# **2. Review of 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. [20] This section focuses on the findings of the studies conducted with regards to cyberbullying.

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

Gonzales (2014) conducted a qualitative study to explore the relationship between social media and cyberbullying. [7] 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:

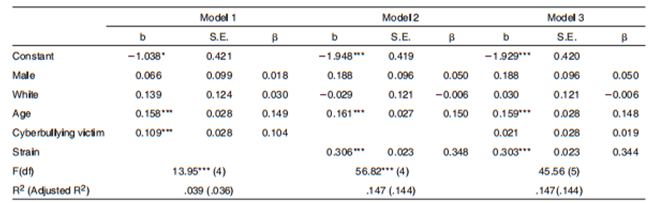
* Social media is the root cause of cyberbullying.
* There is no specific law in the Philippines that clearly defines punishable acts for cyberbullying.
* Self-discipline must be imposed by all social media users.
* Cyberbullying can be avoided, if people have a better understanding of social media.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

### **2.2.2 Offline Consequences of Online Victimization: School Violence and Delinquency**

Hinduja and Patchin (2012) conducted a study to determine the relationship between victimization, strain, and deviant behavioral choices of the cyberbullying victims. [21] 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 a 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 a 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 on the effect of cyberbullying victimization on offline problem behaviors.



*Table 2.0: 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. [22] 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.

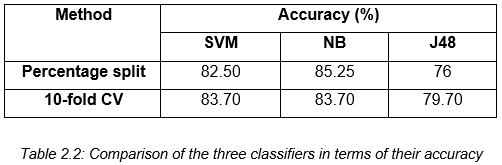
### **2.2.1 Comparative Assessment of the Performance of Three WEKA Text Classifiers Applied to Arabic Text**

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. [23]

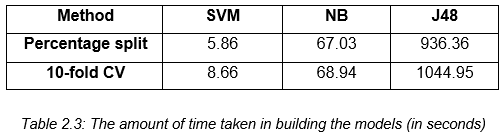
The first phase of their experiment begins with the creation of the corpus by gathering Arabic text documents from different websites: Kooora, news-all, and from Saheeh Al-Bukhari book and other websites. These data were already classified into number of categories such as Sport, Economic, Religion, 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 2.2 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 2.2, the 10-fold cross-validation has significantly improved the accuracy for each classifier.

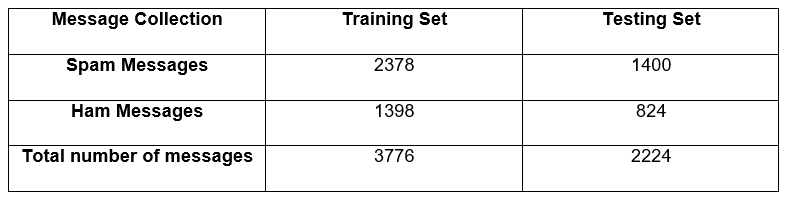
****

Aside from the accuracy of each classifier, they also measured the time taken for constructing each model. As shown in table 2.3, 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.



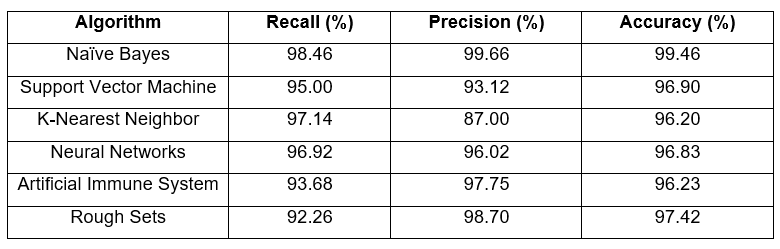
### **2.2.2 Machine Learning Methods for Spam Email Classification**

Awad and ELseuofi (2011) compared the performance of different machine learning algorithms in classifying spam emails. [24] 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 was divided the corpus into two sets: training and testing.

  
*Table 2.4: 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 2.5, Naïve Bayes outperformed the other classifiers in terms of precision, recall, and accuracy.

  
*Table 2.5: The performance of different machine learning algorithms in spam email classification*

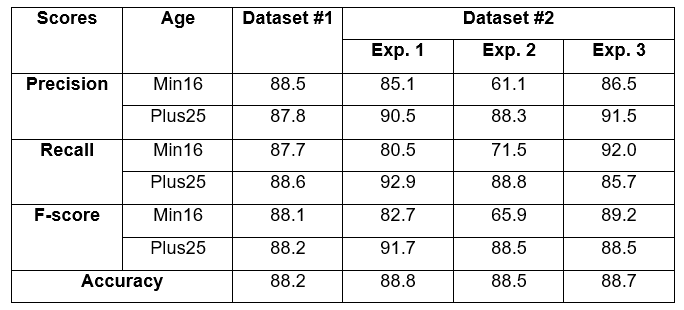
### **2.2.3 Predicting Age and Gender in Online Social Networks**

Peersman, Dalesman, 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. [25]

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 pre-processing 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 and then we first set up our experiments with 10,000 posts per class and 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 2.6 illustrates the overview of the results of the three experiments in comparison with the first dataset.

  
*Table 2.6: The result of the three experiments in comparison with the first dataset*

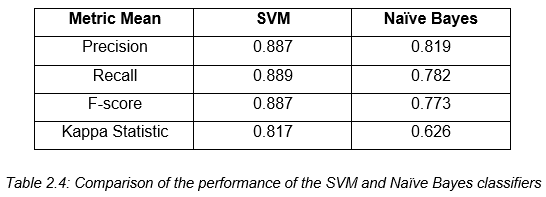
### **2.2.4 Classifying Typhoon Related Tweets**

In a study conducted by Lam, Paner, Macatangay, and Delos Santos (2014), [26] 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 2.7, the SVM classifier outweighs the performance of Naïve Bayes classifier in both metrics.



*Table 2.7: The comparison of the performance of SVM and Naïve 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. [15] This section focuses on the various methods used by different researchers in automating the process of detecting cyberbullying and its multiple forms (racial discrimination and offensive language). It also examines the effectivity of each approach.

### **2.3.1 Locate the Hate: Detecting Tweets against Blacks**

Kwok and Wang (2013) applied a supervised machine learning approach in detecting tweets which pertains to racial discrimination. [27] In their experiment, they designed a survey to gauge the complexity of identifying hate speech with the use of Fleiss’ Kappa to assess the reliability of agreement. They began by compiling a number of 100 tweets that contains keywords that are found in hate speech. Three annotators were assigned to classify whether a tweet was offensive or not, and the severity level of offensive tweets from a scale of 1 to 5 (with five being the most offensive). However, since the calculated percentage of overall agreement was only 33%, they assumed that it would be more difficult for machines to classify tweets accurately.

As for the classification of racist and nonracist tweets, they implemented Naïve Bayes classifier. The experimentation phase begins by constructing a dataset of racist and nonracist tweets. Moreover, a total number of 24,852 tweets were obtained. The preprocessing steps involve the removal of URLs, mentions, stopwords, and punctuation, transformation of each word to a lowercase, and normalization of words. By analyzing the tweets in the survey, they were able to derive labels such as offensive words, reference to painful historical contexts, stereotypes, threatening, and others. In addition to this, their feature set was limited to unigrams which results into 9437 unique words in the racist training dataset and 8401 unique words in the nonracist training dataset.

A 10-fold cross-validation method was utilized to evaluate the performance of their classifier. Furthermore, they were able to achieve an accuracy of 76% and an error rate of 24%.

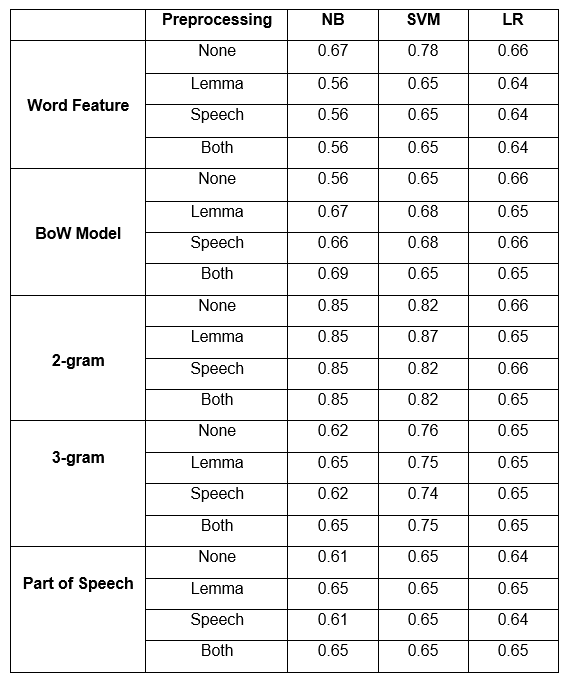
### **2.3.2 Comparison of Machine Learning Algorithms for Offensive Language Filtering**

Buckman (2012) employed different machine learning techniques to identify and filter players of the videogame League of Legends, who are responsible for authoring offensive posts. [28] The experimentation phase begins with the construction a corpus by crawling a Python script which utilizes the BeautifulSoup library in the Riot Games Website. A total number of 30011 cases were downloaded and stored in JavaScript Object Notation. The first 11 cases were used to aid in the development of the software and the remaining 30000 cases were allocated for the test corpus. The dataset was divided into two partitions: training and testing. Each dataset contains a number of 100 cases.

The first step in preprocessing the data involves the use of a control method wherein each text was left unaltered. A spell checker was also applied to fix the common errors and replace misspellings with the proper words. This process was done using Enchant software package and the aspell dictionary. In addition to this, the words were lemmatized using Python’s Natural Language Toolkit.

The process of feature extraction involves the use of n-gram models, grammatical parsing, and word features. The n-gram models were programmed using Python. In grammatical parsing, each word was tagged with one of the Penn Treebank POS tags such as coordinating conjunction, comparative adjective, and personal pronouns. The word features that were extracted are word length, words per line, number of capital letters, and letters per word. A total number of 10 features were obtained.

For their experiment, they employed three machine learning algorithms: Naïve Bayes, Support Vector Machine, and Logistic Regression. As shown in Table 2.8, SVM outweighs the performance of Naïve Bayes and Linear Regression with an accuracy of 87%, achieved by extracting 2-grams from data which had been lemmatized.

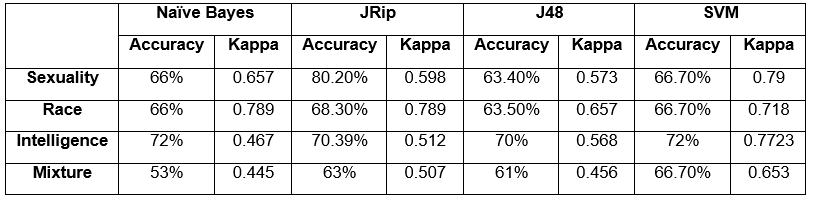


*Table 2.8: Accuracies of Trials*

### **2.3.3 Modeling the Detection of Textual Cyberbullying**

Dinakar, Reichart, and Lieberman (2011) proposed a method in creating a cyberbullying detection model. [20] 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 clusters 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.

  
*Table 2.9: The comparison of the performance of Naïve Bayes, JRip, J48 and SVM*

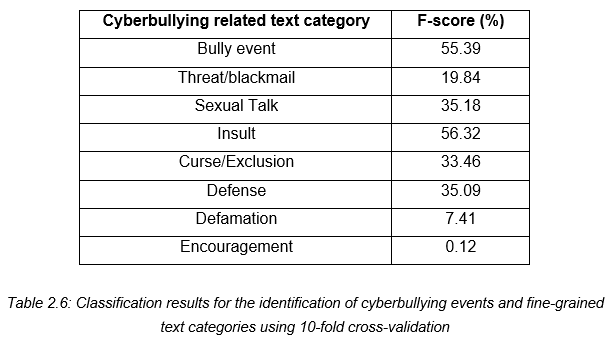
As shown in Table 2.9, JRip yields the best performance in terms of accuracy while SVM is the most reliable as measured by kappa statistics. In addition to this, the binary classifiers trained for each individual labels performed better than multi-class classifiers trained for all the labels.

### **2.3.4 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.  [18] 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, and Threat vary from 0.52 to 0.66.

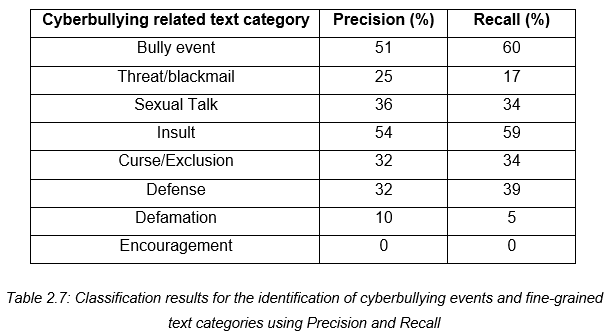
The preprocessing steps involved tokenization, PoS-tagging and lemmatization to the data by utilizing LeTs Preprocess Toolkit. They implemented two types of lexical features for their experiment: bag-of-word features and polarity features based on existing sentiment lexicons. Thus, it results into a 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 2.10 shows the result of their preliminary experiment by using F-score.



*Table 2.10: Classification results for the identification of cyberbullying events and fine-grained text categories in terms of F-score*

Table 2.11 illustrates the performance of both precision and recall with regards to the identification of cyberbullying event and their classification into fine-grained text categories.



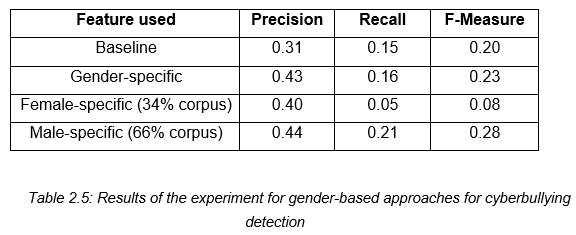
*Table 2.11: Classification results for the identification of cyberbullying events and fine-grained text categories in terms of precision and recall*

### **2.3.5 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. [29]To test this idea, they conducted an experiment on impdadvarroving 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 2.5, the incorporation of gender-specific features improved the overall accuracy measures.



*Table 2.12: The result of improving cyberbullying detection using gender-specific features*

### **2.3.6 Automated Role Detection in Cyberbullying Incidents**

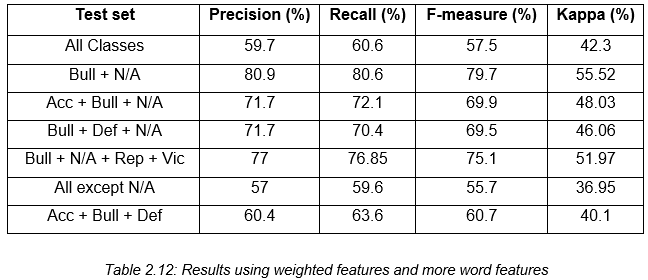
Cheng and Ng (2016) conducted an experiment on the detection of cyberbullying roles. [30] Their experiment begins with the creation of a corpus by gathering data from Facebook and Youtube. In total, 6000 posts/comments written in both English and Tagalog were collected (1500 for YouTube and 4500 for 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 the any of the class).

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)

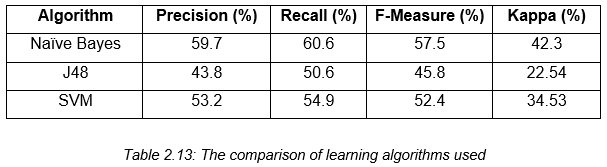
For their first experiment, they used an initial set of 25 word features in each class. They checked the presence of both words that are written in all capital letters and those which contains profanity. From a total number of 150 features, it was decreased into 93 unique word features. Their second 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. As shown in Table 2.9, there was a decrease in all measures. 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 a 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.



*Table 2.12: The result of the last experiment*

Lastly, the experiment that yield the highest accuracy was tested using different algorithms: Naïve Bayes, J48 and Support Vector Machine. As shown in Table 2.12, among the three algorithms that were utilized, SVM yield the highest accuracy.



*Table 2.13: The comparison of the performance of Naïve Bayes, J48 and SVM*

*Note: Ms., so far we only have 10, but we’ll be adding more literatures in the following weeks and add the synthesis part as well. This RRL is still incomplete.*