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

Samantha G. Mallari
Asia Pacific College
Makati, Philippines
sgmallari@student.apc.edu.ph

Faith I. Ballesteros
Asia Pacific College
Makati, Philippines
fiballesteros@student.apc.edu.ph

Eva R. Samillano
Asia Pacific College
Makati, Philippines
vrsamillano@student.apc.edu.ph

Miguel Paulo S. Capuz
Asia Pacific College
Makati, Philippines
pscapuz@student.apc.edu.ph

Lorena R. Rabago
Asia Pacific College
Makati, Philippines
lorenar@apc.edu.ph

Ernesto C. Boydon
Asia Pacific College
Makati, Philippines
ernestob@apc.edu.ph

ABSTRACT

Social media has brought a revolutionary change in terms of sharing information and communicating with other people. However, alongside with the advent of social media platforms, cyberbullying has become more prevalent. Studies on cyberbullying detection employ text classification approach and focused on optimizing the accuracy of the detection model. This research aims to extend the technological feasibility of automating the detection of cyberbullying into the generation of reports once a harmful post has been detected. It begins with the creation of a Support Vector Machine model 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. Once the model has been developed, 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 model as cyberbullying, a report which contains the author's name, 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.

KEYWORDS

Cyberbullying detection, Natural Language Processing, Text Classification, Support Vector Machine, and Crowdsourcing.

1 INTRODUCTION

The emergence of social media has created a new avenue for enabling online interactions. Nowadays, people easily connect with one other through chat rooms, email, instant messaging, forums, and social networking sites. (Sheoran, 2012). However, the increased use of social media, has led to cyberbullying becoming a major problem in the society (Alim, 2016).

Cyberbullying is defined as the use of electronic platforms to harass and harm an individual in a deliberate, repetitive and aggressive manner (Stopbullying.gov, 2014). Cyberbullying includes sending of insulting or threatening messages, disseminating false information, displaying humiliating photos, or excluding others in an online communications (Perren et al., 2012). Cyberbullying has becoming more prevalent than traditional bullying because of its characteristics and impacts. Bullies can easily their identity to their victims through fake profiles (Zimbardo, 1970). Moreover, bullying in the cyberspace can be viewed by a large amount of audience.

Philippines was recognized as the "Social Media Capital of the World", with 83 percent of Filipinos who use social networking sites. As the number of Filipino social media users gradually increase over time, the cyberbullying problem in the Philippines has intensified as well. A survey administered by Stairway Foundation Inc. revealed that 80% of Filipinos have been cyberbullied through social media (Takumi, 2016). Popular examples of cyberbullying cases in the Philippines include "Amalayer" incident, Malinay's prank involvement, and DJ Karen Bordador's cyberbullying experience following her arrest. 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 sites have also adopted 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 huge efforts made by the authority and social media 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. Thus, there is a need for technology to intervene in the process of mitigating online bullying.

In this paper, we explore the feasibility of automating the detection of cyberbullying posts in Filipino on social media sites by applying the concept of text classification and Support Vector Machine (SVM).

The rest of this paper is organized as follows: Section 2 presents the previous work in automating the process of cyberbullying detection while Section 3 provides an in-depth discussion of the concepts that were applied within the current research. Section 4 describes the processes that were involved in creating the cyberbullying detection model. Section 5 discusses the experimental evaluation phase using the dataset described earlier. Lastly, Section 6 presents the conclusions drawn from this study and discusses directions for future work.

2 RELATED WORKS

Throughout the years, several methods have been proposed for detecting cyberbullying instances in social media. Most of these methods approach the problem by treating it as a classification task, wherein messages are categorized into classes such as cyberbullying and non-cyberbullying.

Dinakar, Reichart, and Lieberman (2011) proposed a supervised machine learning approach in detecting cyberbullying instances. A number of 50,000 YouTube comments were gathered and classified into four categories: physical appearance, sexuality, race and culture, and intelligence. Their findings show that JRip yields the best performance in terms of accuracy while SVM is the most reliable as measured by kappa statistics. Moreover, the binary classifiers performed better than multi-class classifiers trained for all the labels.

Van Hee et. al (2015) focused on the linguistic characteristics of cyberbullying by classifying them into fine-grained categories, such as, threat, sexual talk, insult, curse, defense, defamation, and encouragement. They also identified the roles involved in a cyberbullying scenario: bully, victim, bystander-defender and bystander-assistant. They collected 91, 370 Dutch posts from Ask.fm and utilized Support Vector Machine (SVM) algorithm for the classification task. They obtained a Kappa score of 0.69 in detecting cyberbullying instances and a score of 0.52 to 0.66 for cyberbullying categories.

Dadvar, Jong, Ordeiman, and Trieschnigg (2012) used a Gender-Based Approach in detecting cyberbullying in Myspace. They employed Support Vector Machine in training the classifier in WEKA. Their dataset was provided by Fundacion Barcelona Media and it was composed of 381,000 posts. 34% of the posts were written by female while the remaining 64% were from male authors. Their proposed method improved the Baseline result by 39% in precision, 6% in recall, and 15% in F-measure.

A recent study conducted by Cheng and Ng (2016) focused on the detection of cyberbullying roles: accuser, bully, defender, reporter, and victim. A total number of 6000 comments/posts were gathered from Facebook and YouTube. They employed Support Vector Machine for detecting both cyberbullying incidents and roles. The optimal model yields an accuracy of 59.7% using 171 unique word features and a Kappa score of 42.3% in the detection of cyberbullying roles.

3 THEORETICAL BACKGROUND

3.1 Audience Segregation

Ervin Goffman introduced the mechanisms of audience segregation which describes how people play different roles in different situations in order to create a favorable image of themselves. This framework offers a way of thinking about how people may act differently depending on the audience and setting which are relevant to an exploration of cyberbullying. First, people can easily hide their identities through fake pictures, fictitious names, emails and numbers in the cyberspace. First, 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. Second, 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.

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 us 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 which 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

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.

3.3 Support Vector Machines

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.

3.4 Other Machine Learning Algorithms

3.4.1 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.

3.4.2 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.

3.4.3 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.

3.4.4 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.

3.4.5 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 is 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 optimised for the local subspace described by this leaf. Random trees has 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.

3.4.6 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 a 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 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:

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

3.4.7 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.

3.4.8 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.

3.4.9 Hoeffding Tree

Hoeffding Tree was derived from the Hoeffding bound that is used in the tree induction. The main idea behind this algorithm is Hoeffding bound gives certain level of confidence on the best attribute to split the tree. Thus, a model can be built based on certain number instances known. Hoeffding tree is suitable for mining data streaming because the learning time is constant. In order to achieve the streaming classification characteristics, Hoeffding bound was introduced to determine how many example of instances needed to achieve certain level of confidence. Moreover, it has the ability to produce the same results regardless of the probability distribution in generating the observations. However, the number of observations needed to reach certain values of and are different across probability distributions.

3.2.1.11 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.

3.2.1.3 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.

3.5 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). 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. Lastly, 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 1: Confusion Matrix

3.5.1 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.

3.5.2 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.

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

3.5.3 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.

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

3.5.4 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.

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

3.5.5 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.

3.5.6 Matthews Correlation Coefficient

The Matthews Correlation Coefficient (MCC) pertains to the quality of binary classifications. It takes into account all the values in confusion matrix: true and false positives and negatives. Moreover, it is a balanced measure which can be used even if the classes are of very different sizes.

$$MCC = \frac{\text{TP} * \text{TN} - \text{FP} * \text{FN}}{\sqrt{(\text{TP} + \text{FP}) * (\text{TP} + \text{FN}) * (\text{TN} + \text{FP}) * (\text{TN} + \text{FN})}}$$

4 METHODOLOGY

Our methodology contains two parts: the training phase, or the creation of the cyberbullying detection model, and the execution phase.

4.1 Training Phase

4.1.1 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. The data was extracted from these websites through Import.io, a web scraping tool.

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.

The application, however, 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.

4.1.2 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 was procedure was done using regular expressions.

4.1.3 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.

4.1.4 Tokenization

In this phase, all of the statements that were cleaned will be divided per each word within a particular statement based on the whitespaces 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 had 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.

4.1.5 Bag-of-Words

The dataset was transformed into a Bag-of-Words model, in which a set of text documents is converted into a numeric feature vectors wherein the order of word occurrences and grammar are ignored. It is primarily used as a tool for generating features. 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 we removed all words that contained digits.

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.

4.1.6 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.

4.1.7 Cyberbullying Detection Model

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.

4.2 Execution Phase

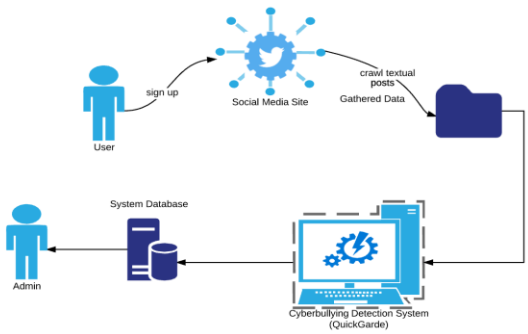

Figure 1: System Architecture

4.2.3 Gathering of public textual posts

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.

4.2.4 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. Through the use of replaceAll function, special characters and numbers were replaced by a space.

4.2.5 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.

4.2.6 Flagging of cyberbullying statements

In this feature, the classified statements will be explicitly annotated using the 3 specified schemes: "C", "NC", and "AC".

4.2.7 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.

5 RESULTS AND DISCUSSION

5.1 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, we increase the number of the training data for each run. To illustrate, we started with 200 data for the first run then added 300 more data for the second run. As for the third run, we added 200 more data and another 300 data for the fourth run. Table 5.10 depicts the accuracy and the kappa statistics of the model.

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 2: Baseline Results

As seen in Table 5.10, 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 5.0, 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).

5.2 Percentage Split

For this experiment, the dataset was divided into two parts: the training and testing. However, for each run, different splits was used. The purpose of this experiment is to determine the right split for our dataset. As shown in Table 5.11, the proper split for our dataset is 80/20.

# 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 3: Result of using Percentage Split

5.3 K-Fold Cross Validation

For this experiment, we 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. In k-fold cross-validation the data is first partitioned into k equally (or nearly equally) sized segments or folds. Subsequently k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k 1 folds are used for learning. 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 5.1 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 4: K-Fold Cross Validation

As shown in Table 5.12, the model can yield the highest accuracy and kappa score when the dataset was divided into

5.4 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 5.0 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 5: 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 5.0, both RandomForest 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), we 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 RandomForest with a score of 0.1712. ZeroR, Decision Stump, and HoeffdingTree have a kappa score of 0.

We also measured the performance of each model using precision (exactness) and recall (sensitivity). Among all models, RandomForest 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, we also take into account 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 5.0, 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, we also included Matthews Correlation Coefficient in measuring the accuracy of each model. 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), we also focused 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 6: Time in building the model

5.5 Issues

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

5.5.1 Evolution of Filipino Language

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

5.5.2 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 works 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 our 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.

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. Thus, other performance metrics must be considered in order to determine the optimal model for our classification problem. For this, we 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 can be 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 our accuracy needs. For this purpose, we used Matthew Coefficient Correlations, which is a more suitable metric in dealing with imbalanced data. 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).

6 CONCLUSION AND FUTURE WORKS

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 remains to be on a conservative level, 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, 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, we presented a novel approach that has a potential for detecting harmful messages and allowing social media administrators to provide timely responses.

We 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 our initial experiments, we only used accuracy and kappa statistics to measure the performance of the classifier despite of our imbalanced dataset because kappa itself is a good indicator of performance. However, in terms of comparing different machine learning algorithms, we used other metrics such as Precision, Recall, F-Measure, and Matthew Coefficient Correlations in identifying the best algorithm for our classification task. We found that 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. Our results show that SVM is still the best algorithm for our classification problem.

As for our recommendations and future works, we are planning to make Quickgarde even better by including additional studies in our work. First, by adding more data into our dataset. This will ensure that we can cover different variations in Filipino language such as Bekimon and Jejemon to improve the classification of cyberbullying statements. We are also planning to use different metrics such as ROC Area to further analyze the performance of our SVM classifier.

For future studies, a study on the effects of using Quickgarde in other data mining techniques such as sentiment analysis are highly encouraged. This will merely help us in identifying improvements for our work.

ACKNOWLEDGMENTS

The authors would like to express their deepest gratitude to Mr. Nicco Nocon for sharing his expertise, and sincere and valuable guidance and encouragement.

REFERENCES

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

[2] Boughorbel S, Jarray F, El-Anbari M (2017) Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. PLoS ONE 12(6): e0177678. https://doi.org/10.1371/journal.pone.0177678

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

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.

[4] 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
[5] 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
[6] 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
[7] 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
[8] 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.
[9] Goffman, E. (1956). The Presentation of Self in Everyday Life. New York, New York: Random House. [4] 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
[10] 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
[11] 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
[12] Lewis, D. D. (1992). Representation and Learning in Information Retrieval (Doctoral dissertation). Retrieved from UMI. (GAX92-19460)
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.
[13] 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
[14] Madnani, N. (N.D). Getting Started on Natural Language Processing with Python. ACM Crossroads, 13(4), pp. 5-5.
[15] 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.
[16]Nasa, Chitra & Suman, Suman. (2012). Evaluation of Different Classification Techniques for WEB Data. International Journal of Computer Applications. 52. 34-40. 10.5120/8233-1389.
[17] 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
[18] 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
[19] Salton, G. & McGill, M.J. (1986). Introduction to Modern Information Retrieval. New York, New York: McGraw-Hill Inc.
Research Papers
[20] 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).
[21] 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.
[22] 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
[23] 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
[24] 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
[25] 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
[26] 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.
[27] Vapnik, V., & Cortes, C. (1995, September). Support-Vector Networks. Machine Learning, 20(3), pp. 273-297.
[24]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.
[28] What is Cyberbullying?. (2011). Retrieved from https://www.stopbullying.gov/cyberbullying/what-is-it/index.html
[29] 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.
Boughorbel, S. Jarray, F. & El-Anbari, M. (2017). Optimal Classifier for Imbalanced Dataset using Matthews Correlation Coefficient Metric. PLos ONE 12(6):e0177678.