# Quickgarde: A Plug-in for Detecting Cyberbullying Occurrences in Filipino Social Media Posts

A Research Paper Submitted
To the Faculty of School of
Computing and Information Technologies
Of
Asia Pacific College

In Partial Fulfillment of the Requirements for the subject
Thesis Presentation

By

Ballesteros, Faith I.
Capuz, Paulo Miguel S.
Mallari, Samantha G.
Samillano, Eva R.

Date
September 11, 2017

**ASIA PACIFIC COLLEGE**

Approval Sheet

QUICKGARDE: A PLUG-IN FOR DETECTING CYBERBULLYING OCCURRENCES IN FILIPINO SOCIAL MEDIA POSTS

Prepared and Submitted by:

Ballesteros, Faith I.
Capuz, Paulo Miguel S.
Mallari, Samantha G.
Samillano, Eva R.

In Partial Fulfilment of the Requirements for the Degree of

Bachelor of Science in Computer Science with Specialization in Systems Software

Examined and Recommended for Acceptance and Approval for Research/Capstone Presentation

______Sir Ernesto Boydon______
Adviser

Panel of Judges

________Ms. Roselle Wednesday Gardon________
Chairperson

Panel Members

______Dr. Manuel Calimlim, Jr.______ ________Ms. Maricel Naviamos________
Panel Member Panel Member

Acceptance and Approved in partial fulfillment of the requirements for the degree of Bachelor of Science

__________Ms. Rhea Valbuena__________
Executive Director
School of Computing and Information Technologies

## Table of Contents

## Abstract

Social media has brought about a revolutionary change in terms of sharing information and communicating with other people online. However, alongside with the advent of social media platforms, cyberbullying has become more prevalent. This phenomenon has been linked to detrimental effects such as suicide and depression. Recent studies on cyberbullying detection employ a text classification approach and they primarily focus on optimizing the accuracy of the detection model. This research aims to extend the technological feasibility of automating the detection of cyberbullying in social media sites into the generation of reports once a harmful post has been detected. It begins with the creation of a Support Vector Machine classifier in WEKA which can detect cyberbullying statements written in English and Filipino. The optimal model was able to achieve an accuracy of 57% and a kappa score of 0.2094. After creating the model, public posts from Twitter were retrieved. Text preprocessing techniques such as cleaning and tokenization were applied on the data. Lastly, they were converted into Bag-of-Words (BoW) representation. Once a post has been classified by the detection model as cyberbullying, a report which contains the author's name, the content of the post, and the time and date it was posted will be generated. This novel approach shows a potential for detecting harmful messages and allowing social media administrators to provide timely responses.

## List of Figures

## List of Tables

# I. Introduction

## 1.1 Background of the Problem

Recent technological advancements have expanded the way people communicate. With the rise of Web 2.0, people can easily connect with one another through chat rooms, email, instant messaging, forums, and social networking sites (Sheoran, 2012). However, alongside the modern advancements in communication, an old pervasive issue arises - cyberbullying (Dadvar & De Jong, 2012). Cyberbullying is defined as an aggressive, intentional act carried out by an individual or a group over an electronic device against someone who cannot easily defend himself/herself (Van Royen, Poels, Daelemans, & Vandebosch, 2015). Furthermore, it is a form of harassment that occurs via the Internet which includes vicious forum posts, name calling in chat rooms, and sending cruel messages (What is Cyberbullying?, 2011). Unlike the traditional form of bullying, perpetrators of cyberbullying may use several types of communication-related technologies such as social networking sites to deliberately inflict harm on someone (Boehm, 2012). Since information spread fast across the cyberspace and the number of audience is limitless, it can leave deeper, long-lasting effects on the victim in comparison to that of physical bullying (Campbell, 2005). According to Smith et al (2008), victims of cyberbullying may experience severe depression, low self-esteem, or even commit suicide attempts.

The Philippines was recognized as the social media capital of the world, with more and more Filipinos being inclined to visit different social networking sites (Ellwood-Clayton, 2006). A study conducted by We Are Social in 2017 found that Filipinos spent an average of 4 hours and 17 minutes per day on social media sites (Digital in 2017: Global Overview, 2017). However, as the number of Filipino social media users continuously increase, it consequently intensifies the problem of cyberbullying in the Philippines (Gonzales, 2014). A survey administered by Stairway Foundation Inc. revealed that 80% of Filipinos have been cyberbullied through social media (Takumi, 2016). Popular cyberbullying incidents in the Philippines are Paula Jaime Salvosa's "Amalayer" incident (Lacuata, 2014), Raymond Malinay's prank involvement (Tulad, 2012), and DJ Karen Bordador's cyberbullying experience, following her arrest with her boyfriend in a drug-related buy bust operation (Torres, 2016). However, these are only few of the cyberbullying instances that have been formally reported.

The growing cases of cyberbullying led to the introduction of Anti Bullying Act of 2013, which requires all elementary and secondary school to adopt policies that will prevent and address cyberbullying in educational institutions (RA 10627: The Anti-Bullying Act, 2015). In 2015, House Bill

5718 was proposed to provide consequences for cyberbullying act wherein perpetrators shall face a penalty of six months to six years of imprisonment (Republic Act No. 10627, 2013). Social media administrators also play a crucial role in the process of combating cyberbullying by ensuring a safe environment, deleting harmful contents, and identifying perpetrators of online bullying. Furthermore, they have adopted various strategies to protect their users by preventing and intervening in cyberbullying situations. Their current practice involves having a moderator that will monitor inappropriate content which will allow them to detect cyberbullying in an early stage and to take actions thereafter.  One of the most common methods used by these sites is introducing a set of privacy settings which allows users to limit the amount of information that can be viewed publicly. A reporting tool page was also used wherein users can report instances of online bullying directly to the administrators. Safety Mode, an opt-in setting, was introduced by YouTube to filter search results. Facebook has moderation and profanity blocklist that can be used to filter a set of harmful words on a page. Twitter offers Mute Feature that allows a user to remove a person's tweets from his timeline without them knowing.

Despite the efforts made by the authority and administrators of social networking sites, these methods were deemed to be inefficient because it is impossible to monitor all activities in the cyberspace given the vast amount of information available online. In addition to this, their methods rely heavily on the users to submit a report before taking an action. Since Philippines remains to be on a conservative level, Filipinos are often reluctant to admit that they have been cyberbullied and report a cyberbullying instance (Takumi, 2016). Thus, there is a need for technology to intervene in the process of mitigating online bullying.

To facilitate the process of monitoring online information and to track cyberbullying instances automatically and accurately, several studies were conducted towards the development of an automatic cyberbullying detection model (Dadvar et al., 2012; Dinakar et.al, 2011). Moreover, several Machine Learning approaches to text categorization were applied to automate this process. Two of the most popular methods were Naive Bayes (Sintaha, M. Satter, S. Zawad, N. Swamaker, C. & Hassan, A, 2016; Marathe, S. & Shirsat, K, 2015) and Support Vector Machines (Van Hee et al, 2015). Their methods significantly reduced the task of the moderator in monitoring the activities in social media.

The aforementioned researches focused on dealing with cyberbullying scenarios occurring within their respective country of origin. This research, on the other hand, aims to create a cyberbullying detection application that is primarily suited to address the problem of cyberbullying as defined by scenarios considered by the Filipinos themselves as cyberbullying. The application will be designed to identify, flag, and report cyberbullying statements automatically, in real-time, and will run in the background (while the user stays online).

## 1.2 Statement of the Problem

How can cyberbullying statements in Filipino languages be detected on social media sites?

## 1.3 Objectives

### 1.3.1 Main Objective

To create an application that can detect cyberbullying statements in Filipino languages on social networking sites such as Facebook, Twitter, etc.

### 1.3.2 Specific Objectives

- To generate a cyberbullying detection model
- To develop a cyberbullying detection system
- To test the system's performance

## 1.4 Scope and Limitations

The corpus (dataset) consists of 2000 statements which were obtained from either public Facebook and Twitter posts or Youtube comments. The totality of these statements pertained to the major controversial issues in the Philippines, given that it presents a negative connotation towards a particular person or groups of people.

Text preprocessing methods that were done on the dataset include cleaning, tokenization – the process of breaking down a statement into smaller pieces, and conversion of the dataset into Bag-of-Words form. The cleaning of the dataset involved the removal of all special characters, non-readable text (e.g. asdfghjkl), emoticons, links, and foreign language characters. Basic Jejemon slang was retained in the dataset.

Three schemes were used for text annotation (or labelling) namely cyberbullying, not cyberbullying, and ambiguous cyberbullying. 2000 statements were randomly distributed among Metro Manila citizens for them to annotate.

The Machine Learning algorithm that was utilized is the Support Vector Machine algorithm. The decision to do so was greatly influenced by the related literatures the proponents of this project have included in the document.

Cyberbullying occurrences in public social media posts expressed using the Filipino language will be detected word per word. 2000 statements were utilized in WEKA in order to form the cyberbullying detection model. 10-fold Cross Validation was used for determining the accuracy, precision, recall, F-measure, and Kappa statistic of the constructed cyberbullying detection model. However, only the accuracy and Kappa Statistic were used to measure the model's performance. Overall, it yielded an accuracy rate of 57.95% and a Kappa Statistic of 20.94%. It was initially experimented on the corpus data before being integrated with the application.

The program for the system, which will allow the automated identification, flagging, and reporting of cyberbullying occurrences on Twitter - the social networking site that will be used in the project as a testing platform - to take place will be hard-coded using the Java programming language. It is expected to come in the form of a plugin or a website extension.

To test the performance of the application, it should be able to interact with Twitter API through Twitter4J. Doing so would enable the application to directly gather tweets from the site, including additional information related to the tweet such as the username of the person who posted it, and the time and date it was posted. Reports will only feature posts that were declared as cyberbullying. Likewise, their accessibility is limited to the administrators of the site. These reports will then be arranged in a tabular format. Statements that will not be flagged as cyberbullying will be disregarded. Procedures to be implemented by the authorities to resolve the issue will no longer be covered in this project.

## 1.5 Significance

Theory is one of the main constituents of computer science. It is typically expressed mathematically, and aims to answer questions relevant to the limits of computation (e.g. whether a certain hypothesis can be proven feasible in terms of numbers) (Fields of Computer Science, n.d.). In this research, the proponents are trying to verify their hypothesis of utilizing technology to mitigate the occurrence of cyberbullying in social media. The process of doing so involves the use of existing algorithms to perform Machine Learning. This will then enable a computer to tell apart cyberbullying statements from non-cyberbullying ones, while they remain in their natural language forms – a feat that many believed to be impossible. Successful or not, this study will contribute to the field of computer science as it tried to channel one of the field's purpose or objective – to challenge the current limit of computation in terms of natural language processing.

The main significance of this research project was aimed towards the improvement of the current procedures being done in the identification and reporting of cyberbullying occurrences in social media sites, most especially among interactions between Filipino citizens residing in Metro Manila, Philippines. As mentioned earlier, the model was designed according to cyberbullying in the Philippine setup, indicating that it will only be able to classify statements expressed in the Filipino language. Doing so would benefit the majority of the people expressing themselves using this particular language, which a great number of Metro Manilans do. They will be able to avail themselves of a more efficient way of dealing with cyberbullying which would then guarantee them a fun and safe experience in social media.

Authorities, who are mainly those people given the right to monitor social media accounts of their organization's affiliates in search of inappropriate content (specifically cyberbullying) and resolve issues related to it, will likewise gain something from the results of this study. After all, the identification and reporting of potential cyberbullying content in the site will be automated with the aid of the algorithm. This will make their work easier and more efficient for them, as less time and effort will be exerted to complete the said task. Likewise, the reports will be done in real-time, indicating that they will be notified right away of the possible cyberbullying occurrences before it escalates.

The findings of this study will redound to the benefit of researchers who would want to explore both cyberbullying and Natural Language Processing (NLP) – a field combining the areas of computer science, artificial intelligence, and computational linguistics to comprehend human languages. The study provides detail on how the process of text classification was conducted with the aid of the linear Support Vector Machine Algorithm. Likewise, cyberbullying in the Philippines was defined in this study. Future researchers can make use of this information to sort cyberbullying from non-cyberbullying occurrences.

Anti-cyberbullying advocates, specifically those willing to help Filipinos, will be assisted by the system in the fulfillment of their advocacies as it will create a significant leap in terms of resolving such incidents through real-time identification and reporting processes to be conducted on each statement.

## 1.6 Definition of Terms

| Terms | Definition |
|---|---|
| **Bag-of-Words (BoW) Model** | A model, used in Natural Language Processing (NLP) and Information Retrieval (IR), to represent a multiset of words, disregarding grammar and word order |
| **Corpus** | A collection of written texts |
| **Cyberbullying Detection Model** | The output of WEKA toolkit when the preprocessed text (from the corpus) is integrated with Support Vector Machine (SVM) algorithm to detect Filipino cyberbullying statements online |
| **Filipino** | Refers to the combination of both Tagalog and English languages (Taglish) in expressing written or spoken statements, typically done by Metro Manila citizens in their everyday conversations |
| **Machine Learning (ML)** | A type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed |
| **Natural Language Processing (NLP)** | A field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages |
| **Social Media** | Websites and applications that enable users to create and share content or to participate in social networking |

| | |
|---|---|
| **WEKA toolkit** | Consists of a collection of Machine Learning algorithms for data mining tasks |

*Table 1: Definition of Terms*

## 1.7 Research Methodology

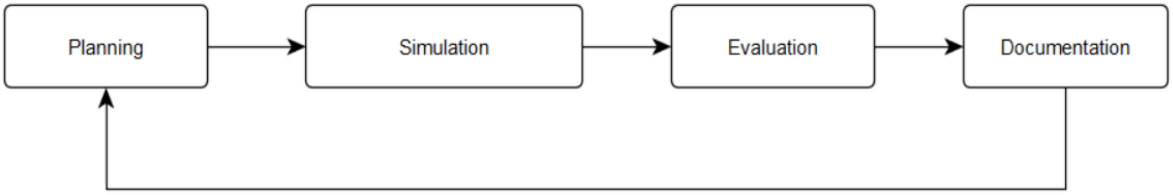


*Figure 1: Quickgarde's Project Development Life Cycle*

The diagram above illustrates the project development life cycle (PDLC) that was adapted by the proponents of the study for them to accomplish the objectives of their research. It was divided into 4 main phases: planning, simulation, evaluation, and documentation. It continues to follow an iterative flow until all the procedures applicable for the development and improvement of the project were successfully executed.

During the planning phase, the research idea was established. Furthermore, the finalization of the scheduling and appointment of tasks, and the flow of these tasks also took place. However, for the proponents to do so, they must first gather related literature and studies involving concepts such as cyberbullying in the Philippines, Natural Language Processing (NLP), text classification, Support Vector Machine (SVM) algorithm, and existing cyberbullying detection applications. These publications gave them an idea on how to design an appropriate architecture for their system in mind. They were able to create the needed diagrams from this information. A research project, unlike that of a software development project, bears no established methods in terms of its creation, therefore the procedures to be adapted into the project depends solely on the material that the researchers have gathered during their initial research. This explains why the planning phase is likewise inclusive of the research process.

The succeeding phase in the PDLC is the simulation phase. This phase can be broken down into two: the creation of the model and the integration of the model to the application (which will come in the form of a plug-in/web extension). This phase is also the longest, as the creation of the model, alone, was comprised of the following processes: data collection (for the textual corpus), cleaning of the data, text preprocessing (which includes frequency and BoW Unigram), and algorithm implementation (which involves both training and testing). On the other hand, the integration of the model to the application is inclusive of the process of establishing communication between the application and Twitter API (as Twitter will serve as the application's testing platform), the encoding of the data gathering, cleaning, text preprocessing, and algorithm implementation processes in Java, the transferring of flagged cyberbullying statements to the database, and the outputting of results (creation of reports).

The performance of the model and the application was subsequently evaluated in the next phase (evaluation phase). In the evaluation of the model's performance, several experiments were performed. These are the following: addition of textual data in the corpus, division of the data set in 60-40, 70-30, and 90-10 percentage splits, utilization of various k-fold cross validation techniques (except for the ten-fold cross validation), conversion of the dataset to both BoW Bigram and Trigram, and implementation of TF-IDF to put weights on the frequently occurring words among annotated cyberbullying instances in the dataset (instead of the frequency). These experiments yielded accuracy and kappa statistics respectively, which is relevant to the model's performance given such setups. These were then compared to the baseline (or initial) results – the model before it was submitted to the aforementioned experiments. The data was also described and analyzed at this stage in order for the researchers to draw inferences as to why the model yielded a particular accuracy and kappa statistic. The performance of the application, on the other hand, was evaluated based on whether or not it performed the identification, flagging, and reporting of cyberbullying statements, as expected, on Twitter.

The documentation phase is the last stage of this research project's PDLC. All the needed papers for documenting the project for future use, such as the thesis document, vision and scope document, change management plan, user manuals, and the like, were being accomplished at this point in time. Since the documents remain open to revisions until the final submission of the research paper, it is highly likely for the proponents to go through the project development life cycle once again.

# II. Related Literature

## 2.1 Cyberbullying Literatures

Several studies in the social sciences has been devoted to understanding the nature of cyberbullying and the extent of its prevalence among children and young adults (Dinakar, Reichart & Lieberman, 2011). This section focuses on the findings of the studies conducted with regards to cyberbullying.

### 2.1.1 Social Media as its Channel and its Implications on Cyberbullying

Gonzales (2014) conducted a qualitative study to explore the relationship between social media and cyberbullying. Through the use of focus interview analysis, he was able to gather information from eight experts from various field of specialization. From his study, he came up with the following conclusions:

- Social media is the root cause of cyberbullying
- There is no specific law in the Philippines that clearly defines punishable acts for cyberbullying
- Self-discipline must be imposed by all social media users
- Cyberbullying can be avoided, if people have a better understanding of social media
- Social media users should be wary of sharing personal information in the cyberspace
- The victim should report to the authority once the bully poses a serious threat to his life or liberty

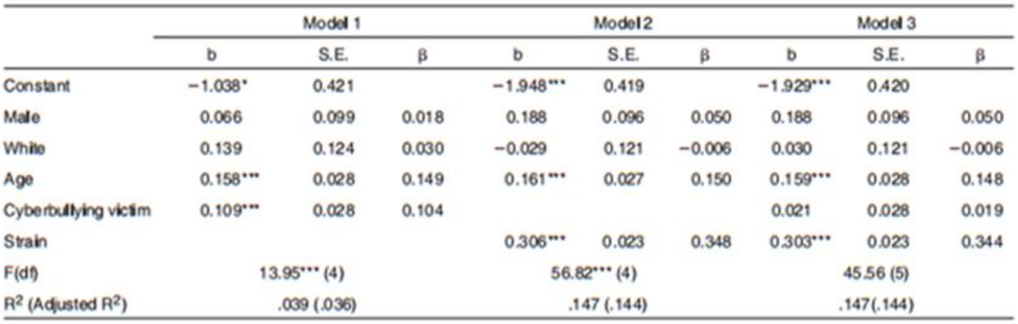
### 2.1.2 Offline Consequences of Online Victimization: School Violence and Delinquency

Hinduja and Patchin (2007) conducted a study to determine the relationship between victimization, strain, and deviant behavioral choices of the cyberbullying victims. Moreover, they used the general strain theory (GST) to identify both the emotional and behavioral effects of cyberbullying.

The proponents conducted an online survey methodology to obtain data from 1,388 adolescents. They used two primary independent measures (cyberbullying victimization and strain), a dependent variable (offline problem behaviors) and three demographic control variables such as age, race, and gender. Cyberbullying victimization is a scale that is composed of eight types of online victimization ranging from relatively minor forms of bullying to more serious forms of harassment.

The strain scale, on the other hand, refers to the common coping mechanism of a victim and is composed of nine items. The dependent variable is composed of an eleven-item index which represents the respondent's behavior for the past six months. It ranges from a minor form of deviance to more serious forms of delinquency.

For their experiment, a series of stepwise ordinary least squares (OLS) were estimated to explore the relationship between cyberbullying victimization, strain, and offline problem behaviors. In total, three models were created. The first model shows the relationship between cyberbullying victimization and offline problem behaviors, the second model illustrates the relationship between strain and offline problem behaviors, and the third model illustrates the relationship between cyberbullying victimization and strain and offline problem behaviors. As shown in Table 2.0, the first model proves that cyberbullying victimization is significantly related to offline problem behaviors which means youth who experience cyberbullying are more likely to participate in problem behaviors offline. The second model shows that strain is positively related to offline problem behaviors. Thus, youth who experience more strain are more likely to engage in offline problem behaviors. The third model illustrates that strain has a significant relationship with delinquency. The result of the third model demonstrates that strain serves as a mediator for the relationship between cyberbullying victimization and offline problem behaviors mainly because strain can be attributed to the effect of cyberbullying victimization on offline problem behaviors.



| | Model 1 | | | Model 2 | | | Model 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | b | S.E. | β | b | S.E. | β | b | S.E. | β |
| Constant | −1.038* | 0.421 | | −1.948*** | 0.419 | | −1.929*** | 0.420 | |
| Male | 0.066 | 0.099 | 0.018 | 0.188 | 0.096 | 0.050 | 0.188 | 0.096 | 0.050 |
| White | 0.139 | 0.124 | 0.030 | −0.029 | 0.121 | −0.006 | 0.030 | 0.121 | −0.006 |
| Age | 0.158*** | 0.028 | 0.149 | 0.161*** | 0.027 | 0.150 | 0.159*** | 0.028 | 0.148 |
| Cyberbullying victim | 0.109*** | 0.028 | 0.104 | | | | 0.021 | 0.028 | 0.019 |
| Strain | | | | 0.306*** | 0.023 | 0.348 | 0.303*** | 0.023 | 0.344 |
| F(df) | 13.95*** (4) | | | 56.82*** (4) | | | 45.56 (5) | | |
| $R^2$ (Adjusted $R^2$) | .039 (.036) | | | .147 (.144) | | | .147(.144) | | |

*Table 2: Ordinary Least Squares Regression - Delinquency Regressed on Strain and Cyberbullying Victimization*

## 2.2 Text Classification

Recently, various Machine Learning approaches for automated text classification has witnessed a surge in terms of application (Sebastiani, 2002). This section presents the different applications of text classification including the methods that were employed by the researchers. It also presents the comparison of each approach when applied to different classification problems.

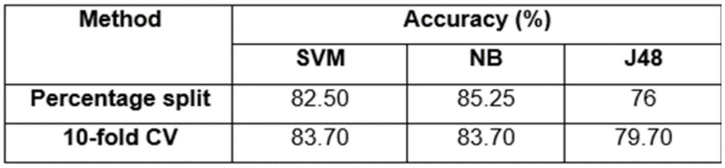
### 2.2.1 Comparative Assessment of the Performance of Three WEKA Text Classifiers Applied to Arabic Text

Wahbeh and Al-Khabi (2012) conducted an experiment to illustrate the performance of three different text classification techniques: SVM, Naïve Bayes, and C4.5 in classifying Arabic text documents.

The first phase of their experiment begins with the creation of the corpus by gathering Arabic text documents from different websites: Kooora, news-all, and from Saheeh Al-Bukhari book and other websites. These data were already classified into number of categories such as Sport, Economic, Religion, Politics, and Mohammed sayings. They gathered a total number of 1000 documents (250 documents for each category) for their corpus.

As for the preprocessing step, any occurrences of digits and punctuation marks were removed. Next, the set of characters were normalized into a canonical form. Third, non-Arabic text, special characters, and stop words were also removed. The last step involved in pre-processing includes the tokenization of the documents. All of the preprocessing steps were done using a tool created in C#. These documents were converted into ARFF format by utilizing WEKA TextDirectoryToArrf converter and StringToWordVector.

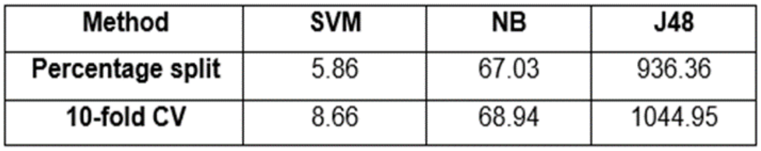
For their preliminary experiment, they utilized the percentage split which involves the process of dividing the data into two partitions: 60% was used for training phase while the remaining 40% was used for testing phase. Furthermore, they used 10-fold cross-validation technique for both dataset. These experiments were done to know if there will be improvements in the accuracy when the 10-fold cross-validation method is applied instead of the percentage split alone. Table 3 shows the comparison of the performance of three classifiers with respect to the percentage split method and 10-fold cross-validation. As shown in Table 3, the 10-fold cross-validation has significantly improved the accuracy for each classifier.



| Method | Accuracy (%) | | |
|---|---|---|---|
| | SVM | NB | J48 |
| Percentage split | 82.50 | 85.25 | 76 |
| 10-fold CV | 83.70 | 83.70 | 79.70 |

*Table 3: Comparison of the Three Classifiers in Terms of their Accuracy*

Aside from the accuracy of each classifier, they also measured the time taken for constructing each model. As shown in Table 4, SVM requires the shortest amount of time to build the model. It was followed by the NB classifier. Lastly, J48 requires the largest amount of time in building the model.

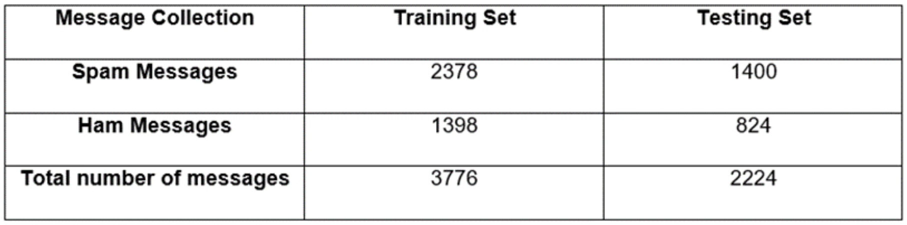


| Method | SVM | NB | J48 |
|---|---|---|---|
| Percentage split | 5.86 | 67.03 | 936.36 |
| 10-fold CV | 8.66 | 68.94 | 1044.95 |

*Table 4: The Amount of Time Taken in Building the Models*

### 2.2.2 Machine Learning Methods for Spam Email Classification

Awad and Elseuofi (2011) compared the performance of different Machine Learning algorithms in classifying spam emails. Their experiment begins with the construction of a corpus by compiling both spam and legitimate emails from SpamAssassin, a collection of publicly available emails. This collection contains a total number of 6000 emails. Their dataset divided the corpus into two sets: training and testing.

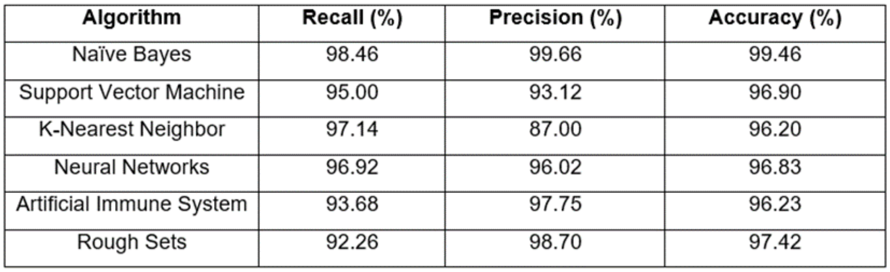


| Message Collection | Training Set | Testing Set |
|---|---|---|
| Spam Messages | 2378 | 1400 |
| Ham Messages | 1398 | 824 |
| Total number of messages | 3776 | 2224 |

*Table 5: Corpora of Spam and Ham Messages*

Each email was further divided into three different parts: subject (the title of the email), from (the name of the sender) and body (the main part of the message). The preprocessing steps involve the removal of common words and case-change, wherein each word in the body is converted into small letters. Each message was converted to a feature vector which results into 21,700 attributes.

They selected a number of 100 features. These features were the most frequent words in spam mails. In addition to this, every email in the training dataset was denoted as a feature vector. Once the preprocessing steps were done, they applied different Machine Learning algorithms: Naïve Bayes, K-Nearest Neighbor, Artificial Neural Networks, Support Vector Machine, Artificial Immune System, and Rough Sets. To evaluate the performance of each classifier, they used precision, recall, and accuracy. As shown in Table 6, Naïve Bayes outperformed the other classifiers in terms of precision, recall, and accuracy.



| Algorithm | Recall (%) | Precision (%) | Accuracy (%) |
|---|---|---|---|
| Naïve Bayes | 98.46 | 99.66 | 99.46 |
| Support Vector Machine | 95.00 | 93.12 | 96.90 |
| K-Nearest Neighbor | 97.14 | 87.00 | 96.20 |
| Neural Networks | 96.92 | 96.02 | 96.83 |
| Artificial Immune System | 93.68 | 97.75 | 96.23 |
| Rough Sets | 92.26 | 98.70 | 97.42 |

*Table 6: The Performance of Different Machine Learning Algorithms in Spam Email Classification*

### 2.2.3 Predicting Age and Gender in Online Social Networks

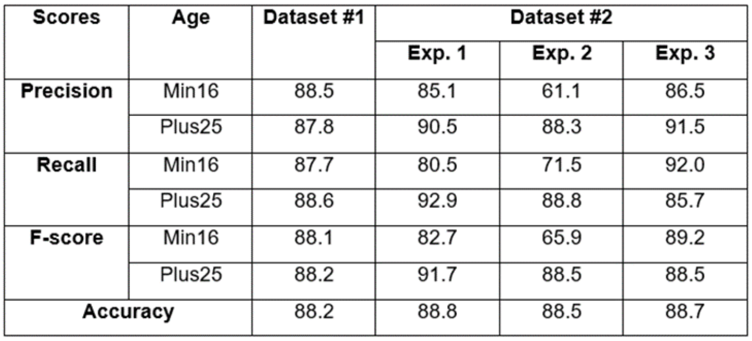
Peersman, Daelemans, and Vaerenbergh (2010) conducted a study to explore the feasibility of detecting age and gender using statistical text classification and the usefulness of this approach when applied to short texts.

The experimentation phase begins by obtaining 1,537,283 Flemish Dutch posts from Netlog. Relevant information such as age and gender of the authors were also identified in the corpus. The first step in pre-processing involves extracting only the last post of each interaction. Tokenization was also applied to the dataset, which results into a total number of 18,713,627 tokens. Moreover, each token was converted to a lowercase and four or more consecutive identical characters were reduced to three. The third step in preprocessing involves grouping the data using the profile data. In this step, the corpus is divided into following subclasses: min16 (from 11 to 15 years old), plus16 (16 and older), plus18 (18 and older) and plus25 (25 and older). The metadata for both genders were also incorporated and the following classes were derived: min16_male, min16_female, plus25_male and plus25_female.

For their experiment, they used 10,000 posts per class then subsequently decreased it to 5000 and 1000 posts per class.

For the feature selection process, they applied the Chisquare (χ2) metric. The feature set was limited to token and character features: word unigrams, bigrams and trigrams, character bigrams and trigrams, and tetragrams. The feature sets were built by selecting the 1000, 5000, 10,000 and 50,000 features with the highest Chisquare values. Once the features have been selected, each document is represented as a binary vector for the SVM classifier. Moreover, the SVM classifier was trained using Liblinear package. The performance was evaluated using 10-fold cross validation as experimental regime.

In their first experiment, they reduced the number of classes in both train and test sets from the four complex classes to two in order to compare the result to those from the first dataset, which was balanced according to age only. In their second experiment, the classifier was trained into four complex classes then the number in the classifier's output was reduced to two classes in order to determine whether the extra gender information the classifier had acquired would generate to a better age prediction on the test sets. The third experiment involves the reduction of the number of classes in both training and test sets to two age classes and gender was included as an extra feature in every instance. Table 7 illustrates the overview of the results of the three experiments in comparison with the first dataset.



| Scores | Age | Dataset #1 | Dataset #2 | | |
|---|---|---|---|---|---|
| | | | Exp. 1 | Exp. 2 | Exp. 3 |
| **Precision** | Min16 | 88.5 | 85.1 | 61.1 | 86.5 |
| | Plus25 | 87.8 | 90.5 | 88.3 | 91.5 |
| **Recall** | Min16 | 87.7 | 80.5 | 71.5 | 92.0 |
| | Plus25 | 88.6 | 92.9 | 88.8 | 85.7 |
| **F-score** | Min16 | 88.1 | 82.7 | 65.9 | 89.2 |
| | Plus25 | 88.2 | 91.7 | 88.5 | 88.5 |
| **Accuracy** | | 88.2 | 88.8 | 88.5 | 88.7 |

*Table 7: The Result of the Three Experiments in comparison to the First Dataset*

### 2.2.4 Classifying Typhoon Related Tweets

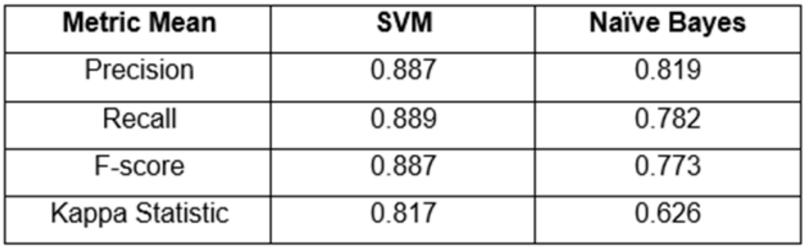
In a study conducted by Lam, Paner, Macatangay, and Delos Santos (2014), they illustrated the classification of typhoon related tweets into six categories:

Resource coordination

- Urgent rescue needed
- Urgent rescue resolution
- Damage reporting
- Missing people
- Media storm coverage

The experimentation phase begins with the collection of 2,356 tweets using Tweet Miner. Furthermore, these data were stored in the SQLite database. The preprocessing steps involve the filtering of tweets that do not contain an official hashtag. Moreover, each data in the set were converted into lowercase for two main purposes: to normalize the tweet by removing duplicate words from inconsistent casing and to remove official hashtags. All of these steps were done using Tweet Filter. Additionally, the filtered tweets are converted into BoW representation in ARFF format.

For their experiment, they trained both SVM and Naïve Bayes classifiers in WEKA. Furthermore, these classifiers were tested using ten-fold cross validation. For the evaluation metrics, they used precision, recall, f-score, and kappa statistics. As shown in Table 8, the SVM classifier outweighs the performance of Naïve Bayes classifier in both metrics.



| Metric Mean | SVM | Naïve Bayes |
|---|---|---|
| Precision | 0.887 | 0.819 |
| Recall | 0.889 | 0.782 |
| F-score | 0.887 | 0.773 |
| Kappa Statistic | 0.817 | 0.626 |

*Table 8: The Comparison of the Performance of SVM and Naive Bayes*

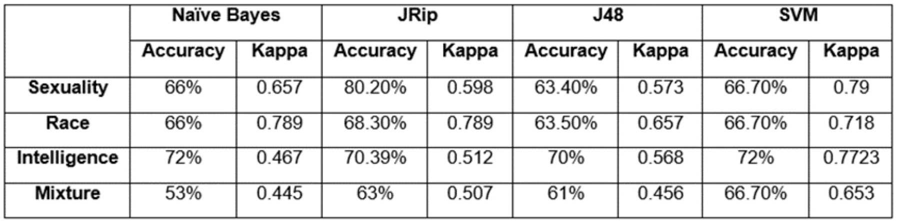
## 2.3 Cyberbullying Detection

Several studies have been conducted in automating the detection of cyberbullying on social networking sites to flag harmful messages and prevent these messages from remaining in the cyberspace by providing timely responses (Van Royen, Poels, Daelemans & Vandebosch, 2015). This section focuses on the various methods used by different researchers in automating the process of detecting cyberbullying. It also examines the effectivity of each approach.

### 2.3.1 Modeling the Detection of Textual Cyberbullying

Dinakar, Reichart, and Lieberman (2011) proposed a method in creating a cyberbullying detection model. Their experiment begins with the creation of a corpus composed of YouTube comments by using YouTube PHP API. They were able to obtain a number of comments that exceeds 50,000. The comments were partitioned into clusters of physical appearance, sexuality, race and culture, and intelligence. In addition to this, 1500 comments from each cluster were annotated to three categories: sexuality, race and culture, and intelligence. As for those comments that were not related to the cluster, they were given a label "none". Each dataset was subjected to four operations: the removal of stop-words, stemming, removal of unnecessary sequence of characters, and cleaning. The dataset for each cluster were further divided into three partitions: 50% training, 30% validation and 20% test data. Moreover, they used four supervised learning methods: Naïve Bayes, SVM, JRip, and J48.

They extracted two kinds of feature from each dataset: general features and specific features. The general features were common across all datasets for both experiments and they are composed of: TF-IDF, Ortony lexicon for negative, list of profane words, and POS bigrams (JJ_DT, PRP_VBP, and VB_PRP). The label specific-features are composed of topic specific unigrams and bigrams. To measure the effectivity of each classifier, they used accuracy and kappa statistics.



| | Naïve Bayes | | JRip | | J48 | | SVM | |
|---|---|---|---|---|---|---|---|---|
| | Accuracy | Kappa | Accuracy | Kappa | Accuracy | Kappa | Accuracy | Kappa |
| Sexuality | 66% | 0.657 | 80.20% | 0.598 | 63.40% | 0.573 | 66.70% | 0.79 |
| Race | 66% | 0.789 | 68.30% | 0.789 | 63.50% | 0.657 | 66.70% | 0.718 |
| Intelligence | 72% | 0.467 | 70.39% | 0.512 | 70% | 0.568 | 72% | 0.7723 |
| Mixture | 53% | 0.445 | 63% | 0.507 | 61% | 0.456 | 66.70% | 0.653 |

*Table 9: The Comparison of the Performance of Naive Bayes, JRip, J48, and SVM*

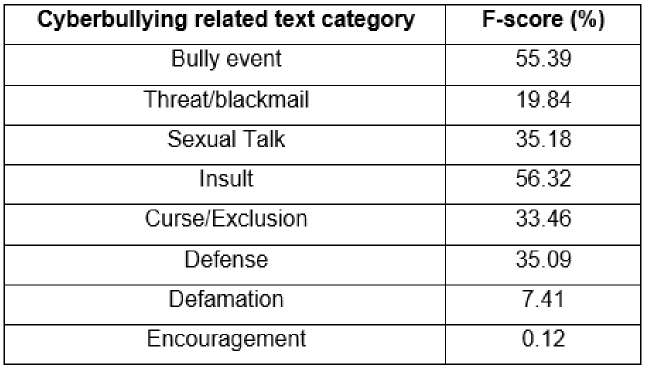
As shown in Table 9, JRip yields the best performance in terms of accuracy while SVM is the most reliable as measured by kappa statistics. In addition to this, the binary classifiers trained for each individual label performed better than multi-class classifiers trained for all the labels.

### 2.3.2 Automatic Detection and Prevention of Cyberbullying

In the experiment of Van Hee et. al (2015), they proposed a method for automating the identification of cyberbullying events and their classification into cyberbullying categories. The experimentation phase begins with the creation of corpus by collecting 91, 370 Dutch posts from Ask.fm. Moreover, they illustrated two levels of annotation: First, the assignment of harmfulness score to the post on a three-point scale wherein 0 indicates non-cyberbullying event, 1 indicates mild cyberbullying event, and 2 indicates severe cyberbullying event. Moreover, the roles in a cyberbullying event were also identified: victim, harasser, bystander-defenders (who discourage the harasser) and bystander-assistant (who take part in the actions of the harasser). At the second level of annotation, each data was classified into fine-grained text categories related to cyberbullying: insults, threats, sexual talk, defamation, defense and curse. In total, 85,462 Dutch posts were successfully annotated using brat rapid annotation tool. Moreover, the interannotator agreement scores were calculated using Kappa. They obtained a Kappa score of 0.69 in the identification of cyberbullying events. Additionally, the Kappa scores for the fine-grained cyberbullying categories such as Threat, Insult, Defense, Sexual Talk vary from 0.52 to 0.66.

The preprocessing steps involved tokenization, PoS-tagging and lemmatization to the data by utilizing LeTs Preprocess Toolkit. They implemented two types of lexical features for their experiment: bag-of-word features and polarity features based on existing sentiment lexicons. Thus, it results into a set of 300,000 features. The proponents utilized a Support Vector Machine (SVM) as their classification algorithm. All of their experiments were carried out using Pattern.

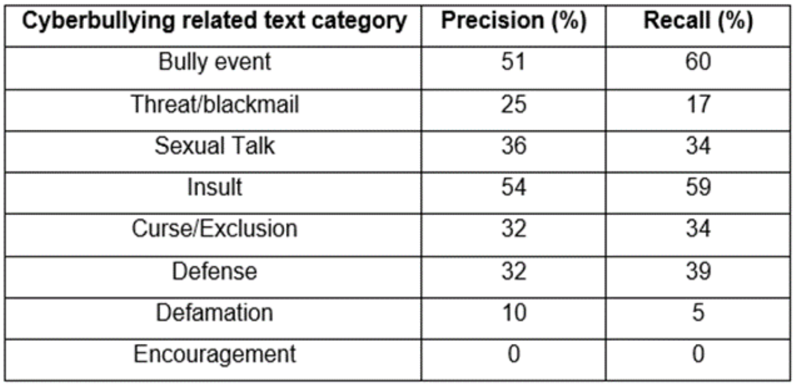
For their preliminary experiment, the evaluation was done using 10-fold cross-validation. Moreover, they used F-score for their evaluation metric. Table 10 shows the result of their preliminary experiment by using F-score.



| Cyberbullying related text category | F-score (%) |
|---|---|
| Bully event | 55.39 |
| Threat/blackmail | 19.84 |
| Sexual Talk | 35.18 |
| Insult | 56.32 |
| Curse/Exclusion | 33.46 |
| Defense | 35.09 |
| Defamation | 7.41 |
| Encouragement | 0.12 |

*Table 10: Classification Results for the Identification of Cyberbullying Events and Fine-grained Text Categories in terms of F-score*

Table 11 illustrates the performance of both precision and recall with regards to the identification of cyberbullying event and their classification into fine-grained text categories.



| Cyberbullying related text category | Precision (%) | Recall (%) |
|---|---|---|
| Bully event | 51 | 60 |
| Threat/blackmail | 25 | 17 |
| Sexual Talk | 36 | 34 |
| Insult | 54 | 59 |
| Curse/Exclusion | 32 | 34 |
| Defense | 32 | 39 |
| Defamation | 10 | 5 |
| Encouragement | 0 | 0 |

*Table 11: Classification Results for the Identification of Cyberbullying Events and Fine-grained Text Categories in terms of Precision and Recall*

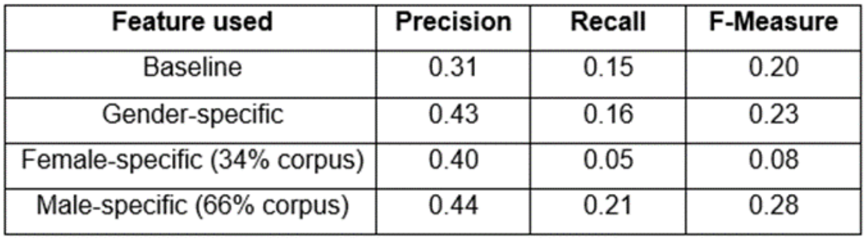
### 2.3.3 Improved Cyberbullying Detection using Gender Information

Dadvar, Jong, Ordeiman, and Trieschnigg (2012) believed that the incorporation of gender specific language features will improve the accuracy of a cyberbullying detection system. To test this idea, they conducted an experiment on improving cyberbullying detection with the aid of gender specific features.

Their dataset was composed of MySpace posts provided by Fundacion Barcelona Media. In total, the corpus contains 381,000 posts wherein 34% was written by male and 67% were from female. However, they were only able to utilize 2,200 posts for their experiment. Furthermore, the dataset was annotated into two categories: harassing and non-harassing. They analyzed the use of foul words among the 100,000 posts and compared the most frequently used foul words by each gender. By utilizing Wilcoxon signed rank test, they were able to determine the different frequencies of foul words in each gender.

For harassment classification, they utilized four types of features: first, profane words (including their acronyms and abbreviations), personal pronouns, second person pronouns, and TFIDF. These features were employed to train the classifier. Moreover, they constructed a Support Vector Machine (SVM) classifier in WEKA. First, they utilized the posts written by both genders as their dataset, then they trained the classifier separately for each respective gender group. In evaluating the accuracy of the classifier, they used 10-fold cross validation and calculated its precision, recall and

F-measure. As shown in Table 12, the incorporation of gender-specific features improved the overall accuracy measures.



| Feature used | Precision | Recall | F-Measure |
|---|---|---|---|
| Baseline | 0.31 | 0.15 | 0.20 |
| Gender-specific | 0.43 | 0.16 | 0.23 |
| Female-specific (34% corpus) | 0.40 | 0.05 | 0.08 |
| Male-specific (66% corpus) | 0.44 | 0.21 | 0.28 |

*Table 12: The Result of Improving Cyberbullying Detection using Gender-specific Features*

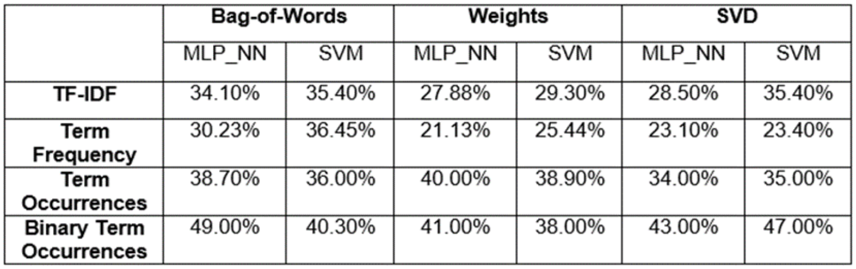
### 2.3.4 Cyberbullying Detection using Time Series Modeling

Potha and Maragoudakis (2014) proposed a method to predict the level of cyberbullying attack through classification and to examine the potential patterns between the linguistic styles of the online predator. For their experiment, they gathered a total number of 33 chat logs from Perverted-Justice. Their dataset was similar to the one that was used by Kontostathis in his study and the same labels for the data were applied. The dataset was further annotated by two students from the Ursinus College and they were categorized into 0, 200, 600, and 900. The category 0 pertains to the non-cyberbullying posts. The category 200 pertains to the posts which contain personal information. The category 600 was assigned to the posts that are associated with sexual meaning. Lastly, the category 900 was assigned to the posts which indicates an attempt from the predator to approach the victim. All predator posts from the set of XML transcripts were parsed and converted into a vector. The preprocessing steps were composed of tokenization, stop-word removal and a case transformation. For the vector representation, four different representation were analyzed: TF-IDF, term frequency, term occurrences and binary term occurrences. The set of predator's questions is ordered by date and time. The features were selected using a Linear SVM followed by an alternative feature selection method called Singular Value Decomposition (SVD). As for the classification, they used both linear and non-linear SVM and compared it to the performance of Multi-Layer Perceptron (MLP) neural net. The classification was done through RapidMiner data mining suite. A window size of K previous posts was chosen and the horizon was set to one in performing the forecasting task. Moreover, two different window sizes were selected (two and three) and horizon was set to the next predator's post category. Moreover, Singular Value Decomposition (SVD) was applied on each time series to create a unified signal that can capture the behavior of individual linguistic information as a whole. The Root Mean Square Forecasting Error (RMSFE) and the Mean Absolute Percent Error (MAPE) were used to measure the performance of the classifier. Table 13 depicts the result of using RMSFE for MLP neural networks and SVM using a window of two.

| | **Bag-of-Words** | | **Weights** | | **SVD** | |
|---|---|---|---|---|---|---|
| | MLP_NN | SVM | MLP_NN | SVM | MLP_NN | SVM |
| **TF-IDF** | 0.171 | 0.226 | 0.147 | 0.143 | 0.143 | 0.177 |
| **Term Frequency** | 0.156 | 0.205 | 0.094 | 0.116 | 0.106 | 0.117 |
| **Term Occurrences** | 0.194 | 0.232 | 0.15 | 0.194 | 0.14 | 0.141 |
| **Binary Term Occurrences** | 0.267 | 0.245 | 0.158 | 0.158 | 0.2 | 0.206 |

*Table 13: RMSFE for MLP Neural Networks and SVM using a Window of Two*

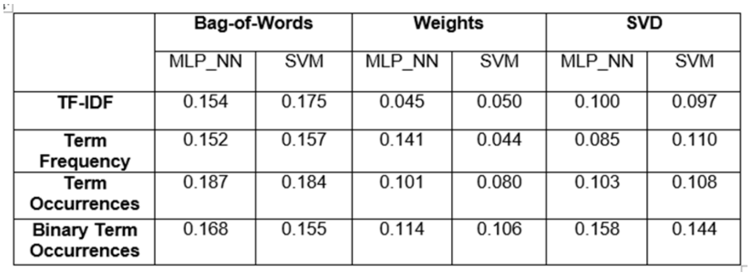
The MLP neural networks for the feature set represented by term frequencies yielded the highest performance. SVM as a predictor also portrays a satisfactory forecasting performance but it was significantly reduced when bag-of-words was used. The SVD method performed better than initial feature set in terms of both neural nets and SVM but it could not outperform the feature selection representation. As shown in Table 14, by using MAPE as a performance metric, MLP neural networks performed better than SVM in all representations while feature selection using SVM and term frequency portrayed the best result.



| | Bag-of-Words | | Weights | | SVD | |
|---|---|---|---|---|---|---|
| | MLP_NN | SVM | MLP_NN | SVM | MLP_NN | SVM |
| **TF-IDF** | 34.10% | 35.40% | 27.88% | 29.30% | 28.50% | 35.40% |
| **Term Frequency** | 30.23% | 36.45% | 21.13% | 25.44% | 23.10% | 23.40% |
| **Term Occurrences** | 38.70% | 36.00% | 40.00% | 38.90% | 34.00% | 35.00% |
| **Binary Term Occurrences** | 49.00% | 40.30% | 41.00% | 38.00% | 43.00% | 47.00% |

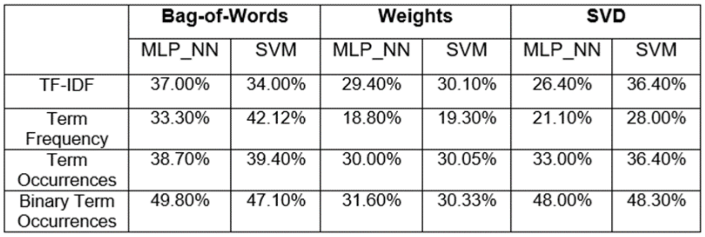
*Table 14: MAPE for MLP Neural Networks and SVM Using a Window of Two*

In Table 15 and Table 16, the same metrics and algorithms were applied for a window size of three. MLP performed better than SVM for all representation but the difference is smaller than before. Their difference ranges from 2% to 3.5%.



| | Bag-of-Words | | Weights | | SVD | |
|---|---|---|---|---|---|---|
| | MLP_NN | SVM | MLP_NN | SVM | MLP_NN | SVM |
| **TF-IDF** | 0.154 | 0.175 | 0.045 | 0.050 | 0.100 | 0.097 |
| **Term Frequency** | 0.152 | 0.157 | 0.141 | 0.044 | 0.085 | 0.110 |
| **Term Occurrences** | 0.187 | 0.184 | 0.101 | 0.080 | 0.103 | 0.108 |
| **Binary Term Occurrences** | 0.168 | 0.155 | 0.114 | 0.106 | 0.158 | 0.144 |

*Table 15: RMSFE for MLP Neural Networks and SVM Using a Window of Three*

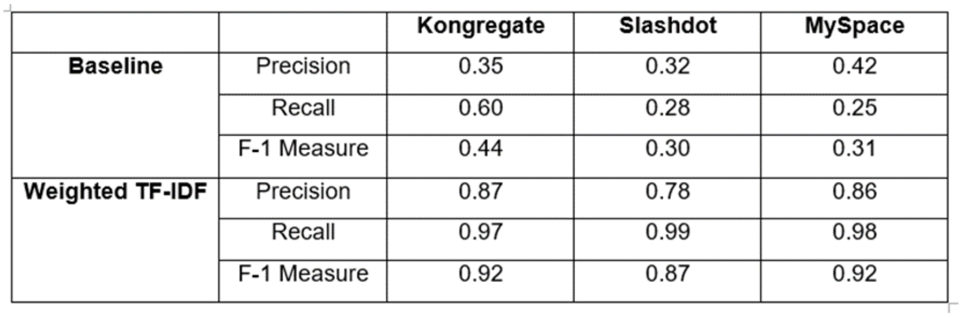


| | Bag-of-Words | | Weights | | SVD | |
|---|---|---|---|---|---|---|
| | MLP_NN | SVM | MLP_NN | SVM | MLP_NN | SVM |
| TF-IDF | 37.00% | 34.00% | 29.40% | 30.10% | 26.40% | 36.40% |
| Term Frequency | 33.30% | 42.12% | 18.80% | 19.30% | 21.10% | 28.00% |
| Term Occurrences | 38.70% | 39.40% | 30.00% | 30.05% | 33.00% | 36.40% |
| Binary Term Occurrences | 49.80% | 47.10% | 31.60% | 30.33% | 48.00% | 48.30% |

*Table 16: MAPE for MLP Neural Networks and SVM Using a Window of Three*

### 2.3.5 An Effective Approach for Cyberbullying Detection

Nahar, Li, and Pang (2013) proposed another approach in detecting cyberbullying instances through a weighting scheme of feature selection. Aside from cyberbullying detection, they also identified the most active cyberbullying predators and victims through ranking algorithms. For their study, they collected three different datasets from the Workshop on Content Analysis for the Web 2.0 and obtained manually-labelled data from the experiment of Yin as the ground truth for evaluating the performance of their proposed methodology. Their dataset was composed of data collected from three different social networking sites: Kongregate, Slashdot and MySpace. The preprocessing of data involves converting uppercase letters to lower case, stemming, removing stop words, extra characters and hyperlinks. The features are generated through Latent Dirichlet allocation (LDA) topic model and weighted TFIDF scheme. Semantic features were also applied in detecting harassing, abusive, and insulting posts. Moreover, they used three types of feature sets: all second pronouns, all remaining pronouns and foul words from noswearing.com. LibSVM was used to classify positive instances (bullying messages) and negative instances (non-bullying messages). Tenfold cross validation was also applied in the dataset. To measure the performance of their proposed methodology, they used Precision, Recall and F-1 Measure. A shown in Table 17, the proposed feature selection method using weighted TFIDF performed better in all aspects.



| | | Kongregate | Slashdot | MySpace |
|---|---|---|---|---|
| **Baseline** | Precision | 0.35 | 0.32 | 0.42 |
| | Recall | 0.60 | 0.28 | 0.25 |
| | F-1 Measure | 0.44 | 0.30 | 0.31 |
| **Weighted TF-IDF** | Precision | 0.87 | 0.78 | 0.86 |
| | Recall | 0.97 | 0.99 | 0.98 |
| | F-1 Measure | 0.92 | 0.87 | 0.92 |

*Table 17: Comparison of the Weighted TF-IDF with Baseline Method on Three Datasets*

The second part of their experiment focused on the identification of predators and victims. To visualize a user's connectivity based on the bullying posts by applying modularity algorithm, Gephi, a graphical interface, was used. The density of messages which pertains to the badness embedded within the post was computed by dividing the total count of bad words to the actual number of words in the post. The Hyperlink-Induced Topic Search (HITS) algorithm is used to identify a predator and the victim and it was also used in computing their respective scores.

### 2.3.6 Automated Role Detection in Cyberbullying Incidents

Cheng and Ng (2016) conducted an experiment on the detection of cyberbullying roles. Their experiment begins with the creation of a corpus by gathering data from Facebook and Youtube. In total, 6000 posts/comments written in both English and Tagalog were collected (1500 for YouTube and 4500 for Facebook). The dataset was cleaned by removing unnecessary symbols. Furthermore, it underwent normalization through the use of NormAPI. Lastly, each data was manually annotated into six classes: Bully, Accuser, Defender, Reporter, Victim, and N/A (which pertains to the instances that do not belong to any of the classes).

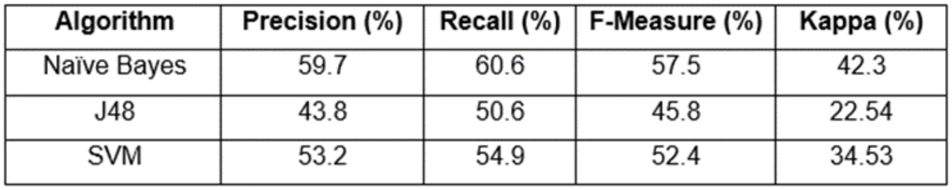
They implemented four types of features for their experiment: bag-of-word, TF-IDF, profane words, and word shape or the instances written in all uppercase. The experiment was conducted 7 times, each with a different set of role classes. The combination of the roles is as follows:

- All classes
- Bully and N/A (Bull + N/A)
- Accuser, Bully, and N/A (Acc + Bull + N/A)
- Bully, Defender, and N/A (Bull + Def + N/A)
- Bully, N/A, Reporter, and Victim (Bull + N/A + Rep + Vic)
- Accuser, Bully, Defender, Reporter, and Victim (All except N/A)
- Accuser, Bully, and Defender (Acc + Bull + Def)

For their first experiment, they used an initial set of 25 word features in each class. They checked the presence of both words that are written in all capital letters and those which contain profanity. From a total number of 150 features, it was decreased into 93 unique word features. Their second experiment involves the removal of both intersecting words and other added features. Thus, if a word feature is found in more than 1 class it will be removed in the feature set. The total number of 150 features was decreased into 63. Their third experiment involves the removal of both profanity and all capital words as features. In this experiment, the model was able to predict more bully and defender roles by removing both profane and full capital words.

For their fourth experiment, they utilized a weighting system that will assign weights to word features. This experiment was done in order for the model to be able to distinguish the respective classes for each feature. There was a significant improvement in the results as compared to the previous experiments. Thus, the assignment of weights can further help the classifier in identifying the features for each of the classes. The next experiment involves adding more features to the current set. Some word features were replaced with more relevant ones such as nouns and proper nouns. More common words were also removed in this phase. The initial number of 25 word features per class was increased into 50. The last experiment obtained the highest accuracy compared to the previous ones. Thus, by adding more relevant features, the roles of the bully, accuser, and victim were able to have more correctly classified instances.

Lastly, the experiment that yield the highest accuracy was tested using different algorithms: Naïve Bayes, J48 and Support Vector Machine. As shown in Table 18, among the three algorithms that were utilized, SVM yield the highest accuracy.



| Algorithm | Precision (%) | Recall (%) | F-Measure (%) | Kappa (%) |
|---|---|---|---|---|
| Naïve Bayes | 59.7 | 60.6 | 57.5 | 42.3 |
| J48 | 43.8 | 50.6 | 45.8 | 22.54 |
| SVM | 53.2 | 54.9 | 52.4 | 34.53 |

*Table 18: The Comparison of the Performance of Naive Bayes, J48, and SVM*

## 2.4 Synthesis

As shown in the previous studies, several approaches were used by the researchers in creating a cyberbullying detection model. Different sources and number of datasets vary from one study to another. Dinakar, Reichart, and Lieberman (2011) used 50,000 data from YouTube, Van Hee et. al (2015), collected 91, 370 Dutch posts from Ask.fm, Dadvar, Jong, Ordeiman, and Trieschnigg (2012) utilized a number of 2,200 posts from MySpace, while Cheng and Ng (2016) gathered 6000 posts from Facebook and YouTube. The present study gathered a total number of 2000 data from YouTube, Twitter, and Facebook. These studies focused on collecting posts written in English aside from Van Hee et al. (2015) who collected Dutch posts for their dataset. The present study further enhances the capability of a cyberbullying detection model by detecting posts written in both English and Tagalog. The researchers also classified their data into different categories: Cyberbullying Roles (Cheng and Ng, 2016), Harassing and Non-Harassing (Dadvar, Jong, Ordeiman, and Trieschnigg, 2012), Insults, Threats, Sexual Talk, Defamation, Defense and Curse (Van Hee et. al, 2015), and Sexuality, Race, and Intelligence (Dinakar, Reichart, Lieberman, 2012). The way they pre-processed their data also varies from one another: For the present study, they only focused on classifying cyberbullying and non-cyberbullying instances. The performance of the models were measured through Precision, Recall, F-Measure, Kappa score, and Accuracy. However, as for this study, the model's performance was measured by Accuracy and Kappa score alone. As seen in the previous studies, the performance of SVM was compared to algorithms such as JRip, J48, and Naïve Bayes. This study aims to improve the previous studies by comparing SVM into several Machine Learning algorithms in WEKA namely: Naïve Bayes, J48, JRip, ZeroR, Decision Stump, RandomTree, RandomForest, RepTREE, HoeffdingTree, DecisionTable, and OneR. The reason for this is to determine if SVM is the best algorithm that can be used for this kind of classification problem. Lastly, the previous studies merely focused on improving the performance of cyberbullying detection model. Thus, the present study aims to further enhance what has been started by the previous researchers by surpassing the technological feasibility of automating the detection of cyberbullying occurrences into automating the generation of reports once a cyberbullying post has been detected.

## III. Theoretical Background

### 3.1 Audience Segregation by Ervin Goffman

In his book "The Presentation of Self in Everyday Life", Ervin Goffman introduced the mechanisms of audience segregation. He describes how people play different roles in different situations. It is a mechanism wherein an individual performs roles, in order to create a favorable image of themselves and leave a good impression to others that is linked to the role they perform. The role that the individual performs is based on who their audience is.

Nowadays, more and more people are being inclined to visit social networking sites because it provides an easier way for social interactions and communications. These sites allow users to share personal information about themselves through text, pictures, and other forms of media which in turn, creates an image for each user; however, the representation of oneself in cyberspace is on a global scale in front of an audience which is possibly unknown and infinite. In social networking sites, the user's privacy is threatened because a large audience might have access to his personal information. In order to handle privacy issues, there were few social media sites that offer limited options for making one's profile visible for a specific set of individuals. In some cases, audience segregation is used as a solution to protect user's privacy; however, Goffman's segregation of audiences is a lot harder in the era of the Internet. Difficulties begin when the audience is used to a certain type of performance from an individual or team but observes another performance which does not create the same impression which results to cyberbullying. The impression created on a social networking profile may not resemble an individual's real-life identity.

The nature of communicating in cyberspace facilitates the potential for anonymous interactions. It was discovered that bullies who choose to use electronic means can easily hide their real identity and make themselves anonymous. Anonymity can be created through the use of temporary email addresses, fictitious names or unknown mobile number. The perception of anonymity in social media serves as a disinhibitor so that people are more likely to do and say things online that they would not do or say in a face to face situation. Another key characteristic of cyberbullying is the potential to reach a limitless audience. Due to the boundless nature of cyberspace, the audience is not confined to a single setting (such as school or office) but has the potential to be viewed by a global audience.

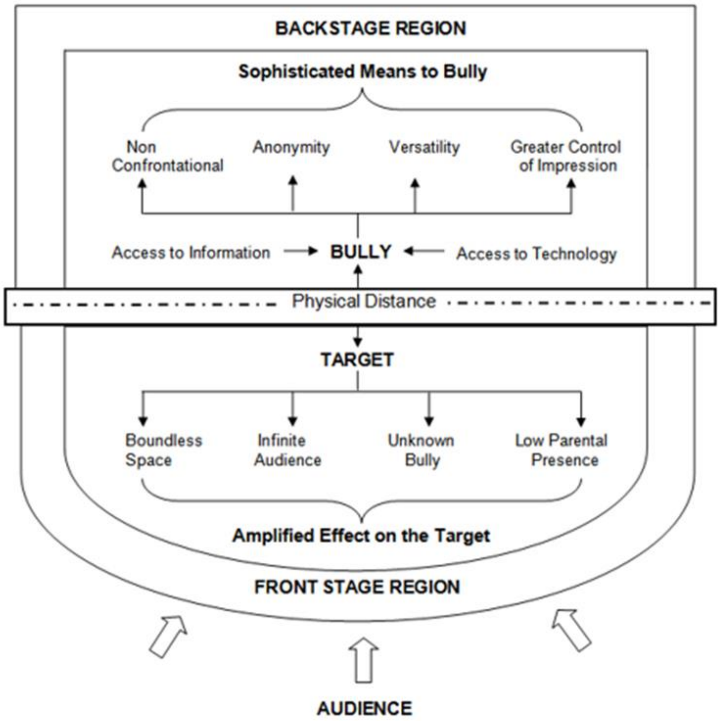


*Figure 2: Model of the Characteristics of Cyberbullying in relation to Bully and Target*

Goffman's framework (Figure 2) offers not only a way of thinking about space in terms of performance but also a way of thinking about how people may act differently depending on the audience and setting which are relevant to an exploration of cyberbullying. Goffman defined three roles in this mechanism: performer, audience, and outsider. These roles can be paralleled to the roles of a target, bully, and bystander. By framing bullying as a performance, a framework is provided that enables one to consider the bystander group as an audience and how different settings may affect how young people act towards others. In order to set the scene for a performance, Goffman made a distinction between the three regions of social space where an individual interacts. The front region is defined as the public performance area. The backstage region is a place wherein the performer can privately prepare for the performance or where members of a group can openly construct the impression they are planning to give. The outside region pertains to those parts which are not covered by backstage and front stage. By using Goffman's framework of performance, cyberspace interactions can be executed by the bully in the backstage region which impacts on the target in the public front stage region. As the backstage region is a place that performers may privately prepare away from the audience, this provides time and space for the bully to plan the ways in which they wish to target others. The physical distance which cyberspace interactions facilitate may also result in the bully managing the impression 'given off', the ability for the bully to conceal their identity and the tone and meaning being open to wider interpretation.

## 3.2 Text Classification

Machine Learning focuses on building systems that can learn from examples. It aims to automate the process of learning in order to make accurate predictions through the use of examples. In relation with NLP, Machine Learning is used to understand the meaning of natural language, therefore, machines have to learn how to do it. One of the examples of how Machine Learning and Natural Language Processing can be leveraged to enable machines to better understand human language is text classification. In text classification, each text document is classified into one or more categories. Since the manual process of categorizing documents can be a laborious task especially if there are several number of documents, Machine Learning automates the process of text classification.

With the aid of Machine Learning, the goal of text classification is to build classifiers by learning the characteristics of the categories from a set of pre-classified documents (Sebastiani, 2002). There are several kinds of classifiers that are suitable for different text classification problems. Therefore, choosing the right classifier is crucial for the performance of the program. The decision criterion of a classifier is learned automatically from the training data. Thus, once the classifier has been trained, it can predict the category of the new data. This approach is also called statistical text classification. Figure 3 illustrates the process of statistical text classification.

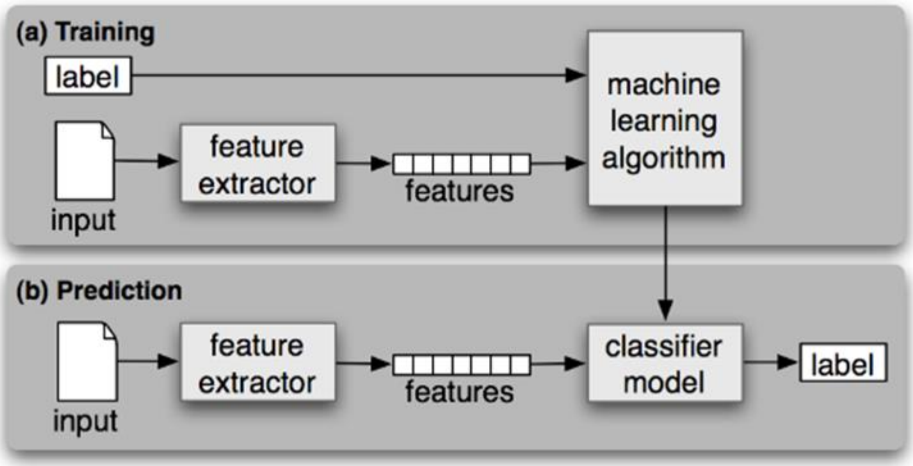


*Figure 3: Statistical Text Classification*

As shown in Figure 3, the process of statistical text classification begins with feature extraction wherein a feature extractor is used to convert each input value to a feature set, which captures the relevant information about each input that will be used in order to classify them. Both features and labels are fed into the machine learning algorithm in order to generate a model. During prediction, the same feature extractor will be used to convert new inputs into feature sets. These feature sets are fed into the model, which in turn, will produce predicted labels.

Some applications of text classification are spam filtering, email routing (Busemann, Schmeier & Arens, 2000), language identification, and genre classification (Litvak & Last, 2008).

### 3.2.1 Machine Learning Algorithms

#### 3.2.1.1 Support Vector Machine

In a machine learning approach to text classification, an algorithm will be used in learning how to classify documents by producing a model to map the input and output. One of the most popular models used in text classification is linear model, which uses the linear combination of feature-values. There are several linear models and one of the most commonly used model is Support Vector Machine (SVM).

Vapnik et al. developed Support Vector Machine, a supervised learning model that is used to analyze data in text classification or regression. It is based on Structural Risk Minimization principle from computational learning theory. SVM performs classification by creating a k-dimensional hyperplane that separates the data into two categories. The number of dimension is equivalent to the number of features an object can possess. In text classification, a feature can be a number of occurrence of particular word in the whole document.

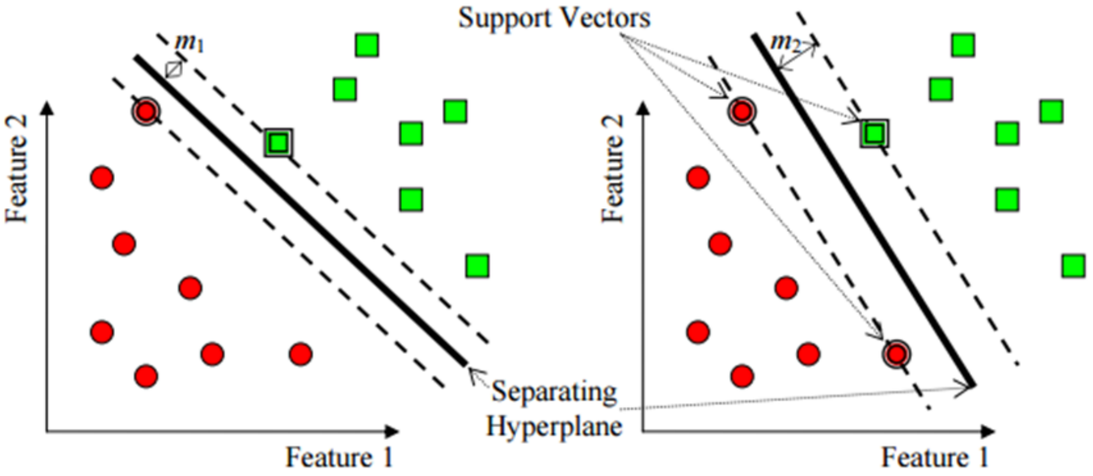


Figure 4: Support Vector Machine with Two Features

In a set of training examples wherein each data has already been labeled, an SVM training algorithm produces a model that will assign new examples to one of the categories which makes it a non-probabilistic binary linear classifier. An SVM model represents the examples (or support vectors) as points in space. SVM seeks to find a line (or hyperplane) that separates the examples based on their labeled classes. The two dashed lines drawn in parallel to the hyperplane represents the distance between the hyperplane and the closest vectors to the line. Moreover, the distance between a dashed line and the hyperplane is called the margin. Thus, whenever a data is added, the side of the hyperplane where it lands will determine the class that will be assigned to it. Figure 4 illustrates how SVM works with two features wherein points are plotted on a 2-dimensional plane.

### 3.2.1.2 Naïve Bayes

Naive Bayes is a group of classification algorithms based on Bayes Theorem. This family of algorithms shares a common principle that every feature is independent of the value of other features regardless of any correlations between them. It assumes that these features independently contribute to the probability that an item belongs to a certain class. Naïve Bayes predicts a class, given a set of features using probability. The principle behind Naïve Bayes rule is that the outcome of a hypothesis (H) can be predicted through the use of some evidences (E) that can be observed from the rule.

$$P(E)=[P(H)*P(H)]/P(E)$$

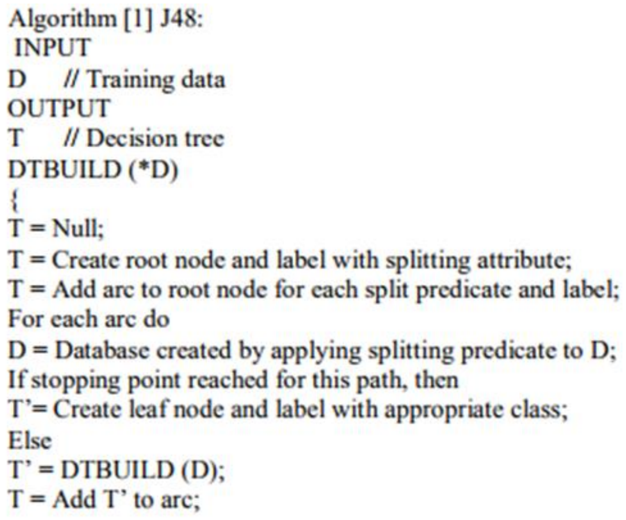
One of the advantages of Naïve Bayes algorithm is that it requires only one pass through the training set to generate a classification model. Moreover, it can be easily trained even with a small dataset. However, since there are cases wherein features are associated with each other, Naïve Bayes may not perform very well in some datasets.

#### 3.2.1.3 J48

J48 is a simple C4.5 decision tree used for classification. It constructs a binary tree. This decision tree approach is deemed useful in dealing with classification problems. The tree will model the process of classifying data. It builds decision trees from a set of labeled training data through information entropy. Moreover, it assumes that each attribute can be used to make a decision by dividing the data into smaller subsets.

J48 examines the normalized information gain that results from choosing an attribute for splitting the data. Thus, the attribute with the highest normalized information gain is crucial in making decisions. Then the algorithm recurs on the smaller subsets. The splitting procedure will stop if all instances in a subset belong to the same class. A leaf node will be created in the decision tree which tells to choose that class.

J48 can handle both continuous and discrete attributes, training data with missing attribute values and attributes with differing costs. It also provides an option for pruning trees after creation.



```
Algorithm [1] J48:
 INPUT
D     // Training data
OUTPUT
T     // Decision tree
DTBUILD (*D)
{
T = Null;
T = Create root node and label with splitting attribute;
T = Add arc to root node for each split predicate and label;
For each arc do
D = Database created by applying splitting predicate to D;
If stopping point reached for this path, then
T'= Create leaf node and label with appropriate class;
Else
T' = DTBUILD (D);
T = Add T' to arc;
```

The main idea is to divide the data into range based on the attribute values for that item that are found in the training sample. It allows classification by using either decision trees or rules that were generated from them.

#### 3.2.1.4 ZeroR

ZeroR is considered as the simplest rule based classifiers which focuses on the target and ignores all predictors. Thus, any rule that works on the non-target attributes will be disregarded. It predicts the majority class by using a frequency table. First, it examines the target attribute and its possible values. Through the use of a frequency table, the most frequent value for a target attribute in a given dataset will be determined. It is specifically used to predict the mean (for a numeric type target attribute) or the mode (for a nominal type attribute). Although ZeroR has no predictability power, it is helpful in determining a baseline performance as a benchmark for other classification methods.

#### 3.2.1.5 Decision Stump

Decision Stump is a machine learning model composed of a one-level decision tree. The tree has one internal node (the root) that is linked to the terminal nodes (its leaves). This model is also called 1-rules because it predicts based on the value of a single input feature (Holte, 1993).

For this model, several variations are possible depending on the type of the input feature. For nominal features, one may build a stump which contains a leaf for each possible feature value or a stump composed of two leaves, one of which corresponds to some chosen category and the other leaf which corresponds to all the other categories. For binary features these two schemes are identical. In addition to this, missing value may be treated as another category. For continuous features, some threshold feature value is selected and the stump contains two leaves — for values that are below and above the threshold.

### 3.2.1.6 Random Tree

Random Tree is an ensemble learning algorithm that constructs several individual learners. In order to produce a random set of data for constructing a decision tree, it employs a bagging idea. This algorithm can be used for both classification and regression purposes. In a Random Forest tree, each node is partitioned using the best among the subset of predicators that were chosen randomly at that node. Random Trees are a collection of tree predictors known as forest. It takes the input feature vector, classifies it with every tree in the forest, and outputs the label that received the majority of "votes". Random Trees are a combination of two existing algorithms in Machine Learning: single model trees and Random Forest. Model trees are decision trees where every single leaf holds a linear model which is optimized for the local subspace described by this leaf. Random Trees have been proven to enhance the performance of single decision tree by constructing two ways of randomization. First, the training data is sampled with replacement for each single tree. Moreover, only a random subset of all attributes is used at every node, and the best split for that subset is computed when growing a tree.

### 3.2.1.7 Random Forest

Random Forest is a collection of simple tree predictors wherein each predictor can produce a response once presented with a set of predictor values. In classification problems, this response may come in the form of a class membership which classifies a set of independent predictor values with one of the categories that are present in the dependent variable. Given a set of simple trees and a set of random predictor variables, it will define a margin function that measures the extent to which the average number of votes for the correct class exceeds the average vote on any other class which are present in the dependent variable. It does not only provide a convenient way of making predictions, but it also provides a way of associating confidence measure along with those predictions.

The mean-square error for a Random Forest is given by:
mean error = (observed - tree response)2

The predictions of the Random Forest are taken to be the average of the predictions of the trees. The formula for Random Forest is as follows wherein the index k runs over the individual trees in the forest:

$$Random\ Forest\ Prediction\ s = \frac{1}{K}\sum_{K=1}^{K} K^{\text{th}}\ tree\ response$$

### 3.2.1.8 REPTree

The Reduced Error Pruning Tree or REPTree is a fast decision tree learner. The goal is to construct a decision tree using information gain and prune it using reduced-error pruning. It utilizes the logic of regression tree and constructs several trees in different iterations then it will select the best among all the trees which will be labelled as the representative. In pruning the tree, the measure used is the mean square error on the predictions made by the tree.

### 3.2.1.9 Decision Table

The Decision Table algorithm summarizes the dataset by using a decision table which is composed of the same number of attributes as the original dataset. A new data item is assigned a category by finding the line in the decision table that matches the non-class values of the data item. It employs the wrapper method in order to find a good subset of attributes that will be included in the table. This algorithm reduces the likelihood of overfitting by eliminating the attributes that does not merely contribute to a test model of the dataset which, in turn, will create a smaller and concise table.

### 3.2.1.10 JRip

JRip (RIPPER) is one of the most simple and popular Machine Learning algorithm. In this algorithm, classes are examined in increasing size and an initial set of rules for the class is constructed through incremental reduced error. Examples of judgments made in the training data are treated as a class and it seeks to find rules that will cover all the members of the class then it will proceed to the next class and repeat the same process. This repetition is done until all classes have been covered.

#### 3.2.1.11 OneR

One Rule or OneR is a simple classification algorithm that is based on one attribute only and it produces a one-level decision tree. It generates one rule for each attribute and selects the rule that will yield the least error rate. However, if there are two or more rules that have the same least error rate, the rule will be selected randomly (Zhao & Zhang, 2007). Rules are created by identifying the most often class, which pertains to the class that appears most frequently for an attribute value. Wolpert and Macready (1995) described OneR as a simple cheap method that can generate good rules for characterizing the structure of data. It often yields a reasonable accuracy on different classification tasks by simply looking at an attribute.

### 3.3 Bag-of-Words (BoW)

In statistical text classification, each input is treated as a feature vector. One of the most common methods used in transforming a text document into a feature vector is through the use of "bag-of-words" representation, in which a set of text documents is converted into a numeric feature vector wherein the order of word occurrences and grammar are ignored. Moreover, it is defined as an order less document representation (Salton & McGill, 1983). In this model, the count of words is given the utmost importance. Each word is represented by a vector of the word counts that appear in the whole document. In this scheme, each individual token occurrence frequency is treated as a feature. Regardless of the simplicity of Bag-of-Words in data representation, it often achieves high performance. (Lewis, 1992).

Once the text has been converted into a BoW model, various measures can be computed to characterize the text. One of the most popular type of features from the BoW model is term-frequency, the number of times a certain term appears in the text. However, term frequency is not considered as the best representation for the text. Oftentimes, insignificant words (such as articles) always yield the highest frequency in the text. These limitations led to the introduction of Term Frequency – Inverse Document Frequency which seeks to diminish the weight of terms that occur very frequently in the document and increases the weight of terms that occur rarely (Jones, 1972).

$$tfidf(t.d.D) = tf(t.d)\ x\ idf(t.D)$$

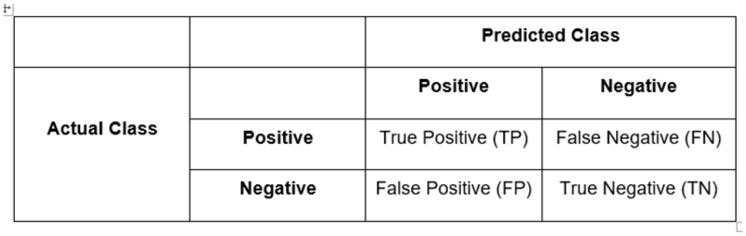
In the concept of TF-IDF, the high weight is conceived by a high frequency and a low term frequency in the whole document. Thus, the weights tend to filter out common terms. The ratio in the idf log function is always higher than or equal to 1, while the value of idf is always higher than or equal to 0. Moreover, when a term appears frequently in the documents, the ratio inside the logarithm approaches 1, bringing the idf and TF-IDF closer to 0 (Josef, 2009).

## 3.4 Performance Measures

Most evaluation for document classifier is conducted experimentally. Thus, it is used to measure its effectiveness or the quality of its predictions on the classification of data. Predictions made are either considered Positive or Negative and expected judgments are called True or False (Pinto, Olieveira & Alves).

As shown in Table 19, a confusion matrix is a table that has two rows and two columns which shows the total number of false positives, false negatives, true positives, and true negatives. Moreover, it allows more detailed analysis than a mere proportion of correct guesses (or accuracy).

- *True positive* refers to the number of examples predicted positive that are actually positive
- *False positive* refers to the number of examples predicted positive that are actually negative
- *True negative* refers to the number of examples predicted negative that are actually negative
- *False negative* refers to the number of examples predicted negative that are actually positive



| | | Predicted Class | |
|---|---|---|---|
| Actual Class | | Positive | Negative |
| | Positive | True Positive (TP) | False Negative (FN) |
| | Negative | False Positive (FP) | True Negative (TN) |

Table 19: Confusion Matrix

### 3.4.1 Precision

Precision is used to measure the exactness of the classifier. Moreover, it refers to the fraction of predicted positive which are actually positive. It is also called positive predictive value (PPV). A high precision indicates less false positives, while a classifier with a low precision means there are more instances of false positives. Precision can be improved by decreasing the recall.

The formula for precision is the number of positive predictions divided by the total number of positive class values predicted.

$$Precision = \frac{TP}{TP + FP}$$

### 3.4.2 Recall

Recall refers to the fraction of those that are actually positive that were predicted as positive. It is used to measure the completeness of a classifier. Moreover, it is also called the true positive rate or sensitivity. Higher recall indicates less instances of false negatives, however, a classifier with lower recall means there are more instances of false negatives. Recall can be improved by decreasing the precision primarily because it is harder to be precise as the number of samples are increasing.

The formula for recall is the number of positive predictions divided by the number of positive class values in the test data.

$$Recall = \frac{TP}{TP + FN}$$

### 3.4.3 Accuracy

The accuracy is the percentage of instances that were correctly classified into their respective classes. It is also called sample accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

One of the disadvantages of accuracy is it can yield to misleading result if the dataset is unbalanced or the number of samples in different classes vary. To illustrate, a model can predict the value of the class with the highest number of samples for all predictions and achieve a high classification accuracy.

### 3.4.4 F-Measure

The F-measure (or F-score) is used to measure the accuracy of the test by considering both precision and recall in computing the score. It conveys balance between precision and recall wherein it reaches its best value at 1 and its worst value at 0.

$$F\ Measure = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}$$

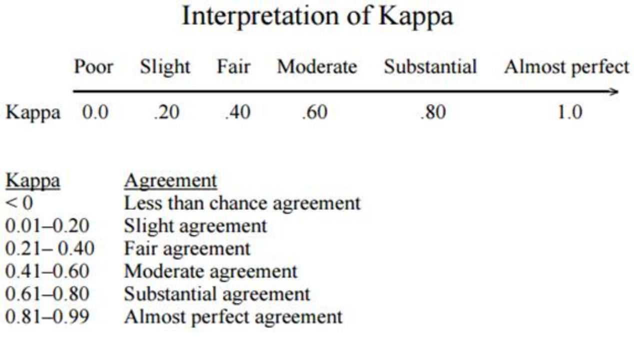
Two of the commonly used F measures are $F_2$ measure and $F_{0.5}$ measure. The $F_2$ measure puts more emphasis on the false negatives by weighing recall higher than precision. $F_{0.5}$ measure puts more emphasis on reducing false negatives by weighing recall lower than precision.

### 3.4.5 Kappa Statistics

Interobserver agreement is a procedure to enhance the believability of data by comparing observations from two or more people who are evaluating the same thing. In evaluating, the observers would agree just by chance. Thus, kappa provides numerical rating of the degree to which this occurs. The calculation is based on the difference between the numbers of agreement that are actually present compared to the numbers of agreement that would be expected to be present by chance.

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

Table 20 illustrates how Kappa measure the differences by standardizing into a -1 to 1 scale.



Interpretation of Kappa

| | Poor | Slight | Fair | Moderate | Substantial | Almost perfect |
|---|---|---|---|---|---|---|
| Kappa | 0.0 | .20 | .40 | .60 | .80 | 1.0 |

| Kappa | Agreement |
|---|---|
| < 0 | Less than chance agreement |
| 0.01–0.20 | Slight agreement |
| 0.21– 0.40 | Fair agreement |
| 0.41–0.60 | Moderate agreement |
| 0.61–0.80 | Substantial agreement |
| 0.81–0.99 | Almost perfect agreement |

Table 20: Kappa Interpretation

# IV. Methodology

## 4.1 System Overview

*Figure 5: Quickgarde Logo*

Quickgarde is a plugin or website extension that can identify possible occurrences of cyberbullying, and subsequently creates a report, regarding the detected post, that is exclusively accessible to some authorized personnel – the person who has the authority to monitor the site. It was designed to work in social media environment, particularly among public conversations expressed in the Filipino language.

## 4.2 System Objectives

### 4.2.1 Main Objective

The aim of the application is to be able to automatically detect, and subsequently flag, statements that imply cyberbullying and produce reports accordingly

### 4.2.2 Specific Objectives

- Identify cyberbullying statements from non-cyberbullying ones in the site
- Flag detected cyberbullying occurrences in the background
- Produce organized reports, that can only be accessed by authorized personnel, in real-time

## 4.3 System Scope and Limitation

Quickgarde is a plugin that can detect, flag, and report cyberbullying occurrences. It covers only harmful posts written in the Filipino language (English and Tagalog). Moreover, it's functionalities are limited only to textual data. The system was programmed in Java. It will be tested in Twitter for the purpose of simulating the plug-in's functionalities.

As for the process of detecting instances of online bullying, the software is incorporated with a cyberbullying detection model which utilizes a Support Vector Machine algorithm deployed through WEKA. Detection is done word per word. Once a cyberbullying post has been detected and flagged, a report regarding the incident will be generated to the authorized personnel. The report is presented in a tabular format composed of the following elements: the post itself, including the name of the person who authored it and the date and time it was written. Non-cyberbullying statements will be disregarded. Procedures to be enforced by the personnel in resolving the issue will no longer be dealt with by the system or the research itself (as it is out of the project's scope).

## 4.4 System Architecture

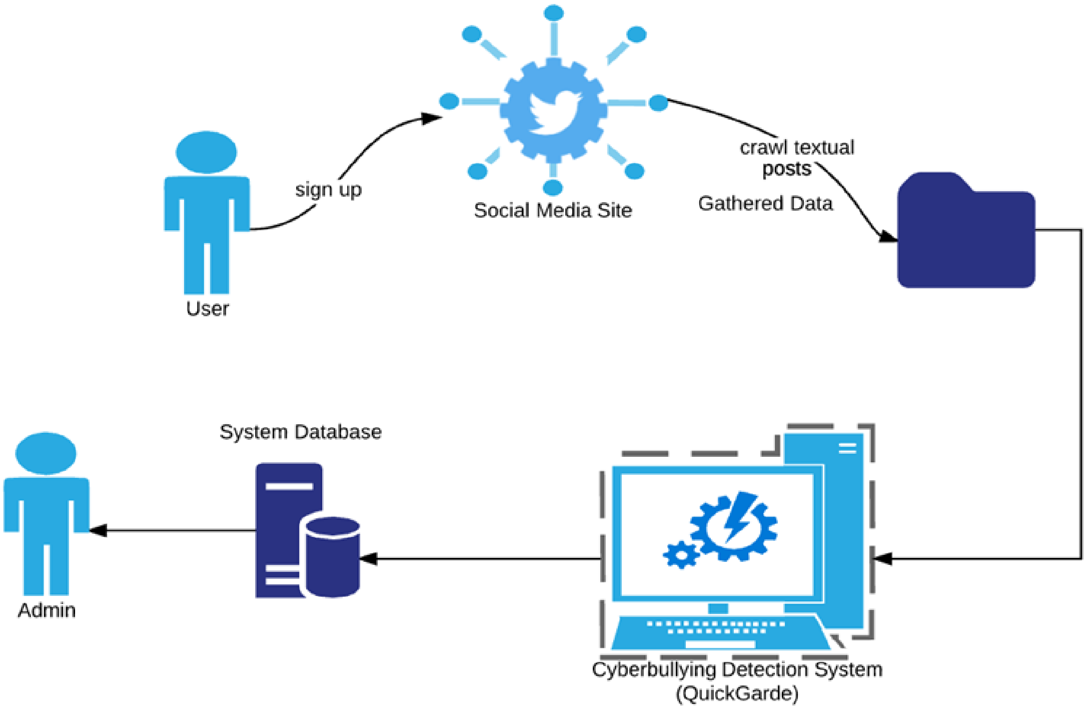


*Figure 6: Quickgarde System Architecture (Generalized)*

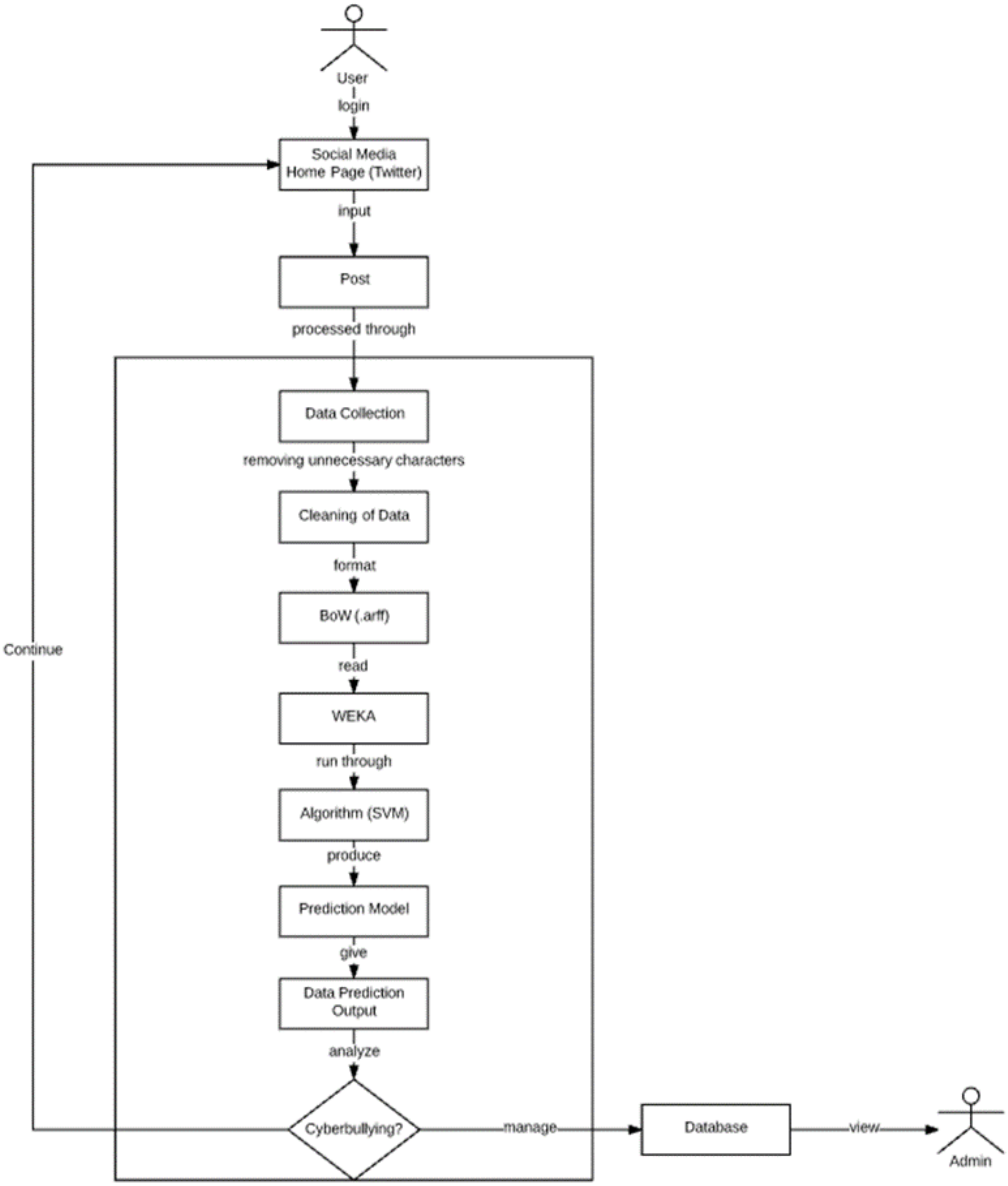


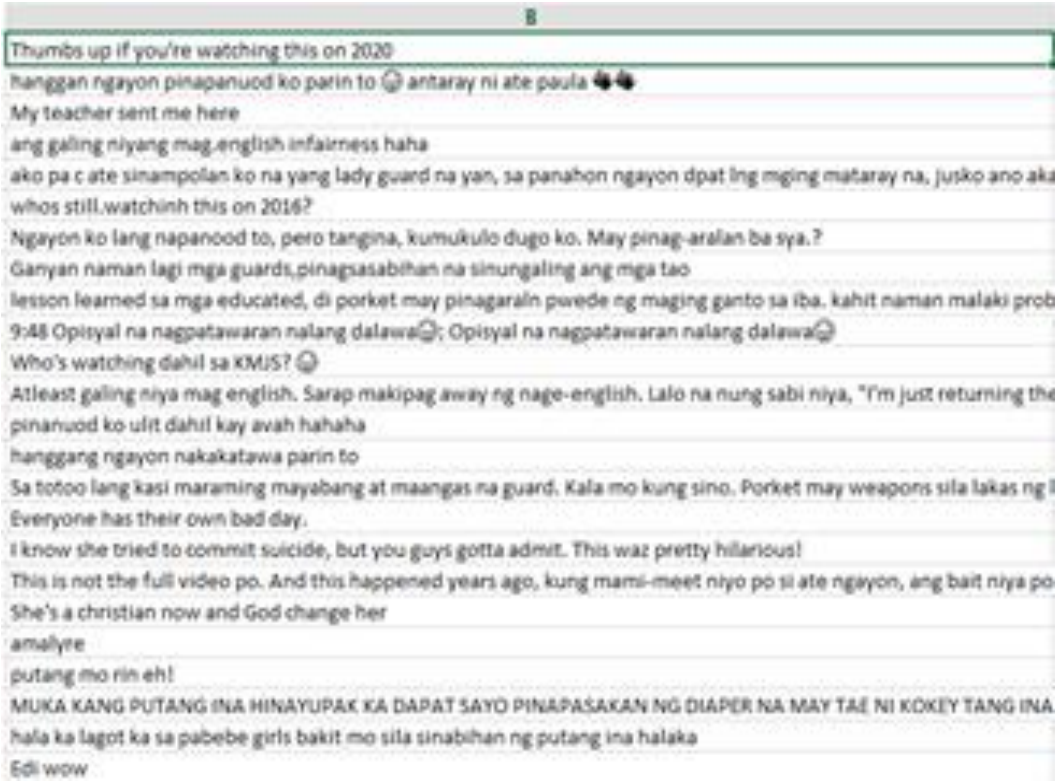
*Figure 7: Quickgarde System Architecture (Expounded)*

### 4.4.1 Data Collection

Social networking sites such as Youtube, Facebook and Twitter were used as sources of data for the corpus. The dataset from Youtube contains comments from videos focusing on controversial events in the Philippines such as cases of bashing against Filipino celebrities and video bloggers, and scandals wherein politicians and celebrities are involved because these topics are often a rich source for objectionable and rude comments (Dinakar, Reichart & Lieberman, 2011).

In Facebook, several posts from the different universities' confession pages were collected because these pages allow anyone to share personal secrets, rumors, gossips, and anything else they might want others to know about but are hesitant to post publicly or in a way that is tied to their identity. Thus, the anonymity of the person posting a confession makes these pages vulnerable to cyber bullying activities. In Twitter, various posts from random Filipino netizens were obtained. Twitter is also prone to cyber bullying attacks since users can easily create fake accounts to launch their bullying cyber-attacks against people they don't like or disagree with. In 2011, a study conducted by the University of Wisconsin-Madison found 15,000 abusive tweets per hour, which equals 100,000 abusive tweets a week.

In acquiring training data from social media posts, Import.io, a web scraping tool, was utilized. It is a tool which allows people to convert unstructured web data into a tabular format and store it in an Excel or CSV file. The only field in the table that was used in collecting data for the corpus was the textual content of the post while the other features such as the user information, links, and others were disregarded. A total number of 2000 statements written in Filipino and English were obtained.

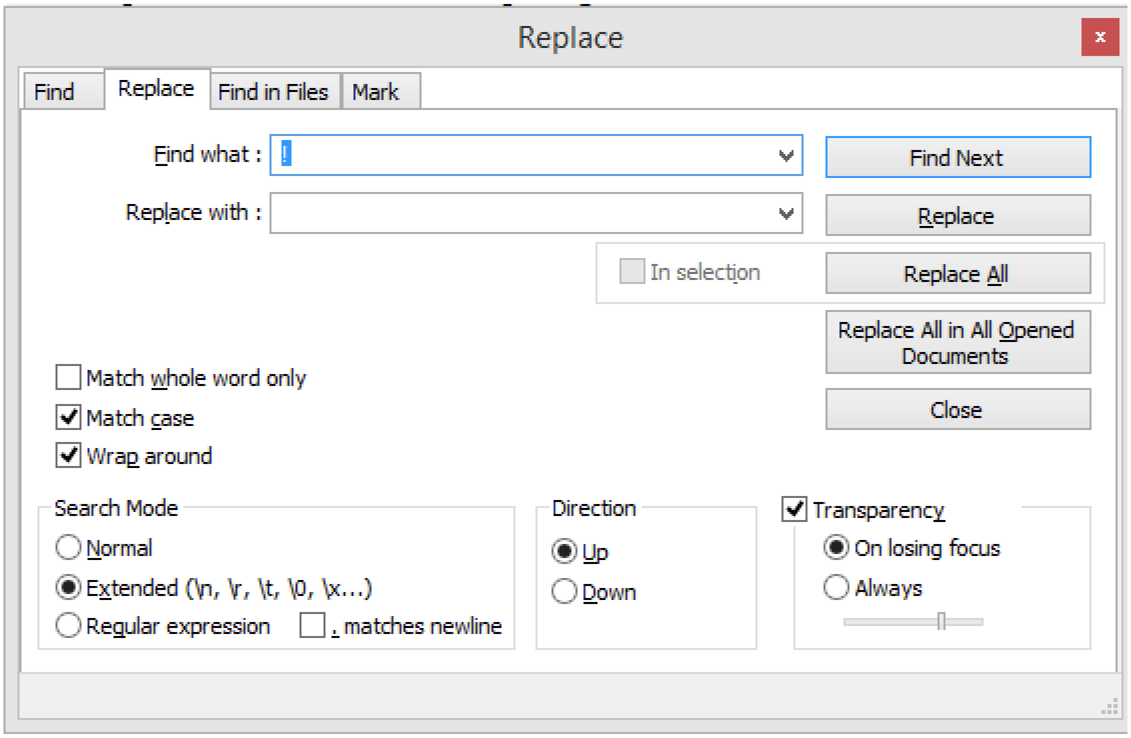


| B |
|---|
| Thumbs up if you're watching this on 2020 |
| hanggan ngayon pinapanuod ko parin to 😂 antaray ni ate paula 👏👏 |
| My teacher sent me here |
| ang galing niyang mag.english infairness haha |
| ako pa c ate sinampolan ko na yang lady guard na yan, sa panahon ngayon dpat lng mging mataray na, jusko ano aka |
| whos still.watchinh this on 2016? |
| Ngayon ko lang napanood to, pero tangina, kumukulo dugo ko. May pinag-aralan ba sya.? |
| Ganyan naman lagi mga guards,pinagsasabihan na sinungaling ang mga tao |
| lesson learned sa mga educated, di porket may pinagaraln pwede ng maging ganto sa iba. kahit naman malaki prob |
| 9:48 Opisyal na nagpatawaran nalang dalawa😂; Opisyal na nagpatawaran nalang dalawa😂 |
| Who's watching dahil sa KMJS? 😂 |
| Atleast galing niya mag english. Sarap makipag away ng nage-english. Lalo na nung sabi niya, "I'm just returning the |
| pinanuod ko ulit dahil kay avah hahaha |
| hanggang ngayon nakakatawa parin to |
| Sa totoo lang kasi maraming mayabang at maangas na guard. Kala mo kung sino. Porket may weapons sila lakas ng l |
| Everyone has their own bad day. |
| I know she tried to commit suicide, but you guys gotta admit. This waz pretty hilarious! |
| This is not the full video po. And this happened years ago, kung mami-meet niyo po si ate ngayon, ang bait niya po |
| She's a christian now and God change her |
| amalyre |
| putang mo rin eh! |
| MUKA KANG PUTANG INA HINAYUPAK KA DAPAT SAYO PINAPASAKAN NG DIAPER NA MAY TAE NI KOKEY TANG INA |
| hala ka lagot ka sa pabebe girls bakit mo sila sinabihan ng putang ina halaka |
| Edi wow |

*Figure 8: Training Data Acquired by Import.io*

### 4.4.2 Cleaning of the Dataset

The cleaning procedure that was applied on the dataset involved the removal of all special characters, non-readable text (e.g. asdfghjkl), emoticons, links, and characters belonging to various foreign countries' writing systems. This was done in order to prevent complications from arising particularly during the experimental phase of the project. Such characters do not make any sense with regard to the detection of cyberbullying occurrences, therefore their appearance may contribute to a probable decrease in the accuracy rate of the model. Basic Jejemon slang was likewise included in the dataset. Since the presence of distinct features were used as basis for the frequency of each word in every statement, it is important to include all words preserved in forms understandable by Filipinos within the dataset. This procedure was done using regular expressions.



Replace

Find | Replace | Find in Files | Mark

Find what : !

Replace with :

□ In selection

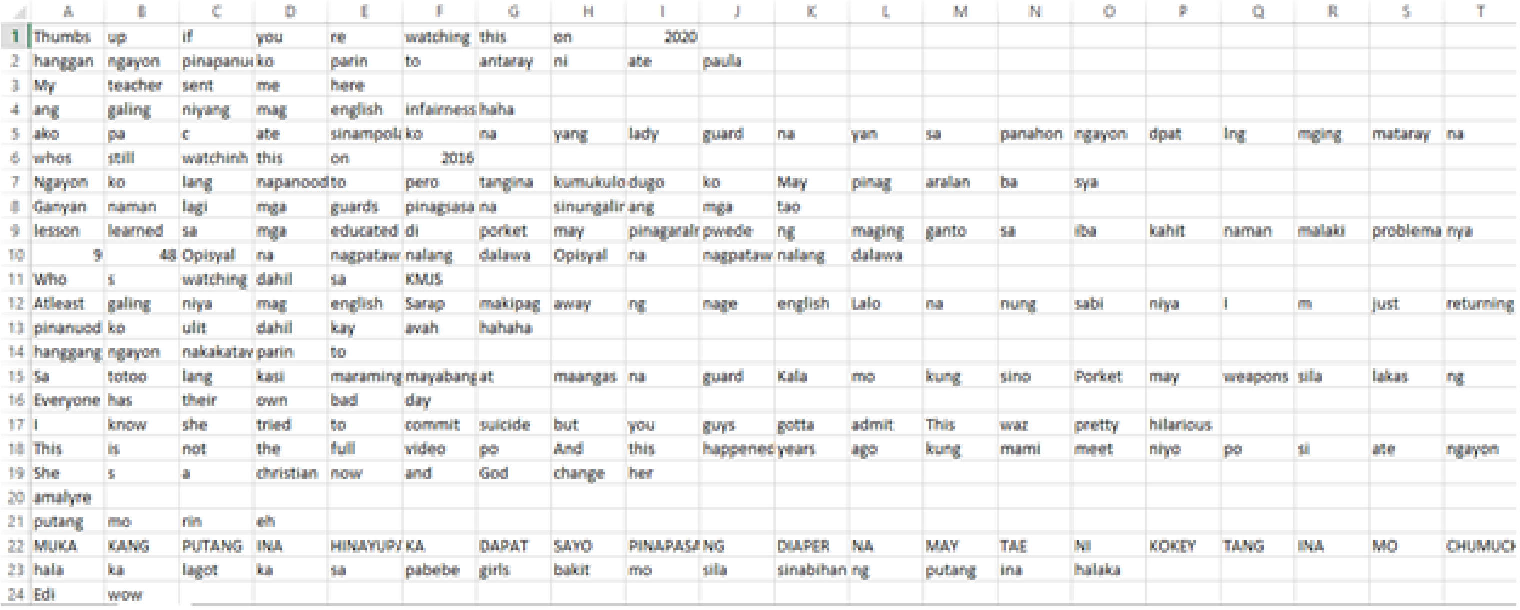
□ Match whole word only
☑ Match case
☑ Wrap around

Search Mode
○ Normal
● Extended (\n, \r, \t, \0, \x...)
○ Regular expression □ . matches newline

Direction
● Up
○ Down

☑ Transparency
● On losing focus
○ Always

*Figure 9: Using Regular Expressions in the Removal of Unnecessary Characters*

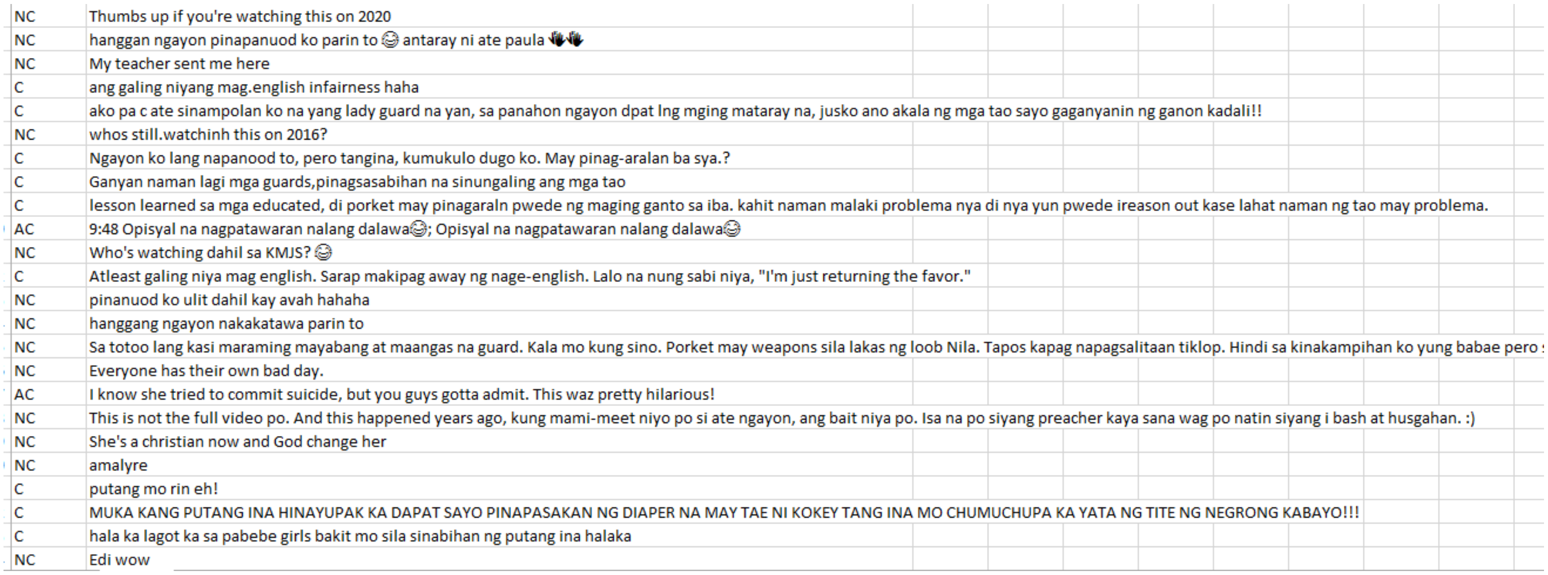


| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Thumbs | up | if | you | re | watching | this | on | 2020 | | | | | | | | | | | |
| 2 | hanggan | ngayon | pinapanu | ko | parin | to | antaray | ni | ate | paula | | | | | | | | | | |
| 3 | My | teacher | sent | me | here | | | | | | | | | | | | | | | |
| 4 | ang | galing | niyang | mag | english | infairness | haha | | | | | | | | | | | | | |
| 5 | ako | pa | c | ate | sinampol | ko | na | yang | lady | guard | na | yan | sa | panahon | ngayon | dpat | lng | mging | mataray | na |
| 6 | whos | still | watchinh | this | on | 2016 | | | | | | | | | | | | | | |
| 7 | Ngayon | ko | lang | napanood | to | pero | tangina | kumukulo | dugo | ko | May | pinag | aralan | ba | sya | | | | | |
| 8 | Ganyan | naman | lagi | mga | guards | pinagsasa | na | sinungalir | ang | mga | tao | | | | | | | | | |
| 9 | lesson | learned | sa | mga | educated | di | porket | may | pinagaralr | pwede | ng | maging | ganto | sa | iba | kahit | naman | malaki | problema | nya |
| 10 | 9 | 48 | Opisyal | na | nagpataw | nalang | dalawa | Opisyal | na | nagpataw | nalang | dalawa | | | | | | | | |
| 11 | Who | s | watching | dahil | sa | KMJS | | | | | | | | | | | | | | |
| 12 | Atleast | galing | niya | mag | english | Sarap | makipag | away | ng | nage | english | Lalo | na | nung | sabi | niya | I | m | just | returning |
| 13 | pinanuod | ko | ulit | dahil | kay | avah | hahaha | | | | | | | | | | | | | |
| 14 | hanggang | ngayon | nakakatav | parin | to | | | | | | | | | | | | | | | |
| 15 | Sa | totoo | lang | kasi | maraming | mayabang | at | maangas | na | guard | Kala | mo | kung | sino | Porket | may | weapons | sila | lakas | ng |
| 16 | Everyone | has | their | own | bad | day | | | | | | | | | | | | | | |
| 17 | I | know | she | tried | to | commit | suicide | but | you | guys | gotta | admit | This | waz | pretty | hilarious | | | | |
| 18 | This | is | not | the | full | video | po | And | this | happenec | years | ago | kung | mami | meet | niyo | po | si | ate | ngayon |
| 19 | She | s | a | christian | now | and | God | change | her | | | | | | | | | | | |
| 20 | amalyre | | | | | | | | | | | | | | | | | | | |
| 21 | putang | mo | rin | eh | | | | | | | | | | | | | | | | |
| 22 | MUKA | KANG | PUTANG | INA | HINAYUP/ | KA | DAPAT | SAYO | PINAPASA | NG | DIAPER | NA | MAY | TAE | NI | KOKEY | TANG | INA | MO | CHUMUCH |
| 23 | hala | ka | lagot | ka | sa | pabebe | girls | bakit | mo | sila | sinabihan | ng | putang | ina | halaka | | | | | |
| 24 | Edi | wow | | | | | | | | | | | | | | | | | | |

*Figure 10: Cleaned Dataset*

### 4.4.3 Data Annotation

Once the preprocessing steps were accomplished, the dataset was further subjected to annotation. For this step, each data was classified into three labels: Cyberbullying, Non-Cyberbullying, and Ambiguous Cyberbullying (a case wherein the annotator was unable to identify whether a certain post implies cyberbullying or not). For this process, 100 questionnaires (that contains 10 sentences (with a total number of 2000 statements) taken from the corpus were distributed among Metro Manila citizens. The participants will manually label each data into three categories. Furthermore, the labeled data will be used in training the classifier.

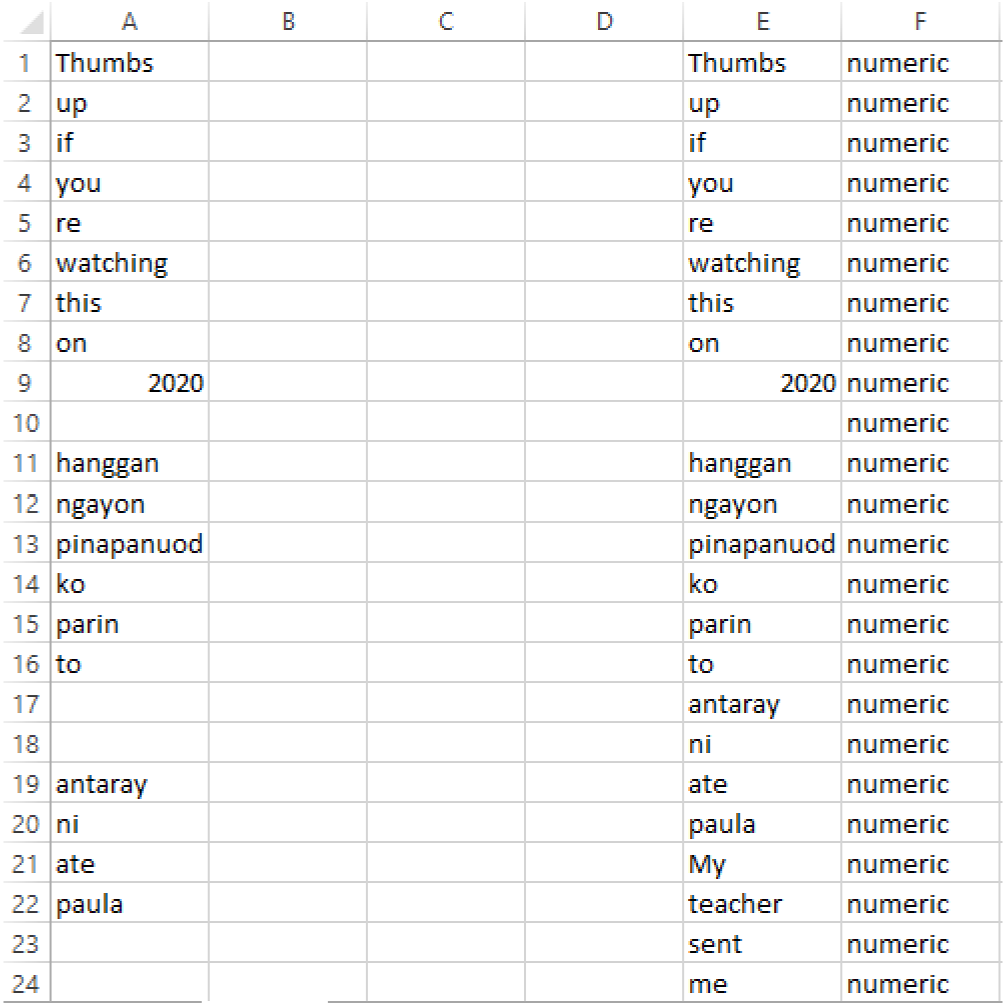


| | |
|---|---|
| NC | Thumbs up if you're watching this on 2020 |
| NC | hanggan ngayon pinapanuod ko parin to 😂 antaray ni ate paula 🙌🙌 |
| NC | My teacher sent me here |
| C | ang galing niyang mag.english infairness haha |
| C | ako pa c ate sinampolan ko na yang lady guard na yan, sa panahon ngayon dpat lng mging mataray na, jusko ano akala ng mga tao sayo gaganyanin ng ganon kadali!! |
| NC | whos still.watchinh this on 2016? |
| C | Ngayon ko lang napanood to, pero tangina, kumukulo dugo ko. May pinag-aralan ba sya.? |
| C | Ganyan naman lagi mga guards,pinagsasabihan na sinungaling ang mga tao |
| C | lesson learned sa mga educated, di porket may pinagaraln pwede ng maging ganto sa iba. kahit naman malaki problema nya di nya yun pwede ireason out kase lahat naman ng tao may problema. |
| AC | 9:48 Opisyal na nagpatawaran nalang dalawa😂; Opisyal na nagpatawaran nalang dalawa😂 |
| NC | Who's watching dahil sa KMJS? 😂 |
| C | Atleast galing niya mag english. Sarap makipag away ng nage-english. Lalo na nung sabi niya, "I'm just returning the favor." |
| NC | pinanuod ko ulit dahil kay avah hahaha |
| NC | hanggang ngayon nakakatawa parin to |
| NC | Sa totoo lang kasi maraming mayabang at maangas na guard. Kala mo kung sino. Porket may weapons sila lakas ng loob Nila. Tapos kapag napagsalitaan tiklop. Hindi sa kinakampihan ko yung babae pero |
| NC | Everyone has their own bad day. |
| AC | I know she tried to commit suicide, but you guys gotta admit. This waz pretty hilarious! |
| NC | This is not the full video po. And this happened years ago, kung mami-meet niyo po si ate ngayon, ang bait niya po. Isa na po siyang preacher kaya sana wag po natin siyang i bash at husgahan. :) |
| NC | She's a christian now and God change her |
| NC | amalyre |
| C | putang mo rin eh! |
| C | MUKA KANG PUTANG INA HINAYUPAK KA DAPAT SAYO PINAPASAKAN NG DIAPER NA MAY TAE NI KOKEY TANG INA MO CHUMUCHUPA KA YATA NG TITE NG NEGRONG KABAYO!!! |
| C | hala ka lagot ka sa pabebe girls bakit mo sila sinabihan ng putang ina halaka |
| NC | Edi wow |

*Figure 11: Annotated Dataset*

### 4.4.4 Tokenization

In this phase, all of the statements that were cleaned will be divided per word within a particular statement based on the white spaces separating them. This function will help provide each distinct occurrence of all the words that were part of the statements stored within the corpus. Once this process has been accomplished, it will determine the number of occurrences (frequency) of each feature as they occur in every statement. The acquired numerical values will then be used in the implementation of the Bag-of-Words.

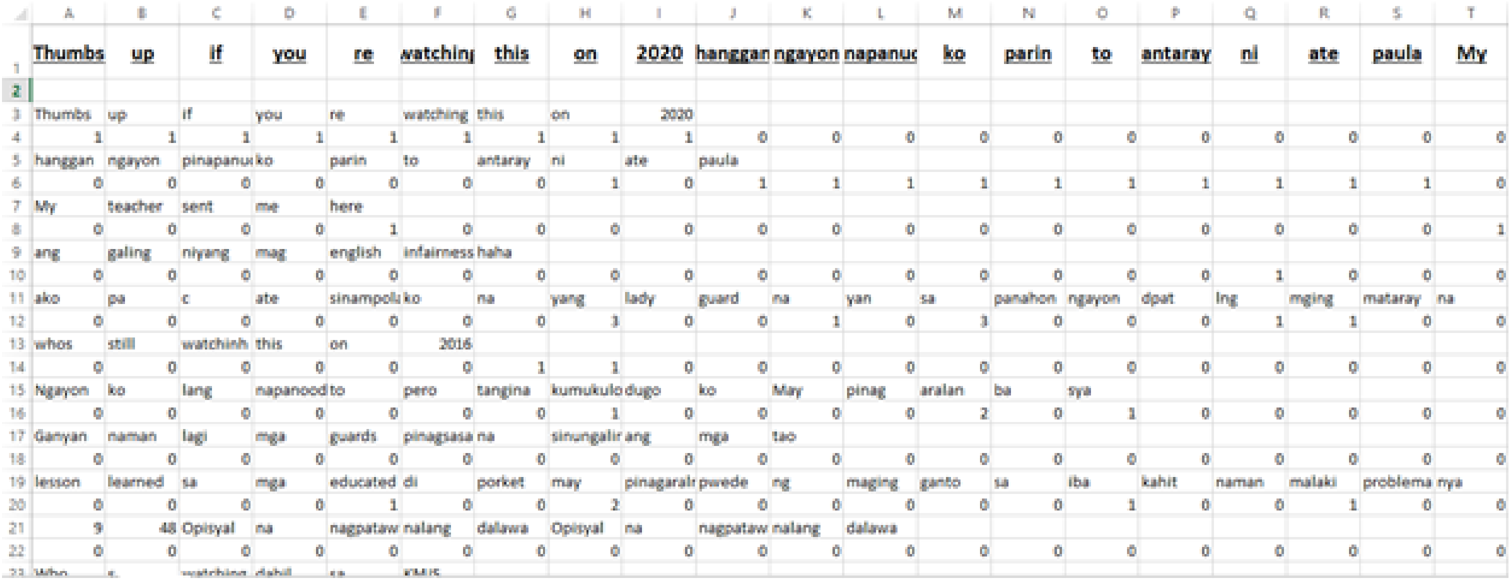


| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Thumbs | | | | Thumbs | numeric |
| 2 | up | | | | up | numeric |
| 3 | if | | | | if | numeric |
| 4 | you | | | | you | numeric |
| 5 | re | | | | re | numeric |
| 6 | watching | | | | watching | numeric |
| 7 | this | | | | this | numeric |
| 8 | on | | | | on | numeric |
| 9 | 2020 | | | | 2020 | numeric |
| 10 | | | | | | numeric |
| 11 | hanggan | | | | hanggan | numeric |
| 12 | ngayon | | | | ngayon | numeric |
| 13 | pinapanuod | | | | pinapanuod | numeric |
| 14 | ko | | | | ko | numeric |
| 15 | parin | | | | parin | numeric |
| 16 | to | | | | to | numeric |
| 17 | | | | | antaray | numeric |
| 18 | | | | | ni | numeric |
| 19 | antaray | | | | ate | numeric |
| 20 | ni | | | | paula | numeric |
| 21 | ate | | | | My | numeric |
| 22 | paula | | | | teacher | numeric |
| 23 | | | | | sent | numeric |
| 24 | | | | | me | numeric |

*Figure 12: Tokenized Dataset*

### 4.4.5 Bag-of-Words (BoW)

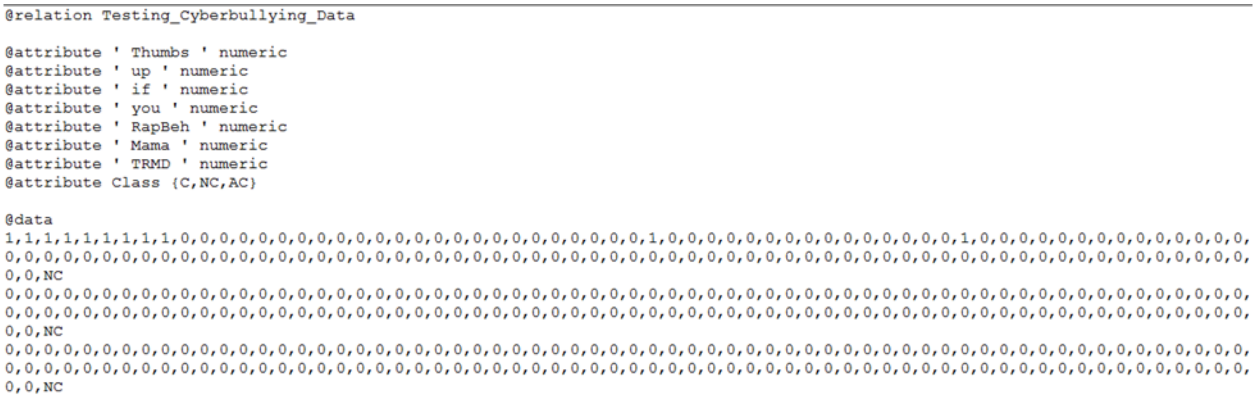
The dataset was transformed into a Bag-of-Words model, in which a set of text documents is converted into a numeric feature vector wherein the order of word occurrences and grammar are ignored. It is primarily used as a tool of feature generation. The process begins by creating a list of unique words from the text. Once a list has been created, the number of times a word appears in a document will be computed. From the Bag-of-Words all the words that contained digits, were removed.



|  | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Thumbs** | **up** | **If** | **you** | **re** | **watchin** | **this** | **on** | **2020** | **hanggan** | **ngayon** | **napanuc** | **ko** | **parin** | **to** | **antaray** | **ni** | **ate** | **paula** | **My** |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Thumbs | up | if | you | re | watching | this | on | 2020 |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | hanggan | ngayon | pinapanu | ko | parin | to | antaray | ni | ate | paula |  |  |  |  |  |  |  |  |  |  |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 7 | My | teacher | sent | me | here |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 9 | ang | galing | niyang | mag | english | infairness | haha |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 11 | ako | pa | c | ate | sinampol | ko | na | yang | lady | guard | na | yan | sa | panahon | ngayon | dpat | lng | mging | mataray | na |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 13 | whos | still | watchinh | this | on | 2016 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | Ngayon | ko | lang | napanood | to | pero | tangina | kumukulo | dugo | ko | May | pinag | aralan | ba | sya |  |  |  |  |  |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 17 | Ganyan | naman | lagi | mga | guards | pinagsasa | na | sinungalir | ang | mga | tao |  |  |  |  |  |  |  |  |  |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | lesson | learned | sa | mga | educated | di | porket | may | pinagaral | pwede | ng | maging | ganto | sa | iba | kahit | naman | malaki | problema | nya |
| 20 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 21 | 9 | 48 | Opisyal | na | nagpataw | nalang | dalawa | Opisyal | na | nagpataw | nalang | dalawa |  |  |  |  |  |  |  |  |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | Who | s | watching | dahil | sa | KMJS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

*Figure 13: Bag-of_Words in CSV File Format*

After cleaning the dataset, the csv (comma-separated values) file was converted into .arff (Attribute-Relation File Format) format since it is the one being used in WEKA. In this format, the distinct features will be represented by the attributes, and the relation as the whole corpus itself. At the bottom part of the file, the number of occurrences (of each word in every statement) along with the annotations placed by both the researchers and their correspondents (in every statement), will be placed. Such data initially came from the .csv file containing the cleaned, parsed, and evaluated words comprising each of the 2000 statements.



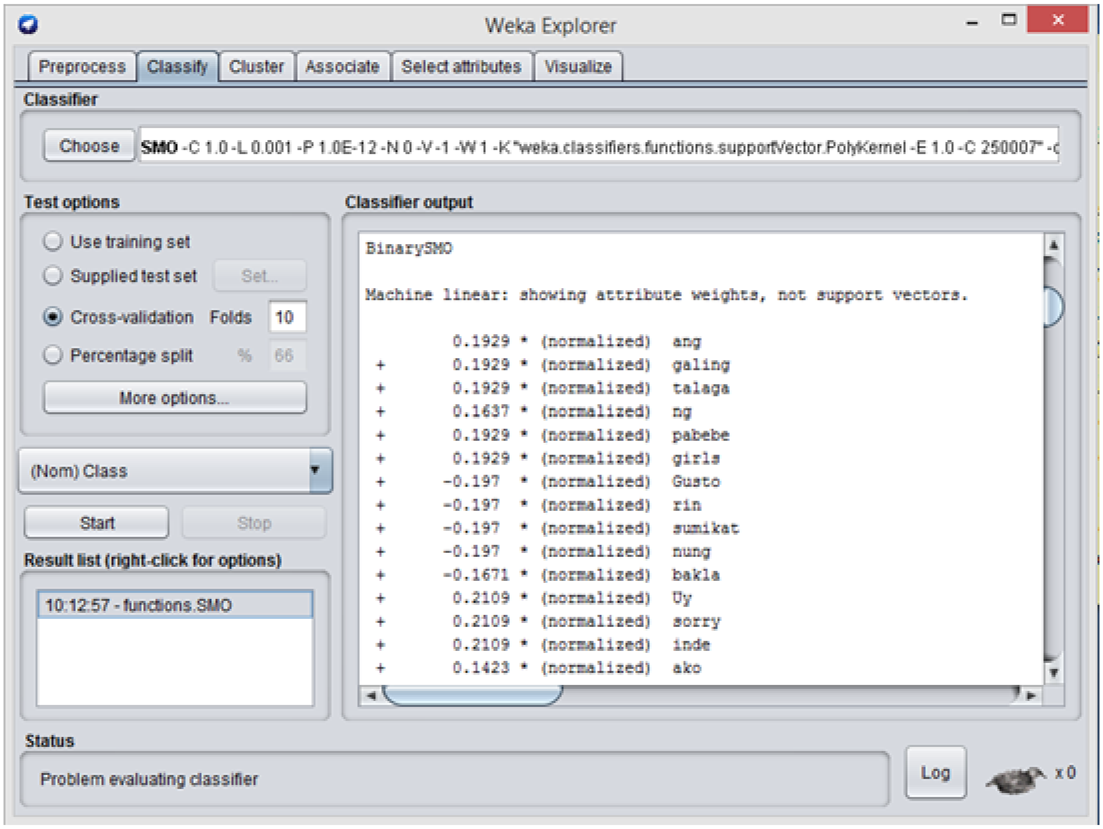
```
@relation Testing_Cyberbullying_Data

@attribute ' Thumbs ' numeric
@attribute ' up ' numeric
@attribute ' if ' numeric
@attribute ' you ' numeric
@attribute ' RapBeh ' numeric
@attribute ' Mama ' numeric
@attribute ' TRMD ' numeric
@attribute Class {C,NC,AC}

@data
1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,NC
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,NC
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,NC
```

*Figure 14: Bag-of-Words in ARFF File Format*

### 4.4.6 Support Vector Machine

Classification is the task of identifying the label for a single entity from a set of data. In order to determine cyberbullying from not-cyberbullying data, an SVM classifier was trained on a set of labeled data. Thus, these words are essentially treated as features that the classifier will use to model the positive instances of cyberbullying as compared to non-cyberbullying and ambiguous cyberbullying.

The Support Vector Machine algorithm was the only text classification algorithm that was used in the research project. It was implemented in the WEKA toolkit, a data processing and machine learning tool.



Classifier

SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -c

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) Class

Result list (right-click for options)

10:12:57 - functions.SMO

Classifier output

```
BinarySMO

Machine linear: showing attribute weights, not support vectors.

         0.1929 * (normalized)  ang
 +       0.1929 * (normalized)  galing
 +       0.1929 * (normalized)  talaga
 +       0.1637 * (normalized)  ng
 +       0.1929 * (normalized)  pabebe
 +       0.1929 * (normalized)  girls
 +      -0.197  * (normalized)  Gusto
 +      -0.197  * (normalized)  rin
 +      -0.197  * (normalized)  sumikat
 +      -0.197  * (normalized)  nung
 +      -0.1671 * (normalized)  bakla
 +       0.2109 * (normalized)  Uy
 +       0.2109 * (normalized)  sorry
 +       0.2109 * (normalized)  inde
 +       0.1423 * (normalized)  ako
```

Status

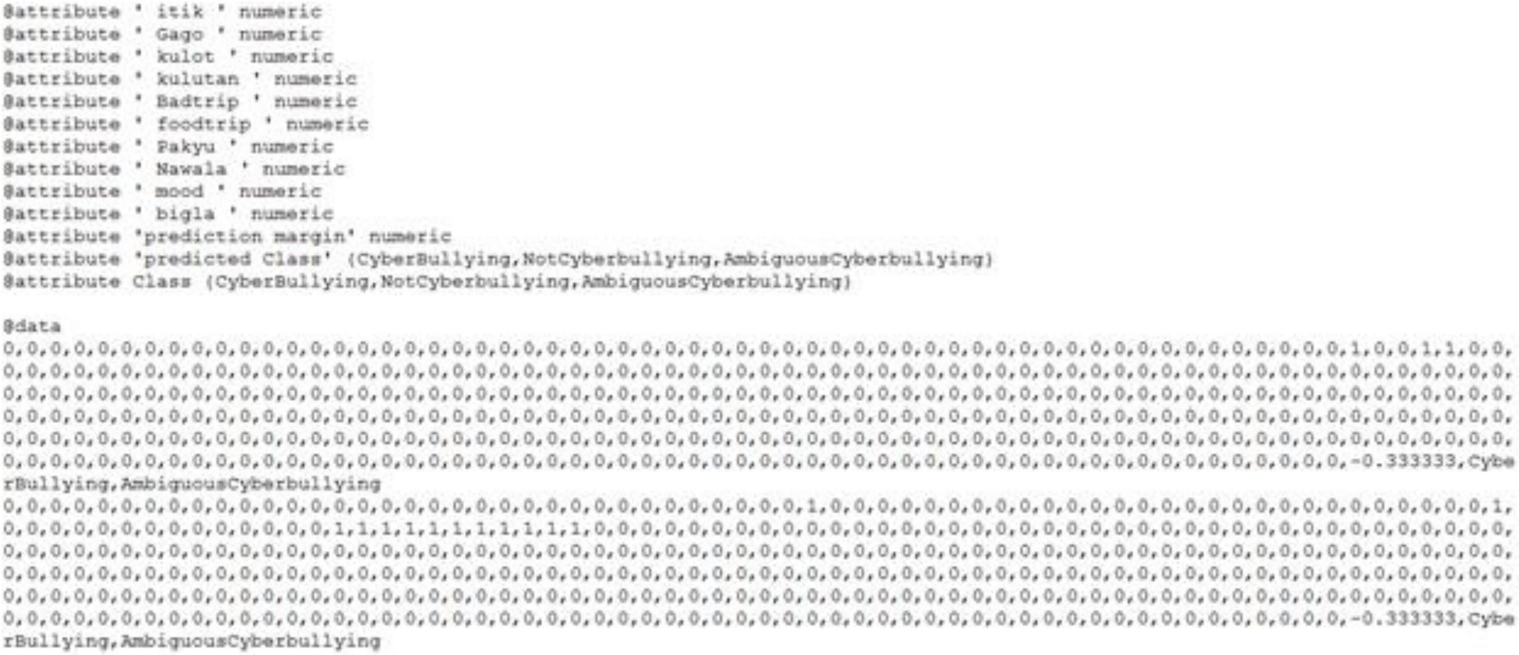
Problem evaluating classifier

*Figure 15: SVM in the process of creating the model*

### 4.4.7 Cyberbullying Detection Model

Among the 2000 statements, a total number of 900 was used for this experiment. The sole experiment that was performed involved the use of the Support Vector Machine (SVM) algorithm on the 2000 statements.

In this phase, the algorithm will be implemented together with the processed data in WEKA. The flagging of cyberbullying statements takes place in this phase. There will be charts that the tool will present to indicate how it classified a particular statement. This step verifies the model's capability of classifying the statements. It is now ready for the next phase – getting hard-coded using the Java programming language.



```
@attribute ' itik ' numeric
@attribute ' Gago ' numeric
@attribute ' kulot ' numeric
@attribute ' kulutan ' numeric
@attribute ' Badtrip ' numeric
@attribute ' foodtrip ' numeric
@attribute ' Pakyu ' numeric
@attribute ' Nawala ' numeric
@attribute ' mood ' numeric
@attribute ' bigla ' numeric
@attribute 'prediction margin' numeric
@attribute 'predicted Class' (CyberBullying,NotCyberbullying,AmbiguousCyberbullying)
@attribute Class (CyberBullying,NotCyberbullying,AmbiguousCyberbullying)

@data
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,1,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-0.333333,Cybe
rBullying,AmbiguousCyberbullying
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-0.333333,Cybe
rBullying,AmbiguousCyberbullying
```

*Figure 16: Classification Results Produced by the Model*

## 4.5 System Functionalities

### 5.4.1 System Requirements

Quickgarde Requirements

- Operating System - Windows, Mac OS, any flavor of Linux that supports Java
- Java Virtual Machine (Java 8 or later) – Weka API, SMO API (for Support Vector Machine), and Twitter4j API 4.0.5 SNAPSHOTS

- Database – MySQL 6.3
- Internet connection - 25mbps (equivalent to normal internet speed connection for download) or highest
- Browser Requirements - Firefox (latest version), Chrome (latest version), Safari (latest version), Internet Explorer 10 or newer
- Social media platform (any) - Facebook, Twitter, etc.

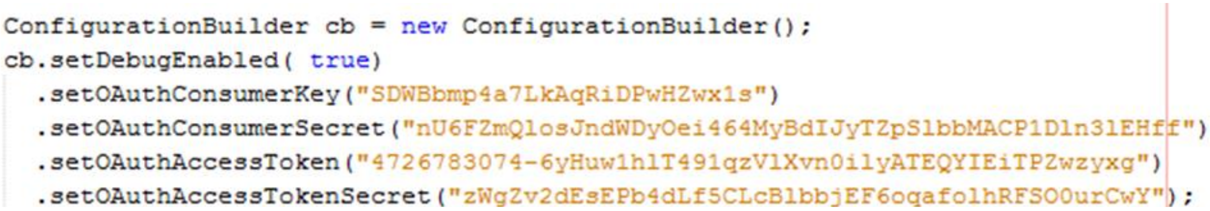
### 5.4.2 Features and Functions

This section presents an in-depth description of the five main features of Quickgarde, along with the functions used (per feature).

#### 5.4.2.1 Gathering of Public Textual Posts

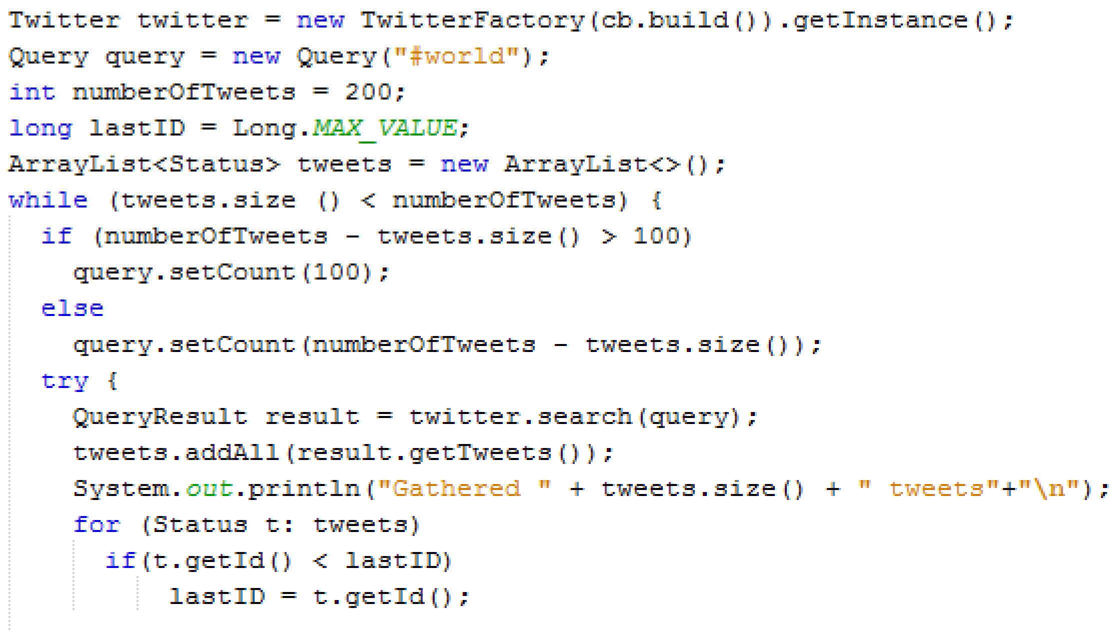
The application will acquire data directly from a social media site. In this project, Twitter will be the platform to be used since it has complete documentation regarding the methods on interacting with its API. A tool known as Twitter4J will be used to gather the tweets and respective information about them. Twitter4J provides a way for developers to integrate their Java application to the Twitter service.

With the aid of Twitter4J, public Twitter posts will be acquired from a Twitter user's timeline automatically and in real-time. Twitter4J does this by first interacting with Twitter's API and then authenticating Quickgarde's access by providing the necessary Twitter API OAuth tokens - which can be acquired via registering the application in Twitter's Application Management webpage. Logging in to an existing Twitter account will suffice for the registration of the application. After successfully providing the needed authentication, Twitter4J can now make use of its built-in functions to acquire data provided by Twitter API. Posts that were gathered will then be added to the corpus.



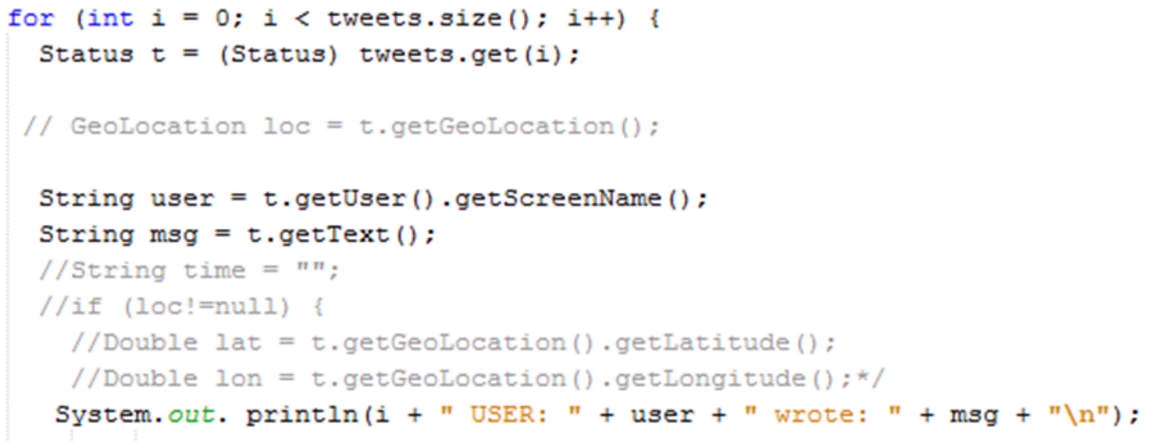
```
ConfigurationBuilder cb = new ConfigurationBuilder();
cb.setDebugEnabled( true)
  .setOAuthConsumerKey("SDWBbmp4a7LkAqRiDPwHZwx1s")
  .setOAuthConsumerSecret("nU6FZmQlosJndWDyOei464MyBdIJyTZpSlbbMACP1Dln3lEHff")
  .setOAuthAccessToken("4726783074-6yHuw1hlT491qzVlXvn0ilyATEQYIEiTPZwzyxg")
  .setOAuthAccessTokenSecret("zWgZv2dEsEPb4dLf5CLcBlbbjEF6oqafolhRFSO0urCwY");
```

Figure 17: Twitter Account Configuration



```
Twitter twitter = new TwitterFactory(cb.build()).getInstance();
Query query = new Query("#world");
int numberOfTweets = 200;
long lastID = Long.MAX_VALUE;
ArrayList<Status> tweets = new ArrayList<>();
while (tweets.size () < numberOfTweets) {
  if (numberOfTweets - tweets.size() > 100)
    query.setCount(100);
  else
    query.setCount(numberOfTweets - tweets.size());
  try {
    QueryResult result = twitter.search(query);
    tweets.addAll(result.getTweets());
    System.out.println("Gathered " + tweets.size() + " tweets"+"\n");
    for (Status t: tweets)
      if(t.getId() < lastID)
          lastID = t.getId();
```

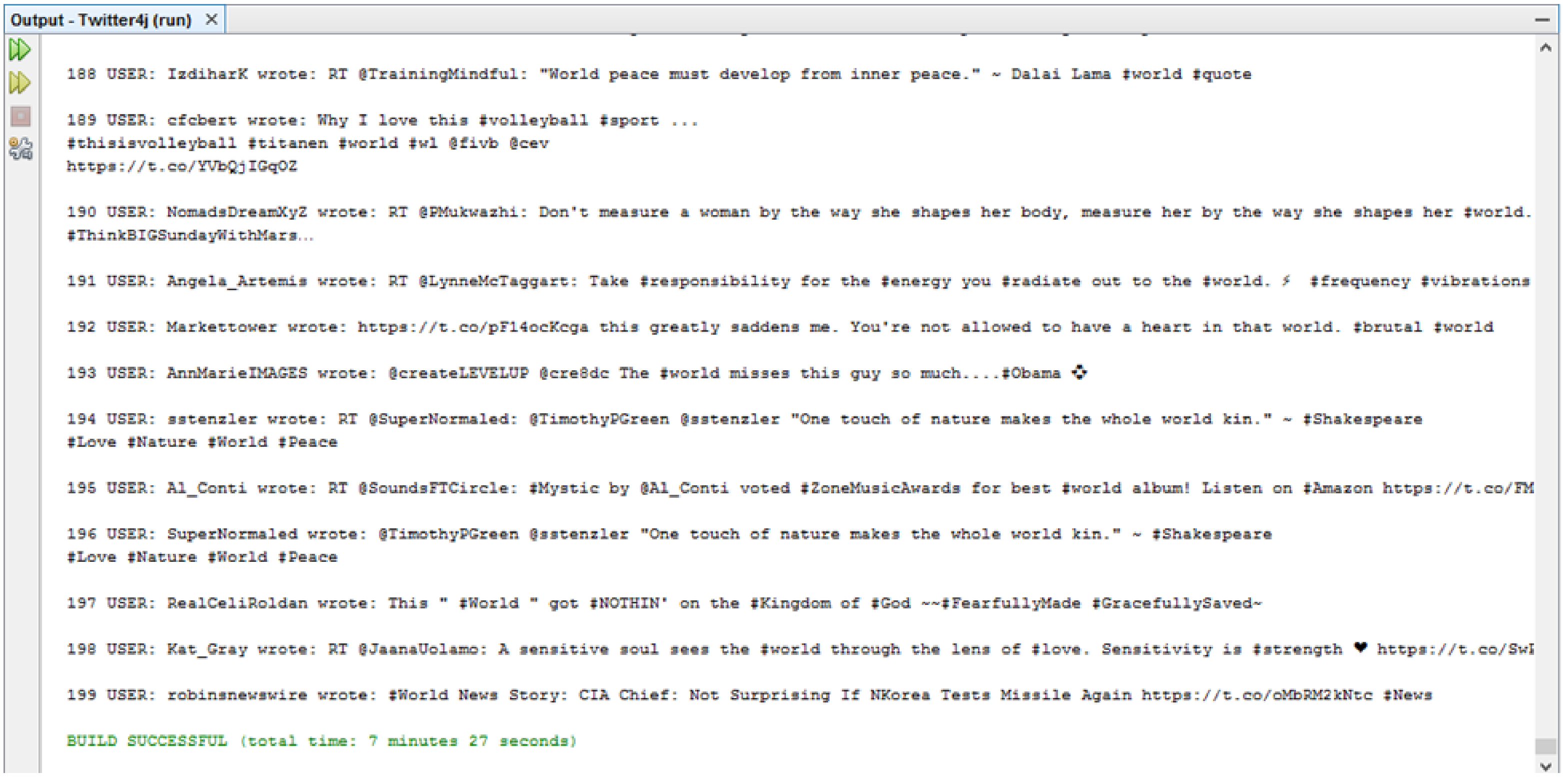
Figure 18: Setting the Number of Tweets to be Gathered



```
for (int i = 0; i < tweets.size(); i++) {
  Status t = (Status) tweets.get(i);

 // GeoLocation loc = t.getGeoLocation();

  String user = t.getUser().getScreenName();
  String msg = t.getText();
  //String time = "";
  //if (loc!=null) {
    //Double lat = t.getGeoLocation().getLatitude();
    //Double lon = t.getGeoLocation().getLongitude();*/
   System.out. println(i + " USER: " + user + " wrote: " + msg + "\n");
```

Figure 19: Outputting the Acquired Posts

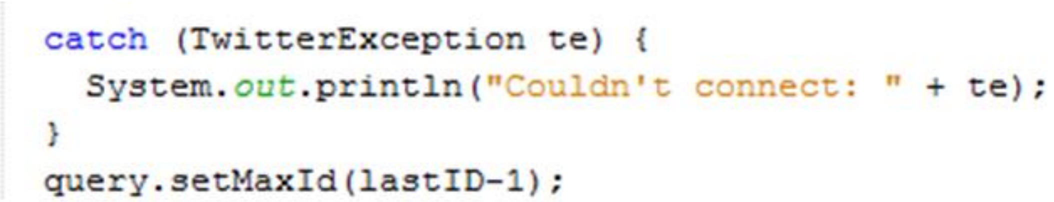
Result:



```
188 USER: IzdiharK wrote: RT @TrainingMindful: "World peace must develop from inner peace." ~ Dalai Lama #world #quote

189 USER: cfcbert wrote: Why I love this #volleyball #sport ...
#thisisvolleyball #titanen #world #wl @fivb @cev
https://t.co/YVbQjIGqOZ

190 USER: NomadsDreamXyZ wrote: RT @PMukwazhi: Don't measure a woman by the way she shapes her body, measure her by the way she shapes her #world.
#ThinkBIGSundayWithMars…

191 USER: Angela_Artemis wrote: RT @LynneMcTaggart: Take #responsibility for the #energy you #radiate out to the #world. ⚡  #frequency #vibrations

192 USER: Markettower wrote: https://t.co/pF14ocKcga this greatly saddens me. You're not allowed to have a heart in that world. #brutal #world

193 USER: AnnMarieIMAGES wrote: @createLEVELUP @cre8dc The #world misses this guy so much....#Obama ❖

194 USER: sstenzler wrote: RT @SuperNormaled: @TimothyPGreen @sstenzler "One touch of nature makes the whole world kin." ~ #Shakespeare
#Love #Nature #World #Peace

195 USER: Al_Conti wrote: RT @SoundsFTCircle: #Mystic by @Al_Conti voted #ZoneMusicAwards for best #world album! Listen on #Amazon https://t.co/FM

196 USER: SuperNormaled wrote: @TimothyPGreen @sstenzler "One touch of nature makes the whole world kin." ~ #Shakespeare
#Love #Nature #World #Peace

197 USER: RealCeliRoldan wrote: This " #World " got #NOTHIN' on the #Kingdom of #God ~~#FearfullyMade #GracefullySaved~

198 USER: Kat_Gray wrote: RT @JaanaUolamo: A sensitive soul sees the #world through the lens of #love. Sensitivity is #strength ♥ https://t.co/SwI

199 USER: robinsnewswire wrote: #World News Story: CIA Chief: Not Surprising If NKorea Tests Missile Again https://t.co/oMbRM2kNtc #News

BUILD SUCCESSFUL (total time: 7 minutes 27 seconds)
```

*Figure 20: Obtained Tweets*

Exceptions:

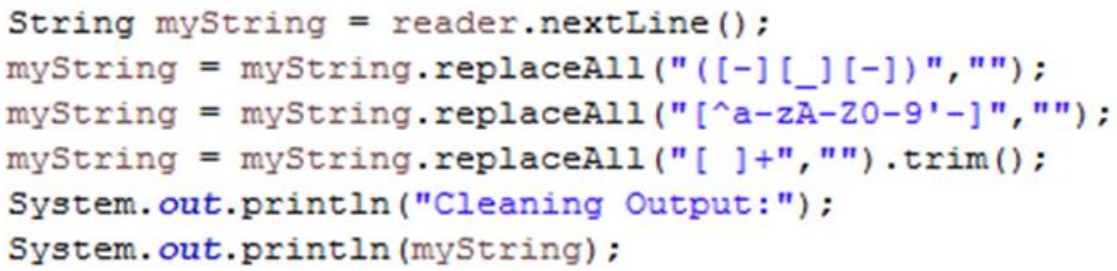


```
catch (TwitterException te) {
  System.out.println("Couldn't connect: " + te);
}
query.setMaxId(lastID-1);
```

*Figure 21: If Twitter4J cannot connect to Twitter (due to slow Internet connection)*

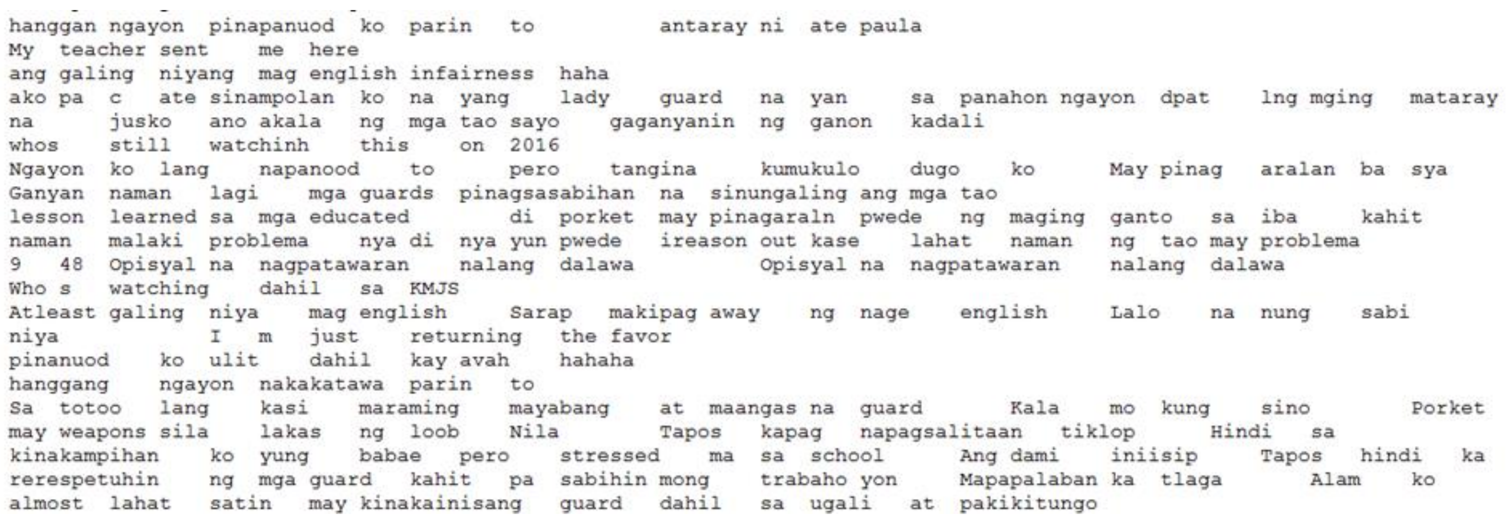
### 5.4.2.2 Preprocessing of Acquired Statements

In order for the classifier to classify each of the obtained statements (into Cyberbullying "C", Not Cyberbullying "NC", and Ambiguous Cyberbullying "AC"), they must first undergo several text preprocessing techniques: cleaning of the dataset, tokenization, and conversion of the dataset into the Bag-of-Words (BoW) unigram model. The cleaning of the dataset (removal of unnecessary words or characters per sentence) will be done automatically, with the aid of String functions in Java, the moment the obtained statement is added in the corpus. It will then be tokenized (chopped into words delimited by white spaces) immediately afterwards using another function. Lastly, the tokenized sentence will be converted to Bag-of-Words (BoW) unigram format - the only format that can be interpreted by WEKA.



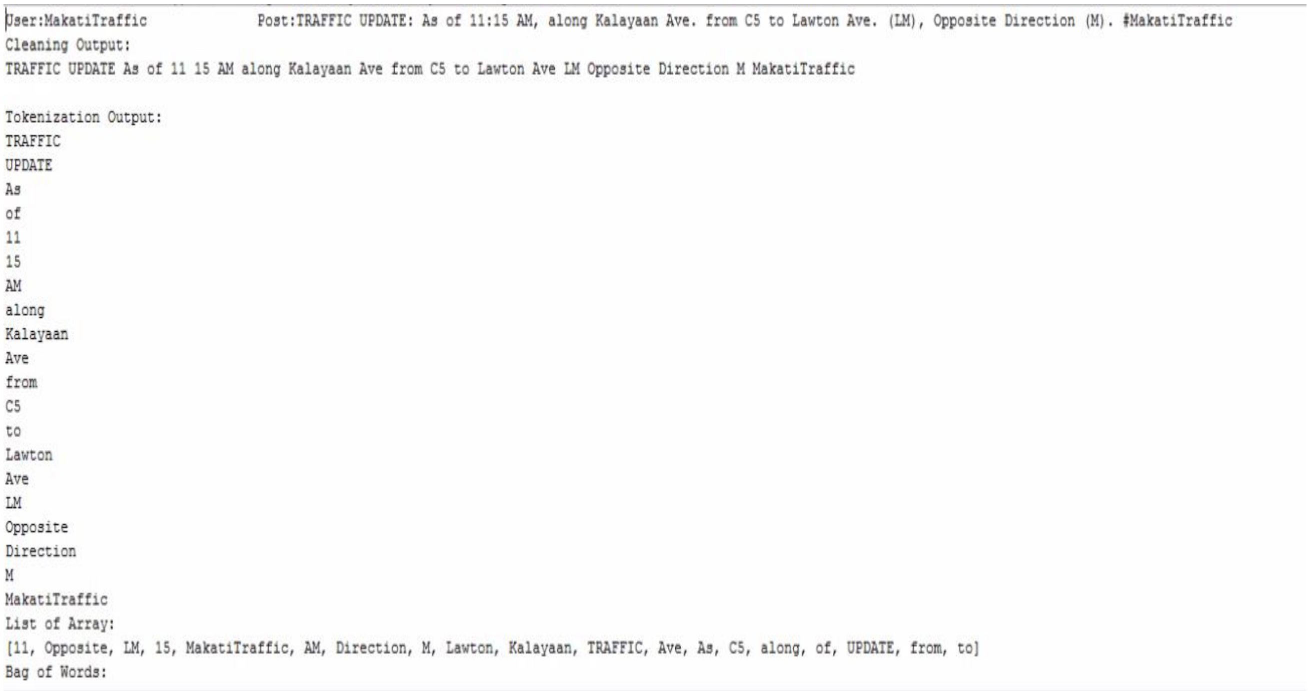
```
String myString = reader.nextLine();
myString = myString.replaceAll("([-][_][-])","");
myString = myString.replaceAll("[^a-zA-Z0-9'-]","");
myString = myString.replaceAll("[ ]+","").trim();
System.out.println("Cleaning Output:");
System.out.println(myString);
```

*Figure 22: Code snippet for cleaning the acquired data*



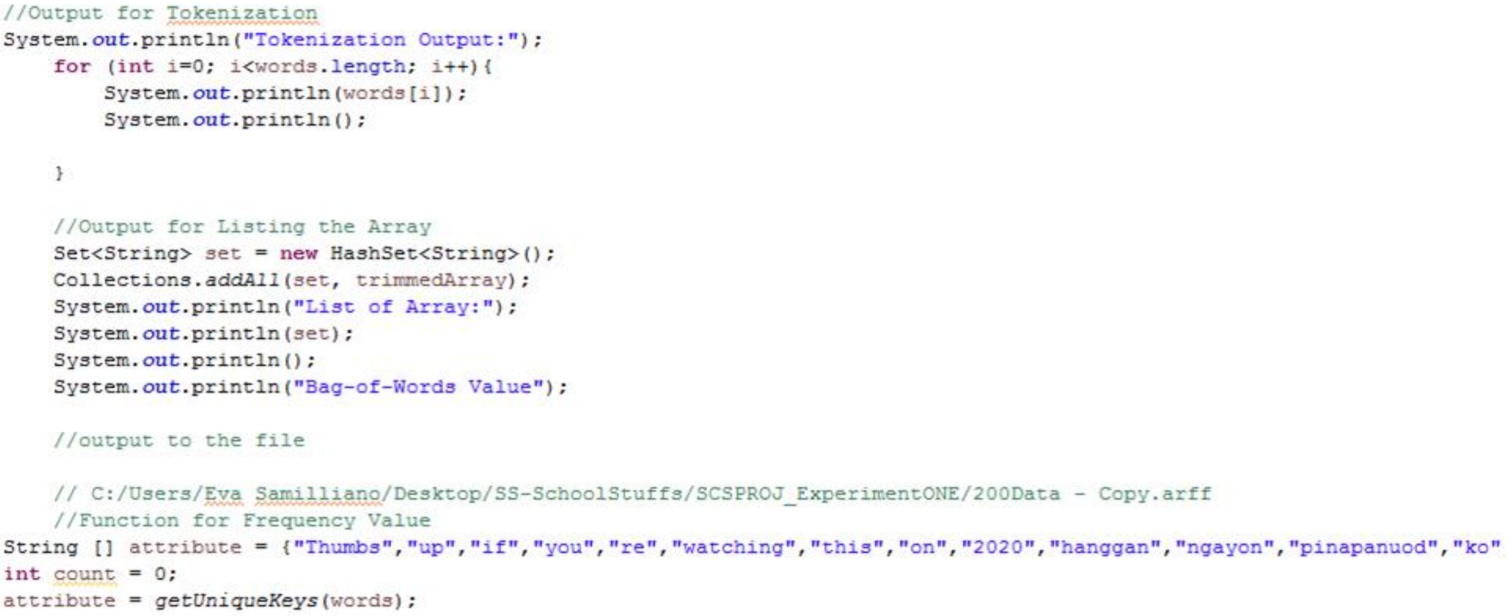
```
hanggan ngayon  pinapanuod  ko  parin   to          antaray ni  ate paula
My  teacher sent    me  here
ang galing  niyang  mag english infairness  haha
ako pa  c   ate sinampolan  ko  na  yang    lady    guard   na  yan     sa  panahon ngayon  dpat    lng mging   mataray
na      jusko   ano akala   ng  mga tao sayo    gaganyanin  ng  ganon   kadali
whos    still   watchinh    this    on  2016
Ngayon  ko  lang    napanood    to      pero    tangina     kumukulo    dugo    ko      May pinag   aralan  ba  sya
Ganyan  naman   lagi    mga guards  pinagsasabihan  na  sinungaling ang mga tao
lesson  learned sa  mga educated        di  porket  may pinagaraln  pwede   ng  maging  ganto   sa  iba     kahit
naman   malaki  problema    nya di  nya yun pwede   ireason out kase    lahat   naman   ng  tao may problema
9   48  Opisyal na  nagpatawaran    nalang  dalawa          Opisyal na  nagpatawaran    nalang  dalawa
Who s   watching    dahil   sa  KMJS
Atleast galing  niya    mag english     Sarap   makipag away    ng  nage    english     Lalo    na  nung    sabi
niya            I   m   just    returning   the favor
pinanuod    ko  ulit    dahil   kay avah    hahaha
hanggang    ngayon  nakakatawa  parin   to
Sa  totoo   lang    kasi    maraming    mayabang    at  maangas na  guard       Kala    mo  kung    sino        Porket
may weapons sila    lakas   ng  loob    Nila        Tapos   kapag   napagsalitaan   tiklop      Hindi   sa
kinakampihan    ko  yung    babae   pero    stressed    ma  sa  school      Ang dami    iniisip     Tapos   hindi   ka
rerespetuhin    ng  mga guard   kahit   pa  sabihin mong    trabaho yon     Mapapalaban ka  tlaga       Alam    ko
almost  lahat   satin   may kinakainisang   guard   dahil   sa  ugali   at  pakikitungo
```

*Figure 23: Output (Cleaning the Dataset)*



```
User:MakatiTraffic              Post:TRAFFIC UPDATE: As of 11:15 AM, along Kalayaan Ave. from C5 to Lawton Ave. (LM), Opposite Direction (M). #MakatiTraffic
Cleaning Output:
TRAFFIC UPDATE As of 11 15 AM along Kalayaan Ave from C5 to Lawton Ave LM Opposite Direction M MakatiTraffic

Tokenization Output:
TRAFFIC
UPDATE
As
of
11
15
AM
along
Kalayaan
Ave
from
C5
to
Lawton
Ave
LM
Opposite
Direction
M
MakatiTraffic
List of Array:
[11, Opposite, LM, 15, MakatiTraffic, AM, Direction, M, Lawton, Kalayaan, TRAFFIC, Ave, As, C5, along, of, UPDATE, from, to]
Bag of Words:
```

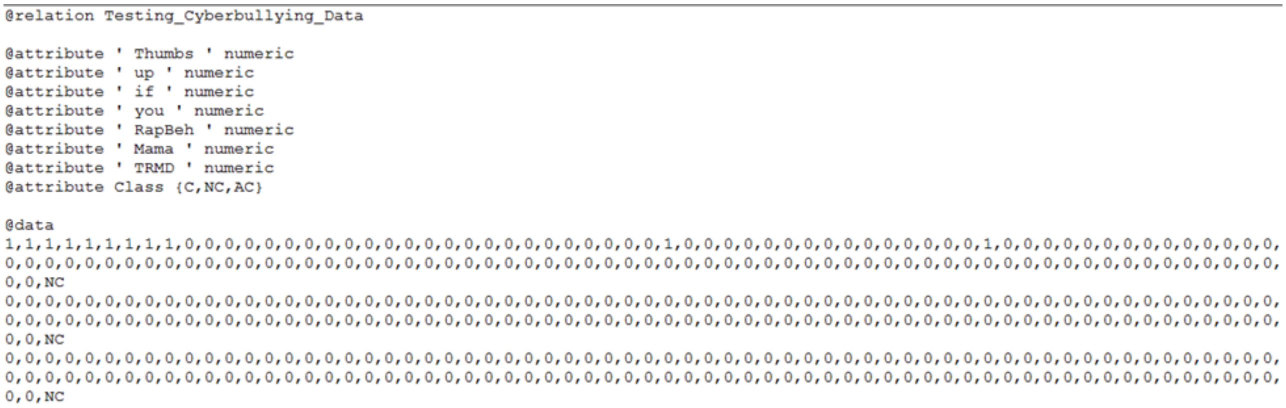
*Figure 24: Tokenized Dataset*



```
//Output for Tokenization
System.out.println("Tokenization Output:");
    for (int i=0; i<words.length; i++){
        System.out.println(words[i]);
        System.out.println();

    }

    //Output for Listing the Array
    Set<String> set = new HashSet<String>();
    Collections.addAll(set, trimmedArray);
    System.out.println("List of Array:");
    System.out.println(set);
    System.out.println();
    System.out.println("Bag-of-Words Value");

    //output to the file

    // C:/Users/Eva Samilliano/Desktop/SS-SchoolStuffs/SCSPROJ_ExperimentONE/200Data - Copy.arff
    //Function for Frequency Value
String [] attribute = {"Thumbs","up","if","you","re","watching","this","on","2020","hanggan","ngayon","pinapanuod","ko"
int count = 0;
attribute = getUniqueKeys(words);
```

*Figure 25: Code snippet for Converting the Acquired Data into Bag-of-Words Format*

Result:

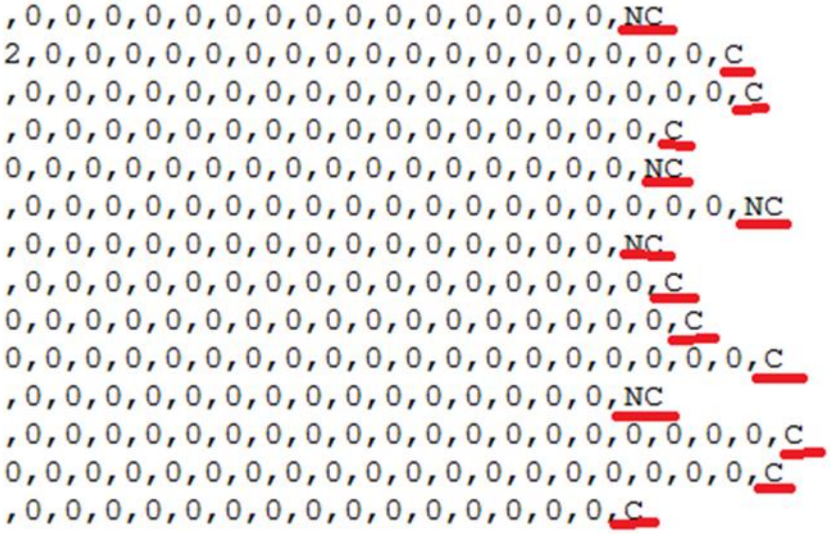


```
@relation Testing_Cyberbullying_Data

@attribute ' Thumbs ' numeric
@attribute ' up ' numeric
@attribute ' if ' numeric
@attribute ' you ' numeric
@attribute ' RapBeh ' numeric
@attribute ' Mama ' numeric
@attribute ' TRMD ' numeric
@attribute Class {C,NC,AC}

@data
1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,NC
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,NC
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,NC
```

Figure 26: Bag-of-Words in ARFF Format

#### 5.4.2.3 Identification of Cyberbullying Statements

This feature involves the automated classification of Filipino cyberbullying statements from non-cyberbullying ones in real-time, made possible by the training of the classifier (model) - which serves as the core knowledge of the application. Statements to be classified are those that have been obtained by Twitter4J (from the user's timeline) and subsequently, added to the corpus and preprocessed.

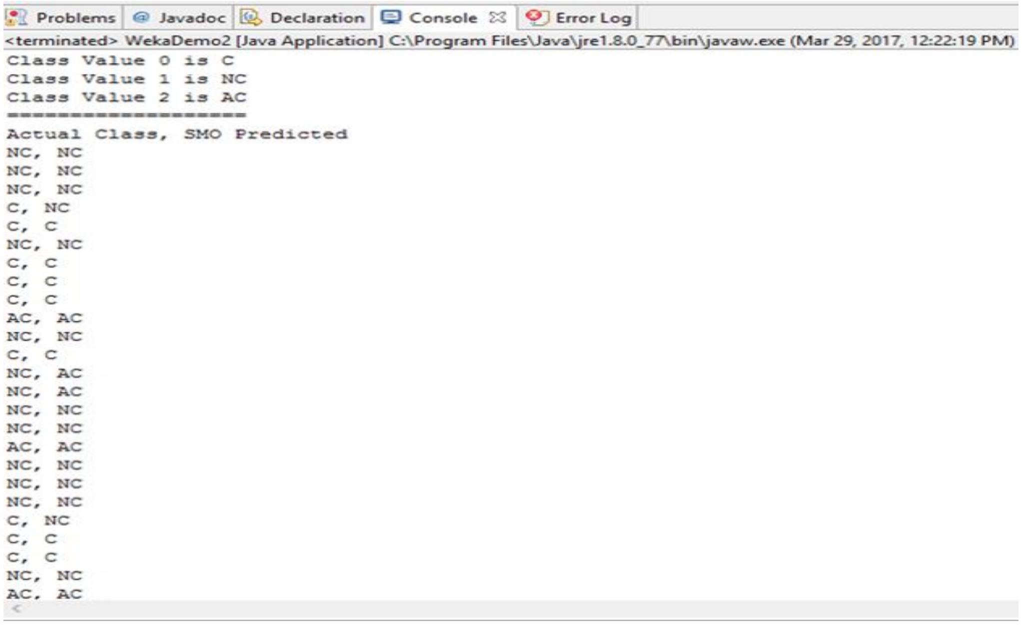


```
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,NC
2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,C
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,C
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,C
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,NC
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,NC
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,NC
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,C
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,C
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,C
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,NC
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,C
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,C
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,C
```

Figure 27: Quickgarde Identifies each Occurrence Based on the Given Annotation

### 5.4.2.4 Flagging of Cyberbullying Statements

In this feature, the classified statements will be explicitly annotated using the three specified schemes: “C”, “NC”, and “AC”.

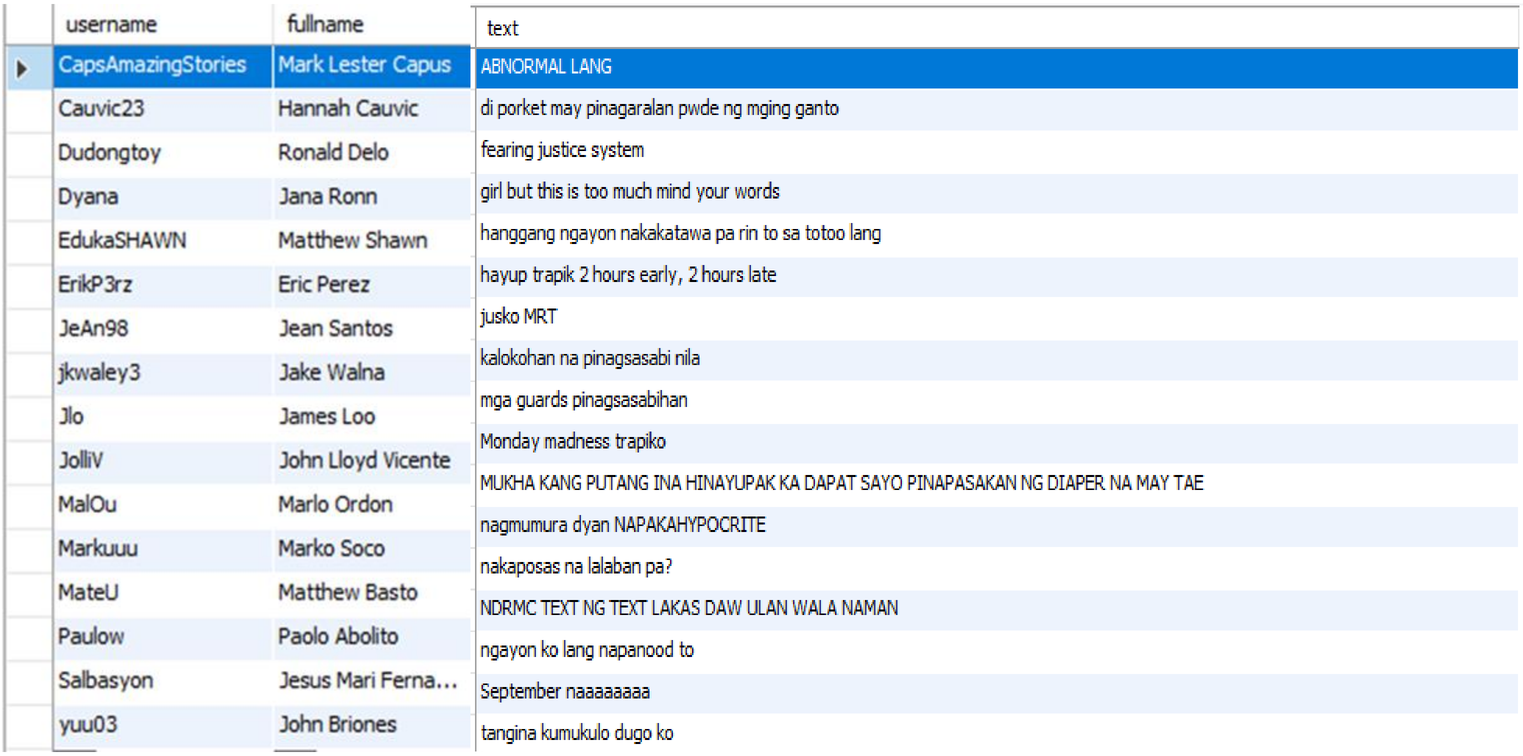


```
<terminated> WekaDemo2 [Java Application] C:\Program Files\Java\jre1.8.0_77\bin\javaw.exe (Mar 29, 2017, 12:22:19 PM)
Class Value 0 is C
Class Value 1 is NC
Class Value 2 is AC
===================
Actual Class, SMO Predicted
NC, NC
NC, NC
NC, NC
C, NC
C, C
NC, NC
C, C
C, C
C, C
AC, AC
NC, NC
C, C
NC, AC
NC, AC
NC, NC
NC, NC
AC, AC
NC, NC
NC, NC
NC, NC
C, NC
C, C
C, C
NC, NC
AC, AC
```

Figure 28: Quickgarde Flags All Cyberbullying Statements

### 5.4.2.5 Reporting of Cyberbullying Statements

After annotating, all cyberbullying statements will be outputted in a tabular format. Information regarding the tweet - username of the poster, time and date it was posted - will be included as well. Statements bearing the “NC” annotation will be disregarded. As for those labeled with “AC”, they will be used as additional features to continuously broaden the knowledge of the application in terms of detection.



| username | fullname | text |
|---|---|---|
| CapsAmazingStories | Mark Lester Capus | ABNORMAL LANG |
| Cauvic23 | Hannah Cauvic | di porket may pinagaralan pwde ng mging ganto |
| Dudongtoy | Ronald Delo | fearing justice system |
| Dyana | Jana Ronn | girl but this is too much mind your words |
| EdukaSHAWN | Matthew Shawn | hanggang ngayon nakakatawa pa rin to sa totoo lang |
| ErikP3rz | Eric Perez | hayup trapik 2 hours early, 2 hours late |
| JeAn98 | Jean Santos | jusko MRT |
| jkwaley3 | Jake Walna | kalokohan na pinagsasabi nila |
| Jlo | James Loo | mga guards pinagsasabihan |
| JolliV | John Lloyd Vicente | Monday madness trapiko |
| MalOu | Marlo Ordon | MUKHA KANG PUTANG INA HINAYUPAK KA DAPAT SAYO PINAPASAKAN NG DIAPER NA MAY TAE |
| Markuuu | Marko Soco | nagmumura dyan NAPAKAHYPOCRITE |
| MateU | Matthew Basto | nakaposas na lalaban pa? |
| Paulow | Paolo Abolito | NDRMC TEXT NG TEXT LAKAS DAW ULAN WALA NAMAN |
| Salbasyon | Jesus Mari Ferna... | ngayon ko lang napanood to |
| yuu03 | John Briones | September naaaaaaaa |
|  |  | tangina kumukulo dugo ko |

*Figure 29: Tabular Data Comprised of All Flagged Cyberbullying Statements*

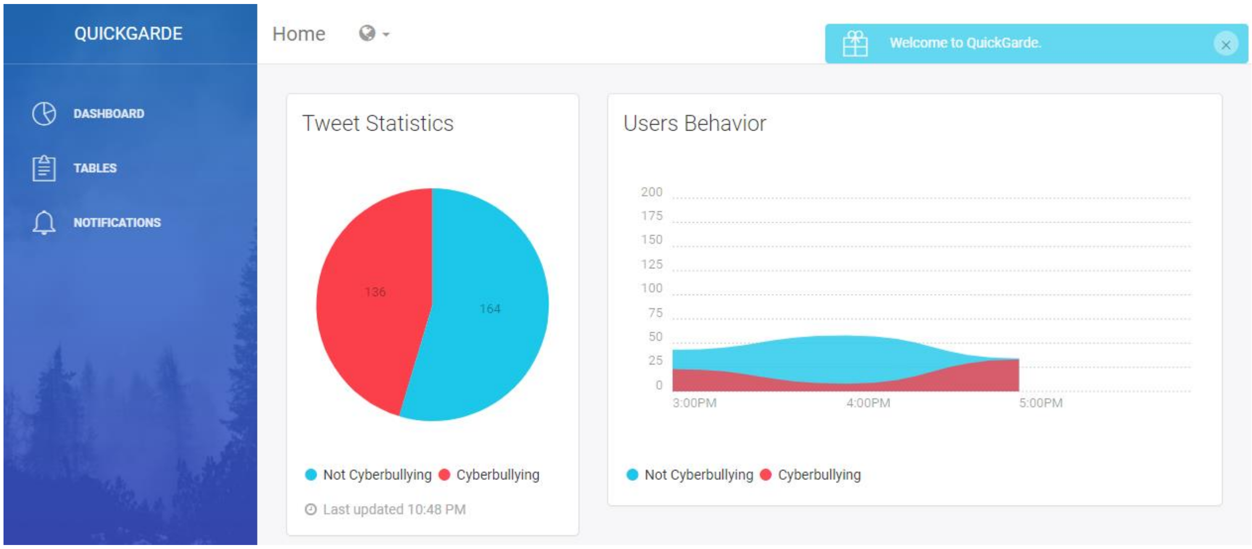


*Figure 30: Cyberbullying Data Represented by Charts and Graphs in the Admin Dashboard*

# V. Results and Discussion

## 5.1 Data Description

This section presents an in-depth description of the pre-processed textual corpus data in terms of its constituents' characteristics and their contribution to the context of the statement (if any).

### 5.1.1 Length of Words per Statement

The length of each word per statement was measured based on character count. Character count functions as a good indicator of how complex the sentences in the dataset are regarding the distribution of each of their words' lengths in that particular sentence. Furthermore, in terms of computer architecture, a "word" pertains to data handled as a single unit. They are considered as "bits" that are processed altogether by the CPU. In the count, all characters - even numeric data, as long as they are not separated by spaces, is then considered a word or part of a word.  The graph below shows the distribution of word lengths in all of the sentences in the corpus.

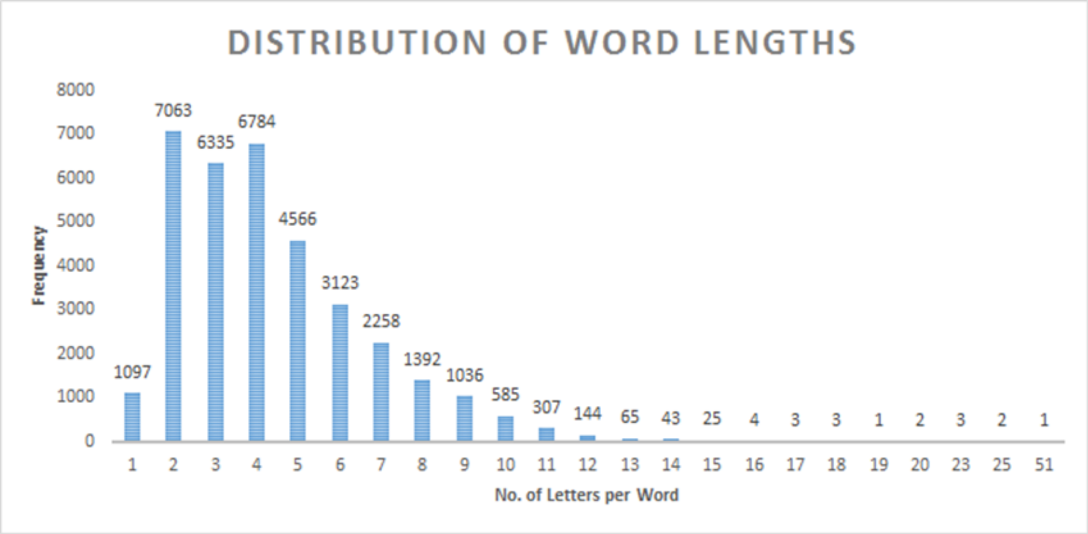


*Figure 31: Graph of the Distribution of Word Lengths in the Dataset*

It can be inferred from the graph that sentences in the corpus are not composed of any complex ones. Two-letter words are the most dominant in the dataset, followed by 4-letter and 3-letter words, considering the fact that every sentence consists of an average of 76 characters or 17 words. It is highly probable to get a sentence made up mostly of these types of words instead of the longer ones out of the corpus.

### 5.1.2 Presence of Numeric Data

The intent of excluding numeric data, specifically number combinations making use of the characters “0” to “9”, from the list of the characters to be removed is due to the fact that there are instances in the written, modern Filipino language when such characters will be combined together with the letters of the alphabet in order to form words. Such cases are present when either two of the Filipino shortcut texting styles or Jejenese - the language of the Jejemon subculture - were applied in the creation of the social media post.

There are two distinct styles of Filipino shorthand messages that make use of combined numbers and words - the phonetic style and repeating units. The phonetic style pertains to the substitution of similar sounding numeric characters, typically adapting the English pronunciation of the numbers, to their syllable counterparts (e.g. i2 for the word ito (it/this) and d2 for the word dito (here)). It can be noticed that the way the number “2” is pronounced is quite the same as that of the syllable “to” in the words ito and dito, and can therefore serve as its replacement in order to shorten the word without losing its context. This type of text messaging shortcut can also be considered as part of light Jejenese. On the other hand, the use of repeating units pertained to the substitution of a numeric character on a repeating syllable in a particular word. The numeral to be substituted will depend on the number of times the syllable will be repeated consecutively. For example, the word aalis (will leave) can be written as a2lis, since the syllable “a” was mentioned two consecutive times in that particular word.

The Jejenese language is defined as the language of the Jejemons. The term Jejemon pertains to individuals, typically Filipinos of the younger generation, who were able to create their own “written language” by forming words out of combined numbers, letters, and special characters (if applicable), typically characterized by alternate uppercase letters, overused Hs, Xs, and Zs, and rearranged characters (which would appear to form a letter when combined). They are often considered to be incomprehensible by those who are unfamiliar with the said “language”. Furthermore, there are no specific syntaxes for writing sentences in Jejenese. Jejemons were able to get a grasp of this “language” due to the influence of other Jejemons as well. Filipino people who are often engaged in social media adapt this simply because it is a trend.

| **Jejemon word/phrase** | **Original word/phrase** (In Filipino) |
|---|---|
| aq0uh | ako |
| bzt4h | basta |
| g34hin | gayahin |
| pWerA LnG iF tiNO2pAk aQ! | Pwera lang if tinotopak ako! |
| it'S harD 2 piCk uP d piEceS oF my liFe | It's hard to pick up the pieces of my life |

*Table 21: Sample Words and Phrases in Jejenese*

The dataset that was utilized in this research project yielded a total of 178 words, each of which consisted of at least 1 number. 120 statements contained those words. The most number of words bearing the same characteristics as mentioned in each of the 2000 statements in the corpus is 6, with 0 as the minimum. The graph below shows the percentage of numeric data in the dataset.

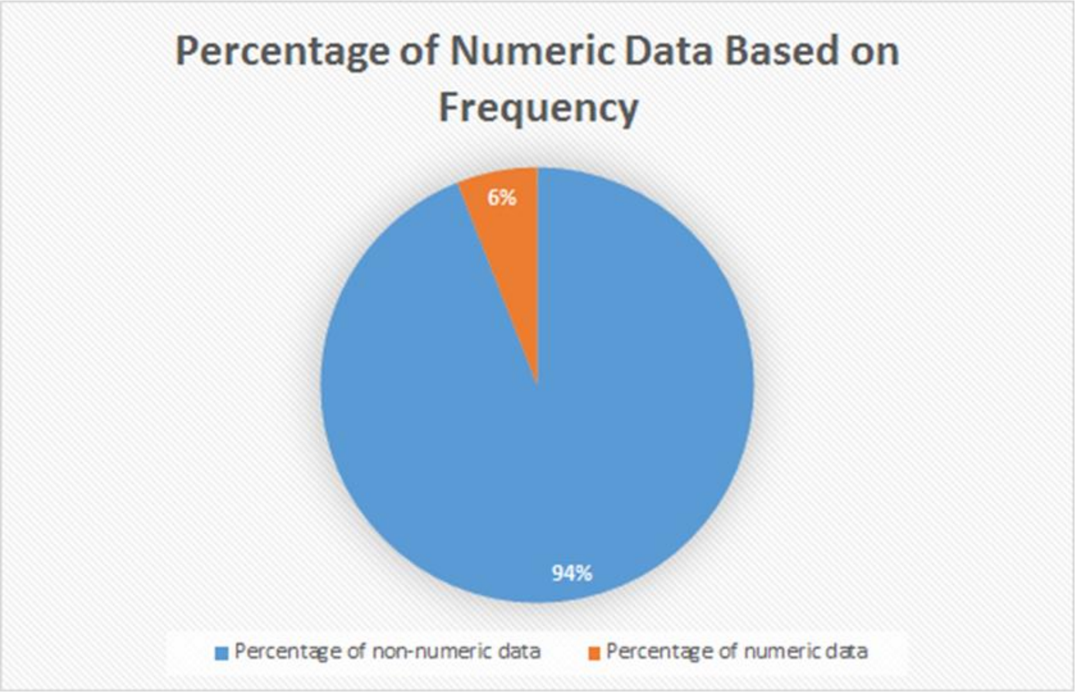


*Figure 32: Percentage of Numeric Data in the Dataset Based on Frequency*

Based on the graph, it can be seen that the number of numeric data that remained in the corpus is very minimal. Having checked each numeric data occurrence per statement, the researchers were able to determine their nature in the dataset and function in the sentence. Numeric data are being used in the corpus to either pertain to a specific date or time (e.g. to mention a year in the post or the date and time the post was posted - it was included by the web scraper), create the cat-smile ( :3 ) emoji, count nouns (e.g. 40 days, 30 years), represent a cellphone model (e.g. 3310), indicate Bible verses and television channels, utilize text messaging shortcuts (e.g. d2), or even drop a random statement.

While there are no concrete evidences of numbers being used as part of a cyberbullying keyword in the dataset, there were however some probable cyberbullying statements in which numbers played a part in. For instance, the statement "Mich bakit mo pinalitan si jam? Hindi pa nga na ka 40 days si jam pinalitan mo na ang landi mo", apparently contained implications of cyberbullying, but the number merely functions as a support to the "..ang landi mo" statement. There were also instances wherein the numbers do not make sense at all unless the topic of the thread will be made known (e.g. "isang 45 lang para sa pamilya ng mga suspek hahaha". Lastly, there were no instances of heavy Jejenese in the dataset - only those which were also used as text messaging shortcuts. These words did not imply cyberbullying as well.

As much as these types of words' impact on the performance of the model remain minimal, there is still a need for them to be kept in the dataset, for when the researchers are able to gather more similar words, then the machine will be able to bear enough knowledge to uncover the differences between a cyberbullying and non-cyberbullying keyword.

### 5.1.3 Presence of Words in Uppercase

Words expressed in all capital letters (or uppercase) likewise took a portion of the dataset. The proponents of the study decided to include them in the analysis part for most people regard them as a means of "shouting" typed text - which typically presents a negative connotation. It was mentioned from a source that words in all capital letters did not necessarily imply "loudness" during the early days. In fact, they were only utilized to convey the importance of a word in the statement. The said implication may have stemmed from the fact that uppercase words occupy larger spaces, in comparison to words in lowercase, giving them more visibility and the feeling of overcrowdedness.

There are currently 2428 uppercase words contained within the dataset. Only 332 out of the 2000 statements acquired contained these uppercase words. The maximum number of uppercase words in a single statement is 114, with 0 as the minimum. The following graph illustrates the percentage of uppercase words in the dataset in terms of frequency.

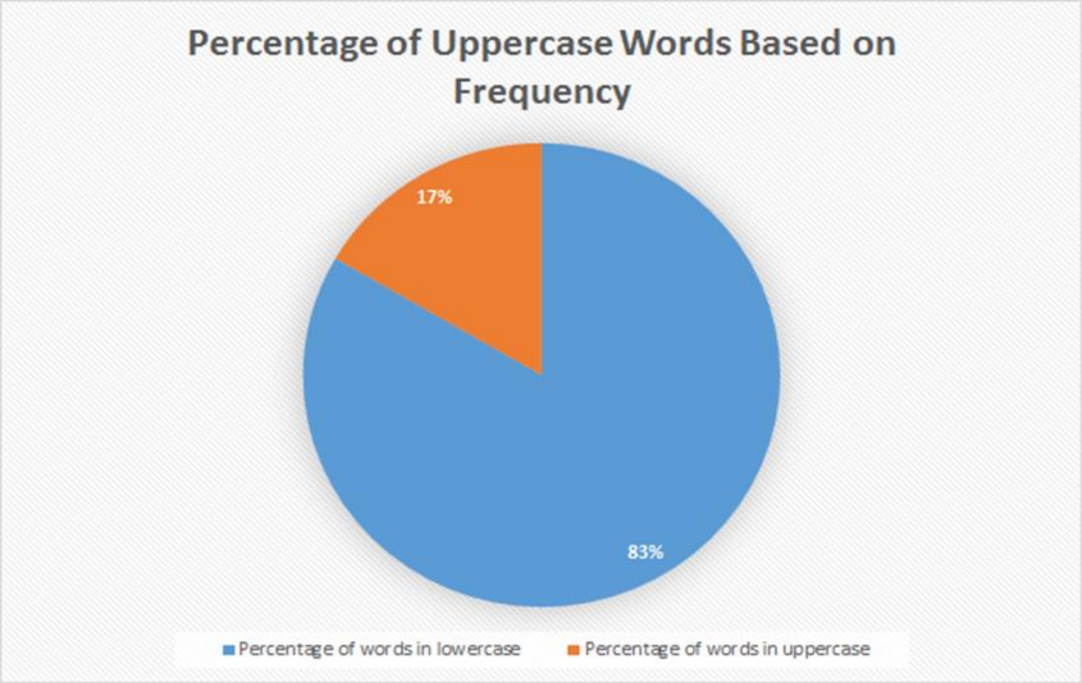


*Figure 33: Percentage of Uppercase Words in the Dataset Based on Frequency*

There are a lot of uppercase words in the dataset that served as indicators of cyberbullying, either on their own or when combined with another set of words. For instance, the statement with the most number of uppercase words in the corpus that goes, "IM HERE AGAIN O PAANO SENATOR NA SI MANNY PAQUIAO MY IDOL PERO KAYONG MGA SUMUSUBO NG ETITS NG KAPWA LALAKE EH NANANATILI PA RING MASAHOL SA HAYOP LALO KA NA BOY ABUNDAT! PURO KA DAKDAK SI MANNY SIKAT PA DIN PERO KAYONG MGA MASAHOL SA HAYOP EH NASAAN NA KAHIT KAILAN HINDI MANANAIG ANG MASAMA SA MABUTI", likewise contained the most number of cyberbullying references in it when compared to the rest of the statements. The harmfulness of the post also appeared to be doubled due to it being in all capital letters. However, not all words in uppercase are semantically offensive. People also type this way to express excitement (e.g. "OMFG SHIT YAAAAAAAAAAS") or mention abbreviated words (e.g. KMJS which stands for Kapuso Mo Jessica Soho - a television segment).

### 5.1.4 Presence of Words Consisting of Single or Double Characters

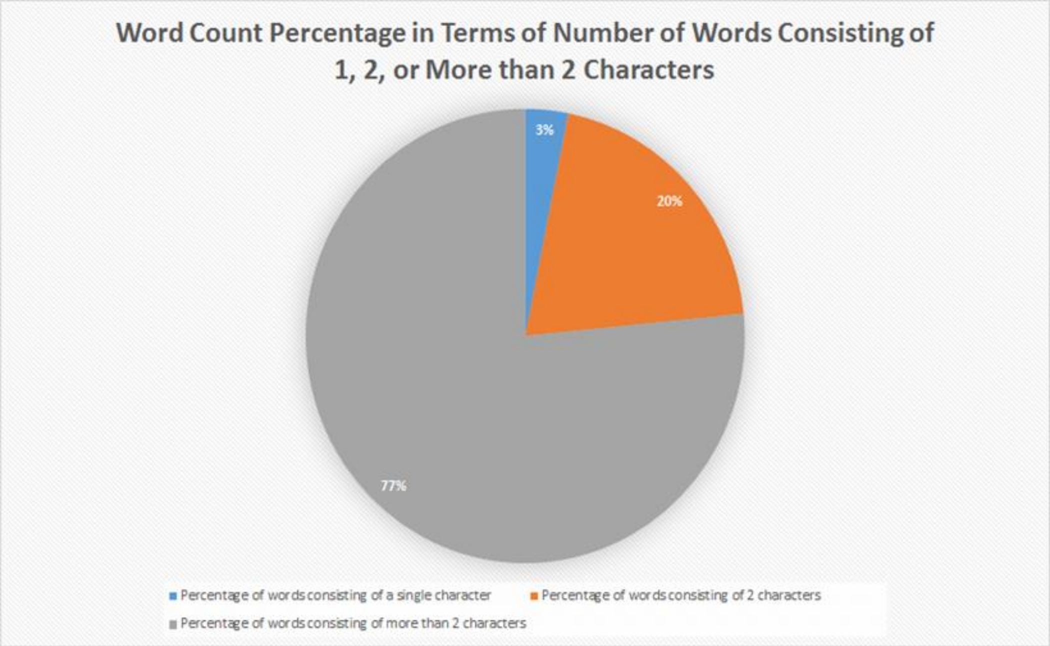


*Figure 34: Word Count Percentage in Terms of Number of Letters per Word in the Dataset*

A total of 8161 words made up of single and double characters remained in the dataset. The number of words with double characters took up the larger portion (with 7063 characters in total), contrary to those in single characters (1098 characters). They were kept in the dataset to train the algorithm to become familiar with the different usages of Filipino shortcut messages, as they are almost always present in every social media post. However, apart from being used as word shortcuts, majority of these characters in the dataset were reduced to meaningless characters due to the removal of special characters such as the apostrophe, period and hyphen, which binds them in their original words for them to convey their statement's context properly.

The tables below illustrate such instances of single characters separated from their original word(s) as a result of the cleaning of the dataset.

| **Separated single characters** (by apostrophe) | **Originally part of the following word(s)** |
|---|---|
| I | I’m, I’ve |
| m | I’m |
| s | who's, she's,it's, that's, God's, Peter's, Manny's, there's, where's, what's, he's, Vice's, everybody's, Pinoy's, attorney's, what's, Rizal's, dean's, night's, father's, China's, Hague's, Duterte's, PHL's (Philippine's), fool's, let's, Pilipino's, one's, someone's |
| t | don't, wasn't, can't, didn't, isn't, doesn't, na't, won't, isa't, couldn't, wouldn't, aren't, gov’t (government) |
| y | y’all |

*Table 22: Single Characters that were Formed due to the Removal of Apostrophes*

| **Separated single characters** (by period) | **Originally part of the following abbreviations** |
|---|---|
| a | a.k.a (also known as) |
| k | a.k.a (also known as) |
| o | o.c. (obsessive compulsive) |
| c | o.c. (obsessive compulsive) |
| U | U.S. (United States) |
| S | U.S. (United States) |
| t | t.v. (television) |
| v | t.v. (television) |
| p | p.i. (putang ina) |
| i | p.i. (putang ina) |

*Table 23: Single Characters that were Formed due to the Removal of Period*

The subsequent table, on the other hand, presents instances of “words” consisting of double characters separated from their original word(s). It is also noticeable that their amount is smaller in comparison to their single character counterpart.

| **Separated double characters**<br>(by apostrophe) | **Original word(s)** |
|---|---|
| re | you're, we're |
| ve | you've |

*Table 24: Double Characters that were Formed due to the Removal of Apostrophes*

Additionally, there were instances in the dataset of both single and double characters being used as shortcuts for different words in each of the statements. The following are some examples.

| **Single characters**<br>(used as shortcuts) | **Original word** |
|---|---|
| b | ba |
| c | si |
| d | hindi |
| f | fuck |
| G | Vice G. (used as shortcut for the surname) |
| g | sige or (literally "game") |
| k | ka or ko |
| m | mo |
| n | na |
| p | pa |
| q | ko |
| r | are |
| s | sa |
| u | you |
| w | with |
| y | why |

*Table 25: Single Characters as Word Shortcuts*

| **Double characters**<br>(used as shortcuts) | **Original word** |
|---|---|
| xa | sa |
| db | hindi ba or di ba |
| ky | kay |
| to | ito |
| kb | ka ba |
| aq | ako |
| di | hindi |
| yn | yan |
| bt | bakit |
| kc | kasi |
| em | I'm |
| nw | now |
| un | yun |
| qt | cutie |
| rt | Retweet (Twitter term) |
| dm | Direct Message (Twitter term) |
| ur | you are or you're |
| tf | the fuck (from "What the fuck") |
| vs | versus |
| te | ate |
| ko | Ako or abbreviation of the word "knockout" |

*Table 26: Double Characters as Word Shortcuts*

So far, the dataset does not contain instances of separated single characters that were used to be conjoined by hyphenated words, similar to that of the double characters. Other functions of words consisted of single characters involve representing numbers, the pronoun "I", the article "a", the Tagalog word "o", ambiguous usage of letters "s" and "G" in a sentence (e.g. "Ngayon ko lang napansin na wala palang S ung girl S" and "May mga kasalanan ka den G" - unless the "G" there functions as someone's initials), the Tagalog expression "e" - sometimes written as "eh", removed ":"

from emoticons - leaving only the letters that follow after it, and the use of the letter "x" to pair people together (e.g. Ibanez x Squier). Lastly, there were inconceivable usages of double characters as well all over the dataset. Characters such as "n1-n9" and "âœ" are some examples.

Words made up of double characters, if not comprehensible, are typically prepositions such as up, in, on, etc., adverbs such as so, etc., pang-ukol (Tagalog prepositions) such as sa, ng, etc., panghalip (pronouns) such as ka or ko, pang-abay (adverbs) such as na, pa, etc., and pangatnig (conjunctions) such as at, ni, etc.

### 5.1.5 Top 50 Cyberbullying Keywords

The following are the top 50 most recurrent words among all statements or sentences annotated as "cyberbullying" in the dataset. They were also based on several sources indicating the top cyberbullying keywords that can be found in posts created by Filipinos in social media.

| Rank | Word(s) | No. of occurrences |
|---|---|---|
| 1 | INA/unu | 566 |
| 2 | TANG/tung | 210 |
| 3 | puta/putang/pota/PUTANGINA/pokeng | 155 |
| 4 | Hahahaha/Hihi/Hehehe | 153 |
| 5 | baba | 137 |
| 6 | gay | 122 |
| 7 | fuck/fucking/FUCKIN/pakyu | 97 |
| 8 | HAYOP/KABAYO/ANIMAL/daga/pet/baboy | 95 |
| 9 | mamatay/hell/Kill/patay/bitayin | 93 |
| 10 | LANDI/itch | 88 |
| 11 | gang | 70 |
| 12 | pusher/druglord/lord/drug | 70 |
| 13 | tawa/Kakatawa | 69 |
| 14 | BIG | 64 |
| 15 | gago/tado | 62 |
| 16 | King | 61 |

| | | |
|---|---|---|
| 17 | tanga/ulol/ulul | 52 |
| 18 | gaga | 47 |
| 19 | malandi/Kalandi/baliw/abnoy | 45 |
| 20 | loko | 42 |
| 21 | bakla/Kadiri | 41 |
| 22 | ass/Butt | 39 |
| 23 | che | 37 |
| 24 | mahina/matanda/slow | 37 |
| 25 | hiya | 35 |
| 26 | bitch/tarantado | 35 |
| 27 | bobo/stupid/kupal | 33 |
| 28 | bad | 30 |
| 29 | paa/TAE | 29 |
| 30 | panga | 26 |
| 31 | wawa/Kaawa | 26 |
| 32 | baho/kati | 25 |
| 33 | punyeta/shit/bwiset/Bullshit | 24 |
| 34 | tangina | 23 |
| 35 | yaya | 23 |
| 36 | adik/salot | 22 |
| 37 | mahirap | 21 |
| 38 | pabebe/epal | 21 |
| 39 | malaki | 17 |
| 40 | pangit | 17 |
| 41 | BLACK/FAT/Blind/sunog | 17 |
| 42 | Bayag/pepe/etits/tuwad/pakantot /boobs/penis | 17 |
| 43 | Kapal | 15 |
| 44 | Yuck/Ewww | 8 |
| 45 | HYPOCRITE/pathetic | 6 |
| 46 | bruha/halimaw | 6 |

| | | |
|---|---|---|
| 47 | demonyo | 4 |
| 48 | yawa | 4 |
| 49 | engot | 3 |
| 50 | inarte | 1 |

Table 27: Top 50 Cyberbullying Keywords

It can be noticed that in the first 10 cyberbullying keywords, profane words are the most dominant, specifically the many variations of the word putangina. Likewise, there were keywords implying threats (e.g. patayin - literally means "to kill" and bitayin - "to submit to torture"), and insults (e.g. landi, baboy,cand baba - which means "chin" in Tagalog but is often used an alias to call people with prominent chin). The last 10, however, are dominant on expressing insults - either physical, behavioral, and mental. It also included a set of words which are meant to pertain to both male and female sex organs, typically used to imply lustful desires toward a person, or perhaps a stranger.

### 5.2 Baseline Results

For the first experiment, the model was evaluated against 500 testing data for each run and the labels assigned by the model were compared against the labels that were assigned to the classes during annotation. The overall number of data that was used in training the model was 1000. However, in order to determine how the number of data can affect the model's performance, the number of the training data for each run was increased. To illustrate, 200 data was used for the first run then 300 more data was added for the second run. As for the third run, 200 more data were included and another 300 data for the fourth run. Table 28 depicts the accuracy and the kappa statistics of the model.

As seen in Table 28, there was a slight increase in the values of accuracy and kappa statistics as the number of training data increases. Thus, the model will be able to classify more correct instances if the number of training data was increased. As for the kappa statistics, the highest value yielded by the model was 0.2312 for the third run which implies that there was a fair agreement between the labels that were assigned by the annotators as well as to those that were assigned by the classifier. However, as shown in Table 28, a larger dataset may not always indicate a higher Kappa score as bias may likely to occur on the side of the annotator (Gwet, 2002).

| # of Training Data | # of Testing Data | Accuracy | Kappa Statistics |
|---|---|---|---|
| 200 | 500 | 49.5 | 0.1571 |
| 500 | 500 | 53.6 | 0.2152 |
| 700 | 500 | 54.5714 | 0.2312 |
| 1000 | 500 | 57.8889 | 0.2294 |

*Table 28: The Effect of Adding More Training Data to the Dataset on the Accuracy and Kappa Statistic of the Model*

## 5.3 Percentage Split

For this experiment, the dataset was divided into two parts: training and testing. However, for each run, different splits were used. The purpose of this experiment is to determine the right split for the dataset. As shown in Table 29, the proper split for the dataset is 80/20.

| Training Data (%) | Testing Data (%) | Accuracy | Kappa Statistics |
|---|---|---|---|
| 60 | 40 | 45.8824 | 0.0911 |
| 70 | 30 | 47.3333 | 0.114 |
| 80 | 20 | 55 | 0.2177 |
| 90 | 10 | 50 | 0.1325 |

*Table 29: The Relevance of the Percentage Split to the Accuracy and Kappa Statistic of the Model*

## 5.4 K-Fold Cross Validation

For this set of experiments, the researchers performed a non-exhaustive cross validation method called k-fold cross-validation wherein multiple rounds of cross-validations were used on the dataset using different partitions. The primary goal of conducting these experiments is to determine the number of folds which can be used that will result into a better predictive model. Table 30 summarizes the result of the experiments that were conducted.

| K-Fold | Accuracy | Kappa Statistics |
|---|---|---|
| 2 | 57.65 | 0.1933 |
| 3 | 57.65 | 0.2007 |
| 4 | 58.2 | 0.2082 |
| 6 | 58.05 | 0.2096 |
| 7 | 58.15 | 0.2081 |
| 8 | 58.85 | 0.2288 |
| 9 | 56.95 | 0.2084 |
| 10 | 57.95 | 0.2094 |

*Table 30: The Relation of each K-Fold Split to the Accuracy and Kappa Statistic of the Model*

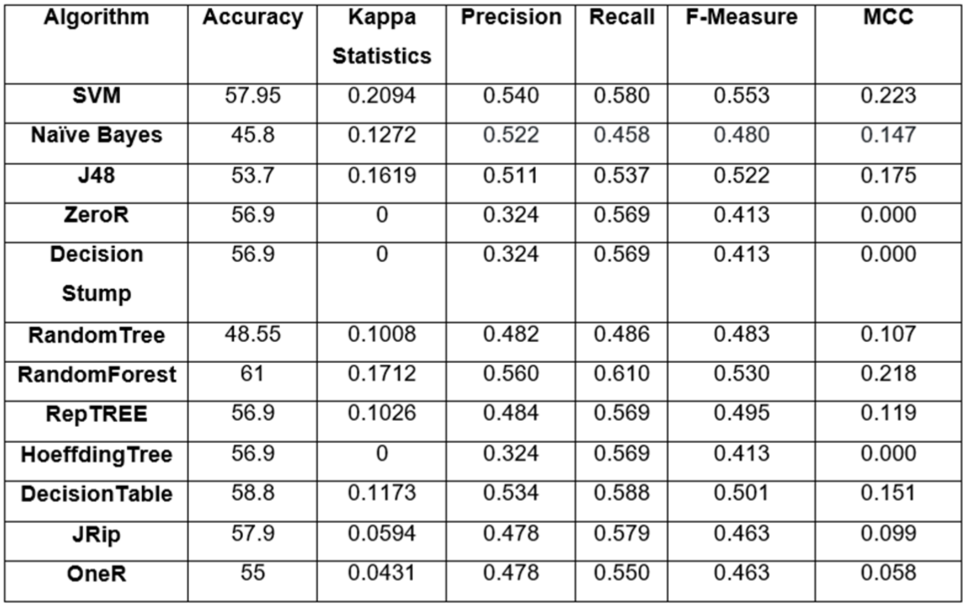
As shown in Table 30, the model can yield the highest accuracy and kappa score when the dataset was divided into 8 folds.

## 5.5 Using Trigrams

By implementing trigrams on 1000 data, the classifier was able to yield an accuracy of 54.7% and a kappa statistic of 0.1002.

## 5.6 Comparison of Machine Learning Algorithms

For this experiment, the Support Vector Machine (SVM) was compared among the different performances of 11 other machine learning algorithms. 10-fold cross validation was performed for each algorithm. In addition to this, each performance was tested against 2000 data. The purpose of this experiment was to determine how each algorithm's performance varies from one another and to identify the algorithm that is best suitable in classifying cyberbullying instances. Table 31 illustrates the comparison of the performance of the 12 machine learning algorithms used for our classification problem.



| Algorithm | Accuracy | Kappa Statistics | Precision | Recall | F-Measure | MCC |
|---|---|---|---|---|---|---|
| SVM | 57.95 | 0.2094 | 0.540 | 0.580 | 0.553 | 0.223 |
| Naïve Bayes | 45.8 | 0.1272 | 0.522 | 0.458 | 0.480 | 0.147 |
| J48 | 53.7 | 0.1619 | 0.511 | 0.537 | 0.522 | 0.175 |
| ZeroR | 56.9 | 0 | 0.324 | 0.569 | 0.413 | 0.000 |
| Decision Stump | 56.9 | 0 | 0.324 | 0.569 | 0.413 | 0.000 |
| RandomTree | 48.55 | 0.1008 | 0.482 | 0.486 | 0.483 | 0.107 |
| RandomForest | 61 | 0.1712 | 0.560 | 0.610 | 0.530 | 0.218 |
| RepTREE | 56.9 | 0.1026 | 0.484 | 0.569 | 0.495 | 0.119 |
| HoeffdingTree | 56.9 | 0 | 0.324 | 0.569 | 0.413 | 0.000 |
| DecisionTable | 58.8 | 0.1173 | 0.534 | 0.588 | 0.501 | 0.151 |
| JRip | 57.9 | 0.0594 | 0.478 | 0.579 | 0.463 | 0.099 |
| OneR | 55 | 0.0431 | 0.478 | 0.550 | 0.463 | 0.058 |

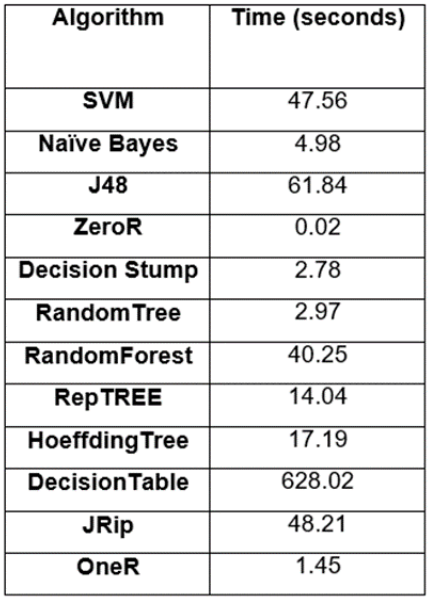
*Table 31: Comparison of the Performances Among 12 Machine Learning Algorithms*

Accuracy or the observed accuracy is simply the number of instances that were classified correctly throughout the entire confusion matrix. In this paper, it pertains to the number of instances that were labeled as cyberbullying via ground truth (annotation) and then classified as cyberbullying by the machine learning classifier. As shown in Table 31, both Random Forest and Decision Table outperformed SVM and other machine learning algorithms in terms of accuracy with a score of 61% and 58.8% respectively. SVM, on the other hand, yields an accuracy of 57.95%. However, since a higher accuracy does not always indicate a greater predictive power than those models with a lower level of accuracy (Tilmann, 2007), the proponents also measured the performance of the models by using other metrics: kappa statistics, precision, recall, and f-measure.

Kappa statistic was used to assess the accuracy of the classification algorithms by distinguishing between the reliability of the data and their validity by comparing the observed accuracy with an expected accuracy, the accuracy that any random classifier would be expected to achieve depending on the confusion matrix. To illustrate, it evaluates classifiers amongst themselves by taking into account random chance (agreement with a random classifier) which means it is less misleading than simply using accuracy as a metric. In terms of kappa statistics, SVM outperforms the other algorithms with a score of 0.2094 followed by Random Forest with a score of 0.1712. ZeroR, Decision Stump, and HoeffdingTree have a kappa score of 0.

The researchers also measured the performance of each model using precision (exactness) and recall (sensitivity). Precision is the proportion of all positive predictions that are correct and recall is the proportion of real positive observations that are correct. Among all models, Random Forest yields the highest precision, with a score of 0.560, and a recall of 0.610. It was followed by SVM with a precision of 0.540 and a recall of 0.580. Precision and recall are always proportional to each other; thus, a higher precision means a lower recall and vice versa. Therefore, also factored in was the combination of precision and recall into a single value, getting their average as a way of measuring the accuracy of each model. This metric is also known as F-measure. As shown in Table 31, SVM yields the highest F-measure with a score of 0.553 which outperformed the other algorithms. However, these metrics do not include True Negative, which pertains to those cyberbullying statements that were not classified as cyberbullying, in their respective equations. To address this, the Matthews Correlation Coefficient in measuring the accuracy of each model was also included. This metric includes true and false positives and negatives and is regarded as a balanced measure which can be used even if the classes vary in terms of sizes. Moreover, it is a correlation coefficient between the observed and predicted binary classifications. SVM yields the highest MCC value of 0.223

Since it is clearly difficult to differentiate the performance of the machine learning algorithms based on their accuracy and kappa scores alone (Williams, N. Zander, S. & Armitage, G., 2006), focus was given on the time taken in building each model known as the computational performance. Among all the algorithms, ZeroR required the fastest time of 0.02 second in building the model. However, it does not contain any predictability power. Instead it is often used in determining a baseline performance as a benchmark for other classification methods (Nasa & Suman, 2012). Thus, other machine algorithm that will be tested on the same dataset must yield a higher accuracy than ZeroR (Brownlee, 2016).



| Algorithm | Time (seconds) |
|---|---|
| SVM | 47.56 |
| Naïve Bayes | 4.98 |
| J48 | 61.84 |
| ZeroR | 0.02 |
| Decision Stump | 2.78 |
| RandomTree | 2.97 |
| RandomForest | 40.25 |
| RepTREE | 14.04 |
| HoeffdingTree | 17.19 |
| DecisionTable | 628.02 |
| JRip | 48.21 |
| OneR | 1.45 |

*Table 32: The Different Processing Speeds of each Algorithm based on the Dataset that was Utilized*

## 5.7 Issues

In this section, the issues encountered throughout the development of Quickgarde were identified.

### 5.7.1 Evolution of the Filipino Language

The dataset was able to cover only a limited number of Filipino and English harmful statements. As for variations in Filipino language, only a little Jejemon harmful statements were covered. Moreover, Bekimon, a popular language used by gays, was not covered at all. As Filipino language continuously evolve, being able to identify as many as harmful statements in different variations of Filipino language is crucial in terms of detecting cyberbullying posts in Filipino social media posts.

### 5.7.2 Imbalanced Dataset

Class imbalance is defined as a problem in machine learning wherein the total number of a class of data is far less than the total number of another class of data. The problem with this is most machine learning algorithms work best when the number of instances of each classes are roughly equal. When the number of instances of one class far exceeds the other, problems arise. In our case, there is an unequal distribution among the three classes: Cyberbullying, Not Cyberbullying, and Ambiguous Cyberbullying. As shown in the graph below, the number of data that belongs to the class Non-Cyberbullying (49%) is enormously larger than those in the Cyberbullying (34%) and Ambiguous Cyberbullying (18%). These numbers are merely dependent on how the annotator perceives each statement.

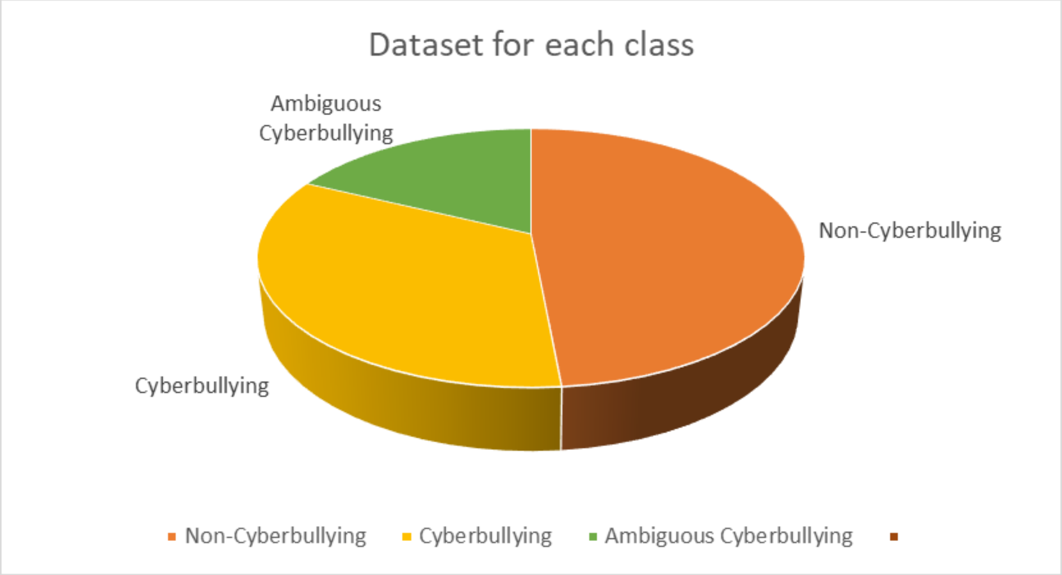


*Figure 35: Pie Graph Illustrating the Total Number of Statements Falling under the Predefined Categories*

Accuracy is not the metric to use when working with an imbalanced dataset because it can be misleading because it can only achieve a higher accuracy on the class which has a larger number of instances (Brownlee, 2015). Thus, other performance metrics must be considered in order to determine the optimal model for the classification problem. For this, the researchers used precision, a measure of correctness achieved in positive prediction, recall, a measure of actual observations which are labeled (predicted) correctly, and F measure, which combines precision and recall as a measure of effectiveness of classification in terms of ratio of weighted importance on either recall or precision as determined by β coefficient. Although these methods are better than accuracy and error metric, they are still ineffective in answering the important questions on classification. To illustrate, these metrics are also ineffective in answering the important questions on classification. For example: precision does not state negative prediction accuracy and recall, on the other hand, only focuses on actual positives (Analytics Vidhya Content Team, 2016). Thus, there is a need to have a better metric to cater to accuracy needs. For this purpose, the proponents used Matthew Coefficient Correlations, which is a more suitable metric in dealing with imbalanced data (Boughorbel, Jarray & El-Anbarri, 2017). This is a powerful metric that considers both accuracies and error rates on both classes, since all the four values in the confusion matrix are included in this formula. A high MCC value indicates that the classifier must have high accuracies on positive and negative classes, and also have less misclassification on the two classes. Therefore, MCC can be considered as the best singular assessment metric (Ding, 2011).

## VI. Conclusions and Recommendations

Philippines was recognized as the social media capital of the world with Filipinos spending an average of 4 hours and 17 minutes per day on social media sites. However, the tremendous growth of social media users has consequently intensified the cyberbullying problem in the Philippines. Current methods employed by social media providers in mitigating cyberbullying rely heavily on user's initiatives to flag and report a harmful post. Since Philippines conservative, Filipinos are hesitant to report a cyberbullying incident. Moreover, the massive information available in the Web makes it difficult for moderators to monitor social media sites manually. Thus, there is a need for an intelligent system to automate the process of detecting cyberbullying instances which will reduce the effort of moderators and individuals in keeping social media a safe environment. For this purpose, a few studies have been conducted to automate cyberbullying detection by incorporating text classification techniques. However, these studies merely focused on optimizing the accuracy of the model by incorporating various techniques rather than defining follow up strategies once a cyberbullying post has been detected. In this paper, the proponents of the study presented a novel approach that has a potential for detecting harmful messages and allowing social media administrators to provide timely responses.

They began by collecting data from Facebook, YouTube, and Twitter. These data undergo text preprocessing techniques such as cleaning and tokenization were applied on the data. Then they were converted into Bag-of-Words (BoW) representation. For the initial experiments, accuracy and kappa statistics were used to measure the performance of the classifier despite the imbalanced dataset because kappa itself is a good indicator of performance. However, in terms of comparing different machine learning algorithms, other metrics were used such as Precision, Recall, F-Measure, and Matthew Coefficient Correlations in identifying the best algorithm for the classification task. Random Forest models may not be the best choice for imbalanced datasets despite the fact that it outperformed

Support Vector Machine in terms of accuracy (0.61), precision (0.560), recall (0.610), and even the speed in building the model (40.25 seconds) because SVM had a higher kappa score (0.2094), F-Measure (0.553), and even MCC (0.223), which is the best indicator of a classifier's performance in dealing with an imbalanced dataset. The results show that SVM is still the best algorithm for the classification problem.

As for recommendations and future works, the researchers are planning to make Quickgarde even better by including additional studies in the work. First, more data will be added into the dataset. This will ensure that different variations in Filipino language such as Bekimon and Jejemon are covered to improve the classification of cyberbullying statements.

Likewise, they are also planning to use different metrics such as ROC Area to further analyze the performance of the SVM classifier.

## VII. Appendices

## BIBLIOGRAPHY

Awad, W., & Elseofi, S. (2011). Machine Learning Methods for Spam Email Classification. International Journal of Computer Science & Information Technology, 3(1), pp. 173–184.

Boehm, C. (2012). Moral Origins: The Evolution of Virtue, Altruism, and Shame. New York, New York: Basic books (Perseus Books Group).

Boughorbel, S. Jarray, F. & El-Anbari, M. (2017). Optimal Classifier for Imbalanced Dataset using Matthews Correlation Coefficient Metric. PLos ONE 12(6):e0177678.

Busemann, S., Schmeier, S. & Arens, R. (2000, April 29). Message Classification in the Call Center, Proceedings of the sixth conference on Applied natural language processing, Seattle, Washington, April 29 – May 4, 2000. Stroudsburg, PA, USA: Association for Computational Linguistics.

Campbell, M. (2005). Cyber bullying: An old problem in a new guise?. Australian Journal of Guidance and Counselling, 15(1), pp. 68-76.

Cheng, C., & Ng, A. (2016). Automated Role Detection in Cyberbullying Incidents, Proceedings of the 16th Philippine Computing Science Congress, Puerto Princesa, Palawan, Philippines, March 16 – 18, 2016. Quezon City, Metro Manila, Philippines: Computing Society of the Philippines.

Dadvar, M., & De Jong, F. (2012). Cyberbullying detection: A step toward a safer internet yard, Proceedings of the 21st International Conference on World Wide Web, Lyon, France, April 16 – 20, 2012. New York, New York: Association for Computing Machinery.

Dadvar, M., De Jong, F., Ordelman, R., & Trieschnigg, D. (2012). Improved Cyberbullying Detection Using Gender Information, Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop, Ghent, Belgium, February 23 – 24, 2012. Ghent, Belgium: Ghent University.

Digital in 2017: Global Overview. (2017, January 24). Retrieved from http://wearesocial.com/sg/blog/2017/01/digital-in-2017-global-overview

Dinakar, K., Reichart, R., & Lieberman, H. (2011). Modeling the Detection of Textual Cyberbullying, International AAAI Conference on Web and Social Media, Barcelona, Catalonia, Spain, July 17 – 21, 2011. Menlo Park, California: The AAAI Press.

Ellwood-Clayton, B. (2006). All we need is love—and a mobile phone: texting in the Philippines, Cultural Space and Public Sphere in Asia 2006, Korea Broadcasting Institute, Seoul, March 17 – 18, 2006.

Fields of Computer Science. (n.d.). Retrieved from http://aihorizon.com/essays/basiccs/general/cs_areas.html

Goffman, E. (1956). The Presentation of Self in Everyday Life. New York, New York: Random House.

Gonzales, R. (2014). Social Media as a channel and its Implications on Cyber Bullying, DLSU Research Congress 2014, De La Salle University, Manila, Philippines, March 6 – 8, 2014. Manila, Philippines: De La Salle University.

Hinduja, S., & Patchin, J. (2007). Offline Consequences of Online Victimization: School Violence and Delinquency. Journal of School Violence, 6(3), pp. 89-112.

Jones, K.S. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 28(1), pp. 11-21.

Kwok, I., & Wang, Y. (2013). Locate the Hate: Detecting Tweets against Blacks, Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, Washington, July 14 – 18, 2013. Palo Alto, California: Association for the Advancement of Artificial Intelligence.

Lacuata, R. (2014, June 27). How cyberbullying changed 'Amalayer' girl. Retrieved from
RA 10627: The Anti-Bullying Act. (2015, January 22). Retrieved from http://www.elegal.ph/republic-act-no-10627-the-anti-bullying-act/

Lam, A., Paner, I., Macatangay, J., & Delos Santos, D. (2014). Classifying Typhoon Related Tweets, 10th National Natural Language Processing Research Symposium, De La Salle University, Manila, Philippines, February 21 – 22, 2014. Manila, Philippines: De La Salle University.

Lewis, D. D. (1992). Representation and Learning in Information Retrieval (Doctoral dissertation). Retrieved from UMI. (GAX92-19460)

Litvak, M., & Last, M. (2008, August 23). Graph-based Keyword Extraction for Single-document Summarization, Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization, Manchester, United Kingdom, August 23 - 23, 2008. Stroudsburg, PA, USA: Association for Computational Linguistics.

Madnani, N. (N.D). Getting Started on Natural Language Processing with Python. ACM Crossroads, 13(4), pp. 5-5.

Marathe, S., & Shirsat, K. (2015). Contextual Features Based Naïve Bayes Classifier for Cyberbullying Detection on YouTube. International Journal of Scientific & Engineering Research, 6(1), pp. 1109–1114.

Minor, M., Smith, G., & Brashen, H. (2013) Cyberbullying in Higher Education. Journal of Educational Research and Practice 2013, 3(1), pp. 15–29.

Peersman, C. Daelemans, W. Vaerenbergh, L. (2011). Predicting Age and Gender in Online Social Network, Proceedings of the 2011 International Workshop on Search and Mining User-generated Contents, Glasgow, Scotland, UK, October 28, 2011. New York, New York: Association for Computing Machinery.

Pinto, A., Oliveira, H.G., & Alves, A.O. (2016). Comparing the Performance of Different NLP Toolkits in Formal and Social Media Text, 5th Symposium on Languages, Applications and

Technologies, Maribor, Slovenia, June 20 – 21, 2016. Saarbrücken/Wadern, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH.

Republic Act No. 10627. (2013, September 12). Retrieved from http://www.gov.ph/2013/09/12/republic-act-no-10627/

Salton, G. & McGill, M.J. (1986). Introduction to Modern Information Retrieval. New York, New York: McGraw-Hill Inc.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34(1), pp. 1–47.

Sheoran, J. (2012). Technological Advancement and Changing Paradigm of Organizational Communication. International Journal of Scientific and Research Publications, 2(12).
Sintaha, M., Satter, S., Zawad, N., Swarnaker, C. & Hassan, A. (2016). Cyberbullying Detection using Sentiment Analysis in Social Media (Unpublished doctoral dissertation). Department of Computer Science & Engineering, BRAC University, Dhaka, Bangladesh.

Sivic, Josef (April 2009). Efficient visual search of videos cast as text retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(4), pp. 591–605.

Smith, P., et al. (2008). Cyberbullying: its nature and impact in secondary school pupils. Journal of Child Psychology and Psychiatry, 49(1), pp. 376–385.

Takumi, R. (2016, March 30). FROM HUMILIATION TO THREATS: Survey says 80% of young teens in PHL experience cyberbullying. Retrieved from http://www.gmanetwork.com/news/story/560886/lifestyle/parenting/80-of-young-teens-in-phl-experience-cyberbullying-survey

Torres, M. (2016). Netizens react to Monster radio DJ Karen Bordador's arrest. Retrieved from https://kami.com.ph/40852-monster-radio-dj-karen-bordadors-arrest-stirs-netizens.html

Tulad, V. (2012). Cyberbullying: A victim's tale of lies and the madness of crowds. Retrieved from http://www.gmanetwork.com/news/story/274156/hashtag/cyberbullying-a-victim-s-tale-of-lies-and-the-madness-of-crowds

Van Hee, C., et. al (2015). Automatic Detection and Prevention of Cyberbullying, HUSO 2015: The First International Conference on Human and Social Analytics, St. Julians, Malta, October 11 – 16, 2015. Wilmington, DE: International Academy, Research and Industry Association.

Van Royen, K., Poels, K., Daelemans, W. & Vandebosch, H. (2015, February). Automatic monitoring of cyberbullying on social networking sites: From technological feasibility to desirability. Telematics and Informatics, 32(1), pp. 89-97.

Vapnik, V., & Cortes, C. (1995, September). Support-Vector Networks. Machine Learning, 20(3), pp. 273-297.

Wahbeh, A., & Al-Kabi, M. (2012). Comparative Assessment of the Performance of Three WEKA Text Classifiers Applied to Arabic Text. ABHATH AL-YARMOUK: "Basic Sci. & Eng.", 21(1), pp. 15- 28.

What is Cyberbullying?. (2011). Retrieved from https://www.stopbullying.gov/cyberbullying/what-is-it/index.html

Williams, N., Zander, S. & Armitage, G. (2006). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Computer Communication Review, 36(5), pp. 5-16.