Cyberbullying Classification using Text Mining

**The purpose of this research is to construct a classification model with optimal accuracy in identifying cyberbully conversation using Naive Bayes method and Support**

**Vector Machine (SVM) then applying n-gram 1 to 5 for the number of class 2, 4, and 11 for each method. Naive Bayes yields an average accuracy of 92.81%, SVM with a poly kernel yields an**

**Average accuracy of 97.11%.**

*Data Collection*

The data used to create a data set is a textual conversation taken from the Kaggle (www.kaggle.com) which provides 1,600 conversations in Formspring.me.

*Preprocessing*

***Data Cleaning & Data Balancing:***The amount of data obtained from *www.kaggle.com* is 12,729 data, including 11,661 data given non-cyberbullying label and 1068 data labeled cyberbully. Data cleaning is done with Microsoft excel by eliminating conversations that have total characters under 15 letters, deleting meaningless words like "haha", "hehe", "wkwk", "emm", "umm". For the purposes of data balancing on the classification of 2 classes􀀃 􀀋cyberbully, non-cyberbully), 4 Classes (non-cyberbully, cyberbully level severity low, cyberbully level severity middle, cyberbully level severity high), and 11 classes (non-cyberbully, cyberbully level severity 1 – 10), then the data used amounted to 1.600 for balancing data (800 labeled cyberbully and 800 labeled non-cyberbully) with the following allocation:

a) 2 Class: each class amounts to 800 data

􀁸 Class No: 800 data with label severity 0

􀁸 Class Yes: 800 data with label severity 1-10

b) 4 Class: each class amounts to 240 data

􀁸 Class No: 240 data with label severity 0

􀁸 Class Low: 240 data with label severity 1 – 3

􀁸 Class Middle: 240 data with label severity 4 – 7

􀁸 Class High: 240 data with label severity 8 – 10

***Tokenization****:* tokenization is the process of cutting or separating each word that compiles a document or conversation.

***Transform case***: Transformation into the lower case.

***Stop Word Removal****:* Delete unnecessary words on every text.

***Filter Token:***The token filter is selecting the word that the number of characters between 3-25.

***Stemming:***The words on the text conversation are transformed into a basic word.

***Generate n-grams****:* The process of generating n-grams into form a set of words from a portable and graph, usually by moving one word forward, in this research a n-gram of 2 to 5*,* because the experiments have been done n-gram over 5 is stable

***C. Extraction***

The preprocessing text conversations will be transformed into a vector space model where text conversations are represented with a vector of extracted features. Features resulting from the extraction are words or combinations of words to form a list of words and the calculation of the weight

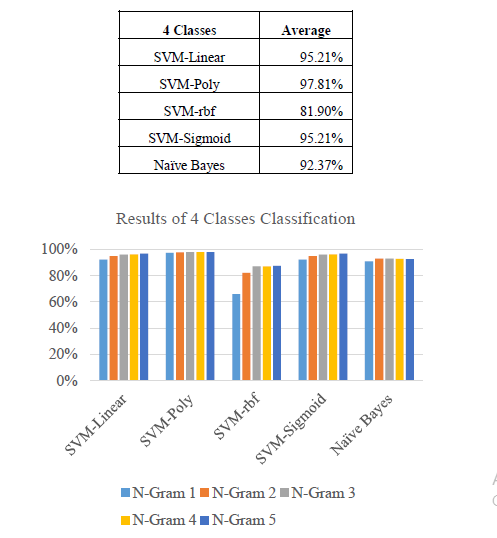
With TF-IDF.

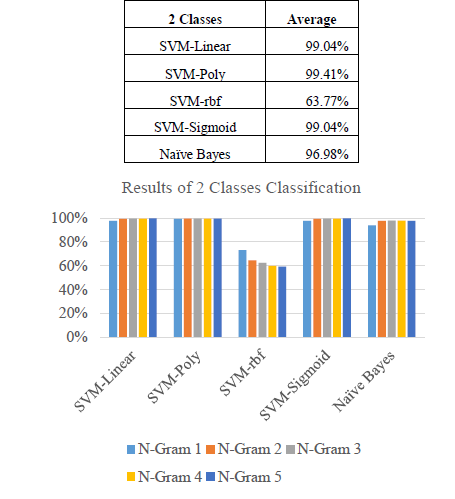
***D. Classification***

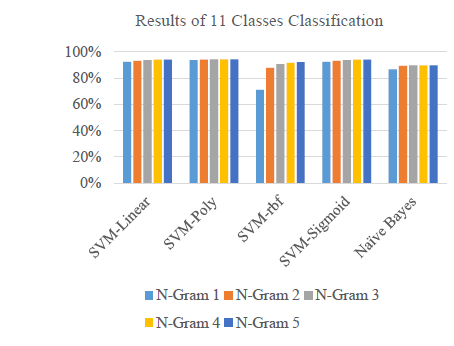
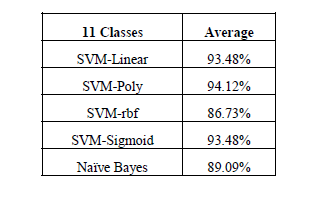
In this stage the classification will use the Naïve Bayes & SVM method with linear, poly, RBF, and sigmoid kernels. Each conversation in the form of questions and answers is combined into one text conversation. The collected text conversations are randomly divided into sets of training and test data.

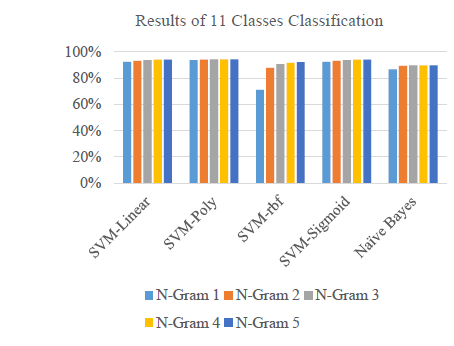
Each text conversation consisting of 1600 conversations is labeled according to the data set and text conversation status. The division of text conversations into data sets is done 10 times

**IV. RESULTS**

SVM 2 Classes SVM and Naïve Bayes 4 Classes



SVM and Naïve Bayes 11 Classes.



The most optimal SVM kernel in classifying cyberbullying is the Poly kernel with an average accuracy of 97.11%, because of the data used in this study are non-linear separable. Therefore, the optimal function for separating the sample into different classes is SVM with poly kernel. The application of n-gram may increase the accuracy level in cyberbullying classification, due to the highest accuracy level at n-gram 5 (92.75%), the lowest accuracy set at n-gram 1 (89.05%).

 TF-IDF is based on the bag-of-words (BoW) model, therefore it does not capture position in text, semantics, co-occurrences in different documents, etc.  
- For this reason, TF-IDF is only useful as a lexical level feature  
- Cannot capture semantics (e.g. as compared to topic models, word embeddings