Using Naïve Bayes Algorithm in detection of Hate Tweets.

Kelvin Kiema Kiilu, George Okeyo, Richard Rimiru, Kennedy Ogada

Department of Computing, Jomo Kenyatta University of Agriculture and Technology

DOI: 10.29322/IJSRP.8.3.2018.p7517
http://dx.doi.org/10.29322/IJSRP.8.3.2018.p7517

Abstract- Social Media has become a very powerful tool for information exchange as it allows users to not only consume information but also share and discuss various aspects of their interest. Nevertheless, online social platforms are beset with hateful speech - content that expresses hatred for a person or group of people. Such content can frighten, intimidate, or silence platform users, and some of it can incite other users to commit violence. Furthermore, social media gives users the freedom to express their thoughts in text without following traditional language grammars, thereby making it difficult to mine social media for insights. Despite widespread recognition of the problems posed by social media content, reliable solutions even for detecting hateful speech are lacking. The main goal of this study is to develop a reliable tool for detection of hate tweets. This paper develops an approach for detecting and classifying hateful speech that uses content produced by self-identifying hateful communities from Twitter. Results from experiments showed Naive Bayes classifier achieved significantly better performance than existing methods in hate speech detection algorithms with precision, recall, and accuracy values of 58% ,62%,and67.47%,respectively.

Index Terms- Hate tweets, Naive Bayes, Text Classification, Sentiment analysis.

I. INTRODUCTION

In recent years, Twitter has become one of the most popular micro-blogging social-media platforms, providing a platform for millions of people to share their daily opinions/thoughts using real-time status updates Conover et al. (2013). Twitter has 270 Million active users and 500 million tweets are sent per day. M.C. Wellons, (2015). Due to high reachability and popularity of social media websites worldwide, organizations also use these websites for planning and mobilizing events for protests and public demonstrations Muthiah et al. (2015).

Twitter is a famous platform for opinion and information sharing and this platform is mostly used before, during and after live events Bollen et al. (2011).

Online spaces are often exploited and misused to spread content that can be degrading, abusive, or otherwise harmful to people. Twitter prohibits users to post violent threats, harassment, and hateful contents. However, there are still tons of users who disobey the rules and use their Twitter account to spread hate speech and negative words.

An important and elusive form of such language is hateful speech: content that expresses hatred of a group in society. Hateful speech has become a major problem for every kind of online platform where user-generated content appears: from the comment sections of news websites to real-time chat sessions in immersive games. Such content can alienate users and can also support radicalization and incite violence Allan, (2013). It is through such access to Twitter where various users have used the platform to propagate and promote hatred tweets to various target groups and individuals Wilkinson, (1997).

No formal definition of hate speech exists but there is a consensus that it is speech that targets disadvantaged social groups in a manner that is potentially harmful to them Jacobs & Potter 2000; Walker 1994). In Kenya, hate speech has been defined as any form of speech that degrades others and promotes hatred and encourages violence against a group on the basis of a criteria including religion, race, color or ethnicity. It includes speech, publication or broadcast that represents as inherently inferior, or degrades, dehumanizes and demeans a group. (KHRC, 2010).

Importantly, the definition does not include all instances of offensive language because people often use terms that are highly offensive to certain groups but in a qualitatively different manner.

Anything tweeted can reach a huge number and the effects can be extensively great.

We were concerned with the task of detecting; identifying and analyzing the spread of hate speech sentiments in the social site and specifically twitter in Kenya. Sentiment analysis is an area of natural language processing which aims at determination of opinions, attitudes of a writer in the text or their attitude towards specific topics. Sentiment describes an opinion or attitude expressed by an individual, the opinion holder, about an entity, the target. The research field of sentiment analysis has developed algorithms to automatically detect sentiment in text Pang & Lee, (2008). Whilst some identify the objects discussed and the polarity (positive, negative or neutral) of sentiment expressed about them Gamon et al. (2005), other algorithms assign an overall polarity to a text, such as a movie review Pang & Lee, (2004). Three common sentiment analysis approaches are full-text machine learning, lexicon-based methods and linguistic analysis. For standard machine learning e.g., Witten & Frank, (2005), a set of texts annotated for polarity by human coders are used to train an algorithm to detect features that associate with positive, negative and neutral categories. The text features used are typically sets of all words, word pairs and word triples found in the texts. The lexicon approach starts with lists of words that are pre-coded for polarity and sometimes also for strength .It uses their occurrence within texts to predict their polarity. A linguistic analysis, in contrast, exploits the grammatical structure of text to predict its polarity, often in conjunction with a lexicon.

For instance, linguistic algorithms may attempt to identify context, negations, superlatives and idioms as part of the polarity prediction process e.g., Wilson, Wiebe, & Hoffman, (2009). In practice, algorithms often employ multiple methods together with various refinements, such as pre-filtering the features searched for Riloff, Patwardhan, & Wiebe, (2006), and methods to cope with changes in data over time Bifet & Frank, (2010).

This paper presents an approach based on Naïve Bayes to detect hate speech/tweets. The approach involves tweet acquisition and streaming using Tweepy API, pre-processing to remove unwanted parts of speech using n-grams, and tweet classification and evaluation using Naïve Bayes. The collection of tweets was selected so that it contained a variety of words, expressions, emotional signals as well as indicative examples of sarcastic, ironic, metaphoric language. We developed a sentiment analysis classifier that processes tweets in real-time and uses supervised learning techniques to analyze and classify their sentiments.

The rest of this paper is organized as follows: Section 2.0 provides an overview of related work. Section 3.0 describes our approach. In Section 4.0, we present the classifier implementation for real-time sentiment analysis using naïve Bayes and the experimental results and discussion and in Section 5.0 we conclude and highlight the future work.

II. Related work

Warner and Hirschberg (2012) detected hate speech on the basis of different aspects including religion. They defined hate speech in their work and then gathered data from Yahoo and American Jews Congress (AJC), where Yahoo provided its data from news groups and AJC gave URL marked as offensive websites. They classified data at paragraph level in their first attempt and then used this data set for annotation by asking annotators to manually annotate the data set. They focused on stereotype and thus decided to make language model for stereotypes to mark hate speech. They made an anti-Semitic speech classifier first. They identified 9000 paragraphs matching to their regular expression and then removed those paragraphs that were not offensive. Then further seven categories were chosen to annotate the data. After this annotation for their gold corpus, they used two fold cross validation classifier to find a refined data set.

Motivated by work done in Kwok and Wang, (2013) proposed a method for detecting hatred speech against black over Twitter. They arranged hundreds of tweets to analyze keywords or sentiments indicating hate speeches. To judge the severity of arguments, a questionnaire was floated to students of different races. A training dataset of 24582 tweets was preprocessed to correct spelling variation, remove stop words and eliminate URL etc. In order to classify tweets, NB classifier highlighted racist and nonracist tweets and prominent feature were identified from those tweets. The classifier showed an accuracy of 86%.

Burnap et al. (2013) developed a rule-based approach to classifying antagonistic content on Twitter and they used associational terms as features. They also included accusation and attributional terms targeted at a person or persons following a socially disruptive event as features, in an effort to capture the context of the term use. Their results demonstrated an improvement on standard learning techniques.

Ting et al. (2013) proposed architecture for discovering hate groups over Facebook with the help of social network and text mining analysis. They extracted features including keywords that are frequently used in groups. Sureka et al. (2012) proposed an approach based upon the data mining and social network analysis for discovering hate promoting videos, users and their hidden communities on YouTube. Chen et al. (2013) presented a framework for identification of extremist videos on YouTube. Author extracted lexical, syntactic and content specific features from user generated data and used various feature based classification techniques in order to classify videos. Agarwal and Sureka proposed a focused crawler (bestfirst search and shark search) based approach for retrieving YouTube user profiles promoting hate and extremism.

Our approach is different from above work because most of available work done on hatred speech is on a single topic or domain depending on the emerging/developing trend on a given period of time. They have tried to focus different subject at time different study like religion, race, ethnicity etc. In this work, we have given an approach to focus on generic issues. Secondly our approach is different from above because we use the stream real time tweets as well as user profile information .In addition carrying experiments with different classifiers, as well as different feature sets consisting of unigrams, bigrams, and the combination of the two is very unique approach.

III. Approach for detecting Hate Tweets

3.1 Description of Approach

The approach consists of the following steps:

Step-1 Creating a dataset**.** First we streamed tweets to build classifier with the help of Tweepy library in python and store the tweets in the database. (Refer to fig 1, Architecture diagram of proposed system).

Step-2 Then we pre-processed these tweets, so that they can be fit for mining and feature extraction.

Step-3 After pre-processing we passed this data in our trained classifier, which then classify them into positive or negative class based on trained results, which will enable in analyzing how hate tweets are promoted, disseminated and how can be curbed.

The steps are realized using the architecture shown in Fig 1.

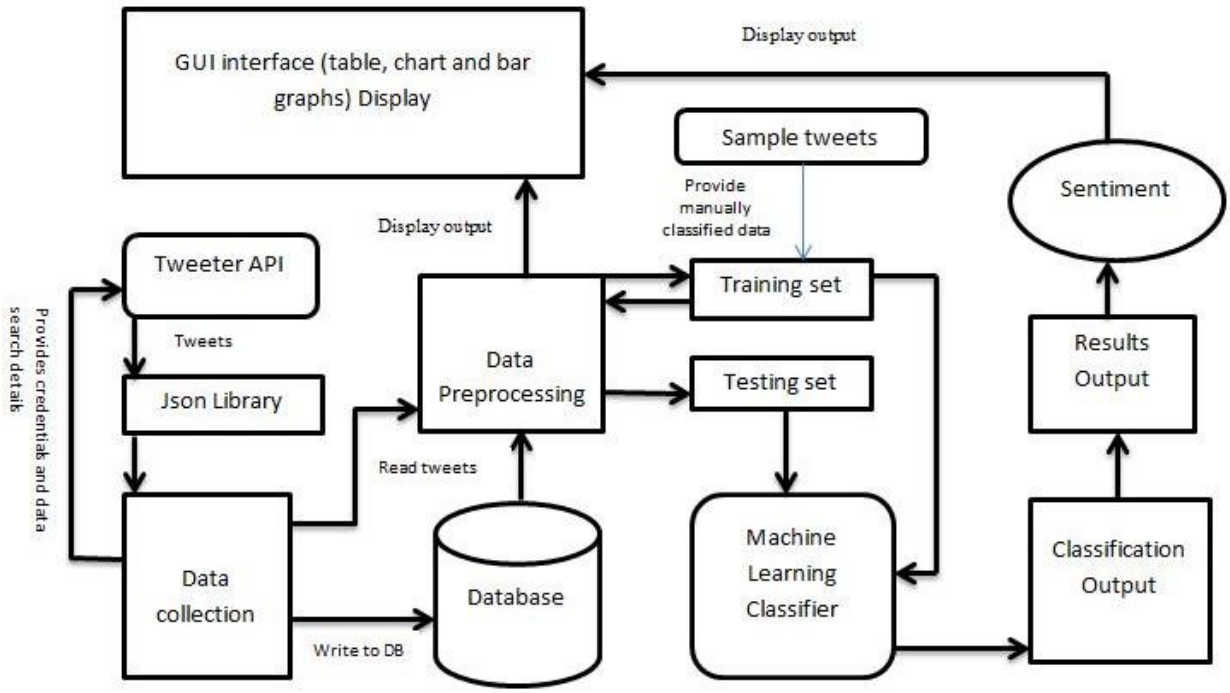

Fig1: Architecture diagram of proposed system

3.2 Training and Test Data

We collected training data of 45645 tweets and test data of 22820 tweets. This training data was obtained from sample sentences and words from a text file, which had been classified manually.

The Test data is fetched from the twitter using tweeter API. These data is loaded to the trained classifier for sentiment analysis.

3.3 Tweets Pre-processing

The language employed in Social Media sites is different from the one found in mainstream media and the form of the words employed is sometimes not the one we may find in a dictionary. Further on, users of Social Media platforms employ a special "slang" (i.e. informal language, with special expressions, such as "lol", "omg"), emoticons, and often emphasize words by repeating some of their letters. Additionally, the language employed in Twitter has specific characteristics, such as the markup of tweets that were reposted by other users with "RT", the markup of topics using the "#" (hash sign) and of the users using the "@" sign.

All these aspects must be considered at the time of processing tweets. As such, before applying supervised learning to classify the sentiment of the tweets, we preprocess them, to normalize the language they contain. We create a code in Python in which we define a function which will be used to obtain processed tweet. This code is used to achieve the following functions:

- remove quotes - provides the user to remove quotes from the text
- remove @ - provides choice of removing the @ symbol, removing the @ along with the user name, or replace the @ and the user name with a word

'AT_USER' and add it to stop words for example @kevoyoung to kevoyoung

- remove URL (Uniform resource locator) - provides choices of removing URLs or replacing them with 'URL' word and add it to stop words
- remove RT (Re-Tweet) - removes the word RT from tweets
- remove Emoticons - remove emoticons from tweets and replace them with

their specific meaning

- remove duplicates – remove all repeating words from text so that there will be no duplicates
- N-Grams-The implementation of creating n-grams in this project is done using the nltk.util.ngrams() function. This process starts by creating a five-gram of the tweet tokens. This means a sequence of five tokens will be created from the array of tokens. The system utilizes a five-gram sequence due to potentially long software names, basing this on the naïve assumption that these names will not exceed five words. This will allow for improved extraction of software names in the next stage. Using the previous tweet as a running example, the outcome of this five-gram modelling process can be seen below.

['I', 'really', 'hate', 'the', 'new', 'Firefox']
)
[(('I', 'PRP'), ('really', 'RB'), ('hate', 'JJ'), ('the', 'DT'), ('Luo', 'JJ')),

- Remove # - removes the hash tag class.

- Lowercase Conversion: Tweet may be normalized by converting it to lowercase which makes it's comparison with an English dictionary easier.
- Stop words- In information retrieval, it is a common tactic to ignore very common words such as \a", \an", \the", etc. since their appearance in a post does not provide any useful information in classifying a document. Since query term itself should not be used to determine the sentiment of the post with respect to it, every query term is replaced with a QUERY keyword.
- Part-of-speech (POS) Tagging-The POS tagger used by this system is taken from the NLTK modules and uses the pos_tag() function which takes a tokenized sentence as its only argument. Continuing from the first example, this process tags as follows:

['I', 'really', 'hate', 'the', 'luo', '']

[('I', 'PRP'), ('really', 'RB'), ('hate', 'JJ'), ('the', 'DT'), ('Luo', 'N')

- PRP Pronoun
- RB Adverb
- JJ Adjective
- DT Determiner

NNP Proper Noun

- Language detection- Since we are mainly interested in English text only. All tweets have been separated into English and non-English data. This is possible by using NLTK's language detection feature.
- Identifying the sentiment of the sentence whether it is positive or negative depending on the number of words filtered from the sentence against the positive and negative text files, which are labeled as either features in the classification or training set. The tokenized sentenced is referred to as test data/set to be used against the training set.

3.4 Tweet Classification

We used scikit-learn to conduct the experiments. The goal of the classification stage is to assign two classes to each tweet, one describing the sentiment of the tweet and one describing the subject matter discussed in the tweet. To build our classifier we used a library of Python called, Scikit-learn. Scikit-learn is a very powerful and most useful library in Python due to its support of multinomial naïve Bayes classifiers, whereas NLTK only has built-in support for naïve Bayes classifiers based on the Gaussian distribution. Scikit-learn also include tools for classification, clustering, regression and visualization. To install Scikit-learn we simply use on line command in python which is 'pip install scikitlearn'. This section focuses on the process of classifying the data taken from twitters API. The tweets were read from the database and converted into Json format so that it could be processed by python. They were then loaded into the python and was created in the previous section was used to classify the sentiment in each tweet into the positive or negative class.

When this classification was complete the results were saved in a text file. As there were a large number of tweets in the dataset a python program was created to calculate the percentage positive and negative tweets in the file and to visualize the results.

Table 1: Showing some sampled data.

	TWEETS DETAILS
1	['Mon Jun 26', '00:34:49', 'Tom', 0, 0, '', b'There are teenaged girls who enjoy this. https://t.co/c9PJ7rJQGk (thanks @vice https://t.co/EKmwPXeATT)', 'negative']
2	2017-09-26 06:46:39 | JP_jokopae | Are u making love to Babu? https://t.co/LYtqfKmtb3 | Nairobi
3	['Thu Jun 15', '13:33:42', 'Tom', 0, 7821, '', b"RT @RandPaul: .@Judgenap: Why do we have a Second Amendment? It's not to shoot deer. It's to shoot at the government when it becomes tyrann\xe2\x80\xa6", 'negative']
4	2017-09-26 06:46:15 | Almost9famous | Tell me...Is been stupid a profession or its just one of your God's gifts??? Babu Owino | Nairobi
5	['Fri Jun 16', '00:16:47', 'Tom', 0, 0, '#FamilyFeud ', b'Bringing small penis to primetime, it\xe2\x80\x99s @AmySchumer with the 1st BZZZZT of the round. #FamilyFeud', 'negative']
6	['Fri Jul 21', '03:08:16', 'Tom', 0, 7417, '', b'RT @BettyBowers: A president pardoning himself for crimes is an admission that he committed them. In other words, the only exhibit necessar\xe2\x80\xa6', 'negative']
7	2017-09-26 06:45:00 | Kamzy___ | @KahindoKaruga Why you big mad someone overtook you □ | Nairobi
8	2017-09-26 06:48:50 | FloOlivia | Women do not scale up their businesses. 58% of SME's in Kenya are owned by women but only contribute to 20% of GDP. #WimKe | Nairobi
9	Freedom of PRESS is Freedom of CONSCIENCE. @Mutahingunyi Our JOURNALISTS are PRISONERS of CONTRACT. They are NOT FREE. What a SHAME!
10	['Tue Jun 06', '20:33:46', 'Tom', 0, 545, '', b'RT @JohnKiriakou: .@theintercept should be ashamed of itself. Matthew Cole burns yet another source. It makes your entire organization untr\xe2\x80\xa6', 'negative']

IV. Experiments and Results

4.1 Experimental setup.

We implemented the python NLTK package for Naive Bayes classification. The training sets need to be labelled in order to recognize the category a corpus is classified upon. The labelled tweets were then stored in MongoDB.

Data collection is done by a few steps, do login twitter, do registering on API twitter to get the access token, and then create scripts for crawling data and input access token that has been obtained in API twitter, then save log data in the database in the form of JSON files. Second, doing the analysis preprocessing and data cleansing with the method described previously to get structured data. Third, classification is done using naïve Bayes classifier and manual classification which is performed on the data that has been cleaned.

The data in this paper was collected using Twitter API (Tweepy). The API helps to retrieves tweets for any user or hashtags from twitter platform .With the help of the API the data containing hashtag hate tweets is collected and saved in the Mongo database. The data collected consists of 45000 raw tweets in Json format .This data contains Full name of person tweeted, Tweet text, Tweet ID, user screen name, date and time of tweet(s). The type of system by which we had uploaded the data, the number of followers the user has, the number of retweets, location of the user and whether user is verified or not. All these functionalities were achieved through a GUI interface.
The interface is executed by interpreting the python codes in the Pycharm-SDK. All the codes for the project are written in the Pycharm SDK platform. Once the interpretation was done the GUI was displayed with all the functionalities present for the project. Furthermore, for the purposes of the experiment, other python libraries were included in the project to add functionality and smooth running of the project .These libraries included:

Json,Csv,NLTK(Corpus,stopwords),PyMongo(MongoDBconnector),Numpy,Matplotlib,Datetime,Pyside (design of the GUI interface)

By iterating through the training set, Naive Bayes classifier finds out the number of occurrences of each bigram word and checks if the test sentence has the same feature words as the training data. After the preprocessing of training set was complete, the bigram feature vectors were extracted from every tweet.

Observing the tweets and converting the data to csv format, it was observed that some of the tweets contained information for instance emotions icons, slangs among others which were filtered to a point in which the tweets were comprehensible. Moreover, this extra information was not fully filtered due to constraint of the python libraries and codes used during the Data preprocessing step.

In experimenting with various Classifiers, we used the metric accuracy, precision and recall measurers for the Classifier. Performance metrics are used for the analysis of classifier accuracy. The proposed system was evaluated using accuracy.

4.2 Results.

Classifying tweets for sentiment analysis in this project, classifiers should have the capability to detect where a sentence is positive or negative. Thus how well it achieves this is relatives to two mistakes that any supervised classifiers can make. These are false positive and false negative.

In the Classification task, for a sentence to be positive means its sentiment is positive and vice verse, hence a false positive indicates that a tweet has been labeled positive while it's not. The Same logic applies to false negative that it is labeled as negative while it's positive.

Give the two mentioned errors, classifiers used need to be vetted using metric to establish their effectiveness.

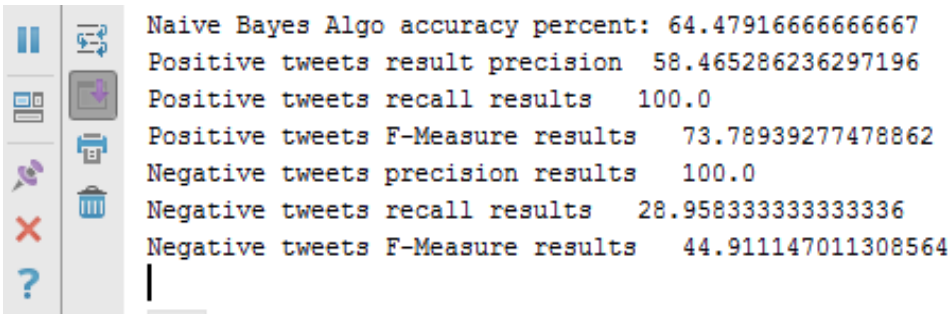

Figure 1: illustrating the metric measures as captured in the experimentation process.

Figure 2: Block diagram of various metrics scores.

Every instance that is positive is correctly identified as such, with 100% recall. This means very few false negatives in the positive class.

1. But, a file given a positive classification is only 53% likely to be correct. Not so good precision leads to **47% false positives** for the positive label.
2. Any file that is identified as negative is 100% likely to be correct (high precision). This means very few false positives for the negative class.
3. But many files that are negative are incorrectly classified. Low recall causes **87% false negatives** for the negative label.

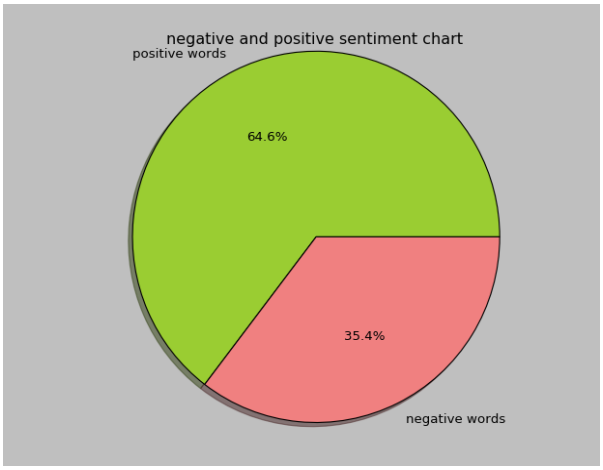

Figure 3: Pie chart for sentiment analysis

The Pie chart in fig 4 represents the total number of words of each and every sentence in the test data that has been classified as either positive or negative from the classifier.

The hate tweets dataset used for this work is provided by table 1. Experimental setup contains simulation environment, parameters and performance metrics. Generally performance metrics are used for calculating metrics like size, execution time, performance accuracy of the system

Table 2: Classification of tweets using sentiment analysis classifiers.

Dataset	Classifier	Accuracy (Unigram)	Accuracy(Bi gram)
Hate tweet data	Naïve Bayes classifier-NLTK	64.47%	70%
	NU Support Vector Classification (NUSVC)	53.69%	59%
	MultinomialNB-(Sklearn)	56.25%	66%
	BernoulliNB(Sklearn)	56.25%	66%
	Logistic Regression(Sklearn)	56.25%	66%
	Linear SVC	51.46%	66%
	SGD classification(Sklearn)	55.21%	53.01%

In addition, we decided to use scikit-learn for supposed robustness in handling big data. Scikit-learn is a free software machine learning library for the Python programming language. This software is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn have an expansive list of available algorithms. Besides, we decided to use scikit-learn for supposed robustness in handling big data. Scikit-learn is a free software machine learning library for the Python programming language. This software is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn has an expansive list of available algorithms.

The results obtained from the three machine learning approaches are based on their precision, recall, accuracy and f-score. It can be observed from Figure 5.4 that we obtained over 78% in all four evaluations metrics by using the features of non-sentiment bearing hashtags to classify tweets.

As shown figure 2, NLTK a positive precision of 56.04% and a recall of 100%.

These are very low values for precision and recall. It means that only 56.04% of the positive tweets retrieved by the classifier were relevant and 100% of the relevant positive tweets were retrieved.

One reason for such high positive precision and recall is because the context of training tweets were mostly cyberbullying related, which means they had a lot of slang and hate words.

On the negative results since our context is hate speech. The negative precision came out to be 98.76% and recall was 22.84%. This means that most of its predicted sentiment was accurate when compared to its training set by a small amount. Hence 98.76% of the negative tweets were relevant and 22.84% of the relevant negative tweets were retrieved. This means very few *false positives* were found for the negative class. However, many tweets that are negative are incorrectly classified.

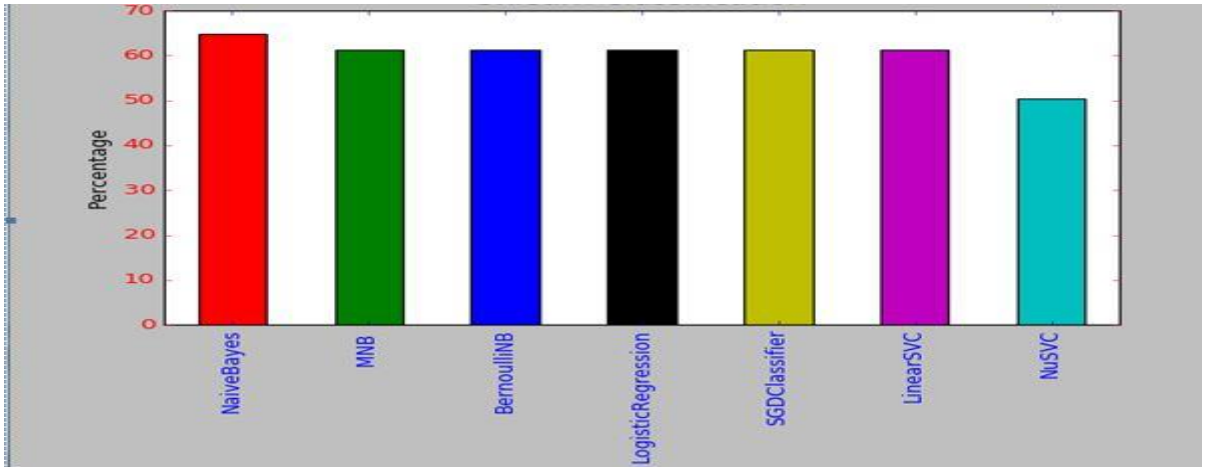

Figure 4:Block diagram showing various accuracy metrics.

As shown in fig 3, the accuracy of Naive Bayes was the best when compared to other approaches MNB being the second. Naive Bayes was able to correctly predict sentiment with an accuracy of 63.39% and Naive Bayes having an accuracy of 60.50%. The lowest performance was 49.6% accuracy which was by Convolutional NuSVC.

Fig 4 illustrates the tweets being fetched from twitter in real time.

It displayed the time of creation, the screen name, tweet text and Location where the user is tweeting from

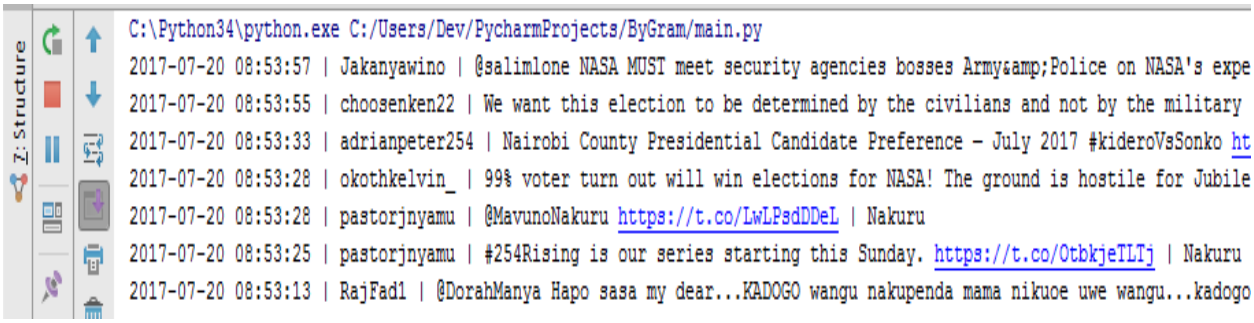

Figure 5: Showing tweets being streamed from tweepy API.

4.3 Discussion.

Table 2 shows prediction results using various sentiment classifiers. With unigram, combined data set had the highest accuracy value of 67.47%. This high performance is because combined data set has many sentiments and these results in high presence of terms features or words in data sets. This accuracy value then decreased to 53.69%, before increasing steadily (56.25%). This is due to higher weighting values resulting from term frequency and inverse document frequency calculations. Combined data set has more sentiments and therefore results into

higher weights being used in the training phases. Naive Bayes uses presence or absence of features in creating classification model. When there is scarcity in training data, absent terms causes zero probability problem. With Boolean indicators, it takes words that do not appear in the sentiments into account.

Table 2 shows predictions results with Bi gram ,combined data set had the accuracy value of 70%.This high performance is because bi gram does not require the assumption that some features are irrelevant and even the lowest ranked features according feature selection methods contain considerable information. In addition, Bigram features incorporate some contextual information which is important for sentiment classification and also generally contain a large number of noisy features and sparse matrix of terms. The size of sentiments used for training has an effect on the classifier performance. To estimate the classifier's performance the following measures can be used: accuracy, precision, recall, and F-measure or F-score (Figure 1.ilustrating the metric measures as captured in the experimentation process).

Accuracy of classifiers is dependent on quantity and quality of training data sets. (Refer to table 1) where the data/tweet is streamed with all the relevant components. This is significant to the final results of the classifiers. Machine learning algorithms rely on selected features from training data to infer the similarities or commonalities that a group of sentiments share and that discriminate them from the rest of the sentiments. The successes of classifiers therefore rely on the relevance of the features for discriminating between class labels. The longer the sentiments, the more the features used in constructing classifier and the better the model. By comparing the performance of different features, we find out that the selections of features are most significant for the sentiment classification tweets on the different dataset sizes, and least significant for the classification of sentiment which associates with the number of classes for sentiment classification. We also observe that the simplest feature, namely bigrams features, in most cases produces the best performance. The size of dataset affects the sentiment classification accuracy. The more sample size the more often single words and phrases are repeated. Hence that the number of unigrams and bigrams significantly reduced on a large dataset.
The accuracy of classification will increase as we increase the training data. The performance of the system depends on training datasets and also content (i.e. Tweets) in these data sets

V. Conclusion and Future Work

The aim of the study was to evaluate the performance for sentiment classification in terms of accuracy, precision and recall. In this paper, we compared various supervised machine learning algorithms of Naïve Bayes' for sentiment analysis and detection of the hate tweets in twitter. Apart from the system's ability to predict for a given tweet whether it is hateful or not, the system also generates a list of users who frequently post such content. This provides us with an interesting insight into the usage pattern of hate-mongers in terms of how they express bigotry, racism and propaganda. The experimental results show that the classifiers yielded better results for the hate tweets review with the Naïve Bayes' approach giving above 80% accuracies and outperforming other algorithms.

This research had a number of key weaknesses that can be addressed. One major consideration would be to include emotions and video images in detecting hate tweets among various users in twitter targeting various groups or individuals.
Another problem that could be addressed is the limitation of twitter API for commercial research where authorization is limited. Currently Twitter allows users to collect approximately 1600 tweets per day and will only provide data that has been uploaded in the last six days. To gain real value from a sentiment analysis it would be required to have massive amounts of data on the product or service which is currently not available without premium accounts or using third parties.

Given the legal and moral implications of hate speech it is important that we are able to accurately distinguish between the two. Thus we can say Naïve Bayes' classifier can be used successfully to analyze movie reviews.

Acknowledgment

First and foremost, praises and thanks to the God, the Almighty, for His providential care throughout my research work to complete the research successfully.

We take this opportunity to thank the School of Computing and Informatics, Jomo Kenyatta University of Agriculture and Technology (JKUAT) for giving me a chance to pursue and successfully complete this course.

We would like to express our deep and sincere gratitude to the co-authors, and the other members of staff in their various capacities for the untiring support, guidance and concern throughout my Thesis or a better way as you seem fit. I am extremely grateful to my parents and family for their love, prayers, caring and sacrifices for educating and preparing me for my future.

I also thank my course mates for their encouragements and team work we have shared during this research. Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

References

[1] W. Warner and J. Hirschberg. (2012) .Detecting hate speech on the world wide web.*Proceeding LSM '12 Proc. Second Work. Lang. Soc.Media*, no. Lsm, pp. 19–26.

[2] I. Kwok and Y. Wang.(2013).Locate the hate: detecting tweets against blacks. *Twenty-Seventh AAAI Conf. Artif. Intell., pp. 1621–1622.*

[3] I. Alfina, D. Sigmawaty, F. Nurhidayati, and A. N. Hidayanto.(2017).Utilizing hashtags for sentiment analysis of tweets in the political domain. In *Proceedings of the 9th International Conference on Machine Learning and Computing*, pp. 43–47.

[4] Freund, Y; Schapire, R.E.(1999) .Large margin classification using the perceptron algorithm. *Machine Learning, 37(3):277–296.*

[5] Kim, Y.H. et al. (2000) .Text filtering by boosting naive Bayes classifiers. *ACM SIGIR Conference:p168-175.*

[6] Parikh R, Movassate M. (2009) .Sentiment analysis of user-generated Twitter updates using various classification techniques. CS224N Final Report;pages. 1–18.

[7] Pak A, Paroubek P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *In: LREC. vol. 10; pages. 1320–1326.*

[8] Gaudette L, Japkowicz N. (2009). Evaluation methods for ordinal classification. *In Advances in Artificial Intelligence. Springer; p. 207–210.*

[9] Go A, Bhayani R, Huang L. (2009) .Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford;p. 1–12.*

[10] Po-Wei Liang, Bi-Ru Dai.(2013).Opinion mining on social mediadata.*IEEE 14th International Conference on Mobile Data Management,Milan, Italy*, pp 91-96.retrieved from http://doi.ieeecomputersociety.org/10.1109/MDM.

[11] Swati Agarwal and Ashish Sureka. (2015). Using KNN and SVM Based one-class classifier for detecting online radicalization on twitter. *Proceedings of the 11th International Conference on Distributed Computing and Internet Technology (ICDCIT'15).*

[12] Leandro Araújo Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. (2016). Analyzing the targets of hate in online social media. *In International Conference on Web and Social Media (ICWSM).*

[13] Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. (2015).A Lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering 10(4), 215–230.*

[14] Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. (2015). Antisocial behavior in online discussion communities. *In International Conference on Web and Social Media (ICWSM).*

[15] Denzil Correa, Leandro Silva, Mainack Mondal, Fabricio Benevenuto, and Krishna P. Gummadi. (2015). The many shades of anonymity: characterizing anonymous social media content. *Proceedings of the9th International AAAI Conference on Weblogs and Social Media (ICWSM'15).*

AUTHORS

First Author: George Okeyo, Dr. Jomo Kenyatta University of Agriculture and Technology, gokeyo@jkuat.ac.ke
Second Author: Ken Ogada, Dr. Jomo Kenyatta University of Agriculture and Technology, Kodhiambo@scit.jkuat.ac.ke
Third Author: Richard Rimiru, Dr. Jomo Kenyatta University of Agriculture and Technology,rimirurm@gmail.com.

Correspondence Author – Kelvin kiema, kelvinkiema@gmail.com, kelvin.kiema@students.jkuat.ac.ke,