaning process, the training data will be stored in a HashMap data structure without any duplicate words and unwanted symbols. The following table shows how the raw data has been pre-processed and converted into our cleaned dataset.

Original data	Cleaned data
572328540890247168 Yes, you put in the wrong way. Cue dumb blonde jokes. #MKR ass	Yes, you put in the wrong way. Cue dumb blonde jokes. ass

Table 1 Comparison between raw data and cleaned data

D. Model Training using Naive Bayes

The cyberbullying data analytics process uses Naive Bayes model to analyze the probability of cyberbullying. Naive Bayes classifier model is widely used in classification areas such as text analysis, artificial intelligence, and web intelligence. It is based on Bayes' theorem and uses independent features of each classification. In order to apply naive Bayes, we need to predefine classification categories and get typical features of each category. The algorithm is based on the rule that all features of each category are independent. In our application, there are two main categories, which are cyberbullying and non-cyberbullying. We extract six hundred tweets (due to crawling limitations set by Twitter) and use eighty percent of this data as training and twenty percent as testing data. All the training data and testing data are cleaned data.

First, the project will load all training data into HashMap data structure. Each individual word is a feature in each category. Each message is one training data. It calculates the number of features in each category and records the amount. We use 0 to represent bullying type, and use 1 to represent normal type. Each Key is an independent feature, and value is a map structure. For example, <“summer”: <0:1, 1:2> > means that the word “summer” appears one time in cyberbullying sentence and appears two times in normal sentence.

Second, it calculates the probability of feature occurrence in each category by the formula,

$$P(C_k \mid F) = P(F \mid C_k) * P(C_k) / P(F)$$

For example, P(bullying | “summer”) = P(“summer” | bullying) * P(bullying) / P(“summer”). All the records come from the data stored in the HashMap. The dataset in our experiment is finite so we can use a HashMap to train our Naive Bayes model. If the size of the dataset increases to petabyte or larger, it needs to store the record into a database and retrieve data from the database. Some common Nosql databases include Big Table, Hbase and they are effective tools for big data storage. After loading the training data, it will get the probability of all features. That’s the preparation process for naive Bayes models. Here are some examples shown in the Table 2, it records the probability of each feature.

	Cyberbullying	**Non-cyberbullying**
reaction	0.33	0.67
protocol	0.8	0.2
SARS-Coronavirus	0.4	0.6
threat	0.67	0.33
development	0.5	0.5
prediction	0.14	0.86
mouse	0.87	0.13
MHC	0.33	0.67
support	0.45	0.55
emerging	0.22	0.77
Immunomics	0.33	0.66
old	0.6	0.4
Problems	0.67	0.33
single	0.625	0.375
health	0.187	0.8125

Table 2 Probability of Features

From the results shown in Table 2, we can see that some aggressive words appear more frequently in cyberbullying dataset, some normal or object words appear equally in cyberbullying category and non-cyberbullying category. We also found that as the coronavirus topic is trending, more and more people talk about the health, virus and treatment.

The third step is to predict whether the test message is cyberbullying or not. It follows the same step to extract useful features from test data. For each message, it will calculate the probability of the message as cyberbullying category and also as the non-cyberbullying category. The calculation of probability is as follows:

$$P(C_k \mid F1F2...Fn) = P(C_k) * P(F1F2..Fn \mid C_k) \; / \; P(F1F2..Fn)$$

To simplify, we just calculate the following,

$$P(C_k) * P(F1 \mid C_k) * P(F2 \mid C_k) \ldots P(Fn \mid C_k)$$

The premise is that each evidence cannot be zero, which means that the probability of features cannot be zero. In order to make sure the probability is valid, it applies Laplace smoothing on the dataset to calculate weighted probability. The constant weight is one so that the occurrence of each feature must be greater than one. The other problem in calculation is that there will be overflow risk when there are too many features and we will calculate decimal values. We can transfer this multiplication to addition by using log function.

Log [P(Ck) * P(F1 | Ck) * P(F2 | Ck) … P(Fn | Ck)] = log [P(Ck)] + log[P(F1| Ck)] + log [P(F2 | Ck)] + … + log [P(Fn|Ck)]

We then calculate the probability of the message under each category and will mark the message as cyberbullying if the probability of cyberbullying is larger, otherwise, mark it as a non-cyberbullying message.

E. Architecture Diagram

The class diagram for our implementation of Naive Bayes is as shown in figure 5. It consists of 2 main classes that include the NaiveBayes class and the test class. The Naive bayes class consists of multiple sub functions. Some of these functions include:

- load data – that processes each cleaned message(it records features and classifier by updating feature count and the classifier count)
- get count of feature – it gets the number of occurrences of features in each category and returns the feature count

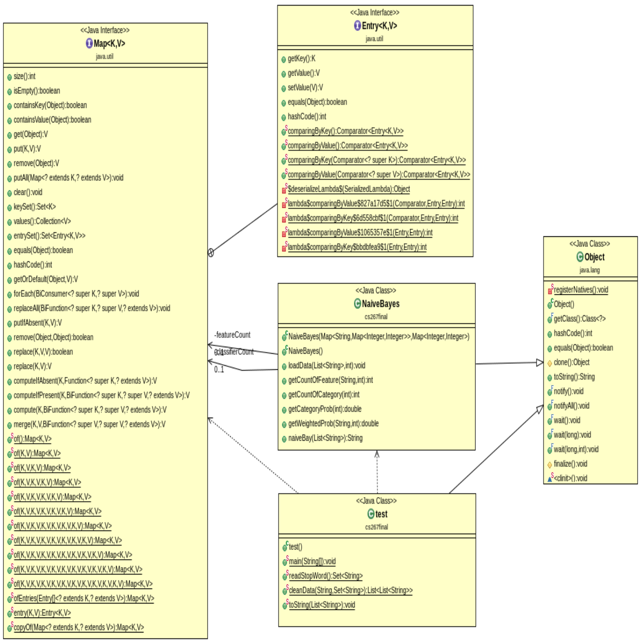


Fig 5. Architecture/Class Diagram

- get count of category – it retrieves the number of messages of a category and returns the classifier count
- probability – that calculates each individaul category probability
- weighted probability – it calculates the probability of each feature in a category and to improve accuracy, a default weight of 1 is added to each count.

The test class consists of the main function,

stopword function – that reads unnecessary words from the specified stopwords file

clean data - that performs data cleaning for the training as well as test data set.

IV. Experimental Evaluation

A. Cyberbullying Data Evaluation

The created models are evaluated based on 2 sets of data: Cyberbullying data and Non-Cyberbullying data. The corpus of Cyberbullying related Twitter tweets were analyzed. Figure 6 shows the percentage of cyberbullying and non cyberbullying tweets based on the training data.

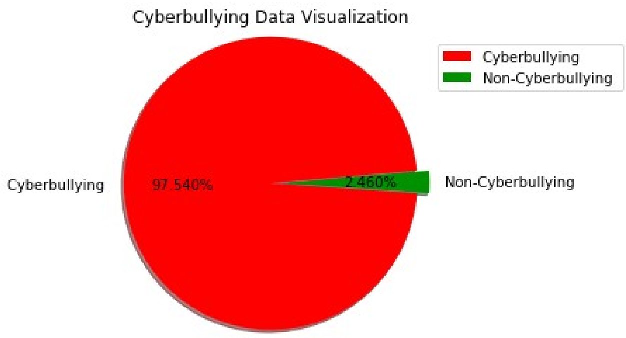


Fig. 6 Evaluation accuracy for Cyberbullying data

Figure 6 shows that 97% of the harvested tweets contain some or the other form of cyberbullying content. Only 2.4% of the total tweets are categorized as normal data that does not contain any form of cyberbully. The corpus used for training consisted of 326 cyberbullying sentences which were pre-processed to filter out any outliers. Out of these 326 sentences 318 contain corona virus related bullying and 8 sentences without any bullying. Thus the accuracy of the model turns out to be 97.546%.

B. Non-cyberbullying data evaluation

Figure 7 shows the percentage of non-cyber bullying and cyberbullying tweets based on the non-cyberbullying dataset.

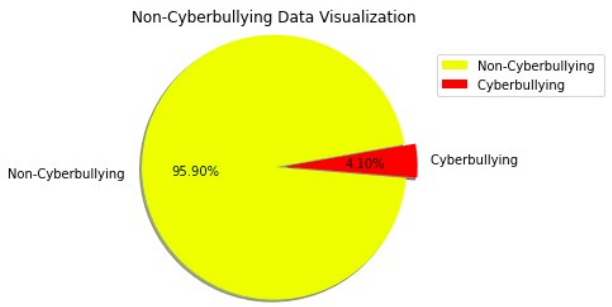


Fig. 7 Evaluation accuracy for Non-Cyberbullying data

Figure 7 shows that almost 96% of the harvested tweets do not contain any form of cyberbullying content. Only 4% of the total tweets are categorized under cyberbullying that contains some or the other form of bully. The corpus used for training consisted of 223 normal sentences which were pre-processed to filter out any outliers. Out of these 223 sentences 214 contain no bullying data and 9 sentences contain corona virus related bullying. Thus the accuracy of the model turns out to be 95.96%.

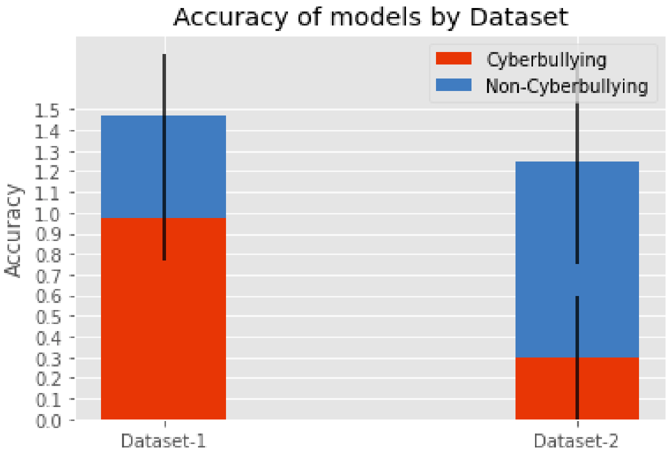


Fig. 8 Accuracy of models by dataset

The Naive Bayes classifier yields high accuracy of detection for training dataset that contains labelled cyberbullying data as well as for test dataset that contains non-cyberbullying or normal data. Initially, for the training dataset that is the dataset 1, containing bullying and non-cyberbullying parts, 0.97 accurate detections are made and 0.23 inaccurate detections as shown in figure 8.

Similarly, for the test dataset that is dataset 2, the model is able to detect 0.95 of the non-cyberbullying sentences accurately and fails to detect 0.5 of the remaining dataset. With exhaustive training the model can be made to perform 0.98 of the detections accurately.

V. Conclusion

In this research, we implemented an approach to detect cyberbullying on Twitter. This approach is based on data analytics using Naive Bayes classifiers to extract cyberbullying and non-cyberbullying features. We collected data by building a custom web crawler for scraping websites and used 80% as training data and 20% as test data. We evaluated the accuracy of the model on the test data. The evaluation results show a high accuracy of detection for cyberbullying content. The data pre-processing is an important step in cyberbullying detection. Based on these results, naive bayes classifier performs well as it only depends on several independent features. We also found that stopwords help improve the performance of detection and accuracy improves significantly as the stopword corpus increases. We believe that our work justifies the high performance of Naive Bayes model and makes a new contribution to cyberbullying detection.

VI. Discussions and Future Work

This new trend of cyberbullying detection we discussed is related to the social media environment. As a next step we would like to consider more about the social network and real time news during the feature chosing step. The features in our model can adapt easily to the latest news and social environment, and it is not limited to the typical ones. We will also use parallel data storage during the data process step as the dataset size increases.

The model built on our approach has some limitations because the corpus does not cover all the possible keywords used in cyber bullying. This is mainly due to restricted usage provided by Twitter as it allows for scraping of only a certain amount of data for a fixed period of time. To overcome this we will have to scrape data everyday for a long period to create a robust corpus of data. This data when analysed will give better results in terms of training as well as test data. The model built on this research generalizes well for a wide variety of cyberbullying sentences but there are 3-4% of indirect cyberbullying or sexual harassment sentences that might be missed by the model as it has accuracy of 97%. For future work we would like to train our model to detect these indirect cyberbullying sentences. Once this is achieved, it can be used as an effective tool to detect and stop cyberbullying completely on Twitter and help save lives by preventing high suicide rates among teens. As an additional measure, we would like to make it real time and deploy our model on Twitter to see how the model performs detections and prevents cyberbullying.

Acknowledgement

We thank Dr. Mike Wu for his guidance on the project and the Department of Computer Science at San Jose State University for the research opportunity.

References

[1] Ting, W. S. Liou, D. Liberona, S. Wang and G. M. Tarazona Bermudez, "Towards the detection of cyberbully based on social network mining tech," 2017 Int. Conf. on BESC, Krakow, 2017, pp. 1-2.

[2] Y. Win, "Classification using Support Vector Machine to Detect Cyberbullying in Social Media for Myanmar Language," 2019 IEEE Int. Conf. On ICCE-Asia, Bangkok, Thailand, 2019, pp. 122-125.

[3] K. D. Gorro, M. J. G. Sabellano, K. Gorro, C. Maderazo, and K. Capao, "Classification of Cyberbullying in Facebook Using Selenium and SVM," 2018 3rd Int. Conf. on Comput. and Commun. Sys., Nagoya, 2018.

[4] B. Haidar, M. Chamoun and F. Yamout, "Cyberbullying Detection: A Survey on Multilingual Techniques," 2016 European Modelling Symposium (EMS), Pisa, 2016.

[5] V. S. Chavan and Shylaja S S, "Machine learning approach for detection of cyber-aggressive comments by peers on social media network," (ICACCI), Kochi, 2015.

[6] Y. N. Silva, C. Rich and D. Hall, "BullyBlocker: Towards the identification of cyberbullying in social networking sites," 2016 IEEE/ACM, San Francisco, CA, 2016.

[7] C. Chelmis, D. Zois and M. Yao, "Mining Patterns of Cyberbullying on Twitter," 2017 IEEE Int. Conf. on DM. Wksh, New Orleans, LA, 2017.

[8] Noviantho, S. M. Isa and L. Ashianti, "Cyberbullying classification using text mining," 1st Int. Conf. on Inf. and Comput. Sci., Semarang, 2017.

[9] Fauman, M. A. (2008). Cyber bullying: Bullying in the digital age. American Journal of Psychiatry, 165(6), 780-781

[10] I. Riadi, "Detection of cyberbullying on social media using data mining techniques," Int. J. of Comput. Sci. and Inf. Secur., vol. 15, no. 3, 2017