SENTIMENT CLASSIFICATION FOR HATE TWEET DETECTION IN KENYA ON TWITTER DATA USING NAÏVE BAYES ALGORITHM

KELVIN KIEMA KIILU

MASTER OF SCIENCE

(Computer Systems)

JOMO KENYATTA UNIVERSITY OF

AGRICULTURE AND TECHNOLOGY

2020

Sentiment Classification for Hate Tweet Detection in Kenya on Twitter Data Using Naïve Bayes Algorithm

Kelvin Kiema Kiilu

A thesis submitted in partial fulfillment of the requirement for the degree of Master of Science in Computer Systems in the Jomo Kenyatta University of Agriculture and Technology

2020

DECLARATION

This thesis is my original work and has not been presented for a degree in any given University.

Signature…………………………………………..Date………………….

Kelvin Kiema Kiilu

This thesis has been submitted for examination with our approval as the university.

Signature……………………………………………..Date………………….

Dr.George Okeyo, PhD

JKUAT, Kenya

Signature……………………………………………..Date………………….

Dr. Richard Rimiru, PhD

JKUAT, Kenya

DEDICATION

First and foremost, praise and God, the Almighty, for His providential care throughout my research work to complete the research successfully. I take this chance to thank the Jomo Kenyatta University of Agriculture and Technology (JKUAT) for allowing me to pursue and complete this course.

To my supervisors, Dr.George Okeyo and Dr.Richard Rimiru, I'm deeply indebted to you for all the assistance, suggestions, and encouragement I received during this report's research and scripting. I am incredibly grateful to my parents and family for their love, prayers, caring, and sacrifices to educate and prepare me for my future.

I also thank my course mates for their encouragement and teamwork during the report's research and writing, finally, to all or any of my friends for their insights, encouragement, and prayers.

TABLE OF CONTENTS

LIST OF TABLES.

LIST OF FIGURES.

LIST OF APPENDICES.

LIST OF ABBREVIATIONS / ACRONYMS.

KNCHR: Kenya National Commission on Human Rights

NCIC: National Cohesion and Integration Commission.

IE: Information Extraction.

NLP: Natural Language Processing.

LDA: Latent Dirichlet Allocation.

SMS: Short Message Service.

WWW: World Wide Web

JSON: JavaScript Object Notation Syntax:

GUI: Graphical User Interface

SDK: Software Development Kits.

API: Application Programming Interface.

TDH-Twitter Data Handler module

HTTP: HyperText Transfer Protocol.

NB: Naïve Bayes.

NLTK: The Natural Language Toolkit.

URL: Uniform Resource Locator.

BSD: Berkeley Software Distribution.

SNA: Social Network Analysis.

REST: Representational state transfer.

ME: Maximum Entropy

BN: Bayesian Network.

JPD: Joint Probability Distribution

NN: Neural Network.

CSV: Comma-Separated Values.

ML: Machine Learning.

IR: Information Retrieval

WSD: Word Sense Disambiguation.

MT: Machine Translation.

MI: Market Intelligence.

DF: Document Frequency.

TS: Term Strength.

IG: Information Gain.

MBR: Minimum Bayes Risk.

IDE: Integrated Development Environment.

MNB: Multinomial Naïve Bayes.

TF-IDF: Term Frequency- Inverse Document Frequency.

KOT: Kenyans on Twitter

ME: Maximum Entropy

SVM: Support Vector Machine

TP: True Positive

FP: False Positive

FN: False Negative

TN: True Negative

PMI: Pointwise Mutual Information.

MaxEnt: Maximum Entropy.

TF-IDF: Term Frequency Inverse Document Frequency

NLP:-Natural Language Processing

HT: Hate Target

HI: Hate Instigator

BOW: Bag of Words

ME: Maximum Entropy.

WCM: Web Content Mining

ML: Machine Learning

DTM: Document Term Matrix

AI:Artificial Intelligence

ABSTRACT

Twitter has flourished to several hundred Million users and could present a rich information source for detecting and classifying hate speech instigator and hate targets using the platform. Microblogging sites are well-known to be suitable for conveying hate speech. As such, hateful wording involves communications that unlawfully demean any group or person based on certain characteristics, including color, race, gender, ethnicity, sexual orientation, religion, or nationality. Such content can frighten, intimidate, or silence platform users, and a few of it will incite different users to commit a crime. The continuing rise of social internet platforms, especially Twitter, has forced the need for more immediate analysis of hatreds and other related antagonistic responses to various trigger events. Twitter users usually air their views about various topics of their interest. The problem is that each tweet is limited in characters and is hence very short. It may contain slang and misspelled words. Thus, it isn't easy to apply traditional NLP techniques designed for working with formal languages into the Twitter domain.Another problem is that the total volume of tweets is extremely high, and it takes a long time to process, thus motivating for analysis within the field. We performed a comparative analysis using various sentiment analysis and machine learning tools using various feature values and model hyperparameters. This thesis developed an approach for collection, preprocessing, and classifying hateful speech that uses content created by self-identifying hateful communities from Twitter. Therefore, this study aims to detect and classify hate speech based on Kenya's context over the Twitter platform, using Natural Language Processing (NLP) techniques, various machine learning methods, and a novel approach for sentiment analysis on Twitter data. These tweets were extracted from Twitter through Twitter API and stored in JSON format in Mongo DB. The Naive Bayes machine-learning algorithm was developed to classify hate tweets into positive and negative sentiments. Experimental evaluations show that the proposed machine learning classifiers are efficient and perform better in accuracy and time. For actual implementation, Python with NLTK and python-twitter APIs have been used. To validate the results of applied offensive tweets, identification, and classification techniques, various performance metrics were used in the study. The experiment results show that Naïve Bayes offered the best performance among other classifiers on Twitter data set classification with an accuracy performance value of 83.1%.

CHAPTER ONE

INTRODUCTION

This chapter gives a background study of social media, specifically Twitter. It also looks at the problem statement, objectives of the study, justification for the study, and finally, the thesis's organization.

1.1 Background of the Study

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, 2013). Twitter has 270 Million active users, and 500 million tweets are sent per day (Wellons, 2015). Due to the high reachability and popularity of social media websites worldwide, organizations also use them to plan and mobilize events for protests and public demonstrations (Muthiah, 2015). The development and use of SNSs have revolutionized the way people share information and keep in touch with friends. People can express their opinions in the form of posts (Facebook), tweets (Twitter), emoticons, etc., concerning many issues that affect their day-to-day lives. According to Ipsos Synovate Report (2009), 79% of internet users in Kenya are Facebook members using it as a primary means to talk to friends, relatives, work-mates, and "follow" their favorite companies' well prominent personalities and politicians. The Twitter community, known as #KOT (KenyansOnTwitter), on the other hand, is particularly active, using this social media platform for online activism, praising corporate brands or calling them out, and even rallying the public to help in cases of famine, protest campaigns, and many others. The Kenyan leadership has understood the power of social networks and has made it an integral part of their communication—the president of Kenya, Hon. For example, Uhuru Kenyatta has a followership of close to a million on Facebook and over 400,000 on Twitter and was recently ranked most followed president in Africa (Standard Digital, 2014).

Wasswa (2013), in his research on "The Role of Social Media in the 2013 Presidential Election Campaigns in Kenya," found that social media played a big role in the 2013 presidential elections with presidential candidates integrating social media into their campaigns. The platform was majorly used for sharing information on campaign activities, debate on issues, share photos, videos, and links, solicit for funds, counter-propaganda, and update followers on the going on. The corporate world is not left behind, with Safaricom, OLX Kenya, Airtel, Samsung Mobile Kenya, Equity Bank, Kenya Airways, and Kenya Power among the leading embracers of social media with the largest followership (Socialbaker, 2014).

Twitter is a famous platform for opinion and information sharing, and this platform is mostly used before, during, and after live events (Bollen, 2011). Online spaces are often exploited and misused to spread content that can be degrading, abusive, or otherwise harmful to people. Twitter prohibits users from posting violent threats, harassment, and hateful content. However, tons of users disobey the rules and use their Twitter account to spread hate speech and negative words. Over the past decade, social media has emerged into a dynamic form of worldwide interpersonal communication.

It is being used as a tool to practice several kinds of mischievous acts with concealed agendas and promote ideologies in a sophisticated manner (Qin, 2011). For example, infiltration of extremist groups, hate groups, racial supremacy groups, and terrorist organizations on social networking (Facebook), micro-blogging (Twitter, Tumblr), image sharing (Imgur, Flickr) and video hosting and sharing (YouTube, Dailymotion, Vimeo) websites are posing grievous threats to our societies as well as the national security. Due to the high reachability and popularity of social media websites worldwide, extremist groups or organizations use these websites to plan and mobilize events for psychological warfare, fund-raising, recruitment, and propagation of their agendas, protests, and public demonstrations (Muthiah S, 2015). The study of civil unrest reveals that now most of the protests are planned and mobilized in much advance (Rowe, 2015). Many hate promoting groups use popular social media websites to promote their ideology among their viewers (Namee, 2010) (Anwar, 2012).

In this era of the internet, there is much to be imagined. The internet plays a central role in the evolution of gangs and radicalization because of the ability to broadcast key symbols, images, and messages worldwide in a matter of minutes (Decker S, 2011). An important and elusive form of such language is a hateful speech: content that expresses a group's hatred in society. Hateful speech has become a major problem for every 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). Through such access to Twitter, 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 Speech targets disadvantaged social groups in a potentially harmful manner (Jacobs, 1994). In Kenya, hate speech has been defined as any form of speech that degrades others, promotes hatred, and encourages violence against a group based on criteria, including religion, race, color, or ethnicity. It includes speech, publication, or broadcast representing as inherently inferior or degrading, dehumanizing, and demeans a group. (KHRC, 2010). The thesis's goal was to detect, identify, classify, and analyze the spread of hate tweets on the social site and specifically Twitter in Kenya. Sentiment analysis is an area of natural language processing that aims to determine 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 detect sentiment in a text (Pang, 2008) automatically. While some identify the objects discussed and the polarity (positive, negative, or neutral) of sentiment expressed about them (Gamon, 2005), other algorithms assign an overall polarity to a text, such as a movie review. The study used a stand-alone classification classifiers approach based on Naïve Bayes to detect hate speech/tweets. The approach involves tweet acquisition and streaming using Tweepy API, preprocessing to remove unwanted parts of speech using n-grams, and tweet classification and evaluation using Naïve Bayes. The tweets' collection was selected to contain various words, expressions, emotional signals, and indicative examples of sarcastic, ironic, metaphoric language. A hate dataset classification framework was developed that processed tweets in real-time and supervised learning techniques to analyze and classify their sentiments.

1.2 Problem Statement

Social media gives users an easy-to-use way to communicate and network, thus generating huge data useful in various fields. Online social networking sites are becoming more popular each day. Among all these sites, Twitter is the fastest growing site than any other social networking site. Kenya is a multicultural country with over forty-two ethnic tribes, each with its unique way of communicating. Almost all ethnic communities in Kenya have some stereotypes about them; these stereotypes may be positive or negative (National Cohesion and Integration Commission, 2013). Over the last decades, people are getting more engaged with widespread social networks. Microblogging applications opened up the chance for people worldwide to express and share their thoughts extensively and in a real-time manner. Such expressions afford researchers the ability to investigate the online social emotions in different events. People now can speak freely; this allowed them to exchange all sorts of thoughts, emotions, and knowledge. However, cyberspace is not always safe; it can be a reason for disseminating aggressive and harmful content. Hate speech is a common online form for expressing prejudice and aggression.

There is an uncontrollable number of comments and posts issued every second on social media platforms, making it impossible to trace or control such a platform’s content. Therefore, social platforms face a problem in limiting these posts while weighing the right to freedom of speech. (Z. Waseem and D. Hovy, 2016). While most of the online social networks and microblogging websites forbid the use of hate speech, these networks and websites’ size makes it almost impossible to control all of their content. Therefore, the necessity to automatically detect such speech and filter any content that presents hateful language or language inciting hatred arises.

Most negative statements depict contempt and general hate towards targeted communities resulting in heightened friction and animosity among various ethnic communities. The negative statements are often expressed in coded language well known to the community members who use it and may or may not be known to the targeted ethnic communities (National Cohesion and Integration Commission, 2013). To be quantified as hate speech, the statement should contain: threatening, abusive, or insulting messages, sometimes using coded language. These messages must be directed towards a targeted group and intended to stir hatred based on the group's identity, including ethnicity, race, color, or national origin (National Cohesion and Integration Commission, 2011).

However, despite increasing evidence that cyberhate is on the rise, the availability of legislation to bring about prosecution and the desire from leading social media companies to reduce harm go largely unpunished, given the multiple difficulties in policing online public spaces. Of these difficulties, classifying cyberhate promptly and at scale is the most challenging given increasing restrictions on policing resources (Giannasi, 2014) and the difficulty with identifying appropriate opportunities to engage in counterspeech. Therefore, automated techniques are needed that programmatically classify cyberhate to lighten the burden on those responsible for protecting the public. (Giannasi, 2014).

This thesis focused on detecting and classifying hate data set on the Twitter platform. Many sentiment analysis tools and applications have been developed to mine the opinions in user-generated content on the Web. However, the performances are abysmal due to natural language (Maynard, 2016). In essence, sentiment analysis is still a problem of natural language processing (NLP), which deals with the natural language documents, which are also called unstructured data (Liu, 2012). Prior researches show that sentiment analysis is more difficult than the traditional topic-based text classification (Pang, 2008). Although various approaches have been proposed to conduct sentiment analysis, it is still difficult to deal with some linguistic phenomena, such as negation and mix-opinion text. This leads to low accuracy of sentiment classification (Khan 2016). Besides, it is insufficient to only determine the opinions' polarity since an opinion without a target is of limited use. Extracting the opinions and their targets simultaneously is also called aspect-level sentiment analysis in the research literature and are more difficult to achieve (Liu, 2012). Current studies show that the methods of dealing with aspect-level sentiment analysis are limited. The first research question concerns the need to manage many online reviews and improve sentiment classification performance automatically. The research question underlines the significance of identifying the opinions' targets, which pursues to help individuals make an informed purchasing decision and provide manufacturers insight to improve their products or services.

1.3 Objectives of the Study

This thesis's main objective was to perform sentiment analysis in detecting, analyzing, and classifying hate tweets in social media, particularly Twitter, by providing a novel framework for sentiment analysis.

1.3.1 Specific Objectives

1. To identify the algorithms and metrics for evaluating the performance of Machine Learning Classifiers while identifying various technologies, Programming Languages, and Libraries that could be applied to implement Machine Learning.
2. Develop a classification framework that can collect, extract, and preprocess hate data of users from their posts on Twitter.
3. To identify various stereotypical /hate words used to propagate and disseminate hate speech on Twitter.
4. To develop a prototype of the hate tweets classification model.

1.4 Research Questions

The main research questions in the current study were:

1. What are the current machine learning techniques applied in hate speech detection in social media?
2. What are the unique constraints that would hinder the effective classification of

hate tweets, and what are some of the words used to disseminate hate speech on Twitter?

3. What are the best machine learning methods in classifying hate tweets, and how can users who promote and disseminate hate messages can be identified?
4. What will be the performance effect of a combined solution by combining lexicon-based approaches and machine learning approaches?

1.5 Justification

Social networks are generators of large amounts of data produced by users, who are not limited to the content of the information they exchange. Twitter is a micro-blogging platform that allows users to make tweets, messages no longer those 140 characters, resembling SMS (Short Message Service). Tweets are synthetic messages containing different kinds of information: links, media attachments, mentions (@), and hashtags (#). First, the World Wide Web (WWW) has made information available more than ever before. Hence, people increasingly take their required information from one another rather than from corporations, media outlets, religion, or political bodies.

The number of registered users is increasing day by day, and the amount of online user-generated content quickly grows. It is difficult to do manual flagging to remove hateful content in online media. Therefore, it is necessary to use accurate, automated methods to flag abusive / hate speech in online media. When looking at the policies and regulations established by different social media websites, we also feel a big need to automatically identify hate speech. People may give their opinions on the shared posts; those opinions may be positive, negative, or controversial. Besides, Twitter has been free to create one profile and does not have stringent restrictions than other social sites such as Facebook, thus attracting a large following. Using the data generated can be a good indicator of trends and topic preferences among users. Twitter has been a prolific environment for sentiment analysis, allowing researchers to dive into real research for text classification.

Most of the tweets posted on Twitter contain hate messages that target people through color, ethnicity, gender, religion, or race. Some have also used the Twitter platform to propagate and disseminate hate, offending, and harming information to other users and the public. Furthermore, the user has no limitations regarding the text's content they can write in a tweet. This freedom and lack of formalism generate issues when analyzing and classifying the text, and classic NLP tools seem almost powerless. Thus, applying text classification techniques in detecting and classifying hate speech in those tweets will go a long way in reducing the vice and contributing immensely to the field sentiment analysis.

1.6 Scope of Study

There are various languages in Kenya, approximately 42 tribes/communities, each with various dialects. The hate detection model will not cover all but only a few. Hate speech is not only propagated through video only but also through text, which this study covered. This introductory chapter describes my research interest in the text mining in the social site's data and particularly Twitter to potentially reduce or curb the promotion or dissemination of hate messages through postings, comments, or likes among various users, the general public. This research's scope involved clustering and classification of the datasets, analyzing the dataset, detecting various users' language, and finally simulating the dataset to identify the language used in the posting or publications to spread hate speech. This research covered how various language dialects promote and disseminate hate speech in social sites in a textual manner and how it can be reduced in the said sites. The data is presented in a graphical format. The research was conducted in Natural Language Processing, considering the task as a text analytic problem. Then the research focused on a lexicon-based approach for the classification of words with the combination of a machine learning technique.

1.7 Thesis Organization

This thesis is divided into five chapters. Chapter 1 is the background of the study. This chapter starts by describing the problem statement, research objectives, justification, and thesis organization. The next four chapters are organized as follows.

Chapter 2 starts with a literature review, followed by a text classification concept and data mining sentiment process. A description of the feature of sentiment classification and feature selection methods is discussed. N-gram document modeling, its application follows. An analysis of various Machine learning algorithms and their applications in text categorization is then discussed. Methods of evaluating classifiers follow this.

Chapter 3 presents the methodology used. The experiment was used in this research. This chapter describes the data collection process, research population, sentiment classification steps, and data preprocessing.

Chapter 4 is dedicated to Experiments and Results. This chapter deals with a discussion of the results and findings described in chapter 3. The results of prediction and classification are analyzed and presented in the form of tables.

Chapter 5 starts with a description of Knowledge contribution, conclusion, and future works.

CHAPTER TWO

LITERATURE REVIEW

This chapter gives an introductory literature review of hate speech in social media, text classification, and machine learning algorithm applications in social media sites, especially Twitter. It also looks at twitter concepts and terminologies and the evaluation metrics for the study.

2.1 Introduction

The rise in use and popularity of the informal language and the adoption of social media platforms, especially Twitter, have made Sentiment analysis of tweets an important research area (S. Vosoughi, 2015.) .Twitter is a famous platform for opinion and information sharing. This platform is mostly used before, during, and after live events. However, most users have used the platform to disseminate hate messages among the users and the general public. According to Alexa, and based upon its panel of toolbar users, Twitter had become the world's ninth hottest internet site by October 2010, alexa.com/top sites (2010) despite only beginning in July 2006. The rapid climb of the location could also be partly thanks to celebrities tweeting regular updates about their daily lives (Johnson, 2009).

We are concerned with detecting, identifying, analyzing, and curbing the spread of hate speech sentiments on the social site and specifically Twitter in Kenya. Most of these hate speech text or messages uses stereotypical language. Several researchers have investigated the detection of flames and virulent messages in social media and the spread of hateful messages in dark web forums. Hate speech uses offensive and threatening language that targets certain groups of people based on their religion, ethnicity, nationality, color, or gender. The hate message source is typically a member of a supposedly rival group or fellow Twitter or another ethnic community in Kenya. The analysis of subjective language has been widely applied to the classification of opinions and emotions in the text (Wiebe, 2005).

Indeed, sentiment analysis, which aims to annotate text using a scale that is a measure of the degree of negative and positive sentiment within the text, has been applied to data collected from social media to determine emotional differences between genders on My Space (Thelwall 2011) and study levels of positive and negative sentiment in Facebook (Ahktar, 2009) and Twitter comments (Bollen, 2011) (Thelwall, 2011) following real-world events. (Burnap, 2013) developed a rule-based approach to classifying antagonistic content on Twitter, and they used associational terms as features. They also included accusational and attributional terms targeted at a person or persons following a socially disruptive event as features to capture the context of the term use.

Chen (2012) also performed simple anaphora resolution by resolving pronouns to the tweet's closest entity. The rule-based algorithm differentiates between imperative declarative and interrogative sentences and can, among other things, handle comparative sentences, negation, and but-clauses. To enhance the recall of the proposed methods, the authors identify additional tweets that are likely to be opinionated and train a support vector machine to assign polarity labels to the contained entities. (Chen, 2012) identified offensive content by using profanities, obscenities, and pejorative terms as features weighted accordingly based on the associated strength of the term, as well as references to people. They also produced rules to model offensive content, showing improved standard machine learning approaches in terms of a much-reduced false-negative rate.

Moreno (2014) proposed an open framework to automatically collect and analyze data from Twitter's public streams. This is often a customizable and extensible framework, so researchers can use it to check new techniques. The framework is complemented with a language-agnostic sentiment analysis module, which provides a group of tools to perform a sentiment analysis of the collected tweets. This platform's capabilities are illustrated with two study cases in Spanish, one associated with a high impact event (the Boston Terror Attack), and another one associated with regular political activity on Twitter. The primary case study involved Twitter's activity around a high impact event, the Boston

Terror Attacks. During this case, they tracked a hashtag. The second case study was focused on regular Twitter usage, tracking the activity around well-known Spanish political actors.

The review of the related work done in this field shows that the models trained after extracting N-gram features from text give better results (Chikashi Nobata, 2016). The TF-IDF approach on the bag of- words also features show-promising results (Williams P. B., 2016). Based on the review of features and the prominent classifiers used for text classification in the past work, we decided to extract n-grams from the text and weight them according to their TFIDF values.

Djuric (2015) detect hate speech in comments collected from Yahoo! Finance, using 1) Paragraph2vec with Bag of Words (BOW) Neural Language Model to get masked insults and to swear; and 2) embeddings-based binary classifier to separate hateful and non-hateful comments. Paragraph2vec was ready to discover some non-obvious swearing words and also obtaining better results than BOW models. In their context, most insults were targeting rich people (Djuric N, 2015). Previous studies have also found that using word embeddings (i.e., distributional semantics) performs well. For instance, (Nobata C 2016) detect hate speech, profanity, and derogatory language. They used N-grams, Linguistic, Syntactic, and Distributional Semantics, finding that combining all feature types gave the simplest performance for Finance and News contexts (Nobata C, 2016). (Mondal M, 2017) used an easy syntax "I," allowing them to spot explicit hate targets. The user intends that intensity is the emotional level and a hate target because it receives dislike or hatred. To avoid false positives, such as: "I really hate owing people favors," they 1) placed a selected word before 'people' to specify hate targets (e.g., black people, Mexican people, stupid people), and, since not all hate contains the word 'people,' they 2) used 1,078 hate words from the Hatebase2. Using this strategy, they identified 20,305 Tweets and 7,604 Whispers as hateful, commonest categories on social networks being race, behavior, and physical. The study by (Mondal M 2017) illustrates the restrictions of using keywords only. The difficulty is that the diversity of hate, which isn't fully captured by the lexicon. Additionally, the tactic is susceptible to error; for instance, it might find "I hate police officers” but miss “police officers are dogs.” (Saleem, 2017).

Further, point out that a keyword used in one as a hate indicator may not represent hate in another community. For example, (Sood S 2012a) used a profanity list with a stemmer, detecting 40.2% of profanity terms at 52.8% precision, concluding that even the best lists would not achieve reliable performance in profanity detection. Moreover, (Sood S 2012a), Point out that adaptability to new terminology and slang is a major challenge since the existing lists are missing the unfamiliar terms. Overall, our literature review shows that 1) earlier taxonomies of hate targets tend to be coarse and that 2) dictionary-based approaches alone are not sufficient in detecting and classifying hateful online comments. Granular classification is important, e.g., to community managers and public policymakers who wish to understand online hate. To address these issues, we a) develop a granular taxonomy of online hate, and then b) use it to classify hateful online comments by their target and type.

The authors have selected controversial accounts to have a good foundation for sentiment analysis. There are several processing layers, and these modules need to interchange data among them, using open data formats such as JSON. Most tools in the framework are implemented in Python. Still, the Classifier and Tester web interfaces run on NodeJS and are programmed in CoffeeScript (a language that can be preprocessed into JavaScript). The chosen back-end database is MongoDB, a good fit for our purposes since its atomic representation is JSON, just like tweets.

The implementation was based on the Natural Language Toolkit (NLTK) framework. A complete procedure of data extraction and sentiment analysis is divided into three separate steps: data acquisition, training for sentiment analysis, and report generation. The first step is gathering data from Twitter with the Miner. Their paper treats the problem as one of binary classification, classifying tweets as either positive or negative. Due to the lack of hand-labeled training data (Go, 2009), employ distant supervision to train a supervised machine learning classier: they download a large number of tweets via the Twitter API. They use emoticons in the tweets as noisy labels. Tweets containing emoticons expressing both positive and negative sentiment are not considered. They also remove messages containing retweets and message duplicates. Their final training data set consists of 1,600,000 tweets: 800,000 for each class.

In the data-preprocessing step, emoticons are removed, as they are used as labels, and generic tokens are inserted for user mentions and links. Furthermore, adjacent repeated letters are collapsed into two letters to reduce the spelling variety introduced through emphatic lengthening to some extent as features (Go, 2009) employed unigrams, bigrams, a combination of both and part-of-speech tags. In their experiments, they compare the Naive Bayes (NB), Maximum Entropy (MaxEnt), and Support Vector Machine (SVM) classification methods.

Their best result is 82.9%accuracy using SVM with only unigrams as features, and adding bigrams results in an increase of the NB and MaxEnt performance but a decrease in SVM. They report that in their experiments, adding negation as an explicit feature and using part-of-speech tags did not improve classification performance, while using bigrams exclusively yields worse results due to the too sparse feature space. (Gamallo, 2014) presented a family of Naive Bayes classifiers for detecting the polarity of English tweets. Two different Naive classifiers have been built, namely Baseline (trained to classify the tweets as positive, negative, and neutral) and Binary (uses a polarity lexicon and classified as positive and negative tweets are not considered). The features considered by the classifiers are Lemmas (nouns, verbs, adjectives, and adverbs), Multiword, and Polarity Lexicons from different sources and Valence Shifters. The training data set of tweets is obtained from Sem Eval Organization-2014 and additional annotated tweets from external sources. Many combinations of the strategies mentioned earlier and features are implemented. It is also concluded that performance is best when a binary strategy is used with multiword and valence shifters features. Twitter sentiment analysis is considered as a much harder problem than sentiment analysis on conventional text such as review documents, mainly due to the short length of tweet messages, the frequent use of informal and irregular words, and the rapid evolution of language in Twitter. Annotated tweets' data are impractical to obtain.

Therefore, feature selection is an important phase for training the algorithms. Based on the features, algorithms need to be trained. Feature selection seeks to select an optimal subset of features by eliminating irrelevant features or offering no additional information than features within the optimal subset. (Forman, 2003), Many available feature selection techniques can reduce irrelevant features while improving classifier performance for a wide range of text classification problems. (Pang and L. Lee, 2004), successfully utilize the sentiment information such as “thumbs up” or “thumbs down” to accurately classify documents (Guyon, 2003) demonstrated that performance increases from feature selection are in part due to the reduction of overfitting. (E. Kouloumpis, 2011), developed using word polarity based on prior probabilities as additional features.

Saif (2012) examined sentiment-topic features and semantic features to be used in conjunction with unigrams to achieve higher accuracy than unigrams. Sentiment classification has been used to address real-world problems such as election prediction (Wang, 2011) and product sales. Emotions are also used in sentiment classification. The tweets with emotions are treated as negative sentiment, and the tweets with emotions are treated as positive sentiments. The algorithms of these are implemented by (A. Go, 2009). (A. Go, 2009), examined Twitter API to classify tweets and to integrate sentiment analysis classifier functionality into web applications. (E. Kouloumpis, 2011), examine Twitter sentiment classification. With N-gram features, they include a sentiment lexicon, part of speech features, and features that capture information about the informal and creative language used in microblogging, such as emoticons, abbreviations, and the presence of intensifiers.

Their findings show that Part of Speech features decreases the performance. Moreover, they claim that features from an existing sentiment lexicon were somewhat useful in conjunction with microblogging features. In this work, we use Sanders’s data set that is hand classified. We use a prominent feature selection technique with N-grams. We examine the classification algorithms performance by providing the different combinations of feature selection and sentiment lexicons. This work uses a supervised machine-learning model. This work aims to improve the level of performance. The level of accuracy improves with low execution time. We implement the combinations of features, which use the sentiment lexicon dictionary, and extracted n-gram tweets of high information gain. By using these features, we evaluate the performance of machine learning algorithms. Sentiment analysis is an important research area that identifies the people's sentiment underlying a text. Sentiment analysis is widely studied in data mining. Sentiment analyses of tweets are widely studied. After reviewing and studying the current research on sentiment analysis, the study aims to get more effective sentiment analysis results on tweets. At the same time, to improving the performance to classify the tweets with sentiment information. We use a feature combination scheme that uses the sentiment lexicons and extracted tweets n-gram of high-performance gain.

2.2 Definition of Hate Speech

Deciding if a portion of text contains hate speech is not simple, even for humans. Hate speech is a complex phenomenon, intrinsically associated with relationships between groups, and relies on language nuances. This is notorious in the low agreement found between annotators in building new collections (Zeerak, 2016). Therefore, it is crucial to clearly define hate speech to make the task of its automatic identification easier (Bjorn Ross, 2017). What has considered hate speech has no formal, legal definition. Still, there is a consensus that Speech is Speech that carries expressions of hatred toward specific groups based on characteristics like race, gender, religion, sexual orientation, or disability. In the following, we provide a list of four dimensions to define hate speech uniquely and contextually. We clarify the difference between hate speech and another related concept for each dimension, respectively:

1. Hate speech attacks specific targets and groups of people identified based on specific characteristics like religion, sexual orientation, gender, ethnic origin, etc. According to this dimension, hate speech differs from toxic language (defined as" toxic comments which are rude, disrespectful or unreasonable messages that are likely to make a person leave a discussion" (Kalchbrenner N, 2014) and profanity (defined as" offensive or obscene word or phrase" since those can also be

perpetrated without a specific target. Hate speech also differs from cyberbullying, where" the aggressive and intentional act is carried out repeatedly and over time, against a victim who cannot easily defend him or herself," while hate speech is more general and not necessarily focused on a specific person more about stereotypes.

2. Hate speech is a speech inciting violence or hate. Very close concepts are radicalization and extremism. To clarify the difference between those three concepts, we need to point out that online radicalization is similar to extremism. Still, radical discourses have usually related a subset of terrorism, anti-black communities, or nationalism (Agarwal S, 2015), while extremism can be on any ideology. However, in both radical and extremist discourses, you can find topics like religion and war (Agarwal S, 2015), recruitment of new members, social media and institutions demonization, and even persuasion (Prentice S, 2011) while hate speech does not usually touch those topics. It can be more grounded in stereotypes and hence more subtle. That means that the kind of violence incited by hate speech discourses can also be subtle, as in the case of stereotypes that are gradually reinforced to such an extent that they can justify discrimination, violence, and hate against groups of people.
3. Hate speech is a speech aiming to attack or diminish specific groups of people. This definition makes hate speech almost indistinguishable from discrimination. However, the latter can be used as the basis of unfair treatment in every environment and can also refer to discriminating behaviors, while hate speech is more about discrimination through verbal means.
4. Hate Speech is not Humour, and Humour is not Hate speech, even if this latter can carry subtle forms of discrimination, e.g., through jokes playing on stereotypes. In this work, we consider these kinds of jokes as hate speech because, in case of a long exposure of users to them, the consequences could certainly be harmful towards some groups of people who could decide to leave the conversation (Douglass S, 2016).

In conclusion, in this paragraph, we used the four dimensions listed above to clarify the hate speech concept while underling its difference with other very close concepts. This analysis is well summarized by (Fortuna P 2018), which proposed a complete and unambiguous definition of hate speech that we are going to use in the following of this document:

“Hate speech is a language that attacks or diminishes, that incites violence or hate against groups, based on specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity or other, and it can occur with different linguistic styles, even in subtle forms or when humor is used.” (Fortuna P, 2018)

2.2.1 Studies on Hate Speech Detection

The societal impact of the internet and social media has increased over the past years, and perhaps this is why there has also been growth and interest in research covering hate speech detection. While the amount of research increases, the field still faces several challenges, both in the actual task of detecting hate speech and the research area in general. (Chikashi Nobata, 2016) have summarized the following challenges for the task of detecting hate speech. First, what is interpreted as offensive or hate speech is subjective and can differ from person to person. This can be a problem for annotating data for training hate speech detection systems, as the annotators do not necessarily agree.

Despite differences, some recent approaches found promising results for detecting hate speech in textual content (Fortuna P, 2018). The proposed solutions employ machine-learning techniques to classify text as hate speech. One limitation of these approaches is that their decisions can be opaque and difficult for humans to interpret *why* the decision was made. This is a practical concern because systems that automatically censor a person’s speech likely need a manual appeal process. To address this problem, we propose a new hate speech classification approach that allows for a better understanding of the decisions and show that it can even outperform existing approaches on some datasets. Some of the existing approaches use external sources, such as a hate speech lexicon, in their systems.

This can be effective, but it requires maintaining these sources and keeping them up to date, that is a problem in itself. Due to the societal concern and how widespread hate speech is becoming on the internet, there is strong motivation to study automatic hate speech detection. By automating its detection, the spread of hateful content can be reduced. (Mondal M, 2017).

Correspondingly, when identifying hate speech, we need to exclude some conditions. In like manner, (Waseem 2016) has proposed 11 parameters to distinguish hate speech specifically in the Twitter platform, some of which are: usage of sexiest and racial terms, attacking and criticizing minority, promoting violence, distorting the truth with lies, and supporting suspicious hashtags. Given these characteristics, a reasonable list can be derived for a particular culture with certain adjustments to deal with the controversy. From that list, hate speech can be reliably identified and recognized. Consequently, the choice depends on the features that can be extracted from the corpus.

Our approach is complementary to the referred study, and at the same time, our survey has specificities that we present here. First, we provide more detailed definitions: we compare hate speech with other related concepts, subtypes of hate and enumerate rules that help hate Speech classification. Moreover, we utilize a systematic method and analyze documents focusing on algorithms and descriptive statistics about hate speech detection. Complementarily, we also give an overview of the evolution of the area in recent years. Regarding the procedure of feature extraction, we use two categories, the “generic text mining features” and the “specific hate speech detection features.” In our approach, this distinction is relevant because the second category of features focuses on this problem's specificities. We also enumerate existing data collections for this task in a more exhaustive way than the previous study.

2.2.2 Hate Speech on Twitter

There is no formal definition, but there is a consensus that it is speech that addresses disadvantaged social groups in a way that is potentially harmful to them (Jacobs, 2000), (Walker In Kenya, hate speech has been defined as any form of speech that degrades others and promotes hatred and encourages violence against a group based on criteria including religion, race, color or ethnicity. Includes speeches, posts, or broadcasts that portray as inherently inferior, or degrade, dehumanize, and degrade a group. (KHRC, 2010). Importantly, the definition does not include all offensive language instances because people often use terms that are very offensive to certain groups but in a qualitatively different way. Everything that is tweeted can reach a large number, and the effects can be outstanding. On August 14, 2013, a Twitter user tweeted: “The Luo Nation and CORD appendages should be liberated from the User 2 bondage. He is a rocking chair - keeps them busy, takes them nowhere”. This was retweeted 938 times with 208 favorites, and the user currently has 687,000 followers. Given these numbers, the expected circulation of similar tribal tweets is alarmingly extensive. According to Alexa, and according to its toolbar user panel. The site’s rapid growth may be due in part to celebrities tweeting regular updates about their daily lives (Johnson, 2009). We were concerned with detecting, identifying, and analyzing the spread of hateful feelings on the social site and specifically on Twitter in Kenya.

Most of these hate speech texts or messages use stereotypical language. Various researchers have investigated the detection of flames and virulent messages on social media and the spread of hateful messages on dark web forums. Hate speech uses offensive and threatening language that targets certain groups of people based on their religion, ethnicity, nationality, color, or gender. The hateful message source is usually a member of a supposedly rival group or a fellow Twitter member of another ethnic community in Kenya. It is in this sense that the use of sentiment analysis on Twitter is essential to detect and analyze hateful tweets. Evaluating our approach using multiple data sets is essential.

The collection of tweets was selected to contain a variety of words, expressions, emotional signals. As well as indicative examples of sarcastic, ironic, and metaphorical language. Also, apply four evaluation measures and demonstrate that two are more suitable for evaluating sentiment classifiers. Additionally, the same measurements used to assess the quality of the training data provide a means to monitor the annotation process.

Additionally, the development of a sentiment analysis classifier that processes tweets in real-time and uses supervised learning techniques to analyze and classify their Additionally, the classifier showed how hate tweets used by multiple Twitter users tend to target multiple groups using various Twitter features. Hate speech is an artistic term in legal and political theory that is used to refer to verbal behavior - and other symbolic and communicative actions - that intentionally express an intense antipathy towards some group or towards an individual due to their belonging to the speech hate speech, therefore, includes things like abuse and harassment that damages identity, certain uses of insults and epithets, some extremist political and religious speeches, and certain manifestations of individual, resulting in humiliation, anguish, and psychological or emotional pain. Twitter plays an important role in social media and communication: Hate speech in the age of Twitter is the way people communicate current issues and discuss them, adding to and altering our daily communication practices. (Cammaerts, 2009), in his study, he demonstrated how the internet works as space where conversations about racial hatred and discrimination are hosted by focusing on extreme right-wing discourses on blogs and mentioning that the Web provides fascists, fundamentalists, and other with the same opportunities as activists, allowing them to connect and interact through their social networks.

However, there is a consensus in the literature that the internet facilitates the spread of hate speech in the digital world and particularly on Twitter. Also, some reasons support the claim. First, given the options to retweet, use a specifically tagged hash, and link content to other platforms, there is no question that Twitter contributes to the spread of a message to a wider audience on and off the platform. This means that there are more opportunities for people to see a hate message, adopt it, and republish. Nemes (2002) further mentions the harm of hate speech on individuals, groups, and society as a whole. As far as it concerns the individuals, he mentions that hate speech can provoke pain, distress, fear, embarrassment, isolation, etc. Hate speech towards groups of people can bring inequality problems and lead the group members in isolation. It creates feelings of fear and discourages them from participating in their community or expressing their opinions.

Moreover, this degradation and humiliation can silence the ‘victims’ and reinforce existing hierarchies in society (Nielsen, 2002). It can also lead to hate speech victims becoming aggressive and dangerous (Parekh, 2006). Given the facts above that concern the harmful nature of hate speech, there is no doubt that discriminatory and offensive expression is an undesirable and negative phenomenon in a democratic society.

Therefore, the thesis designed a multilingual sentiment classifier. The classifier analyzed hate tweets using sentiment analysis in English and extend it to the multilingual setting by employing a standard news machine translation system. The system will accept input from the user. Using the keyword entered by the user, the system will search for the tweets containing that keyword. It will also provide data in graphical presentation. It will also be able to the data received may contain a huge amount of unlabeled or noisy labeled data. Therefore, it is necessary to clean the data before applying sentiment analysis to it. The output is the graphical representation of the data classified based on sentiment.

2.2.3 Features for Hate Speech Detection

When the amount of training data is large enough, different classification methods' performance becomes more similar. The distinguishing impact on performance will then come from the features chosen to employ in the methods. This section describes common types of features used in NLP, based on a survey of existing research on features used in hate speech detection (Wiegland, 2017)

1. **Simple Surface Features**

   The presence and frequency of the words in a document are simple and easily retrievable features. These features can be derived without advanced methods and include bag-of-words, n-grams, appearances, and frequencies (URL, words, characters, etc.). (Chikashi Nobata, 2016) found that n-gram features are predictive and perform well on their own in noisy data sets and that a combination with other features is shown to be powerful.

2. **Word Generalization**

   Data sparsity in text representation can be approached by applying a form of word generalization, such as word embeddings or word clustering. In word clustering, each cluster with a set of words can be used as a feature. Word embeddings can be considered both as word representations and features. As features, embeddings may replace particular words' presence or frequency by establishing the similarities between words in the representations.

3. **Lexical Resources**

   Lexical resources, or word lists, are necessary when specific words are used as features. Several word lists are available on the internet for general tasks, and some are made publicly available. For particular NLP tasks, it may be convenient to create a new list or dictionary. Lexical features are considered insufficient as stand-alone features and are recommended to be combined with other types.

4. **Syntactic Features**

   Syntactic or linguistic features consider the structure of the text and the relationship between words, enabling a better understanding of the underlying meaning. Part of- speech tagging is a method for marking words in a text according to their respective part of speech, such as nouns, verbs, or adjectives. The challenge is assigning part-of-speech tags to words that can have different meanings. Dependency relationships are employed to capture relationships and dependencies between words, which is useful for capturing relations between non-consecutive terms.

5. **Context-Based Features**

   The textual context can provide useful information in understanding the meaning and opinions in a text. However, the context may be difficult to both retrieve and represent. Knowledge-based features are useful in specific domains but require creating the knowledge base, which can be comprehensive to implement. Meta information provides information that is not directly present in the text but can be derived from the surroundings, such as information about the author or an article referenced. Online posts are often a combination of multiple modalities, such as text, video, and image.

2.3 Data Mining

Data mining is the computational process of finding patterns in large datasets, and its methods are at the intersection between computer science, Machine Learning, technology, database technologies, and statistics. The target of information mining is to extract information or knowledge from a dataset and transform it into a structure that may be understood. Most of the work on sentiment analysis focuses on review-based domains like movie and product reviews. Thus, the online discourse domain of sentiment analysis includes evaluating web forums, newsgroups, and blogs. This domain has, however, become a typical source of flames, virulent messages, and rants. (Abbasi, 2008)

For instance, extremists and terrorist groups communicate share ideologies and use the Twitter forum to promote radicalization and spread of hatred and doctrine. The thesis is concerned with determining hate speech sentiment on Twitter. However, many researchers have investigated the detection of flames and alarming messages in social media and the spread of hateful messages in dark web forums. With the emergence and enormous usage of Twitter, hate speech is being propagated at an alarming rate. Hate speech uses offensive and threatening language that targets certain groups of people based on their religion, ethnicity, nationality, color, or gender. The hate message source is typically a supposedly rival group, such as belonging to another ethnic community. The dissemination of hate messages may be through dedicated web sites associated with a cohesive group of members. Still, it may also be through popular sites such as Yahoo!, Twitter or Facebook, where topical issues or news articles may elicit responses laced with stereotypical language.

Lately, some works have introduced the domain of hate speech detection, but their focus is generally limited to supervised approaches to detecting racist speech (I. Kwok and Y. Wang, 2013). Typical definitions of hate speech refer to the content of speech, tone of speech, an evaluation of the nature of that speech, and targets of that speech, and the potential consequences or implications of speech act. (Raphael, 2011) The reasoning that this sits well with the sentiment analysis domain and envisages a design model that could capture the content and evaluative aspects of hate speech and develop a system that can detect and reduce the severity of hate messages.

2.4 Text Classification

Many researchers have come with different ways of defining or describing text classification.

Text Classification is the act of taking a set of labeled text documents, learning a correlation between a document's contents and its corresponding labels, and then predicting the labels of a set of unlabeled test documents as best as possible. Text Classification is sometimes referred to as Sentiment Analysis or Opinion Mining, or Text Mining. Sentiment analysis aims to identify the orientation (positive or negative) of opinions or emotions expressed in documents (Dulac, 2011). Sentiment analysis can be utilized to identify the favorable or unfavorable public opinions expressed in blogs about tribe, race, or gender that might cause violence, animosity between people from different tribes, races, or gender. The sentiment analysis task focuses on detecting and extracting opinions, feelings, and emotions in a text to a certain subject or object. A subtask of sentiment analysis is the sentiment categorization because of certain polarities.

It distinguishes between positive, neutral, or negative expressions or statements of extracted textual or spoken elements. Classical statistical Text Classification approaches are based on well-known machine learning models such as generative models (e.g., Naive Bayes) or discriminant models such as Support Vector Machines (Dulac,2011). Document classification or document categorization can be stated simply to assign a document to one or more predefined categories. This can be done either manually by intellectuals or automatically by the software. The manual classification has mostly been the province of library science, while the automatic classification is mainly performed using machine learning and information retrieval techniques (Yasotha, 2015).

Text mining is a division of Data mining. Data Mining refers to extracting informative knowledge from a large amount of data, expressed in different data types, such as transaction data in Electronic Commerce applications or genetic expressions in the bioinformatics research domain. Data mining’s main purpose is to discover hidden data or unseen knowledge, normally in patterns, from available data repositories (Xu, Zhang, & Li, 2011).

Web Mining or Web Data Mining utilizes data mining methods to induce and extract useful information from Web data information. Web mining may be classified into three categories based on the mining goals, which determine the Web’s part to be mined: Web content mining, web structure mining, and Web usage mining. Web content mining tries to discover valuable information from Web content. Web content mining is generally referred to as textual objects; thus, it is also alternatively termed text mining (Xu et al., 2011).

Information retrieval (IR) involves matching the text contained in a query or a document to a set of other documents. Often, the task involves finding the documents from a corpus of documents that are relevant to a user's query. The idea behind information retrieval is that if a user enters a query such as what is the capital of Sri Lanka?, then a good approach to finding the answer is to find a document that contains all (or some) of the words contained in the query (Coppin, 2004). The widespread and increasing availability of text documents in electronic form increases the importance of using automatic methods to analyze the content of text documents because the method of using domain experts to identify new text documents and allocate them to well- defined categories is time-consuming and expensive, has limits, and does not improve the continuous measure of the degree of confidence with which the allocation was made (Vinodhini, 2012).

Sentiment analysis is a type of natural language processing for tracking the public's mood about a particular product or topic. Sentiment analysis, which is also called opinion mining, involves building a system to collect and examine opinions about the product made in blog posts, comments, reviews, or tweets. Sentiment analysis can be useful in several ways. For example, in marketing, it helps judge the success of an advertising campaign or new product launch and determines which versions of a product or service are popular, and even identify which demographics like or dislike particular features (Vinodhini, 2012). 2.3.1 Basic Concept of Text Classification

Text categorization is the task of assigning a Boolean value to each pair $(d_{j,}c_{i)} \in D \times C$, where D is a domain of documents and $C = \{ C_{i,\ldots,}{}^{c}|^{c}\}|$is a set of predefined categories. A value of T assigned to $(d_{j,}c_{i)}$ indicates a decision to file $d_{j,}$ under c_i, while a value of F indicates a decision not to file $d_{j \text{ under}}$ c_i. More formally, the task is to approximate the unknown *target function* $\Phi :_{D \times C} \rightarrow \{T,F\}$ (that describes how documents ought to be classified) by means of a function $\Phi:_{D \times C} \rightarrow \{T,F\}$called the *classifier* (aka rule, or hypothesis, or model) such that Φ and Φ "coincide as much as possible". Classification $C = \{ C_{i,\ldots,}{}^{c}|^{c}| \}$ is viewed as consisting of $|c|$ independent problems of classifying the documents in D under a given category C_i, for i = 1; ,,,,,, $|c|$: A classifier for C_i is then a function $\Phi i :_{D \times C} \rightarrow \{T,F\}$that approximates an unknown target function $\Phi i :_{D \times C} \rightarrow \{T,F\}$ (Sebastiani, 2002).

2.4.1 Process of Text Classification

The text classification usually talks about the supervised classification, which has two stages: the training stage and the testing stage. Usually, the training stage includes creating the labeled corpora dataset, preprocessing the training text, vectorizing the text, and training the classifier. The testing stage includes preprocessing of testing text, vectorization, and classification of the testing text.

1. Creating Corpus

Twitter allows users to collect tweets with the help of Twitter API. To collect data from Twitter, Tweepy API was used. To use Twitter API, we must first have a Twitter account. Twitter provides two kinds of APIs: Rest API and Streaming API. The difference between these is REST APIs support connections for a short time interval, and only limited data can be collected at a time.

In contrast, Streaming API provides tweets in real-time and connections for a long time. We used Streaming API for our analysis. To collect a large number of tweets, we need Long-lived connections and a limited data rate. The data is obtained from Twitter in the .csv file; the data is stored in the database using MongoDB to be used for sentiment analysis. Comma-Separated Values (CSV) format was used for the collected data files because data consists of many fields.CSV separate each field with a comma, making it very easy to access the particular field text.CSV files also provide faster read/write time as compared to others. The corpus was divided into two sets: the training set and the testing set.

2. Pre-processing

Remove all the unnecessary elements in the text, such as stop words, punctuation, or unreadable text. This step is very important because it will affect the training of the classifier. Preprocessing should be performed to reduce the training dataset's size, leading to speeding up the training process.

3. Vectorization of Text

Transform the text into a vector that can be recognized by the computer. All text will be represented as a feature vector based on the features selected. Feature selection allows building a set of unique terms (features) across the corpus by excluding ambiguous terms.

4. Building and Training of the Classifier

To classify tweets in a different class (positive and negative), we build a classifier consisting of several machine learning classifiers to build our classifier. A library of Python called scikit-learn was used. Scikit-learn is a very powerful and most useful library in Python, which provides many classification algorithms. Scikit-learn includes classification, clustering, regression, and visualization; to install scikit-learn, we simply use Python's online command to install scikit-learn.

5. Classification

After getting the training model, the testing data was fed into the model and the classification. The classification is done using the Naïve Bayes algorithm and compared using other machine-learning algorithms, namely Support Vector Machine and Maximum Entropy (MaxEnt).

2.4.2 Text Representation

Transformations in this area are performed on the whole data. When training machine-learning models, however, the transformations are first to fit and applied to the training data, after which they are applied to the test data. To train machine-learning models, textual data needs to be transformed into numerical data using words as features and counting their occurrences in a sentence. Optionally, normalization of the text entries is performed before transforming text entries into numerical data to reduce the number of unique words or overcome the data's sparseness. The document representation is one of the preprocessing techniques used to reduce the documents' complexity and make it easier to handle; the document has to be transformed from the full-text version to a document vector. Text representation is an important aspect in document classification, which denotes the mapping of documents into a compact form of its contents. A text document is typically represented as a vector of term weights (word features) from a dictionary. Each term occurs at least once in a certain minimum number of documents. A major characteristic of the text classification problem is the extremely high dimensionality of text data. The number of various features often exceeds the number of training document(s). A document is that it is made of joint membership of terms with various patterns of occurrence. Text classification is an important component of many informational management tasks. However, with the explosive growth of the web data, algorithms that can improve the classification efficiency while maintaining accuracy are highly desired (Shang, 2006)

1. N-Grams

After performing morphological normalization, the resulting text entries may be further pre-processed into n-grams: text input of size n (either n words or n characters). N-Grams are widely employed in automatic text algorithms. At every position, the sequence of words or characters within the window is stored. During this study, we are going to work with subsets of n words. The foremost appropriate value for n depends on the dataset. When using bigrams (2-grams) or trigrams (3-grams) rather than unigrams (1-grams, Bag-of-Words), negations like ’niet leuk’ or ’niet zo leuk’ are often captured. However, the larger the worth for n, the sparser the resulting feature set will be. Therefore, unigrams and bigrams are all incorporated during this study. The Scikit-Learn package, a machine-learning package for Python, has the function CountVectorizer for transforming text entries into a matrix of token (n-grams) counts. The function has, amongst others, the parameter ngram_range that may be accustomed to indicate which sizes of n-grams are produced. Other parameters are the utmost number of features (max_features), the minimum document frequency (min_df), and, therefore, the maximum document frequency (max_df). When a maximum number of features is given, a subset of tokens (grams) is chosen in step with their term frequency across the corpus. Minimum document frequency is accustomed to remove highly infrequent words. Maximum document frequency is often wont to remove corpus-specific stop words. During this study, different values for these parameters are going to be evaluated. Appropriate values to contemplate rely on the dataset.

2. Bag-of-words

In natural language processing, the most widely used symbolic representation of features, but also quite effective, is the Bag of Words (Harris). This method's main idea is to simply create a feature for each word in the training texts. That feature for a text vector will then be valued 0 if the word does not appear in the corresponding text.

For this thesis, we utilize the simple bag-of-words method for feature extraction. Bag-of-words is arguably the most common feature extraction method for sentences and documents. The method considers a vocabulary of known words and a measure of the presence of known words. It is called a “bag” of words because any information about the order of the words is discarded, and we are only concerned with the presence of a word in a document, in this case, a tweet. The intuition for this method comes from the idea that similar tweets will usually contain similar content; e.g., negative tweets will usually contain the same negative words. Table 2.1 shows an example of the use of bag-of-words; in this case, a bag-of-unigrams. We considered unigrams, bigrams, and trigrams with their frequencies for our model. That is because combinations of certain words could give different meanings and sentiments in text

Table 2.1: use of a bag of words (unigram)

Tweets	Tweet 1: “This is a tweet.” Tweet 2: “This tweet is also a tweet.”
Bag-of-unigrams	Tweet 1: [a, is, this, tweet] Tweet 2: [a, also, is, this, tweet]
Feature Names	[a, also, is, this, tweet]
Feature Vectors	Tweet 1: [1, 0, 1, 1, 1] Tweet 2: [1, 1, 1, 1, 2]

Several Scikit-Learn methods made feature extraction a simple process. We used the method TfidfVectorizer, which contains more features than just converting the tweets into bags-of-words. TfidfVectorizer is equivalent to using CountVectorizer, followed by TfidfTransformer. CountVectorizer converts a collection of text documents (tweets in our case) to a matrix of token counts, as shown in Table 2.1. Tfidf Transformer transforms a count matrix to a normalized TF or TF-IDF representation. TF-IDF, which is short for term frequency-inverse document frequency, is a numerical statistic method to filter features by weighting and scoring each of the n-grams using the text's frequency of words. Since our feature vectors could be large from all the tweets we use, which would significantly increase our vector space's dimensionality, we use TF-IDF to extract significant words for each tweet. The feature vectors returned by TfidfVectorizer are fed into the LinearSVC model.

3. TF-IDF Vectorizer – Term Frequency Features (TF-IDF)

Term frequency-inverse document frequency vector maybe thanks to measuring the importance of a word or term. How rare a word is present in an exceedingly document can be checked using TF-IDF. So, using this vectorizer, the words with the highest importance as a feature can be obtained. The specialty of TF-IDF is the frequency of the term is offset by the frequency of the word in the corpus that clearly says that some words appear more frequently generally. All the TF-IDF features are extracted using Tfidfvectorizer in the Scikit-learn package. The same feature vector was passed for five different models, and the models' user testing data performance was evaluated.

Example of TF-IDF

A document has 100 words, and the word Samsung appears 3 times. The term frequency (i.e., tf) for Samsung is then (3 / 100) = 0.03. We assume we have 10 million documents, and the word Samsung appears in one thousand of these. The researcher then calculated the inverse document frequency (IDF) using this formula:

Log (10,000,000 / 1,000) = 4

Thus, the Tf-idf weight is the product of those quantities:

0.03 4 = 0.12.

2.5 Text Categorization

Text Categorization is the automatic classification of text documents under predefined categories or classes. Information Retrieval (IR) and Machine Learning (ML) techniques are used to assign keywords to the documents and classify them into specific categories. Machine learning helps us to categorize the documents automatically. Information Retrieval helps us to represent the text as an attribute. The task of automated text categorization has witnessed a thriving significance for a decade from both the researchers and the developers. (Montejo-Raez). Manually organizing large document bases is extremely difficult, time-consuming, error-prone, expensive, and is often not feasible. Automated text categorization is a viable option for larger organizations with time and money as the main constraints. Automated text categorization has reached the highest accuracy levels with a combination of IR and ML techniques when compared with trained professionals and comes as a rescue for

Modern Classification. Categorization is classifying the data for its most effective and efficient use. It is one of the most popular and important supervised learning techniques in data mining. Let (d_j, c_i) € D >> C, where D is the collection of documents and C= {c_1, c_2….$c_{|C|}$} are set of categories which are predefined. Then the main task of Text Categorization is to assign a Boolean value to each pair in D (Sebastiani, 2002).

2.5.1 General Approach to Text Categorization

A Categorization technique could be a systematic approach to create the categorization model from an input set of information. The technique requires a learning algorithm to spot a model that understands the connection between the attribute set and the input file's sophistication label. This learning algorithm should fit the input file well and predict the category labels of previously unknown records. For developing any categorization model, a set of computer file set is employed. This data set is subdivided into Training Data. This data set is subdivided into Training Data Set and Test Data Set (Pang-Ning Tan, 2005).

Training Data Set refers to gathering records whose class labels are already known and are employed to create the categorization model. It is then applied to the test data set. Test Data Set refers to gathering records whose class labels are known but should return the records' accurate class labels when given input to the built categorization model. It determines the model's accuracy supported by correct and incorrect test record predictions (Pang-Ning Tan, 2005).

There are many categorization techniques in use. They are:

1. Bayesian Categorization.
2. K Nearest Neighbor Categorization.
3. Decision Tree Categorization.
4. Rule-Based Categorization.
5. Support Vector Machines.
6. Neural Networks

In this thesis, we discussed and implemented Bayesian categorization methodology

2.5.2 Bayesian Categorization

Bayesian is one of the most well-known techniques of categorization. They can predict class membership probabilities, such as the probability that a given sample belongs to a particular class. i.e., the probability of a given record belongs to a particular category, which is based on Bayes Theorem. Bayes theorem is a simple mathematical formula used for calculating conditional probabilities.

The Bayesian Classification represents a supervised learning method as well as a statistical classification method. Assumes an underlying probabilistic model, and it allows us to capture uncertainty about the model in a principled way by determining the outcomes' probabilities. It can solve diagnostic and predictive problems. A classifier is a rule that assigns to an observation a guess or estimate of what the unobserved label was;

1. Bayes Theorem

Here we consider the relationship between supervised learning, or function approximation problems, and Bayesian reasoning. Let us study about Bayes Theorem using a small example. Consider a supervised learning problem in which an approximate of the unknown target function f: X → Y, or equivalently P(Y|X). To begin, assume Y is a boolean-valued random variable, and X is a vector containing n boolean attributes. In other words, X = X_1, X_2. . . , Xn, where X is the boolean random variable denoting the ith attribute of X. Applying the Bayes rule, see that P(Y = yi |X) can be represented as

$$P(Y=yi|\ X=x_k)=\frac{P(X=xk|Y=y_i)P(Y=y_i)}{\sum_j P(X=x_k|Y=y_j)\ P(Y=y_i)} \qquad (2.1)$$

Where y_m denotes the mth possible value for Y, x_k denotes the kth possible vector value for X, and where the summation in the denominator is over all legal values of the random variable Y. one way to learn P(Y |X) is to use the training data to estimate P (X|Y) and P(Y). These estimates, together with the Bayes rule above, to determine P (Y|X = x_k) for any new instance x_k

2.5.3 N-grams

N-gram phrases (or collocations) are fundamentally important in many areas of natural language processing (e.g., parsing, machine translation, and information retrieval). The phrase carries more information than the sum of its components; thus, it is much more crucial in determining the topics of document collections than individual words. However, most of the topic models (such as Latent Dirichlet Allocation (Blei, 2003)) assume that words are generated independently, i.e., under the bag of words assumption.

An N-gram is a token consisting of a series of characters or words. A token is generated by moving a sliding window across a corpus of text where the window's size depends on the size of the token N. Its displacement is done in stages; each stage corresponds to either a word or a character. Based on the different types of displacements, N-grams can be classified into two categories: Character-based and word-based (Kešelj, 2003). An n-gram is defined either as a textual sequence of length n, or similarly, as a sequence of n adjacent `textual units', in both cases extracted from a particular document. A `textual unit' can be identified at a byte, character, or word level, depending on the context of interest.

The simplest n-gram is the so-called unigram, where n = 1, which falls back to the single-minded \bag-of-words" (BOW) representation. Typically, n is a _xed number, highly dependent on the particular corpus of documents and the queries made against that corpus. Each of the n-grams is a coordinate in a vector representing the text under study, and the frequency that this n-gram appears in the text can be the number of this coordinate (Bouras, 2016). (Dan Jurafsky, 2014) defined an n-gram as a sequence of n words: a 1-gram (unigram), a 2-gram (or bigram) is a two-word sequence, and a 3-word (or trigram) is a three-word sequence of words. This unigram representation has been called the bag of words model.

Unigram model can be thought of as putting the training corpus's words in a bag and then selecting words one at a time. The notion of order of the words is lost; a unigram model gives the same probability to any text's permutation. Higher-order n-gram models maintain some local notion of word order (Russel, 2010). N-Grams are the basic method for text categorization. It is also a statistical-based approach for classifying text. The N is the number of keywords used for dividing the input text. Based on the number of keywords used, the N-grams are called 2-grams, 3-grams, etc. It can classify the unknown text with the highest certainty (Yadav, 2015). In n-grams, initially, the general pre-processing is carried out. Then the document is divided into N-grams or N-shingles. This refers to a sequence of consecutive words of size `N,' where `N' is user-specified. Both suspicious and source documents are converted to their N-gram profiles, and similarity is calculated using Dice's coefficient. This is similar to the Jaccard coefficient, but it reduces shared terms between the documents (Vani, 2014).

2.5.4 Applications of N-grams

Shannon (1948) used character-based N-grams for analyzing and predicting printed English. Since then, his approach with character-based N-grams has been applied to other areas like spelling and error correction, text compression, language identification, and text search and retrieval (Mohan, 2010). There are two obvious bases for the characterization and manipulation of text. These are either the individual characters that form the basis for the byte-level operations available to computers or the individual words used by people. These basic units can then be assembled into larger text segments such as sentences, paragraphs, chapters, etc. N-grams provide an intermediate level of text characterization with advantages in terms of efficiency and/or effectiveness over the conventional character-based or word-based approaches to several important applications in textual computing. Applications that can be implemented efficiently and effectively using sets of n-grams include spelling error detection and correction, query expansion, information retrieval with serial, inverted and signature les, dictionary look-up, text compression, and language identification (Robertson, 1998). The use of n-gram probability distribution and n-gram models in Natural Language Processing (NLP) is a relatively simple idea, but it is effective in many applications

1. Language Identification

The basic idea is to identify N-grams whose occurrence in a document gives strong evidence for or against identifying a text as belonging to a particular language. Although this has been done before, it makes a good test case for our text categorization method. The acquaintance algorithm is a commonly used method for language identification that uses n-grams. An n-gram is simply a collection of n letters. Still, detailed statistics indicate the likelihood of a particular set of letters occurring in any given language. When the acquaintance algorithm is presented with sufficient text, it can identify the language with a surprisingly high degree of accuracy (Coppin, 2004).

2. Spelling-error Correction

The first stage of the Multi Spell algorithm is to compare the keywords given by the user with the correct words contained in the dictionary. First of all, check based on the used dictionary if the word is misspelled. If this is the case, the algorithm builds n-grams for the misspelled word. Then, selecting correction candidates from the dictionary. To keep the number of correction candidates as small as possible (Akshay, 2016) carried out next-word predictions by implementing n-gram language models. N-grams were implemented by adding additional nodes to an existing suffix tree and sorting them to their frequencies. N-grams were developed in two phases. In the first phase, predictions were made based on a statistical model, consisting of default frequencies of words in a dictionary. In the second phase, frequencies were modified with each iteration according to the user's typing style. MultiSpell has also been integrated as a pre-processing approach. It can be applied to queries and documents to support users during a keyword-based and semantic-based search. The first is an important task for retrieving the relevant documents related to the query identifying the misspelled words, and correct them for a correct interpretation.

3. Content-based filtering

Content-based filtering is mainly based on the use of the Machine Learning (ML) paradigm. In ML, a classifier is automatically induced by learning from a set of pre-classified examples. The feature extraction procedure maps text into a compact representation of its content, uniformly applied to training and generalization phases. Bag-of-Words (BoW) approach yields good performance and exists in general, over more sophisticated text representation that may have superior semantics but a lower statistical quality (Thilagavathi, 2014).

4. Machine Language Translation

One of the most successful system combinations for MT is based on confusion network decoding, as described in (Rosti 2007). Given translation hypotheses from multiple MT systems, one of the hypotheses is selected as the backbone for hypothesis alignment. This is usually done by a sentence-level Minimum Bayes Risk (MBR) re-ranking method. The confusion network is constructed by aligning all these hypotheses against the backbone. Words that align to each other are grouped into a correspondence set, constituting the confusion network's competition links. Each path in the network passes exactly one link from each correspondence set. The final consensus output relies on a decoding procedure that chooses a path with the maximum confidence score among all paths that pass the confusion network.

5. Spelling-error detection

Spell checking is a process including detecting, correcting, or providing spelling suggestions for misspelled words'-grams can be used for spelling check and correction processes. The first step to use n-grams is to find the language-specific n-grams by using a corpus. Spelling is very important because the misspelled words can cause some misunderstandings. Hence, a spelling checking system is necessary to make these documents are correct. This is one of the pre-processing steps for Natural Language Processing (NLP) problems. In (Samani 2015) developed a real-word error checker and detector system for the Persian language using a predefined candidate set and made content-aware choices based on frequent adjacent n-grams of each potentially real word error. They defined real-word errors as the misspelled words that have been converted wrongly to another lexicon word. Detection of these errors required semantic analysis of the content. Their results indicated that their proposed system had a high performance, which they attributed to good coverage of Persian language real-words error and using proper algorithms and parameters.

2.5.5 Applications of Text Categorization

Since Maron's (1961) work, TC has been applied in several different applications. The borders between the different classes of applications are fuzzy and somehow artificial, and some of these may be considered special cases of others (Sebastiani, 2002). With the explosive growth of social media (e.g., reviews, forum discussions, blogs, micro-blogs, Twitter, comments, and postings in social network sites) on the Web, individuals and organizations are increasingly using the content in these media for decision making (Liu, 2007). The following section describes application areas of text categorization.

1. Document Organization

A document organization could be a collection of documents composed of labeled clusters that contain similar documents. Note that a group of non-clustered documents isn't a document organization. If the document organization contains clusters with nested clusters, it's called a hierarchical document organization. If its clusters don't have any nested clusters, it's called a flat document organization. It's necessary to create a document organization, manually or automatically, to manage documents efficiently. There are two sorts of document organizations, static document organization and dynamic document organization. If the clusters of the document organization are fixed permanently, it's called a static document organization. If it adapts by itself to the present situation, we seek advice from the document organization as a dynamic document organization. Indexing with a controlled vocabulary is an instance of the overall problem of the document base organization.

2. Text Filtering

Text filtering is the activity of classifying a stream of incoming documents dispatched asynchronously by an information producer to an information consumer (Belkin, 1992). A typical case could be a news feed filter where the producer could be a wire service, and therefore, the consumer could be a newspaper. During this case, the filtering system should block the delivery of the documents the patron is probably going, not fascinated by (e.g., all news not concerning sports, within the case of a sports newspaper). Filtering is often seen as a case of single-label TC, that is, the classification of incoming documents into two disjoint categories, the relevant and also the irrelevant. Additionally, a filtering system may further classify the documents deemed relevant to the buyer into thematic categories; within the example above, all articles about sports should be further classified consistent with which sport they handle, to permit journalists specialized in individual sports to access only documents of prospective interest for them. Similarly, an e-mail filter may well be trained to discard \junk" mail and further classify non-junk mail into topical categories of interest to the user (Androutsopoulos, 2000).

It is necessary to filter the corpus of two million tweets before the manual labeling process. The filter is intended as a two-step process that produces the use of seven files. Six of them are dictionaries of words representing different hate types, whereas the last contains generic insults. Words inside each of the primary six files are tagged. A word is alleged to contain hate if the word unequivocally expresses hate, irrespective of its context. Otherwise, if hate depends on the context, the word is alleged to contain relative hate. Otherwise, if hate depends on the context, the word is claimed to contain relative hate. For example: “f**CK” contains absolute hate, whereas “s*x,” which also denotes intimate English, may contain hate or not reckoning on the context.

3. Hierarchical categorization of Web pages

Text categorization aroused lots of interest for its possible application to automatically classifying sites, or sites, under the hierarchical catalogs hosted by popular Internet portals. When Web documents are cataloged this way, instead of issuing a question to a general-purpose Web program, a searcher may find it easier to navigate the hierarchy of categories and then restrict the search to a selected category of interest. Classifying sites automatically have obvious advantages since the manual categorization of an outsized enough subset of the net is infeasible (Sebastiani, 2002). Automatic categorization of web documents (e.g., HTML documents) denotes the task of automatically finding relevant categories for a (new) document, which is to be inserted into an online catalog.

4. Automatic survey coding

Survey coding is assigning a symbolic code from a predefined set of such codes to the solution that an individual has given in response to an open-ended question in a very questionnaire. This task is sometimes disbursed to group respondents per predefined scheme that supported their answers. Survey coding could be a difficult task since the code that ought to be attributed to a respondent-based on the solution she has given could be a matter of subjective judgment and thus requires expertise. The matter will be formulated as a single-label text categorization problem, where the answers play the documents' role. Therefore, the codes that apply to the answers returned to a given question play categories. Text categorization(TC) is that the task of automatically building, utilizing machine learning (ML) techniques, automatic text classifiers, i.e., programs capable of labeling linguistic communication texts from a website D with thematic categories from a predefined set C=C. Each document in D must be tagged with exactly one category from C is named multiclass TC.

Since the survey, coding is usually a multiclass TC task. In multiclass, TC effectiveness is measured in terms of accuracy, defined because of the ratio between the number of correct classification decisions and, therefore, the total number of classification decisions. The development of an automatic text classifier relies on a labeled corpus Ω=Ωof documents pre-classified under C. A general inductive process (called the learner) automatically builds a classifier for C by learning C's characteristics from a training set Tr=Trof documents. Once a classifier has been built, its effectiveness (i.e., its capability to require the proper categorization decisions) is also tested by applying it to the test set Te=Ω−Tr and checking the results' accuracy. In our survey-coding context, the set of all answers to a given question play the role of D, and the set of all possible codes that will be attributed to a solution to an issue play the role of C. Once they need been built, the learners (and to the classifiers) consist of a solution DJ represented as a vector of term weights dj=(w1j,...,w|T|j). Here, T is that the dictionary, i.e., the set of words that occur a minimum of once within the training set, and 0≤wkj≤1 quantifies the importance of tk in characterizing the semantics dj. Automatic indexing with controlled dictionaries is closely associated with automated metadata generation. In digital libraries, one is typically curious about tagging documents by metadata that describe them under a spread of aspects (e.g.creation date, document type or format, availability, etc

5. Word Sense Disambiguation

Word sense disambiguation (WSD) is an enabling technology for applications such as IR, Information extraction, Machine Translation (MT), question answering systems, cross-language applications, and document classification. Word Sense Disambiguation is the process of selecting the correct sense for a word in a context. Knowledge-based supervised and unsupervised approaches are used in WSD. In a supervised approach, classifiers are used for assigning the correct sense of each word (Dwivedi, 2014). Word sense disambiguation (WSD) is the activity of finding, given the occurrence in a text of an ambiguous (i.e., Polysemous or homonymous) word, the sense of this particular word occurrence. For instance, a bank may have (at least) two different senses in English, as in the Bank of England (a financial institution) or the bank of River Thames (a hydraulic engineering artifact). Thus, it is a WSD task to decide which of the above senses the bank's occurrence in Last week, I borrowed some money from the bank has. WSD is very important for many applications, including natural language processing and indexing documents by word senses rather than words for IR purposes. We partially used a manually prepared list of words and sent word as the lexicon to determine our experiments' correct sense. For example, synsets of “whore” are listed in table 2.2. Moreover, the sense concept is expanded with its synset relationship such as hypernym, hyponym, meronym, holonym, etc., except for antonym relation. This expansion is called the signature of the sense.

Table 2.2: Synsets of “whore”.

Synet	**Synonyms**	**Definition & Example**
Whore	Malaya	A person who engages in promiscuous sex for money.
Wore	prostitute	A woman whose behaviour in her sexual relationships is considered immoral.
W*hore	sex worker	A woman who has many sexual partners.
Hore	**slut**	A person who is regarded as willing to do anything to get a particular thing.

2.6 Machine Learning

Machine learning techniques are accustomed to classify text documents in keeping with their sentiment (Sharma, 2012). The foremost widely used technique for sentiment analysis is supervised machine learning. (Alexander Pak, 2010) studied the Naïve Bayes classifier's behavior employing a Twitter corpus and reported that Bayesian analysis works well compared to Support Vector Machine analysis (Alexander Pak, 2010). (Eibe Frank,2006) studied the Naïve Bayes classifier by varying the number of words present in each class and concluded that unbalanced classes negatively affect the classifier's performance. They also proposed a centroid based class normalization technique to counter the matter. The altered classifier works on two out of 4 datasets utilized in the experiment. They also state that the required results will be observed by altering the algorithm's parameters' values.

The machine learning (ML) approach applicable to sentiment analysis mostly belongs to supervised classification. It includes two sets of data: a training and a test set. An automatic classifier uses the training set to learn the differentiating characteristics of documents, and a test set is used to validate the performance of the automatic classifier. Machine learning techniques like Naïve Bayes (NB), maximum entropy (MaxEnt), and support vector machines (SVMs) have achieved great success in text categorization. The other most well- known machine learning methods in the natural language processing area are K- Nearest neighborhood, ID3, C5, centroid classifier, winnow classifier, and the N-gram model (Vinodhini, 2012). Then a model is built from some. The machine learning (ML) approach applicable to sentiment analysis mostly belongs to supervised classification. It includes two sets of data: a training and a test set.

An automatic classifier employs the training set to find out the differentiating characteristics of documents. A test set is employed to validate the performance of the automated classifier. Machine learning techniques like Naïve Bayes (NB), maximum entropy (MaxEnt), and support vector machines (SVMs) have achieved great success in text categorization. The opposite most well-known machine learning methods within the linguistic communication processing area are K- Nearest neighborhood, ID3, C5, centroid classifier, winnow classifier, and the N-gram model (Vinodhini, 2012). A model is then made from some training data using the extracted features. the rationale for using machine learning is that machine-learning techniques outperform humans within the sense that humans training data using the extracted features. Using machine learning is that machine-learning techniques outperform humans, in the sense that humans cannot handle a large volume of data.

A machine-learning model's performance and results usually rely upon the standard of the training data or databases. Overall, the main goal of machine learning is to spot complex structures and automatically make intelligent decisions. Twitter data are sequences of string characters. To use automatic classification algorithms, special representation must be wont to make data suitable for computation. In our work, two varieties of presentations were used: Bag-of-Words, N-grams, and linguistic features. A training set is employed by an automatic classifier to be told the differentiating characteristics of documents, and a test set is employed to test how well the classifier performs. Text Classification Problem Definition: a collection of coaching records D = where each record is labeled to a category. The classification model is expounded to the underlying record features to at least one of the category labels. For a given instance of an unknown class, the model is employed to predict a category label for it. The hard classification problem is when just one label is assigned to an instance. The soft classification problem is when a probabilistic value of labels is assigned to an instance.

Sentiment analysis is a vicinity of tongue processing that aims to determine opinions, attitudes of a writer within the text, or their attitude towards specific topics. Sentiment describes an opinion or attitude expressed by a private, the opinion holder, about an entity, the target. It's also called opinion mining because it derives the speaker's opinion or the user about some topic. Sentiment analysis helps research into online communication because it gives researchers the flexibility to automatically measure emotion in online texts. The research field of sentiment analysis has developed algorithms to automatically detect sentiment within the text (Pang, 2008). While some identify the objects discussed and also the polarity (positive, negative, or neutral) of sentiment expressed about them (Gamon, 2005), other algorithms assign an overall polarity to a text, like a movie review (Pang,2004).

Three common sentiment analysis approaches are full-text machine learning, lexicon-based methods, and linguistic analysis. For normal machine learning (Witten, 2005), a group of texts annotated for polarity by human coders is wont to train an algorithm to detect features that accompany positive, negative, and neutral categories. The text features used are typically sets of all words, word pairs, and word triples found within 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.

In contrast, a linguistic analysis exploits the text's grammatical structure to predict its polarity, often in conjunction with a lexicon. For example, linguistic algorithms may identify context, negations, superlatives, and idioms as a part of the polarity prediction process (Wilson, 2009). In practice, algorithms often employ multiple methods along with various refinements, like pre-filtering the features hunted for (Wilson, 2006) and methods to deal with over some time (Bifet, 2005).

A few algorithms detect sentiment strength additionally to sentiment polarity (Lee, 2005) (Strapparava, 2008), including some for casual online text (Neviarouskaya, 2007). These work on the idea that humans can differentiate between mild and powerful emotions in text. As an example, hate could also be thought to be a stronger negative emotion than dislike. Sentiment strength algorithms try to assign a numerical value to texts to point to the strength of any sentiment detected.

2.6.1 Supervised Machine Learning Methods

Supervised machine learning methods assume the presence of labeled training data that are used for the learning process. The latter estimates the input dataset's output, referring to when the classifier defines the label the object belongs to. As the training data set, labeled documents have to be used. Usually, the bag-of-words model by Tang, 2016) is employed to represent a document as a feature vector $d=(w_1,w_2,\ldots,w_i,\ldots,w_N)$, where N is set of all the unique terms in the training dataset, and w_i is the weight of the i-th term. To convert the training dataset to a feature vector, vocabulary with N unique words must be created from the training data.

1. Naive Bayes (NB)

The Naive Bayes classifier is that the simplest and most ordinarily used classifier. Naıve Bayes classification model computes the posterior probability of a category, supported the distribution of the words within the document. All tweet texts were transformed into term vectors to be processed by the classifier. To perform classification, selected features from the info were used first. For text classification, the feature vector is called the term vector, which is the most vital structure during the training and classification process. All tweet texts are transformed into term vectors to be processed by the classifier. A vector is generated supported by a novel vocabulary generated from the training dataset, and there aren't any duplicate words within the vocabulary. The term vector's scale is that the term vector's size is the vocabulary size. There are two types of Naive Bayes implementations:

Multinomial. The biggest difference between them is how features are extracted from the documents. Bernoulli implementation as an example, for a sentence sort of a tweet, a term vector will have initialized with all elements adequate zero. Then check each word within the vocabulary to determine if the word exists within the tweet. If it exists, mark the term vector's corresponding element to 1; if not, mark the term vector's corresponding element to 0. If the vocabulary is sufficiently large, every tweet is often represented by employing a term vector with 0s and 1s. The model works with the BOWs feature extraction, which ignores the word's position within the document.

It uses Bayes Theorem to predict the probability that a given feature set belongs to a selected label (label) is that the prior probability of a label or the likelihood that a random feature sets the label. P (features label) is that the prior probability that a given feature set is being classified as a label (features) is the prior probability that a given feature set occurs. Given the Naive assumption, which states that each one feature is independent. Bayesian Network (BN). The most assumption of the NB classifier is that the independence of the features. The opposite extreme assumption is to assume that every one of the features is fully dependent. This results in the Bayesian Network model, a directed acyclic graph whose nodes represent random variables, and edges represent conditional dependencies. BN is taken into account an entire model for the variables and their relationships. Therefore, an entire probability Distribution (JPD) over all the variables is specified for a model. In-Text mining, BN's computation complexity is extremely expensive; that's why it's not frequently used.BN was employed by (Hernández 2012) to contemplate a real-world problem within which the author's attitude is characterized by three different (but related) target variables. (Hernández, 2012), proposed the utilization of multi-dimensional Bayesian network classifiers. It joined the various target variables within the same classification task to use the potential relationships between them. They extended the multi-dimensional classification framework to the semi-supervised domain to require advantage of the massive amount of unlabeled information available during this context. (Hernández, 2012). Their semi-supervised multidimensional approach outperforms the foremost common Sentiment Analysis approaches. Their classifier is that the best solution is a semi-supervised framework because it matches the particular underlying domain structure.

1) Naïve Bayes Categorization

Naive Bayes categorization is one of the simplest probabilistic Bayesian categorizations. It is based on the assumption that the effect of an attribute value on a given category is independent of the values of other attributes, which is called conditional independence. It is used to simplify complex computations. The Naïve Bayes Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the inputs' Trees dimensionality is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods. The Naive Bayes classifier is a probabilistic classifier that is based on the Naïve Bayes assumption. The Naive Bayes algorithm is a classification algorithm based on the Bayes rule, which assumes the attributes X_1….. X_n is all conditionally independent of one another, given Y. This assumption's value is that it dramatically simplifies the representation of P (X|Y) and the problem of estimating it from the training data. Consider, for example, the case where = X_1, X_2. In this case

$$\begin{aligned} P(X|Y) &= P(X_1, X_2|Y) \\ &= P(X_1|X_2, Y)P(X_2|Y) \\ &= P(X_1|Y)\,P(X_2|Y) \end{aligned} \tag{2.2}$$

The second line follows from a general property of probabilities, and the third line follows directly from our above definition of conditional independence. More generally, when X contains n attributes which are conditionally independent of one another given Y, then;P (X_1...X_n|Y) =∏i=1 P (Xi|Y). Now when Y and the X_i are Boolean variables. Only 2n parameters to define P (Xi= xik|Y = y_j) for the necessary i, j, k. This is a dramatic reduction compared to the 2(2_n− 1) parameters needed to characterize P (X|Y) if make no conditional independence assumption. Deriving the Naive Bayes algorithm, assuming in general that Y is any discrete-valued variable and the attributes X_1. . . X_n is any discrete or real-valued attributes. Our goal is to train a classifier that will output the probability distribution over Y's possible values for each new instance X that asked it to classify. The expression for the probability that Y will take on its kth possible value, according to Bayes rule (Equation1), is

$$\mathrm{P}\,(\,\mathrm{Y} = \mathrm{yi} \mid \mathrm{x1}\ldots\mathrm{Xn}\,) = \mathrm{P}\,\frac{\mathrm{P(Y{=}yk)\ P(x1\ldots Xn|Y{=}yk)}}{\sum \mathrm{jP(Y{=}yj)\ P(x1\ldots xn|Y{=}yj)}}$$

(2.3)

Where the sum is taken over all possible values y j of Y. Now, assuming the X_i are conditionally independent given Y, equation (2.3) is used to rewrite (Equation 2.4) this as

$$\mathrm{P(Y = yi \mid X1\ldots Xn)}\,\frac{P(Y = \mathrm{yk})\prod \mathrm{iP\ (Xi \mid Y = yk)}}{\sum \mathrm{jP(Y = yj)}\ \prod \mathrm{iP\ (Xi \mid Y =}}$$

(2. 4)

Equation (2) is the fundamental equation for the Naive Bayes classifier. Given a new instance X^{new}=X_1. . . X_n, this equation shows how to calculate the probability that Y will take on any given value, given the observed attribute values of X new and given the distributions

P(Y) and P (X_i|Y) estimated from the training data. With the only interest in the most probable value of Y, then the Naive Bayes classification rule will be:

$$Y \leftarrow \text{argmaxyk} \frac{P(Y = yk)\prod iP\ (Xi \mid Y = yk)}{\sum jP(Y = yj)\ \prod iP\ (Xi \mid Y = yj)} \qquad (2.\ 5)$$

Which simplifies to the following (because the denominator does not depend on y_k).

$$Y \leftarrow \text{argmax}y_k\ P(Y=y_k) \prod_i P\ (X_i|Y=y_k) \qquad (2.6)$$

Unigram Naive Bayes

For unigram Naive Bayes, the probability of a term belonging to a class is given as the empirical counts of that term in messages with the same class.

$$P(\text{tk}|\text{c}) \frac{\text{TxC}^2\text{k}}{|\text{Vc}|} \qquad (2.7)$$

Tctk is the number of times the term is associated with the class, and Vc is the total number of terms seen. In contrast to the multinomial model, the Bernoulli multivariate model deals with the number of documents containing the term for that class divided by the class's total number of documents. The binarized variation of the multinomial model clips the word count in each document as one (Sagar, 2014).

Bigram Naive Bayes

The bigram Naive Bayes classifier calculates the probability that a document belongs to a class based on the number of times word pairs are seen. The training set becomes sparse, linear interpolation, and the back o_ model can be used. The linear interpolation weighs the unigram as well as bigram probabilities to calculate the overall probability of the document using the following equation;

$$p(cjd) = wPunigram\left(\frac{c}{d}\right) + 1 - w)Pbigram(C/d)/ \tag{2. 8}$$

The back off-model uses the bigram probability if seen with the class or else backs off to the unigram probability (Sagar, 2014). One of the main reasons that make the NB model works well for text-domain is because the pieces of evidence are \vocabularies" or \words" appearing in texts, and the size of the vocabularies is typically in the range of thousands. The large size of pieces of evidence (or vocabularies) makes the NB model work well for the text classification problem, as cited in (Swamy 2014). Naive Bayes works best with textual and numeric data formats and is easy to implement and compute.

Algorithm Implementation

The equation of Naive Bayes for sentiment classification

$$\mathrm{P}(₡\,\mathrm{D})\ \frac{\mathrm{P(D|C)\ P(C)\infty P(D|C)\ P(C)}}{\mathrm{P(D}} \tag{2.9}$$

From the equation that the two most important values needed to compute the posterior probabilities are class prior probabilities P(c) and likelihoods P (D/C). For a training set, it seems that to get these two values, some counts are needed. For instance, use C to refer to the tweet label and X to refer to the features (words). What its needed are:

1. The number of tweets: the number of tweets.
2. Number of C: the number of unique labels.
3. The number of C = c: the number of tweets with label c. This should be a global variable and can be used with the previous number to compute label prior probabilities.

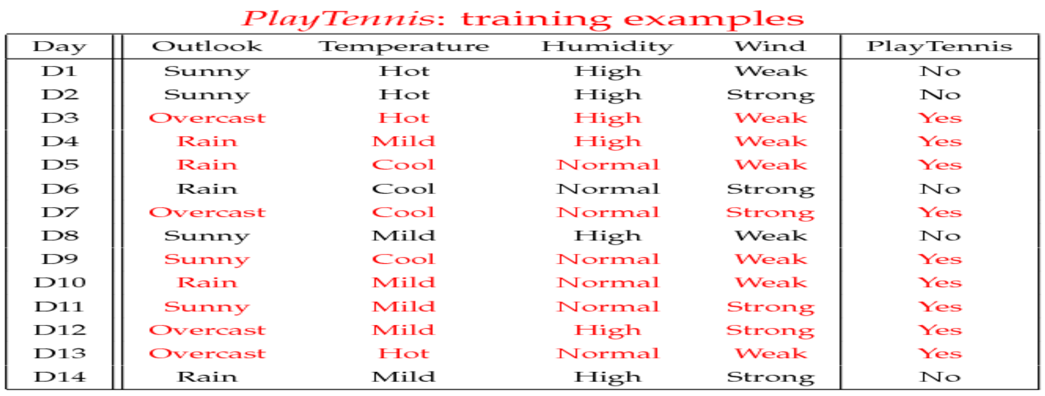
4. Number of X in C= c: the number of words in all the tweets with the label c.
5. The number of X= x in C= c: the number of word x that is in all the tweets with label c. This number and the previous number are combined to compute the probabilities of a single word x under a label c.
6. The total number of unique words in all tweets. This is the size of our vocabulary. From the equation, getting the prior probabilities P(c) and likelihoods P (D/C) is attained. And the goal is to build the prediction model, which means test the training phase is needed.

1. Bayesian network

Bayesian networks (Pearl, 1998) are powerful and widespread tools for modeling uncertainty in a few domains. These probabilistic. Appropriate inference algorithms may query a Bayesian network to extract probabilistic information about interest variables after their specification. Among others, classification represents a very important application of Bayesian networks. A number of the foremost used classifiers proposed within the Bayesian theory of probability, just like the naive Bayes classifier and the tree-augmented naive Bayes classifier, will be considered learning/inference algorithms Bayesian networks with particular topologies.

Bayesian networks are precise models, within the sense that exact numeric values should be provided as probabilities needed for the model parameters. This requirement is typically too narrow. There are situations where one probability distribution cannot properly describe the uncertainty about a variable's state. Therefore, Bayesian networks provide a probability mass function specification, describing the probabilistic relations among the variables' full set. The specification is compact within the sense that only probability mass functions for the variables conditional on (any possible value of) the fogeys should be assessed. Once a Bayesian network has been specified, a typical task we'd consider consists of querying the model to assemble probabilistic information about the state of a variable given evidence about the states of some others.

Table 2. 3: Example of Naive Bayesian Network diagram.



PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Table 2.4: Learning phase

Outlook	*Yes*	*No*	Temperature	*Yes*	*No*
Sunny	2/9	3/5	*Hot*	2/9	2/5
Overcast	4/9	0/5	*Mild*	4/9	2/5
Rain	3/9	2/5	*Cool*	3/9	1/5

Humidity	*Yes*	*no*	Wind	*Yes*	*No*
High	3/9	4/5	*Strong*	3/9	3/5
Normal	6/9	1/5	*Weak*	6/9	2/5

P (play =yes) =9/14 P (play =no) =5/14

Testing phase (inference)

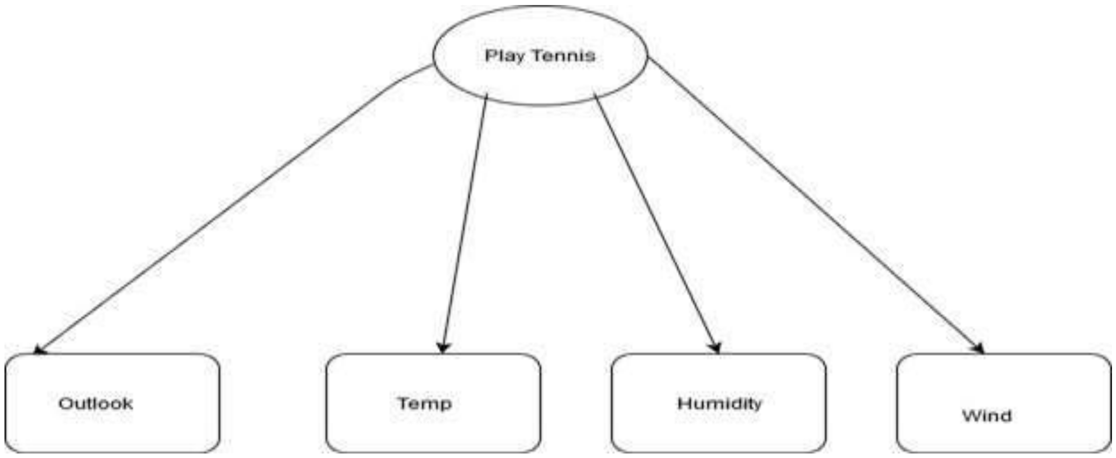
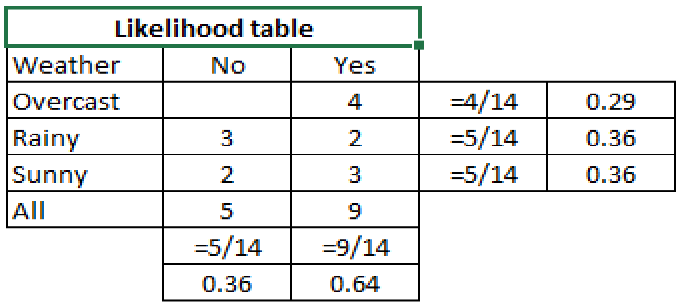


Figure 2.1: Testing phase (inference)

Table 2. 5: Likelihood table



Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

2. Maximum-Entropy

Other names also know this model as Maximum-Entropy (MaxEnt) classification or log-linear classifier. A logistic function is used in this model, where probability describes the outcome of a single trial. The logistic regression can be implemented from the Scikit-learn library of Python, in which there is a class named Logistic Regression. MaxEnt models' idea is that one should prefer the most uniform models that satisfy a given constraint (Nigam, 2009). Given an independent variable, the model predicts possible outcomes for a categorically distributed dependent variable. (Pang, 2002) reports that from previous studies, MaxEnt outperformed NB sometimes, but not always. This is because, unlike NB, MaxEnt makes no assumptions about the relationships between features; therefore, it might potentially perform better when conditional independence assumptions are not met. McDonald's, Ryan's (2009) experiment and analysis gave significant support for the mixture weight method for training very large-scale conditional maximum entropy models with L2 regularization. The idea behind the centroid classification algorithm is simple and straightforward (Songbo, 2008). A classifier is the centroid classifier approach applied to text classification using Term Frequency-Inverse Document Frequency (TF-IDF). (Erik, 2013) I applied the centroid algorithm to perform open domain sentiment analysis. They blended the largest existing taxonomy of common knowledge with a natural-language-based semantic network of commonsense knowledge then applied multi-dimensional scaling on the resulting knowledgebase.

Max Entropy does not assume that the features are conditionally independent of each other. The MaxEnt is based on the Principle of Maximum Entropy, and from all the models that fit our training data, selects the one with the largest entropy. Probability distribution equals

$$P(x,y)=\frac{1}{N} \times number\ of\ times\,(x,y) occurs\ in\ the\ sample$$

Its estimate of P (Ck | x) takes the following exponential form

$$P(Ck \mid x) = \frac{1}{z(d)} \exp = \left(\sum_i \lambda\ F(d,c)\right)$$

Where Z (d) is the normalization function. F is the feature function for each sentence and is defined as follows

$$F(d,c) = (x+a)^n = \begin{cases} 1\ n(d) \\ 0\ otherwise \end{cases} > 0$$

3. Support Vector Machine (SVMs)

SVM are supervised machine learning methods used for classification, regression, and detection models. SVM is more effective for high dimensional space. SVCs are capable of multi-class classification. SVC and NuSVC are similar, whereas LinearSVC is based on linear kernels. Support vector machines (SVM) are a method that considers that each set of features represents a position inside a hyperspace. The SVM tries to divide it using a hyperplane, maximizing the distance between this hyperplane and each vector, minimizing the objective function. This space division is hard to accomplish and sometimes impossible; for this, the SVM can use a margin that allows misclassifying some examples but increases the overall performance. Figure 2.1 illustrates linearly separable training data, along with labels for the most important SVM elements. When the classification problem is not linearly separable, the algorithm can use kernel functions. Kernel functions transform low-dimensional input space into a higher-dimensional space, which is then linearly separable.

The main principle of SVMs is to see linear separators within the search space, which may best separate the various classes. In there are 2 classes x, o, and there are 3 hyperplanes A, B, and C. Hyperplane A provides the simplest separation between the classes because the traditional distance of any of the information points is that the largest, so it represents the most margin of separation. Text data are ideally suited to SVM classification thanks to the text's sparse nature, during which few features are irrelevant. Still, they tend to be correlated with each other and are usually organized into linearly separable categories.

SVM can construct a nonlinear decision surface within the original feature space by mapping the information instances non-linearly to a dot product space where the classes will be separated linearly with a hyperplane. SVMs are employed in many applications; among these applications are classifying reviews in line with their quality. SVMs were employed by (Yung-Ming Li, 2013) as a sentiment polarity classifier. Unlike the binary classification problem, they argued that opinion subjectivity and expresser credibility should be considered.

Yung-Ming Li (2013) proposed a framework that provides a compact numeric summarization of opinions on micro-blogs platforms. They identified and extracted the topics mentioned in the opinions associated with users' queries and then classified the opinions using SVC. They worked on twitter posts for their experiment. They found out that the consideration of user credibility and opinion subjectivity is essential for aggregating micro-blog opinions. They proved that their mechanism could effectively discover market intelligence (MI) for supporting decision-makers by establishing a monitoring system to track external opinions on different aspects of a business in real-time (Yung-Ming Li, 2013).

The researcher then plotted the training vectors (xi) into a higher dimensional space by the function φ in the Kernel functions. SVM finds a linear splitting hyperplane with the utmost margin in this higher-dimensional space. $C > 0$ is the penalty parameter of the error term.

Furthermore, $K (xi, xj) \equiv \varphi (xi) T \varphi (xj)$, (S. Kotsiantis, 2006). The following are the four basic kernels use in SVM:

$K(x1,xj)= x T I Xj$

Linear: $K xi, xj = x T i xj$

Polynomial Kernel: $K \ , xj = [(xi . xj + 1)]$

Radial Basis Kernel: K (, xj)=exp (- Y $\frac{IIXi - XjII}{2a^2}A$)

Sigmoid Kernel: K(X,Y)=tanh(γ·XTY+r)

Sigmoid Kernel: (i , xj = tanh$f0$(v xi , xj + C)

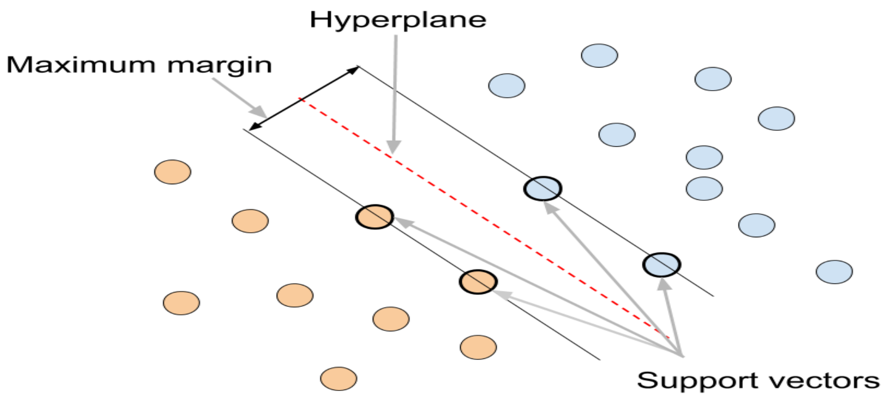


Figure 2.2: SVM hyperplane separation

2.6.2 Unsupervised Machine Learning Methods

Unsupervised learning shows how systems can learn to represent particular input patterns that reflect the overall collection of input patterns' statistical structure. With unsupervised, there are no explicit target outputs or environmental evaluations associated with each input; rather, it brings to bear prior biases as to what aspects of the input structure should be captured in the output. Unsupervised learning is important since it is likely to be much more common in the brain than supervised learning.

The only things that unsupervised learning methods have to work with are the observed input patterns xi, which is often assumed to be independent samples from an underlying unknown probability distribution PI[x], and some explicit or implicit a priori information as to what is important. The larger class of unsupervised learning methods consists of maximum likelihood (ML) density estimation methods. All of these are based on building parameterized models P[x; G] (with parameters G) of the probability distribution PI[x], where the forms of the models (and possibly prior distributions over the parameters G) are constrained by a priori information in the form of the representative goals

1. Lexicon based techniques

In the unsupervised technique, classification is done by comparing a given text against sentiment lexicons whose sentiment values are determined before their use. The sentiment lexicon contains lists of words and expressions used to express people's subjective feelings and opinions. For example, start with positive and negative word lexicons and analyze which sentiment they need to find. If the document has more positive word lexicons, it is positive; otherwise, it is negative. Sentiment analysis's lexicon-based techniques are unsupervised because they do not require prior training to classify the data. Lexicon based techniques use the following approach;

a) Dictionary-based approach

Kim (2004) Presented the main strategy of the dictionary-based approach. A small set of opinion words is collected manually with known orientations. Then, this set is grown by searching in the well-known corpora WordNet or thesaurus for their synonyms and antonyms. The newly found words are added to the seed list, then the next iteration starts. The iterative process stops when no new words are found. After the process is completed, the manual inspection can be carried out to remove or correct errors. The dictionary-based approach has a major disadvantage: the inability to find opinion words with domain and context-specific orientations. (Qiu, 2012), he used a dictionary-based approach to identify sentiment sentences in contextual advertising. They proposed an advertising strategy to improve ad relevance and user experience. (Qiu, 2012), used syntactic parsing and sentiment dictionary and proposed a rule-based approach to tackling topic word extraction and consumers' attitude identification in advertising keyword extraction. They worked on web forums from automotvieforums.com. Their results demonstrated the effectiveness of the proposed approach to advertising keyword extraction and ad selection.

b) Corpus-based approach

The Corpus-based approach helps to solve the problem of finding opinion words with context-specific orientations. Its methods depend on syntactic patterns or patterns that occur together and a seed list of opinion words to find other opinion words in a large corpus. Using the corpus-based approach alone is not as effective as the dictionary-based approach because it is hard to prepare a huge corpus to cover all English words. Still, this approach has a major advantage in finding the domain and context-specific opinion words and their orientations using a domain corpus.

The basic steps of the lexicon-based techniques are as follow;

1. Preprocess each text (i.e., remove HTML tags, noisy characters).
2. Initialize the total text sentiment score: $s \longleftarrow 0$.
3. Tokenize text. For each token, check if it is present in a sentiment dictionary.
    (a) If the token is present in the dictionary,
4. If the token is positive, then $s \longleftarrow s + w$.
5. If a token is negative, then $s \longleftarrow s - w$.
6. Look at total text sentiment score s,
7. If $s >$ threshold, then classify the text as positive.
8. If $s <$ threshold, then classify the text as

2.7 Sentiment classification

Sentiment Classification can be classified into three categories, namely supervised learning, unsupervised learning, and semi-supervised learning. (Pang, 2002) Suggests different supervised machine learning approaches (Naïve Bayes, Maximum entropy classification, and Support Vector Machine) for sentiment classification. (Collie, 2004), introduced a supervised sentiment classification technique that assigned values to selected phrases and words and used the technique for bringing them together to create a model for the classification of texts. (Gamallo, 2014), proposed a supervised method for sentiment analysis of tweets written in the English language. The system used a Naïve Bayes classifier for sentiment classification. In this work, sentiment analysis is performed using Twitter data. Naïve Bayes classifier has low computational overhead and high performance (Gamallo, 2014). The proposed work uses a Naïve Bayes classifier for sentiment classification.

2.7.1 Evolution of sentiment analysis

1) Text interpretation

Although the studies of sentiment analysis and opinion mining have thrived with the development of the Internet and Web 2.0, the previous works or projects before also laid the foundation for the current research activity. At the early stage of opinion mining extraction, the studies focused on text interpretation in a few areas, such as psychology and politics. Text interpretation is defined as analyzing various texts using simple computerized techniques with the need for human interaction. Simple language approaches were used to understand the subjectivity, point of view in the narrative, and metaphor interpretation (Anbananthen, 2013).

Carbonell (1979) Points out that "modeling human understanding of natural language requires a model of the processes underlying human thought. No two people think exactly alike; different people subscribe to different beliefs and are motivated by different goals in their activities." He proposed a theory of subjective understanding to explain the subjectively-motivated human thinking ranging from ideology belief to human discourse and personality traits. Then he examined people's subjective beliefs in comprehension through a process-model called "POLITICS" in the domain of politics. (Carbonell, 1979). Believes that the domain of politics exposes peoples' subjectivity and political perspectives. He classified them into two categories of opinion based on their conversations: either conservative or liberal. He claims that: "each ideology produces a different interpretation of the input event."

The literary theorist (Banfield, 1982) has been instrumental in proposing subjective and objective sentences as indicators and searching the text by providing simple queries and using the psychology elements as essential factors for natural language processing. According to Benfield's theory, the sentences of narration have been categorized into subjective and objective sentences. Subjective sentences refer to those that portray a character's thought or consciousness (represented thought) or present a scene as a character perceives it (represented perception), including the character's emotions, judgments, beliefs, attitudes, and effects. Objective sentences only narrate the event objectively and directly, rather than through the thoughts or perceptions (ibid). Benfield's theory has been extensively used in opinion mining's early history (Anbananthen, 2013).

2) Text annotation for information extraction

As seen from (Carbonell's 1979) thesis to (Hearst's 1992) study, although some researchers have developed different text interpretation models, they require major human interaction and insertion of search queries, thus there remained the difficulty of extracting information from the unstructured corpus and conducting classification. To solve this problem, text annotation was produced to replace the traditional text interpretation technique. Text annotation is the act of tagging and labeling the corpus by adding new notes to the text. The purpose of the annotation is to produce new attributes and rebuild the corpus; thus, further classification could be processed (Anbananthen, 2013). Text annotation research has taken place at several institutions, such as Xero research center and Grenoble, and computerized text annotation tools such as footnote and endnote have been used in Microsoft Word and OpenOffice (Shabajee, 2012).

Wiebe (2004) concludes that an information extraction system shall distinguish between factual and non-factual information, while question-answering systems should distinguish between factual and speculative answers. Much of the early studies on information extraction mainly focused on the fact-based question answering (Rebecca, 1999) tried to extend the field of the studies in another direction by presenting a probabilistic classifier developed on the resulting annotations, which aims to 'distinguish sentences used to objectively present factual information from sentences used to present opinions and evaluations.' This approach could be applied in news reporting to distinguish between facts andopinions presented in an article or news. This research took a large step toward developing a reliable gold standard of annotation.

However, the text annotation problem requires extensive effort and time to annotate and define the attributes manually to build the corresponding corpus from unstructured text. Therefore, advanced unsupervised learning techniques were proposed for opinion-oriented information extraction to work on different opinion classification levels to improve classification accuracy (Anbananthen, 2013).

One of the important approaches is to extract keywords from the unstructured text using subjectivity or objectivity as indicators. Each word in the text would be individually identified as one of the parts of speech called 'part-of-speech tagging' (POS tagging or POST). Words are traditionally grouped into equivalent classes in grammars called parts of speech (POS) such as noun, verb, adjective, preposition, adverb, conjunction, etc. Thus it can provide a significant amount of information about the word and its neighbors (Jurafsky, 2008).

Dave K (2003) has developed a method for automatically distinguishing between positive and negative reviews by training a classifier using a corpus of self-tagged reviews from websites. (Yi, 2003), presented a sentiment analyzer (SA) that extracts opinions about a subject from the online text by detecting all the online references to the given subject. The sentiment analyzer is considered one of the first attempts on sentiment analysis, which used word sense disambiguation and bag of words (Anbananthen, 2013).

3) Web text mining

Web pages' amount and complexity have been increasing explosively, as has the Web pages' information. In today’s world, firms’ Web data must be analyzed to gain a competitive advantage in the topic sector. Web text mining (TM) is gaining a lot of importance because it is increasingly used in business applications to understand and predict valuable information. It plays a key role in organizing huge web unstructured (textual) data and condensing it into valuable knowledge. Web mining's main objective is to provide data mining algorithms that can improve the content, structure, usage, performance, and categorization of web documents, snippets, and user sessions. Web data mining can be classified into Web Structure Mining, Web Content Mining, and Web usage mining. All these three categories focus on discovering unknown data and potentially very useful information from the web. Though each of them focuses on the same attribute, each may be using it with different mining objectives.

Web Structure Mining involves mining the structure of web documents and links. Useful insights can be given by mining structural information on the web. WSM is very useful in generating visible web documents, luminous web documents, and luminous paths. A path common to most of the results returned, linkage information was useful to improve search engine results, hyperlink structure analysis, link analysis, graph, categorization, and mining the document structure.

Web Content Mining examines the contents of web pages as well as the results of web searches. Web Content Mining (WCM) is described as the automatic search of information resources available on-line. It represents structured, unstructured, semi-structured documents and builds a model for interactive retrieval view and DataBase View. It is all about extracting and integrating useful data with information and knowledge discovery from Web page contents. Web Usage Mining focuses on several techniques that could help learn or predict user behavior and navigation pattern of users using the web around the clock. It includes the data from server access logs, user registration or profiles, user sessions or transactions, etc. It also depends on the collaboration of the user to allow access to the weblog records. Several new techniques and applications were introduced for sentiment analysis in the arena of Web 2.0.

Liu (2005) proposed an analysis system called Opinion observer to identify positive and negative product features from online reviews and compare consumer opinions of competing products.

2.7.2 Framework for Twitter Sentiment Analysis on Hate Speech.

1. **Research gaps**

In pursuance of automatically mining the opinions, two divergent approaches for sentiment analysis have been reviewed in detail. The approach of machine learning has achieved reasonable accuracy, which, to a considerable degree, relies on the quality and size of the training data (Agarwal, 2016). This indicates that the machine learning approach is indubitable domain-dependent, which is a decisive factor that leads to a better performance of sentiment classification in the corresponding domain. However, it is difficult to catch the sentiment strength by implementing the machine learning approach because the output mechanism is usually binary (‘*positive*’ and ‘*negative*’) that counts on the training data. The problem raised here is most of the real-life data contains varying degrees of sentiment.

Another limitation of the machine learning approach is that it is more appropriate for document-level analysis, where there are more textual features to make predictions. Due to the nature of the machine learning techniques, it is not suitable for fine-grained sentiment analysis, such as clause level or aspect level analysis (Khoo, 2015). Thereby, this research follows the semantic orientation approach to explore a better way for fine-grained sentiment analysis.

Over the last decade, the semantic orientation approach for sentiment analysis has been focused on and explored either in academia or in the business industry. Still, it seems it runs into a bottleneck in the matter of (Khan, ‘SentiMI: Introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection.’, 2016). A few research gaps are identified in this research. First, the domain-dependent sentiment lexicon is limited. The general-purpose sentiment dictionary has been widely applied in most studies, whether it is an automatic generated lexicon, such as *SentWordNet*, or a manually built dictionary, such as General Inquirer and Hu & Liu Lexicon (Cernian, 2015). The general-purpose sentiment dictionary problem is that the polarities of words and expressions depend on the context of use.

Due to the tongue's characteristics, prior research indicates that domain-dependent methods can improve accuracy (Liu, 2015). However, it's hard to automatically get domain-dependent sentiment lexicons because that approach relies on seed words with obvious sentiment orientations. Thus, the domain-dependent lexicon requires manual work for choosing and annotating sentiment words. Thanks to the value in terms of your time and energy, a reliable domain-dependent sentiment lexicon is prohibiting. Additionally, the prevailing sentiment dictionaries mainly contain adjectives, adverbs, and verbs. The number of nouns and phrases that indicate sentiments are rare,

Finally, the present hate speech algorithms and model, while incorporated singularly, have failed significantly in detecting hate tweets propagated by various Twitter users in Kenya using various dialects and corded text to a given group because their meaning faces geographical and linguistic challenges and therefore, the classifiers inbuilt rules weren't comprehensive and faced limited functionality of corpus.

2. Framework for Twitter Sentiment Analysis

The rise of machine learning techniques in natural language processing has led to increased research sentiment analysis. In machine learning, a textual feature representation has been utilized coupled with several algorithms such as Naïve Bayes, Support Vector Machines (SVM), Maximum Entropy (ME), which are commonly used to build the classifiers for sentiment analysis. These classifiers built based on different algorithms can learn the rules or decision criterion of sentiment classification based on training data. They are used to conduct sentiment analysis (Ghiassi, 2013) automatically.

This indicates that the machine learning approach for sentiment analysis is a kind of supervised learning paradigm. Many labeled training data are required to train the classifier before it is used for classifying the new data (Waila, 2012). A detailed discussion on how these machine-learning algorithms work is beyond the scope of this work. However, the logic behind the machine learning approach for sentiment analysis is straightforward. It is based on supervised classification (shown in Figure 2.3) and composed of two stages: 1) learning the model from a corpus of labeled training data via algorithms; 2) classifying the new data based on the trained model. In general, the whole process of the classification task involves several sub-tasks, such as data preprocessing, feature selection, representation, classification, and post-processing (Khairnar, 2013).

In this computational framework, various steps are data preprocessing, feature extraction, classification based on sentiments, and positively or negatively labeling a tweet. The data set is separated into training and testing sets. In the first phase, various n-gram features such as unigrams, bigrams, and trigrams are extracted from the data. The test instances are created based on these features. A classification model is built from training data to be tested on unknown test data in the second phase. The training is numerical feature vectors representing the term frequency of every selected feature. The sentiment label attached with it and test instances are numerical feature vectors representing every selected feature's term frequency. Labeled training data are used to train a machine learner, as it allows the evaluation of classification. The supervised text classification consists of two phases: Training and Predicting. For the training phase, a corpus of annotated documents is needed. A document representation process extracts the features. These features will be presented to a supervised classification method to learn a classification model that will predict new documents' labels.

For the predicting phase, a new document is presented to the same document representation process to extract the features that represent the document to use the classification model to predict the document's label. Figure 2.3 shows these two phases;

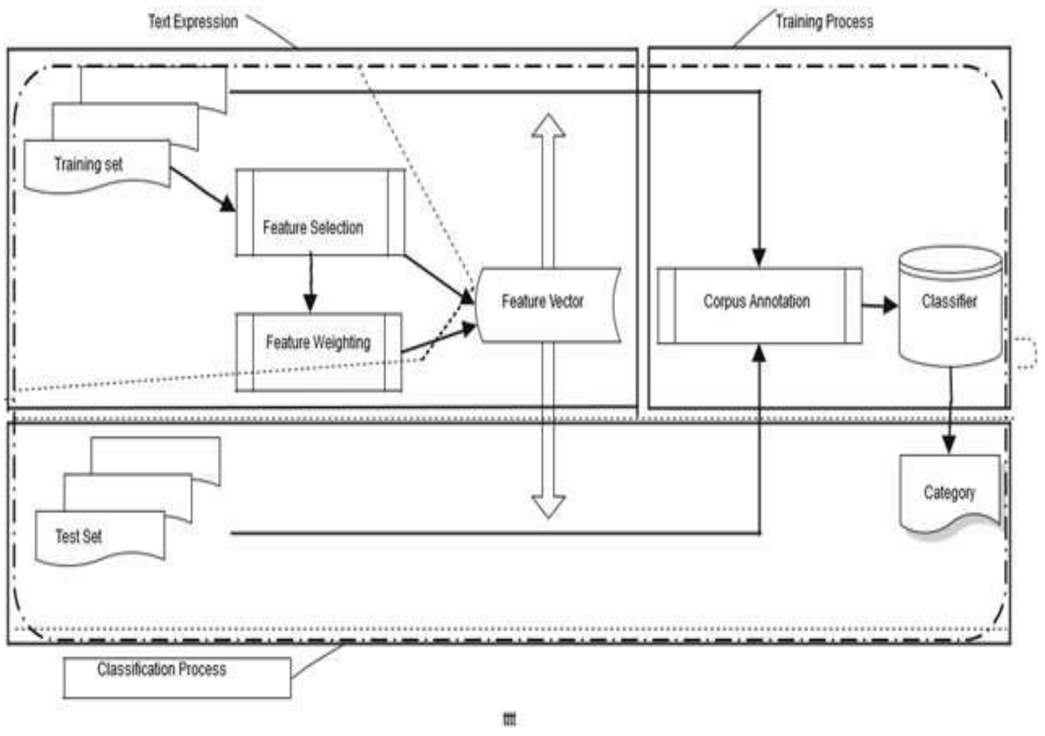


Figure 2.3: Framework for Training and predicting phases of a text classification system

Based on the literature reviewed and the various gaps identified, this work proposes the following conceptual framework to detect Twitter's hate speech. Hate speech relevant data was being collected from Twitter and used to create the corpus necessary for learning. The tweets were annotated as hate speech or non-hate speech and then went through several steps in the pre-processing phase, including removing punctuation marks, removing stop words, and converting to lower case. The tweets were represented in a document-term matrix, using unigram terms with TF-IDF feature weighting to work well in classification problems. For its suitability in text classification problems, the Naïve Bayes algorithm was applied to learn a model for detecting hate speech from the training set. The model's performance was evaluated based on the metrics: accuracy, precision, recall, and the F-Score. Once the model has reached an acceptable level of performance, it can detect new hate speech instances in other tweets. A user will specify keywords to retrieve unobserved tweets from the Twitter Search Application Programming Interface (API).

As shown in the research, gaps in sentiment analysis have been identified based on prior researches. To fill the research gaps and provide an effective approach to sentiment analysis, an innovative framework has been proposed in this research (see Figure 2.4). The framework provides fine-grained sentiment analysis on customer reviews at different levels. It can analyze not only single-opinionated text but also mixed-opinion text. The framework provides a novel way to detect phrases or multi-word expressions in the text via a sentiment lexicon to gain more context. Furthermore, the framework offers an effective approach to conduct aspect-level or feature-based sentiment analysis.

This thesis introduces a classification model based on supervised machine learning techniques and word-based N-gram analysis to classify Twitter messages automatically. The research investigated feature selection based on TF-IDF and different word N-gram ranges. The best performance is achieved using both unigrams and bigrams, Naïve Bayes as a classifier, and TF-IDF as a feature extraction technique. The obtained results indicate that word N-gram features are more relevant for credibility prediction than content and source-based features, compared with character N-gram features. We also incorporate n-gram representation into our classification approach, based on the assumption that n-grams can capture more local context information in text, thus enhancing topic similarity analysis. Unlike most studies that only consider the presence or frequency count of n-grams in their applications, we use TF-IDF weighted n-grams in building the content classification models. Feature selection is usually integrated as an important part of treating the corpus training data in the machine learning approach (Kummer, 2012). Feature selection is an important sentiment analysis field because its efficiency will determine the sentiment analysis accuracy. Several attempts have been made for feature selection using different approaches; however, the approaches can broadly be grouped.

Statistical approaches are fully automatic techniques, while Lexicon based approaches need human intervention. First, the training data are labeled as positive, negative, or neutral, and then a set of features is extracted from the labeled training data. The collection of features can then be encoded using simple value types, such as Booleans, numbers, and strings. Adding individual words to the feature vector is usually referred to as the unigrams approach (Pang, 2002). The feature selection's main purpose is to decrease the feature space's dimensionality and make computational processing easier.

The classifier based on the Naïve Bayes algorithm is a simple probabilistic classifier, which "applies to a certain class of problems, namely those that can be phrased as associating an object with a discrete category." Thus, the Naïve Bayes algorithm could perform poorly when the features are co-related to each other; as (Dhande 2014) points out, the major limitation of the Naïve Bayes classifier is "real-world data may not always satisfy the independence assumption among attributes." Thereby, (Dhande 2014) proposed combing the Neural Network algorithm with Naïve Bayes for sentiment analysis. The Neural Network algorithm can handle the correlation and dependence thus. They proposed a Naïve Bayes Neural classifier that combines Naïve Bayes with Neural Network algorithm to improve sentiment classification accuracy in real-world datasets.

On the other hand, the machine learning approach is considered a supervised learning technique, subject to data training. The text features and selected algorithms determine the polarity of a given text. Both approaches have their strengths and weaknesses. The semantic orientation approaches are generally efficient and require little training, but the performances are generally lower than those of machine learning techniques are.

However, the machine learning approach is very time-consuming because the model built by machine learning techniques depends on the quality and size of training data (Zhou, 2008). Different combinations of feature selections applied in the machine learning approach could lead to various performances. For example, whether the N-gram feature can increase sentiment classification accuracy has been heavily debated by different researchers. (Pang, 2002), have tested a number of feature types, such as unigrams and bigrams (two words), with or without POS tags.

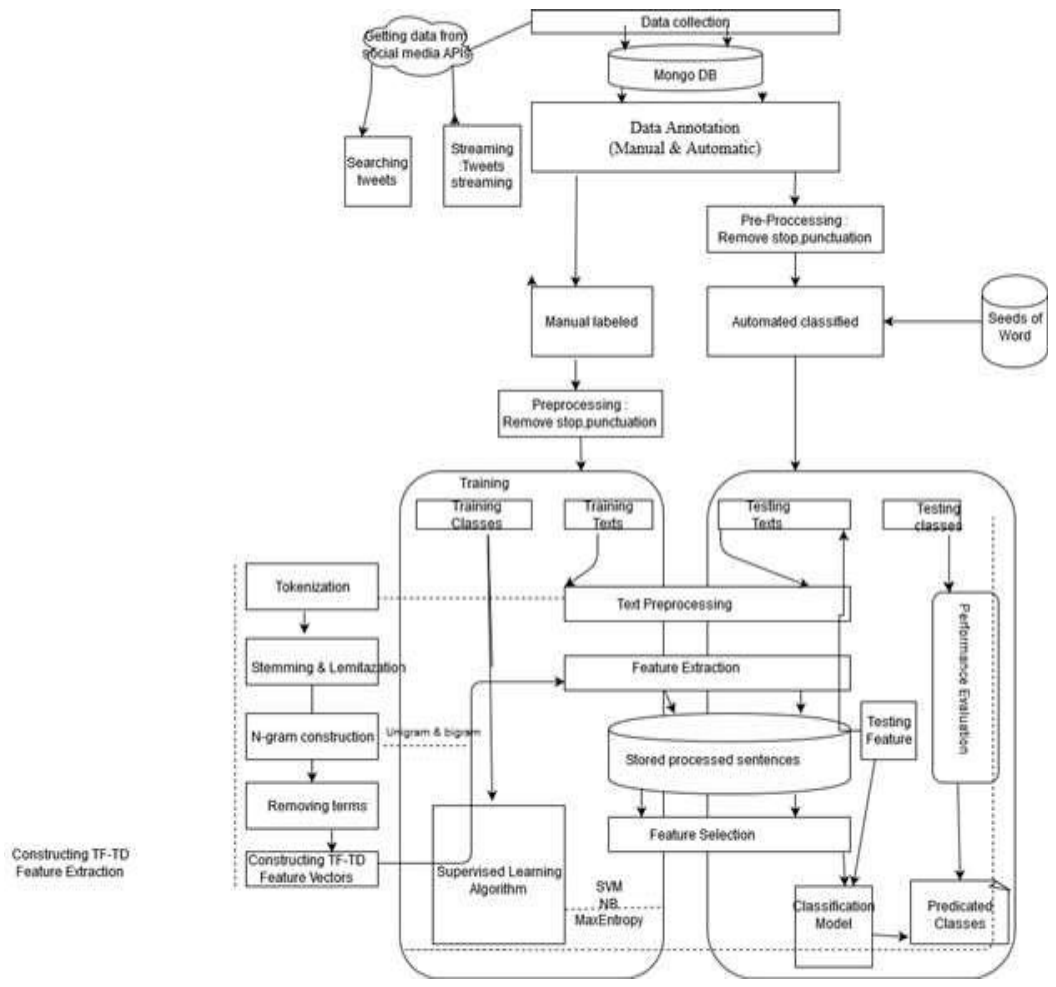


Figure 2.4: Proposed Framework for sentiment analysis

3. Data Preprocessing and Feature Reduction.

N-Grams are an essential method for text categorization. It is also a statistical-based approach for classifying text. The N is that the number of keywords used for dividing the input text. Supported the number of keywords used, the N-grams are called 2-grams, 3-grams, etc. It's able to classify unknown text with the highest certainty (Yadav, 2015).

An n-gram is defined either as a textual sequence of length n, or similarly, as a sequence of n adjacent `textual units', in both cases extracted from a selected document. A `textual unit' is identified at a byte, character, or word level, reckoning on the context of interest.

The only n-gram is that the so-called unigram, where n = 1, which falls back to the only minded \bag-of- words" (BOW) representation. Typically, n may be a fixed number, highly captivated with the actual corpus of documents and the queries made against that corpus. In n-grams, initially, the final pre-processing is applied. Then the document is split into N-grams or N-shingles. This refers to a sequence of consecutive words of size `N,' where `N' is user-specified. Both suspicious and source documents are converted to their N-gram profiles, and similarity is calculated using Dice's coefficient. N-grams are a part of large strings generated from a larger text. During this approach, the input text is split into substrings with n maximum length, and so the frequency of occurrence is counted. After the count of all n-grams, the smallest amount is discarded, and the rest is written to a profile.

The profile of unknown text is then compared to every language (category) listed. This approach helps small text be categorized like comments on social network sites, conversations in a very forum, etc. Each sentence is split into small parts. First, the dataset of the language that must be classified is ready. After that, the input text is split into small tokens until it matches with the language dataset. It is used with the Bag-of-words model for better lead to document classification. The result gives good performance for the statistical approach, but semantic similarity doesn't yield an expected output (Yadav, 2015).

Using N-gram frequency profiles provides straightforward and reliable thanks to categorizing documents during a wide selection of classification tasks. The key benefit that N-gram-based matching provides derives from its very nature.

4. N-grams Approach for Language Identification

In this approach, the classification was done by building n-gram language models. N-gram of the token was used for identifying language. N-gram sequence of n items from a given sample of text or speech. A gram may be a token or lexicon taken into consideration for training, and a classification-gram represents a collection of such chosen lexicons.

Generally, during this approach frequency of n-grams is used. In traditional information retrieval and topic-oriented classification, the frequency of n-grams gives better results. The frequency is converted to TF-IDF to require the term's importance within the document to be classified. The N-grams approach for language identification the research was achieved through NLTK (Natural language toolkit) Python. Good language Identification algorithm decides the success of sentiment analysis tasks. Linguistic communication processing and machine learning techniques require data that is annotated with its language.

Natural language processing algorithms must be modified according to the grammar of the language. Natural language tool kit (NLTK) in Python is the most popular natural language processing package for English. Language identification is very useful for various text processing, text-mining tasks such as Named entity recognition, Parts of speech tagging, machine translation, and multilingual sentiment analysis. The proposed Language Identification process every token by passing it through all available language models. Language dictionaries are used to classify the tokens for different languages

5. Twitter Data Extraction

The goal was to provide an easy-to-use Twitter data extraction and analysis tool for research purposes. To this end, the platform is modular, with several independent modules implemented as different programs. Adding new modules or customizing them is straightforward, making it easy to adapt the platform to the researcher's needs. The central piece in this architecture is a database that keeps the tweets mined by the application and makes them available for further processing. There are several processing layers, and these modules need to interchange data among them, using open data formats such as JSON. By default, the framework includes modules for generation report and sentiment analysis. The platform is divided into the following components;

1. **Data collection:** Tweets are mined and collected from Twitter by use of a Twitter API.

2. **Preprocessing and Data Preparation**: The core of the platform. It listens indefinitely to a filtered Twitter stream and stores the whole status update into the database. In this step, documents are collected, cleaned, and properly organized, the terms (features) are identified, and a vector space representation is created. In this step, data may be divided into two subsets:

    - Training Set. This part of the data is used to create the model. In some cases, this is then split into two: the actual model construction subset and a model evaluation subset needed to tune the learner parameters.
    - Test Set. This part of the data is used for testing the model.

3. **Creating a Bag-of-Word Corpus:** The individual words inform of text are taken into account, and a Bag-of-word corpus is created. Bag-of-word is a feature vector representation where each dimension of text corresponds to a feature. The assumption is that all features are independent, given the class labels. In this model, texts are represented as a bag of words, disregarding grammar, semantics, context, and even word order but keeping multiplicity. The occurrence of each word is used as a feature for classifier training. The Bag-of-word model learns a vocabulary from all of the documents and then models each document by counting each word's number of times by implementing a Document Term Matrix (DTM) process. For further Data preparation procedures, the following activities;

    i. **Removal of Stop words -** This is where commonly used words such as ‘me,’ ‘a’, ‘the,’ ‘who,’ ‘them,’ ‘shall,’ ‘has,’ ‘have,’ among others, are not meant for the analysis are removed. These words are referred to as Stop words.

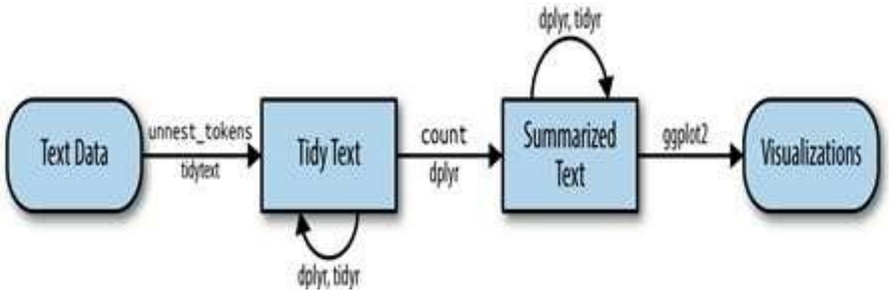


Figure 2.5: A flowchart of a typical text analysis that uses tidy data principles

ii. **Stemming -** Stemming is that the process of reducing a word to its word stem (parent word) that affixes to suffixes and prefixes or the roots of words referred to as a lemma. Stemming is a component of knowledge extraction (feature extraction), a process of linguistic normalization, during which the variant varieties of a word are reduced to a typical forum

4. **Create an n-gram tokenizer function**

N-grams are sequences of n consecutive words from a given text. In linguistic communication processing, tokenization is that the process of breaking human-readable text into machine-readable components. The foremost obvious thanks to tokenizing a text is to separate the text into words. The function allows us to specify unigram terms within the Term-Document Matrix. Additionally, the speed of Sparsity is determined. This is often done by determining the smallest number of times a term appears in an exceedingly document or entry.

5. **Classifier**

A Web interface that permits collaborators to assist with the supervised classification of tweets. The collaborators should decide the tweet's sentiment and classify it accordingly. The collaborators must decide the tweet's sentiment and classify it accordingly. It builds the model; this is often the particular learning (also called training) step, which utilizes the training algorithm. It's usually an iterative and interactive process that will include other steps and should be repeated several times so that the simplest model is created:

6. Trainer

Simplified interface to the NLTK library, which helps build corpora and models out of tweet collections.

7. Evaluation

Set of tools that help assess the trained model's suitability, including tools for manual classification of tweets via CLI and web interfaces, classification of a Twitter stream in real-time.

8. Reports

The report generation module assists by aggregating statistics from the database, both quantitative and sentiment variables. To decouple the statistical analysis from the backend, an intermediate module generates a set of CSV files from the database. This module can handle quantitative reports, provide some basic statistics, and sentiment reports resulting from the sentiment classification. The chosen back-end database is MongoDB, a good fit for our purposes since its atomic representation is JSON, just like tweets. MongoDB is known for its fast write throughput and especially for fast document access. Besides the obvious advantages, it eases congestion in Twitter's stream reading since the blocking time writing is very Low. Most framework tools are implemented in Python, but the Classifier and Tester web interfaces run on Scikit-learn. MongoDB plays a central role as a persistent data store with which modules can communicate. Reports are generated in CSV format and stored in Jason format. Open formats are important, especially in research, because they allow seamless integration with external tools and easy file format conversion, and future-proof compatibility. Trained classifiers are stored as Python pickles (Python's serialization format).

A complete procedure of data extraction and sentiment analysis is divided into three separate steps: data acquisition, training for sentiment analysis, and report generation. The first step is gathering data from Twitter with the Miner. Then the classifier is trained, and the sentiment analysis is carried out. Finally, the platform generates a set of reports. Figure 2.6 illustrates the procedure. During the data acquisition phase, tweets are captured from Twitter's public streams under some (optional) filtering conditions. Twitter's filter stream allows filtering for several parameters: track (a set of phrases), follow (a set of Twitter user IDs), and locations.

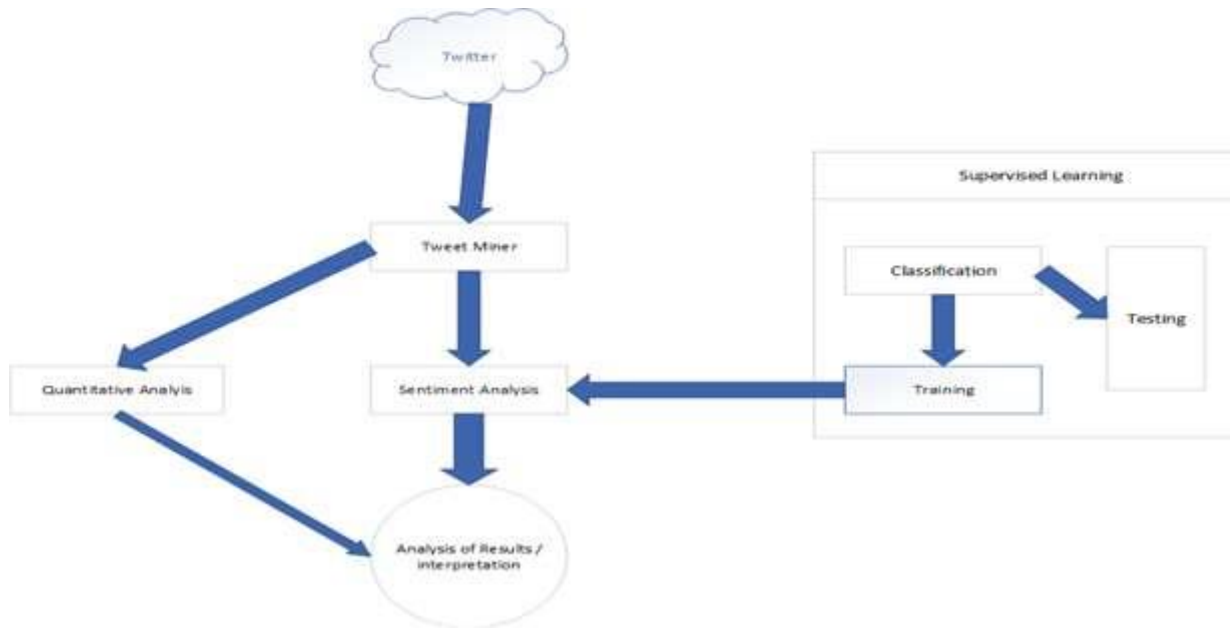


Figure 2.6: Platform's general usage flowchart, including sentiment analysis.

The training phase is separated into two branches: manual classification carried out by humans; and training, which builds a statistical model (using these manually classified tweets) to measure the membership probability of new tweets. Since massive amounts of data are required for proper classifiers training, we needed a large pre-classified dataset. Our approach to overcome the problem is swarm-like: a simple web interface where several collaborators can help with classification. Tweets are presented one by one to collaborators, which then must decide, based on their tacit knowledge of the language, the sentiment expressed in the message. Finally, the report generation phase is divided into quantitative analysis and sentiment analysis. The first one is a traditional statistical analysis involving quantitative variables like the tweet's length, frequency, or mentions. The latter (only possible after the statistical model is trained) involves an automated analysis of the message sentiment to be analyzed later.

9. Laplace Smoothing (add-1)

According to (A. Y. Liu and C. E. Martin, 2011), Naïve Bayes Classifier includes a problem with maximum likelihood training. an example, the matter of 'unknown word' particularly where if a feature (or word) doesn't occur in any document within the training set, all documents within the test set that contain this same feature are going to be zero for all classes 'c,' causing Multinomial Naïve Bayes to lose all discriminative power. Additionally, rarely occurring features can also be problematic if smoothing is not performed. As an example, a rare feature which will occur in some classes within the training set but doesn't occur within the test set will dominate probability estimates since it'll force P(wi|c) to be zero, no matter the values of the remaining word features.

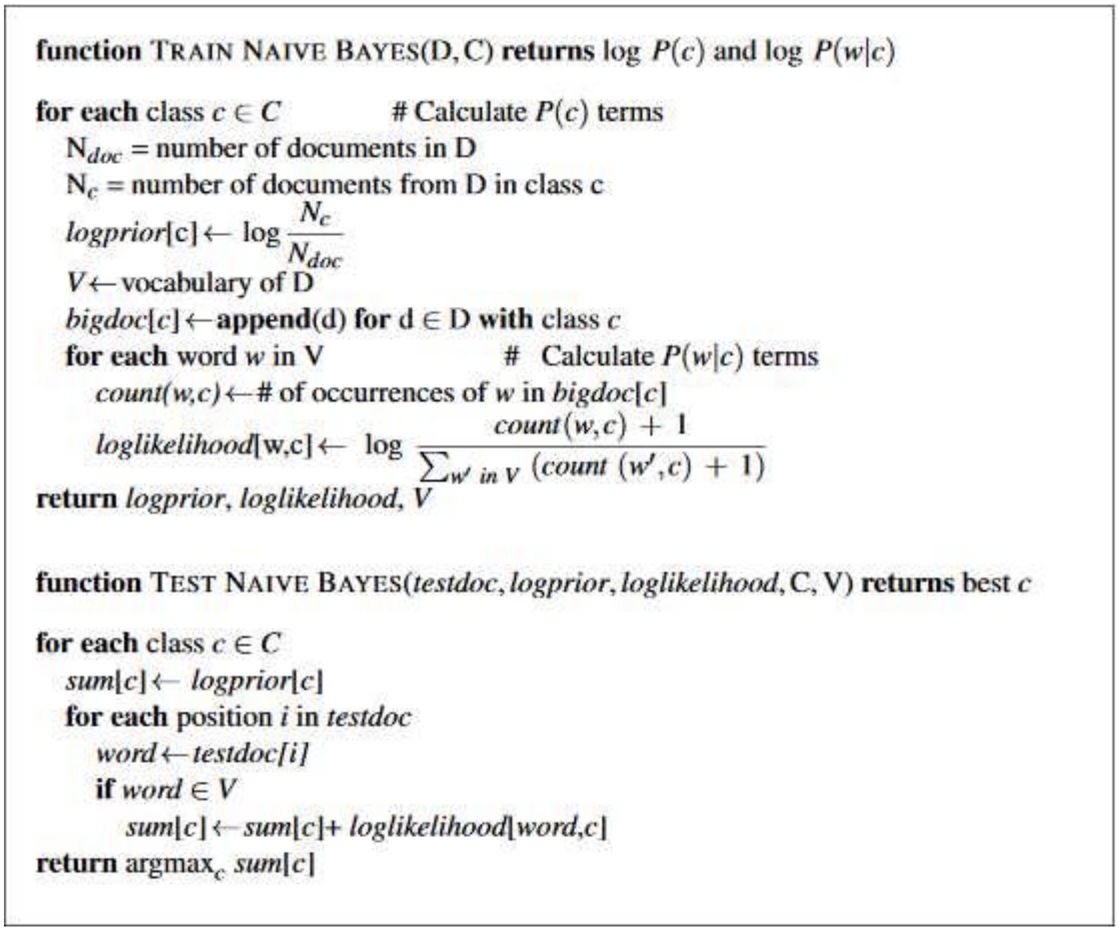
For instance, when trying to estimate the likelihood of the word "great" given the class a positive, there could also be no training documents containing the word "great" and are classified as positive. The word "great" may have occurred sarcastically in the class negatively. This word feature's probability will be zero in such a scenario, as shown in figure 2.7.

$$\mathrm{P(great \mid positive)} = \frac{\mathrm{count(great \mid positive)}}{\Sigma_{w \in V}\mathrm{count(w,positive)}} = 0$$

Naïve Bayes naively multiplies all the feature likelihoods together, and zero probability in the likelihood word for any class will cause the class to be zero, despite the evidence.

The solution to these limitations is parameter smoothing. For this study, we apply Laplace smoothing (add-1 smoothing) to prevent cases where missing, unknown, or rarely occurring features inappropriately dominate the probability estimates in Multinomial

Naïve Bayes. This is commonly used in Naïve Bayes text categorization:



```
function TRAIN NAIVE BAYES(D, C) returns log $P(c)$ and log $P(w|c)$

for each class $c \in C$          # Calculate $P(c)$ terms
   $N_{doc}$ = number of documents in D
   $N_c$ = number of documents from D in class c
   $logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}$
   $V \leftarrow$ vocabulary of D
   $bigdoc[c] \leftarrow$ append(d) for d $\in$ D with class $c$
   for each word $w$ in V                  # Calculate $P(w|c)$ terms
      $count(w,c) \leftarrow$ # of occurrences of $w$ in $bigdoc[c]$
      $loglikelihood[w,c] \leftarrow \log \frac{count(w,c) + 1}{\sum_{w' \text{ in } V} (count(w',c) + 1)}$
return $logprior$, $loglikelihood$, $V$

function TEST NAIVE BAYES($testdoc$, $logprior$, $loglikelihood$, C, V) returns best $c$

for each class $c \in C$
   $sum[c] \leftarrow logprior[c]$
   for each position $i$ in $testdoc$
      $word \leftarrow testdoc[i]$
      if $word \in V$
         $sum[c] \leftarrow sum[c] + loglikelihood[word,c]$
return $\text{argmax}_c\ sum[c]$
```

Figure 2.7: The Naïve Bayes algorithm, using Laplace Smoothing

10. Training the Multinomial Naïve Bayes Classifier.

To learn the possibilities of P(c) and P P(fi|c), that is, the prior probability of a given class ‘c’ and therefore the probability likelihood of a given feature fi given a category ‘c’; we consider the most likely estimate by using frequencies within the data. For P(c), we identify what percentage of documents within the training set are in each class ‘c.’ Let Nc

be the number of documents in our training data with class ‘c,’ and Ndoc be the entire number of documents. A textual feature is chosen during this phase because the measurement criterion then some numeric values extracted from documents with known classes are assigned to the textual feature (these documents are called training set). This phase is brought up training phase. After receiving the feature vectors of some documents with unknown classes (test set), at the testing phase, a classification algorithm predicts the foremost likely class of them by comparing the test set's feature vectors with those of documents in the training set.
This description will be formulated in several ways. It may take the shape of a classifier, neural network, decision tree, production rules, mathematical equations, etc. The information modeling methods were developed within the kind of mathematical equations within the area of statistics. Typically, a learning system is supplied with a collection of coaching data classified by hand. The system then attempts to be told from these training data a way to classify the identical data and also a way to classify new data that it's not seen. Concept learning involves determining a mapping from a group of input variables to a Boolean value. Since we don't have direct access to pre-labeled Twitter data, we planned to crawl it manually.

11. Data Preprocessing.

Twitter Developer API used to collect tweets with certain settings Twitter API allows developers to pass keywords of interest to limit interest collection. Data was collected from Twitter using Twitter API, as shown in figure 2.6. We created a corpus by downloading tweets in real-time using Twitter'sTwitter's streaming API. To retrieve tweets, a tweeter application was generated to get ConsumerKey, Consumer_Secret, Access_Token, and Access_Token_Secret. These keys are used to connect R Studio and tweeter applications. The collected data was stored in a database in JSON format. Data preprocessing is an important step in sentiment analysis since with the appropriate selection of preprocessing techniques, the classification accuracy can be improved (Haddi, 2013).

We applied both Twitter-specific and standard preprocessing on the info. The particular preprocessing is particularly important for Twitter messages since the Twitter community has created its unique phrases and forms to write down messages. Moreover, user-generated content in social media often contains slang (Petz, 2012) and frequent grammatical and spelling mistakes (Petz G. K., 2013). Therefore, with Twitter-specific text preprocessing, we attempt to handle these properties of the Twitter language and improve the standard of features. Besides the Twitter-specific text preprocessing, we also consider standard text preprocessing techniques (Feldman, 2007) to define and reduce the feature space. These involve applying text tokenization (text splitting into individual words/terms), testing whether stop word removal (removing words which don't contain relevant information, is helpful, performing stemming (converting words into their base or root form), and n-gram construction (concatenating1 to n stemmed words appearing consecutively during a tweet.). We set the worth of for n-gram construction to 2, meaning that we construct unigrams (an n-gram of size 1, which may be a single-stemmed word) and bigrams (an n-gram of size 2, made by concatenating two stemmed words which appear successively during a tweet).

The resulting terms are used in the construction of feature vectors that represent the tweets. The standard approach to feature vector construction is TF-IDF-based, where TF-IDF stands for the "term frequency-inverse document frequency" feature-weighting scheme (Yang Y. &., 1999). In the TF-IDF scheme, a weight reflects how important a word is to a document in a document collection (TF-IDF increases proportionally to the number of times a word is present in the document but decreases the number of documents in which the word occurs). However, in the classification setting, TF (term frequency)-based approach, where a weight reflects how often a word is found in a document, performs better than the TF-IDF-based approach (Martineau, 2009). Moreover, in our study (Smailovic, 2014), we showed that the TF-based approach is statistically significantly better than the TF-IDF approach in the Twitter sentiment classification setting. Therefore, the feature vector construction in our experiments was based on the TF weighting scheme.

Given our feature set of specific words and syntactic features, we aimed to create a set of results and related models that could be used to inform policymakers of the risk of cyberhate spreading online following events that are likely to incur a hateful or antagonistic response toward a specific social group. The classifier was implemented on Pycharm IDE using Python as the programming language. To produce experimental results, we used python Scikit-learn. Machine learning libraries to develop several supervised classifiers that were trained and tested using the features discussed in the previous section. Each tweet was transformed into a feature vector—a list of attributes representing the tweet to train a classifier.

The classification can be done by supervised machine learning and unsupervised machine learning. In this classifier, we have used supervised machine learning techniques for the classification. Supervised learning is a very important technique for classification. The language chosen is Python (python 2.7.9.), mainly because of its available libraries. NLTK offers most of the preprocessing activities that are very important in text analytics. Scikit Learn offers support vector machine implementations, Naïve Bayes algorithm, Decision tree algorithm, Logistic regression algorithm Kmeans algorithm, and feature extraction techniques like BoW and Tfidf.

To classify tweets, we build a classifier that consists of several machine-learning classifiers. 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 that provides many classification algorithms. Scikit-learn also includes tools for classification, clustering, regression, and visualization. To install Scikit- learn, we simply use the online command in python, ‘pip install scikit- learn.’ We used the following classification techniques to build our classifier, which come in Scikit- learn library.

1. Naïve-Bayes Classifier
2. Support Vector Machine
3. Maximum Entropy

The reason we are used three classifiers so that we can get a more reliable output. To use these classifiers, we write a script in Python, in which we first import the classifier and then pass the training set to each classifier. We developed a sentiment analysis classifier of Twitter data based on supervised learning techniques. The main components of the classifier were:

- A preprocessing module helps refine the data collection and select the features that properly represent the Twitter data.
- A supervised learning module aims to identify, detect, and analyze hate tweets by various Twitter users targeting a certain group(s) of people.
- Component for Searching tweets published by a certain user in real-time.
- Component for analyzing hate tweets of a given user where twitter features are extracted.
- The user bio information.

12. Evaluation metrics

To judge the implemented system's accuracy and performance, the quality performance metrics were utilized in this research: accuracy, precision, recall, and F-score. Those metrics' employment has also crossed over into acting on evaluating language Processing (NLP) models, which incorporates sentiment analysis (Khan M. T., 2016). The accuracy measures the general correctness of classification, which shows the proportion of the documents that are correctly classified. The precision indicates the exactness of a system, that is, the share of the chosen documents that are the targeted ones. High precision indicates fewer false positives. On the opposite hand, a lower precision means more false positives. The precision is sometimes used with recall. Recall measures the completeness or sensitivity of a system, which is the percentage of the system's target documents. The tradeoff between precision and recall is often plotted since an increase of 1 can often decrease the opposite. For this reason, F-measure (also called F-score) is employed to mix precision and recall into one metric to live the general performance.

2.7.3 Sentiment Analysis

According to (Jebaseeli 2012), Opinion Mining or Sentiment Analysis refers to identifying and classification the perspective or opinion expressed within the text span, using information retrieval and linguistics (Jebaseeli, 2012). Sentiment analysis is that the field of study analyses people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities like products, services, organizations, and their attributes (Liu, 2012). Social media has pervasively played an increasing role. They became a vital alternative information channel to traditional media within the last five years during emergencies and disasters. They rank because the fourth preferred sources to access necessary information during emergencies (Alfredo, 2015). specifically, individuals and communities have used social media for several tasks, from warning others of unsafe areas to fundraising for disaster relief (Alfredo,2015).

Social media involves tongue Processing (NLP), computational linguistic and textual analytics to get the subjective and mawkish information from a given text. The sentiment analysis process entails various tasks or procedures, after which the polarity or the subjectivity of the content is often specified. Sentiment analysis is predominantly implemented in software that may autonomously extract emotions and opinions in text. Its many universe applications allow companies to investigate how their consumers perceive their products or brand. This usage is especially applicable to the present project.

It is difficult to classify sentiment analysis mutually specific field of study as it incorporates many various areas like linguistics, Natural Language Processing (NLP), and Machine Learning(ML) or Artificial Intelligence (AI). Because most of the sentiment uploaded to the net is unstructured, it's a difficult task for computers to process it and extract meaningful information from it. Language processing techniques were accustomed to transform this data into a form that a computer can process efficiently.

2.7.4 Sentiment analysis of Twitter

Twitter is a web microblogging service created in March 2006. It enables users to send and skim text-based posts, referred to as tweets, with a 140-character limit for compatibility with SMS messaging. As of July 2014, Twitter has about 270 million active users generating 500 million tweets per day.10 Tweet messages usually convey sentiment (Paroubek, 2010). However, unlike conventional data, tweets have certain characteristics that analyze sentiment over Twitter data a harder problem than conventional analyzing data.

Tweet messages are too short, encouraging the utilization of abbreviations, irregular expressions, poor grammar, and misspellings. Such characteristics make it hard to spot the opinionated content in tweets using sentiment analysis approaches designed for conventional text (Hassan Saif, 2012). Research work has been conducted to target the above problem, especially within the past four years, specializing in the actual problem of sentiment analysis over Twitter data. Like conventional sentiment analysis, existing Twitter sentiment analysis approaches are divided into machine learning and lexicon-based techniques.

2.7.5 Levels of Sentimental Analysis

Sentiment analysis can be divided into three levels, which are Document Level, Entity or Abstract level, and Sentence level.

1. **Document-level Analysis**

It essentially operates on an opinionated text in the unit of a document. In this document-level classification, a single review of a single topic is considered (Varghese, 2013). This type of analysis is also called document-level-sentimental-analysis. This analysis is used to evaluate positive or negative opinions about a single product. This is only applied to a single entity or product, not applicable to multiple entities.

2. Sentence level Analysis.

Sentence level sentiment analysis evaluates the polarity of every sentence. Sentence-level subjectivity classification is beneficial because most documents contain a combination of subjective and objective sentences (Wiebe, 2005). Just in the case of straightforward sentences, one sentence bears one opinion about an entity. Complex sentences don't seem to be desirable to sentence-level sentiment analysis (Varghese, 2013).

3. Entity level analysis.

Sentence level sentiment analysis evaluates the polarity of every sentence. Sentence-level subjectivity classification is beneficial because most documents contain a mixture of subjective and objective sentences (Wiebe, 2005). Just in the case of easy sentences, one sentence bears one opinion about an entity. Complex sentences aren't desirable to sentence-level sentiment analysis (Varghese, 2013).

2.7.6 Multi-Lingual Sentiment Analysis

Today's sentiment analysis systems must be able to cope with an abundance of sentiment-carrying user-generated content. As different approaches are required for distinct languages (Boiy,2009), existing work doesn't typically specialize in devising one sentiment analysis approach for multiple languages, but rather on analyzing the sentiment conveyed by documents in selected languages, mainly by means of applying sentiment analysis techniques tailored to every specific language. Existing work is primarily focused on devising sentiment analysis methods for other languages with minimal effort, without sacrificing an excessive amount of accuracy, instead of constructing new frameworks for languages aside from the reference language(Abbasi,2008).

Boiy, (2009) recent work uses artificial intelligence techniques to be able to re-use many existing tools when performing automated sentiment analysis on multi-lingual textual content.

Sentiment analysis of machine-translated texts could seem a rather ineffective approach, as MT typically fails to translate substantial amounts of text properly and tends to scale back well-formed texts to sentence fragments. Nevertheless, recent work on sentiment analysis of stories messages in nine languages demonstrates that sentiment classification accuracy is essentially independent of the standard of the machine translator used (i.e., the translator doesn't necessarily must produce well-formed texts) which sentiment analysis of texts that are translated into English is consistent across languages, after normalizing sentiment scores to permit for meaningful cross-cultural comparisons (Bautin,2008).

Other existing work suggests that, in some cases, sentiment analysis of machine-translated texts can yield even better results than sentiment analysis of the initial texts, especially when state-of-the-art tongue processing tools don't easily interpret the first language. As an example, (Wan 2008) uses a Chinese sentiment analysis framework for classifying Chinese reviews' sentiment and an English framework for classifying those Chinese reviews after MT into English. (Wan, 2008) shows that sentiment analysis of the translated texts outperforms sentiment analysis of the initial texts. An ensemble of both methods further improves performance.

Machine translation will be utilized in our way and facilitate automated sentiment analysis in multiple languages. Many researchers concentrate on automatically generating sentiment lexicons using AI instead of performing sentiment analysis on machine-translated texts. A standard approach automatically translates an existing sentiment lexicon (Mihalcea, 2007) and, possibly, subsequently propagates the sentiment to semantically related words (Jijkoun, 2009). An alternate approach, which has proven to outperform artificial intelligence of sentiment lexicons, is to automatically generate a sentiment lexicon from a set of (automatically) translated and annotated texts (Banea, 2008), (Lin, 2012) (Mihalcea, 2007). However, research suggests that the subjectivity of most of the words in sentiment lexicons is lost in translation – subjectivity appears to be a property associated not with words but with word meanings (Mihalcea, 2007). Semantic lexicons are employed to handle this issue.

2.7.7 Challenges in Sentiment Analysis

Over the past years, there has been substantial growth in microblogging services such as Twitter and access to mobile phones worldwide. Thus, there is tremendous interest in sentiment analysis of short informal texts, such as tweets and SMS messages, across various security and health areas, just to mention. This calls for sentiment analysis systems that can process large amounts of data and handle the text genre's special challenges of so-called microblogs. Because of the interest in utilizing this freely available information by research and industry, microblogs' sentiment analysis has become popular in recent years. Sentiment analysis of microblogs such as Twitter has to handle the following difficulties;

1. **Big Data-related Issues**

As information from the information sources is available in many shapes and sizes, the tongue's complexity can make it very difficult to access the knowledge within the opinion text (Pak, 2010). Unlike humans, machines cannot identify relevant information as easily as humans can do. NLP tools are still trying to make a general-purpose representation of meaning from unrestricted text. Big Data-related Issues. The proliferation of web-enabled devices offers new mediums for people to form, communicate, and share content on social web platforms, including blogs, social networks, forums, etc. At the same time, the users of those web communities generate an enormous amount of heterogeneous data. The generated data, or because it is called ‘‘big data,’’ offers an unprecedented opportunity for people or organizations to mine and analytics big data content using advanced technologies and analytics techniques, which enable in providing valuable information for decision-makers. Sentiment analysis is one in every of the clear text analytics techniques that extract social web users’ opinions and classify sentiment polarity that is feasible and applicable in numerous domains. In general, huge data analysis could be challenging thanks to volume, variety, velocity, variability, and veracity of knowledge, which characterize the large data. Sentiment analysis of big data is challenging by the common characteristics of massive data. Following are the common sentiment analysis challenges associated with big data:

2. Data Collection

Data collection is a preliminary step for any sentiment analysis task but is one of the main challenges for researchers. Benchmark data sets are not available free for interested researchers in the sentiment analysis field; most of the available social user data are commercial. Twitter provides APIs for enabling data collection from their sites. Although, due to the volume, variety, velocity of big data, the collection of data set through using APIs is still a challenging task, since the APIs like Twitter API enables the user to retrieve only 100 tweets each time, compared to the volume of data available online regarding the selected user's keyword/target the retrieval of relevant data from a huge volume data using APIs is difficult task and the relevancy of the collected data set is a major issue for researches in sentiment analysis.

3. Data Preprocessing

Data volume restricts the filtering of relevant data from non-relevant data that may compromise the sentiment analysis results. Big data variety and velocity limit the feature extractions critical in preprocessing the sentiment analysis data set.

Extraction of opinion words and sentences, POS tagging challenge when the volume of a dataset is huge.

4. Data Storage and Analytics

Another sentiment analysis issues in big data is the memory size required for the preprocessed dataset for analysis. The abundant size of the data with different format storage is one of the technical issues addressed by some advanced storage techniques. Another challenge is the velocity of big data since sentiment analysis on dynamic and real-time events in the big data world is challenging and needs to be addressed efficiently, considering that people's opinions change over time.

5. Text Length

Microblog posts are typically very short. It poses the challenge that the expressed opinion may well be keen about one word only. The word may not be available within the used lexical resource or won't have occurred within the training data, which might cause the loss of the opinion. A discussion of the phenomenon is often found in (Birmingham 2010). Thanks to spontaneity, the informal context, and length restrictions, the spelling in microblog posts tend to possess much greater variability than in other text genres. As an example (e.g. \b4" - \before").

6. Domain Dependency

Many classifiers trained to classify opinion polarities in a specific domain and perform relatively well may produce miserable results if applied in a different domain. (Wollmer, 2013) studied how to perform domain-independence sentiment analysis for movie reviews. Essentially this is still an unresolved challenge in sentiment analysis. Sentiment analysis is a highly domain-sensitive task in which the sentiment classification is highly dependent on the domain the training data has been extracted from. Using a training dataset from one domain usually performs poorly when testing a test dataset from another domain. The challenge is that the opinion words and constructs used to describe an event on a domain are often different from one domain to another. In addition, the orientation of the opinion word may be revered from one domain to another. Existing research is trying to overcome the domain dependence challenge using domain transfer (Lue, 2012). A small amount of training data is labeled from the new domain called the target domain, where it is used to test the original/source domain training dataset.

7. Slang and informal language

Many generations and groups of people use different informal languages concerning various contexts. In Kenya, the difference in diversity and recognition, so many people, especially the youth, use informal language. These languages differ from region to region. Below are the common challenges for informal language sentiment analysis.

8. Lack of Corpora and Dictionaries Lexicon

Due to the different characteristics of non-English languages, the number of other languages corpora and dictionaries lexicons is limited compared with English language-oriented corpora and dictionaries. It is a difficult task based on each language morphologies, characters but still required. More numbers researches in other languages are needed.

9. Different Writing Style

The writing style is another issue of non-English languages when performing sentiment analysis. In some of these languages, like Luo, language-writing style is from right-to-left, and the same word is written in different styles or formats. This issue is also applicable in other languages and needs to be addressed efficiently.

10. Different Word Meaning

This is the case when the same word has a different meaning in different contexts. This is another important issue in sentiment analysis since it extends the efforts to build language-oriented lexicons and dictionaries. It may compromise translation accuracy when sentiment analysis is performing by translating other languages into the English language.

2.7.8 Approaches on Sentiment Analysis

The following are some of the approaches and algorithms that are currently used in sentiment analysis on Twitter.

1. **Polarity Classification**

The classification here at a simple notation is the process of classifying the opinions in an opinionated piece of text, under two opposite sentiment polarities. The opinion in the text is assumed to be covering a single topic, field, or context. The process may be utilized for summarization processes or simply to extract the overall sentiment, say negative. The opinionated text may address subjective or objective issues. The author may address an issue by stating his/her own opinions or stating a fact or a piece of news that might be considered as polarized within a specific context.

Several related problems to the polarity classification include the Related Categories, which explore why the users had an overall positive or negative sentiment. Identifying the pros and cons in this manner improves the overall polarity and enriches individual reviews' helpfulness, where judgments reinforced by reason are more trustworthy. Other problems include Rating Inference, measuring the "degree of positiveness," and having a multi-point scale for the overall output.

2. **Subjectivity Detection**

Polarity classification operates under the idea that the content of the text being opinionated. However, real-life texts can be objective or objective and subjective content associated with the given context. Subjectivity detection is often thought of as an independent process. However, it had been found that having subjectivity detection as a proceeding process for polarity classification increases the efficiency of the sentiment analysis. Subjectivity Detection could be a harder problem than polarity classification, so improvements at subjectivity detection promises better results for the sentiment analysis process as a full.

Sentiment Analysis Techniques

There exist four main categories to perform sentiment analysis:

1. **Keyword spotting**

It is the most naïve approach. It categorizes text based on the presence of affect words. These words are usually given some sentimental values in a given linguistic annotation scheme. Hence, the presence of the keywords most probably indicated the orientation of sentiment polarity. It is not robust to negation and relies on surface features (Cambria, 2013).

2. **Lexical Affinity**

This approach is much more powerful than keyword spotting. In addition to detecting affect words, it assigns arbitrary words a probable "affinity" to particular emotions (Cambria, 2013). This approach is not robust to negation and sentences with other meanings.

3. **Statistical Methods**

This method utilizes a machine-learning approach. (Pang, 2002), observed machine learning techniques (Naïve Bayes, Support Vector Machines, and Maximum Entropy) outperformed human-produced baselines. Some earlier studies with machine learning algorithms on movie reviews. Feeding a machine-learning algorithm, a large training corpus of affectively annotated texts, the system might learn the affective valence of affect keywords and other arbitrary keywords (Cambria, 2013).

4. **Concept-based techniques**

These methods employ web ontologies or semantic networks as it heavily relies on knowledge bases. Superior to purely syntactical techniques, a concept-based technique can detect multi-word expressions (Cambria, 2013). They can analyze multi-word expressions related to concepts that explicitly convey emotion.

2.8 Feature vectors

In machine learning, a feature vector is an n-dimensional vector of numerical features representing some object (Frank, 2005). Usually, machine learning algorithms require a numerical representation of feature vectors because it facilitates mathematical computation and statistical analysis. The instance is often represented by an n-dimensional feature vector x = (x1,...,xn) Rn, where each dimension is called a feature. The feature vector's length n is known as the feature vector's dimensionality (Goldberg, 2009).

Examples of features are:

- Count of words
- Presence of words
- Presence of punctuation marks
- Count of punctuation marks
- Time-based features like the hour when a post was published

2.8.1 Features/Aspects of Extraction

The feature extraction processes are one of the fundamental steps for data-driven analysis of textual content. Converting a plain text into a features vector enables more comprehensive and rigorous data mining and sentiment analysis procedures.

1. **Twitter Dictionary-**We constructed a dictionary for the abbreviations and the slang words used in Twitter to overcome these terms' ambiguity. This dictionary maps certain Twitter expressions and words by their meaning or their corresponding sentiment class.
2. **Dictionaries and lexicons-** The work categorizes the features as the features commonly used in text mining: dictionaries and lexicons. This approach consists of making a list of words that are searched and counted in the text. In the case of hate speech detection, this has been conducted using content words such as insult and swear words, reaction words, and personal pronouns (Liu, 2015), number of

disrespectful words in the text, with a dictionary that consists of words for the English language including acronyms and abbreviations (Njagi Dennis Gitari, 2015) label specific features which consisted in using frequently used forms of verbal abuse as well as widely used stereotypical words (Nemanja Djuric, 2015), Ortony lexicon was also used for negative affect detection (list of words denoting a negative connotation and can be useful because not every rude comment necessarily contains bad language and can be equally harmful) (Silva, 2016).

3. **Twitter-specific features-**When the use-case is restricted to Twitter; things like URL retweets, hashtags, mentions can be indicators of sentiments. Usually, a Twitter reply (mention) is made to criticize or refute an opinion, so the apparition of '@' will give the tweet more probability of being Subjective. Also, the apparition of links gives a different dimension to the text inside a tweet. Sometimes a user can include a link to an image or a link to a text to complement the content on its opinion.
4. **Stylometry or style marker**- is one of the most popular methods for authorship recognition. Stylometry considers writing style, statistics of different word categories, word length, character lengths in words, use of unusual words and symbols, etc. We implemented a set of stylometric features from state-of-the-artwork. Stylometric information is extracted from the collected dataset, and in the second stage, different classification algorithms are trained to predict authors of the unseen text. Stylometry is an important analysis method. The topic all Twitter authors write about is similar, and the purpose of the messages is the same, mainly to spread propaganda or hate messages. It is reasonable to believe that the style of writing they have might be similar. A common approach to the writer's invariant method is the frequency of function words. A function word is a word that has little lexical meaning or ambiguous meaning and is used to link other parts of speech in a sentence. Function word features are commonly used in text recognition problems. In this work, we focus only on stylometric statistics applied for words. This since a tweet cannot be longer than 140 characters. Also, the frequency of hashtags is analyzed.

2.8.2 Using SentiWordNet for Affect Analysis

Chalothorn (2012), in their study, analyzes an existing technique to answer the effectiveness of SentiWordNet in detecting hate speech and emotions on the net. Montada and Qawem web forums were chosen as both use the Arabic language and possess Islamic ideological content. The method involved collecting hate speech data, model building, and data analysis of collected results. The information collection phase included takes the ripping words from the forums. Five hundred sentences of the ripped data are translated manually with Python artificial language employed for model building. The phase for model building involves splitting sentences as words and reducing the high-frequency text (stop words. To better evaluate a sentence's polarity by utilizing "Sentiwordnet," a lexical resource for sentiment analysis and Lexicon. Using the mixture of a part of speech and the word itself, SentiWord gives it a numeric score between −1 and 1. Lower value refers to more negative sentiment, and better value refers to higher sentiment.

Like a tweet, the text consists of some words; we can take the SentiWord score for every one of these words, so sums them up to induce a numeric score for every tweet. Another issue here is that SentiWord doesn't recognize sentences; it only takes words and their corresponding part of speech as input. The part of speech the word will have will depend completely on the sentence itself. Therefore, the simplest way must be devised to map each word within the sentence to its corresponding part of speech. This was done using Parts of Speech tag extraction. This can also be bundled with the SNLP and is employed to spot the parts of speech a word has within a given sentence. So each tweet must first be analyzed using the POS tagger, which will separate the tweet. The language Toolkit (NLTK) could be a platform for building programs for text analysis. one of the more powerful aspects of the NLTK module is Speech tagging, which we incorporated in our study. Lexicon-enhanced sentiment analysis supported the Rule-based classification scheme as an alternative approach to improving users' tweets' sentiment classification.

2.8.3 Feature Selection Methods

Plenty of feature selection methods are available in literature thanks to information with many variables resulting in a very high dimension. Feature selection methods provide a way of reducing computation time, improving prediction performance, and a far better understanding of the info in machine learning or pattern recognition applications. The target is to introduce variable elimination applied to a good array of machine learning problems. A standard feature reduction approach for text categorization is feature selection (Tang, 2015). Feature selection is finished consistent with some specified set of rules to settle on feature words with a bigger contribution from high dimensional feature subspace apply to the following classification process to boost the text classification system's efficiency and space (Wu & Xu, 2015). The subsequent section describes some feature selection methods;

1. **Chi-square**

Chi-square ($\chi 2$) statistic is a measure of association. In statistics, chi-square Measure is formulated as:

$$\frac{\chi 2 = \sum i \sum j\ (\mathrm{fij} - \hat{}\mathrm{fij})2}{fij}$$

(2.10)

Here, Fijis the observed frequency of the cell in row I and column j, and ˆfijis the expected frequency of that cell. A more detailed discussion of the$\chi 2$statistic can be found. The $\chi 2$statistic measures the degree of dependence between a certain term and a certain category. It measures to what degree a certain term indicates membership or non-membership of a document in a certain category. The$\chi 2$statistic is reformulated and used for the task of document categorization by Yang and Pedersen Ng et al., and Spitters as follows:

$$\chi 2(t, \frac{N \times (AD - CB)2,}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

(2.11)

Here, have a 2×2 contingency table. The first row stands for the number of documents that contain term t, and the second row stands for the number of documents that do not contain term t, the first column stands for the number of documents that belong to category c. The second column stands for the number of documents that do not belong to category C.So, Act is the number of documents that belong to category c and contain term t, Bct is the number of documents that do not belong to category c but contain terms, Cct is the number of documents that belong to category c but do not contain terms,Dctis is the number of documents that do not belong to category c and do not contain term t, and N is the total number of documents in the corpus. Two different measures can be computed based on theχ^2statistic

$$X2avg(t) = m\sum i = 1Pr(ci) \times \chi 2(t, ci)$$

(2.12)

Terms that have lower$\chi 2$values than a predetermined threshold are eliminated

2. Document Frequency (DF)

Document frequency (DF) of a term is the number of documents that the term appears. In this technique, each unique term's document frequency is computed, and terms whose document frequencies are less than a predetermined threshold are eliminated. This technique's basic assumption is that rare terms are either non-informative for document categorization or do not have much weight in global performance. This technique can also lead to an improvement in categorization accuracy in case rare terms are noise terms. However, DF is usually not used for aggressive term elimination because there is another widely accepted assumption in information retrieval that low-DF terms are distinctive and thus relatively informative and, for this reason, should not be removed aggressively.

A comparative study of feature selection in text categorization is presented by (Yang 1997). It has been reported that IG and χ^2statistic performed the best. However, DF, the simplest and the most efficient method in terms of computational complexity, performed similarly to IG and χ^2statistics. It has been suggested that DF can be reliably used instead of IG and χ^2statistics when computation performances of the latter two are too expensive. Another point to consider is that IG, MI, and χ^2statistics are supervised techniques and use information about category associations. As our main focus is on unsupervised techniques for document organization, these methods are not suitable to be applied in our study. To reduce the dimensionality of the data, applying DF. Thresholding is paramount by defining the DF threshold as one and removing the terms that appear in only one document.

3. Term Strength (TS)

The term strength method estimates term importance based on how commonly a term is likely to appear in closely related documents. The first step in this method is to use a training set of documents to find document pairs that have a similarity larger than a predetermined threshold. In the next step, TS is calculated based on the estimated conditional probability that a term appears in the second document, given that it appears in the first one. Supposed and dj are any pair of distinct but related documents. Then the TS of term t is defined to be.

$$TS\ (t) = P_r\ (t \in dj | t \in di)$$

Term Strength is an unsupervised dimensionality reduction technique where document categories are not used. It is based on document clustering and assumes that documents with many shared words are related, and the terms that are heavily shared among these related documents are relatively informative.

4. Knowledge Gain

Information gain measures the number of bits of information gained for category prediction when the presence or absence of a term in a document is known. When the set of possible categories is {c1,c2, ..., cm},the IG for each unique term t is calculated.

$$\mathrm{IG(t)=-}\sum_{i=1}^{|c|} P(ci)\log P(Ci) + P(t)\sum_{n=1}^{c} \mathrm{P(cijt)logP(cijt)} + \mathrm{P(_{t}p(t)}\sum_{i=1}^{c} \mathrm{P(cijt)logP(cij\_t)}$$

(2. 13)

IG calculates the decrease in entropy when the feature is given vs. absent (ci) is the prior probability of category ci. It can be estimated from the fraction of documents in the training set belonging to category ci. P (t) i. The prior probability of term t. It can be estimated from the fraction of documents in the training set in which term t is present. Likewise, (t) can be estimated from the fraction of documents in the training set in which term t is absent. Terms whose IGs are less than some predetermined threshold are removed from the feature space.

5. Information gain

Information gain (IG) is widely used in machine learning. (Yan, 2014) state that it is an entry by the presence or absence in an article in the amount of information to calculate the category value's contribution. The information gain (IG) heuristic is adopted due to its reported effectiveness in online text classification. IG(C; A) measures the amount of entropy decrease on class C when providing a feature A. The decreasing amount of entropy reflects the additional information gained by adding feature A, and higher values between 0 and 1 indicate more information gained by providing certain features (Suh, 2016).

IG is used for feature selection as it is easy to use, computationally efficient, and widely used in sentiment classification. Since IG is a filter-based feature selection technique based on a threshold value, all the features whose information gain value is greater than 0 are taken as prominent features (Agarwal, 2013). Although the Information Gain feature selection algorithms can significantly improve the classification performance, its performance usually declined sharply when dealing with the situation in which there is an imbalanced state of the term and classes (Wu & Xu, 2015).

2.9 Twitter

Twitter could be a micro-blogging site founded in March 2006 by Biz Stone, Jack Dorsey, and Evan Williams. Its logo could be a bluebird, and one tweet can include a maximum of 140 characters. Twitter’s core functions represent a really simple social awareness stream model. Twitter users can post short messages, or tweets, which are up to 140 characters long. The messages are displays as a “stream” on the user’s Twitter page. Although messages on microblogging sites and tweets have a personality limit of 140, their popularity grows at an unlimited rate. (Kaplan, 2011) mention three factors are explaining the success of micro-blogs. Through Twitter, one can receive updates on even the most trivial matters in other people’s lives. Secondly, Twitter allows a push push-pull combination. Finally, it offers a “platform for virtual exhibitionism and voyeurism for both active contributors and passive observers. The subsequent is the user twitter process. Yet, with ample active users, many of whom are quite passionate, Twitter supports a lively community with its own set of unique practices that are valuable to look at. Twitter combines elements of social network sites and blogs but with some notable differences. Like social network sites, profiles are connected through an underlying articulated network, but these connections are directed instead of undirected; participants can link to (“follow”) others and see their tweets, but the opposite user needn't reciprocate. Like blogs, participants’ Twitter pages show all of their tweets in reverse chronological order, but there's no ability to treat individual posts. User profiles are minimal and public, but users can make their tweet stream public or protected (private); the default and norm are public. Twitter's central feature, which users see after they log in, maybe a stream of tweets posted by people they follow, listed in reverse chronological order. Participants have different strategies for deciding who they follow. Although people can interact with Twitter directly through the website, many third-party applications are available, starting from mobile and desktop Twitter clients to tools that allow participants to trace popular topics, who un-follows whom, and how popular different users are.

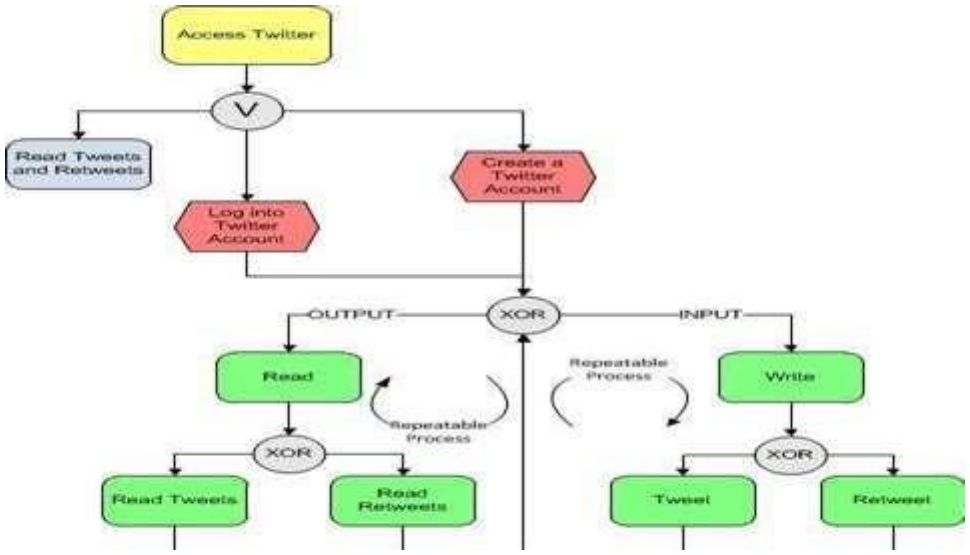


Figure 2.8: Twitter Architecture

2.9.1 Information Dissemination and Socializing with Twitter

Twitter is described as a microblog or social network site. It is for microblogging because the central activity posts short status update messages (tweets) via the net or a mobile. Twitter is additionally a social network site because members have a profile page with some personal information and might connect with other members by "following" them, thus gaining quick access to their content. It seems to be wont to share information and to explain minor daily activities (Java, 2007). However, it can even be used for information dissemination, as an example by government organizations. About 80% of Twitter users update followers on what they're currently doing, while the rest have an informational focus (Naaman, 2010). There are clear differences between users in terms of connection patterns: although most seem to be symmetrical in getting similar numbers of followers to the number of users followed, some are heavily skewed, suggesting broadcasting or primarily information gathering/evangelical function (Krishnamurthy, 2008). An unusual feature of Twitter is retweeting: forwarding a tweet by posting it again, the aim of this is often to disseminate information to the poster's followers, perhaps in modified form (Boyd, 2009), and this reposting seems to be extremely rapid (Kwak, 2010).

The reposting of the identical (or similar) information works because members tend to follow different sets of individuals. However, retweeting also serves other purposes, like helping followers to seek out older posts. The potential for information to flow rapidly through Twitter can even be seen from the very fact that the typical path length between a pair of users seems to be just over (Kwak, 2010). Moreover, if retweeted, a tweet can expect to succeed in a mean of 1000 users (Kwak, 2010). Moreover, a vital event may be expected to trigger more informational tweeting (Hughes, 2009), which suggests that it might be possible to detect important events through the automated analysis of Twitter. Another communicational feature of Twitter is that the hashtag: a metatag beginning with # designed to assist others in finding a post, often by marking the Tweet topic or its intended audience (Efron, 2010). This feature seems to possess been invented by Twitter users (Huang, 2010). The use of hashtags emphasizes the importance of widely communicating information on Twitter.

In contrast, the @ symbol is employed to handle a post to a different registered Twitter user, allowing Twitter to be used effectively for conversations and collaboration (Honeycutt, 2009). Moreover, about 31% of Tweets seem to be directed at a particular user using this feature (Boyd, 2009), emphasizing the social element of Twitter instead of the data broadcasting function related to hashtags. Although Twitter is employed for social purposes, there is considerable evidence that its significant use for information dissemination of assorted kinds, including personal information, could be its major use. Therefore, it’s reasonable to spot, analyze, and detect tweets that are accustomed disseminate hate speech within Twitter.

2.9.2 Introduction to Twitter Concept

A tweet may be a string with a maximum length of 140 characters, containing standard text, links, and hashtags (strings starting with the # symbol). As a microblogging and social networking website, Twitter has become very talked-about and has grown rapidly. An increasing number of individuals are willing to post their opinions on Twitter, which is now considered a valuable online source for opinions. Twitter reportedly had 241 million active users at the top of 2013, and these users generated quite 500 million tweets in a mean day (Goel, 2014). As a result, sentiment analysis on Twitter could be a rapid and effective way of gauging and analyzing vox populi on various issues. (Kaplan, 2010) define social media as “a group of Internet-based applications that hinge upon the ideological and technological foundations of Web 2.0 which allow the creation and exchange of User Generated Content”. In their paper, they divide social media into six different categories: collaborative projects (e.g., Wikipedia), Blogs, Social Networking Sites (SNS), content communities (e.g., Youtube), visual social worlds (e.g., Second Life), and virtual game worlds.

Despite its relatively small life span, Twitter and its potential in political communication and discussions received lots of educational interest. Therefore, it's well established that Twitter plays a big role in communicating current political issues and discussing them. By default, Twitter messages are public, accessible online, and searchable. This feature of Twitter in itself releases many opportunities to try to research microblogging. Furthermore, from the start, Twitter has provided some public messages to businesses and academia through the company’s Application Protocol Interfaces (APIs). Simultaneously, the increasing popularity and importance of the platform as an information source also attracted various information manipulation styles from spreading false rumors, propagating hatred between and among various Twitter users that focus on their ethnicity, color, religion, economic status, and even political alignments.

Twitter supports posting messages via SMS, Web, and mobile Web services and allows users to use different “third party” applications to post (and consume) Twitter messages.

As a result, an array of applications and avenues for posting to Twitter are available, starting from mobile, web-based, desktop, and other applications, including posts on behalf of the user from automated agents. Finally, Twitter users can reference other users in posted messages using the @ symbol, effectively creating links from their message to the referenced user's account. Such reference messages (known as "reply" or "mention," looking on the use) appear within the referenced user's account so that users can keep track of messages mentioning them.

2.9.3 Twitter Data

Twitter provides three public APIs for developers to access their massive corpus of data. These are the REST, Search, and Streaming APIs and shall now be further explored.

Search API-The Search API is the simplest tool provided by Twitter. This API is designed to allow users to query for Twitter content and works very much as the search bar found on the Twitter website. This content may include a set of tweets with specific keywords or tweets from or mentioning a specific user. A simple search would yield up to 1500 of the latest tweets in the last seven days, cached over a 60 second period. There are, however, restrictions on the rate at which programs can utilize this API.

1. **Rest API-**The REST API enables programs to access more of the core Twitter functions. This API retrieves not only the information taken from the Search API but also allows building timelines and retrieves more specific user information such as the user's name, profile avatar, tweet count, and the number of followers and friends they have. The REST API also allows programs to post on Twitter and carry out other functions like retweeting or favoring tweets. These extra functions, however, are not required in this project.
2. **Streaming API**-Twitter's Streaming API is a real-time sample of all public tweets posted on the sample. It allows filtering in various ways such as user id, keywords, or even random sampling and is regarded as the default option for data mining operations. This is because the Streaming API allows a long-lived HTTP

connection, unlike the other APIs, and as such, programs can constantly remain connected to retrieve a running stream of tweets, as the name itself suggests. This removes the overheads associated with reconnecting every time you want to make a query, and the API removes all rate limitations. Hence, there is no worry about exceeding your quota. Unlike the other APIs, programs must be authenticated to use the Streaming API.

2.9.4 Twitter Twitter's Functions and Tools

Himelboim (2013) mentioned that this length restriction of only 140 characters on Twitter serves a fast production, distribution, and message consumption. Hence, news can be published and consumed in a matter of seconds, in contrast to traditional media or other new media channels. The following list explains some

Twitter-specific vocabulary, which will be used throughout the rest of the thesis:

1. **Tweet**: This is the name for one post on the Twitter platform. The length of a tweet is restricted to 140 characters.
2. **User and username**: To post tweets on Twitter, an author must register with the platform first and is afterward known under a freely chosen pseudonym. To interact with other users on Twitter, authors can mention other users' pseudonyms using the @ symbol (e.g.@kevoyoung), which leads to the mentioned user being notified about the tweet. This is often used in conversations to indicate that a post made is meant to answer another tweet.
3. **Followe**r: Users on Twitter can connect by \following "other people, meaning that they get notified about new tweets by users they follow. Following is no bidirectional connection such as a Facebook friendship, so every user has separate lists of other users she follows and users she is followed by.
4. **Retweet (RT)-**Twitter's function that is important to discuss is the re-tweet mechanism. The power of Twitter as a medium for disseminating information and messages, especially by stressing the importance of the re-tweet option. A more

representative example of the retweet power is demonstrated in research conducted by (Kwak 2010), in which the entire Twittersphere10 was explored and analyzed. The study showed that Twitter people do not always acquire information and news from the users who follow, but it is more common to be informed via retweets. Theysupport their claim by presenting findings revealing that any retweeted tweet reaches an average of 1.000 users no matter the number of followers of the original tweet.

5. **Hashtag**- According to (Bruns,2011) the Twitter hashtag is “a short keyword, prefixed with the hash symbol ‘#,’ as a means of coordinating a distributed discussion between more or less large groups of users, who do not need to be connected through existing ‘follower’ networks.” One important beneficial element of hashtags is responding at once to emerging issues or events and the flexibility to create a new hashtag thread as and when needed, without any restrictions.
6. **Twitter Privacy-**Twitter allows users to see and explore the information shared by other users. Unless a user protects his information, any user may view the information shared without receiving permission from the user who posted the information. As a confounding factor, users' privacy concern is important to Twitter, and Twitter takes great effort to protect privacy.
7. **Twitter Handles-**After a user registers with Twitter, creates an account and obtains a Twitter handle. They may participate as a “reader” or a “user” who may submit information to the network. To obtain a Twitter handle (username), a user must submit personal information and accept the policies [30] before Twitter provides them with an account. The personal information includes the full name and e-mail address by which to confirm personal identity. The Twitter handle consists of the username preceded with the symbol @.
8. In most cases, the user selects his unique username. If the desired username is taken, Twitter provides similar alternatives. The username can be composed of any combination of letters, numbers, or symbols. The username does not have to be associated with the user’s name or personal information, although it often is. “**@** ”

is an example of a Twitter handle made of symbols, “**@kevoyoung”** is an example of a Twitter handle that the user chose that has no association with the user’s name.

9. **Twitter and Audience-**As in much computer-mediated communication, a tweet's actual readers, differ from its producer's imagined audience. Twitter allows individuals to send private messages to people they follow through direct messages (DMs), but the dominant communication practices are public. A convention is known as the '@reply' (consisting of the @ sign and username) lets users target a conversation to or reference a particular user. Still, anyone can view these tweets through search.twitter.com, the public timeline or the sender's Twitter page (Honeycutt,2009) provide a detailed discussion of @replies).On Twitter, there is a disconnect between followers and followers. For instance, musician John Mayer (JohnMayer) is followed by 1,226,844 users but follows only 47. While followers indicate the audience, this is imprecise.

When an individual's account is public, anyone – with or without a Twitter account – can read their tweets through the site, RSS, or third-party software. The vast majority of Twitter accounts are public. Those who choose to protect their accounts can restrict their audience. Still, the lists of followers on both public and protected accounts indicate only a potential audience since not everyone who follows a user reads all their tweets. Tweets are also spread further when participants repost tweets through their accounts. This practice, commonly referred to as 'retweeting,' can introduce content to new audiences (Boyd, 2010). While the dominant norm is to use @username to cite the original author or attribute the person who spread the message, retweeted messages are often altered. They may lose any reference to the original.

Additionally, it is not uncommon for people to forward tweets via email or copy and paste them into new communication channels. Given the various ways people can consume and spread tweets, Twitter users can't account for their potential audience, let alone actual readers. Yet, this inability to know the exact audience does not mean that infinite numbers see people's tweets. As with blogs (Shirky, 2005), nearly all tweets are read by relatively few people – but most Twitter users.

10. **Preprocessing Data-**As Twitter has a special nature of its tweets; there should be some preprocessing. For example, Twitter does not support embedding pictures, audio, or video in its tweets. Still, it supports hosting them externally and attaches the link of such media to the text message. This requires excluding the URL attachments or parsing it to know its nature. Twitter comments are not in the form of formal English structure. Since it contains user comments, it cannot apply the algorithm/techniques directly by applying text mining; raw tweets have to be cleaned based on the data structure that we like to apply. The preprocessing task contains the technique which has identified the latent pattern through that concept provided by the interestingness and applied in available data. Tweeter's pattern has been modeled. (Restricted by the chosen hashtag). By feeding the data and populate the data extractor and classifier.

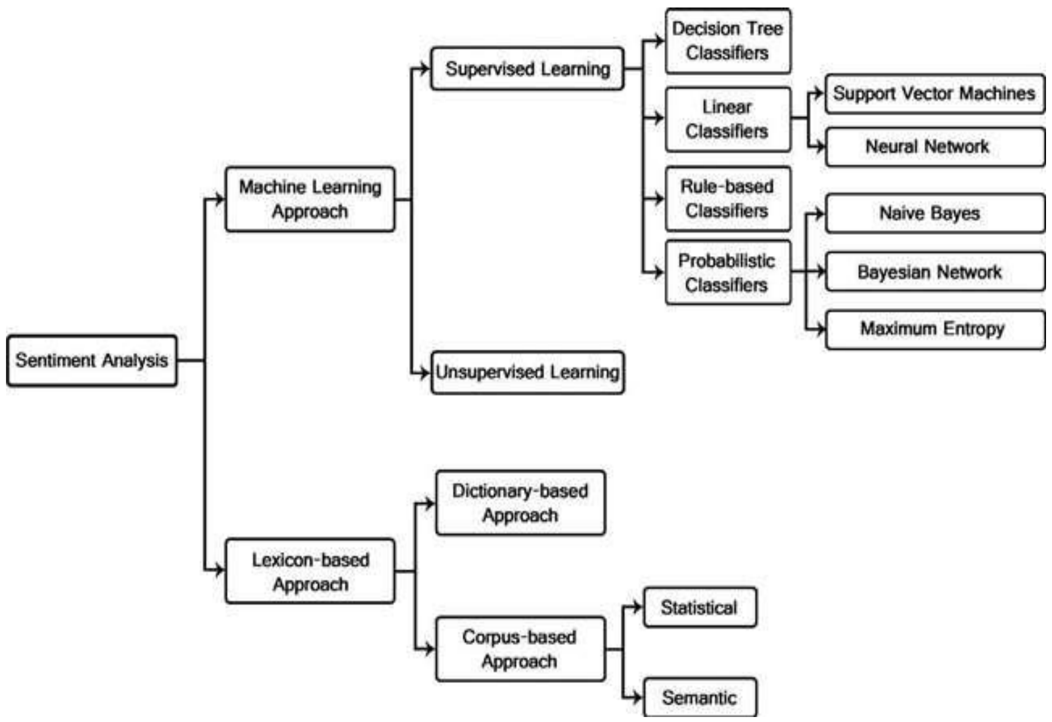


Figure 2.9: Showing General Sentiment Analysis Techniques (Medhat, 2004)

2.9.5 Evaluating Performance of Text Classifiers

After a classifier is built, it needs to be evaluated for its performance to be known. Appropriate evaluation is crucial because it cannot be used in real-world tasks without knowing the classifier's approximate accuracy. Several criteria may be used to evaluate classification algorithms' performance in supervised Machine Learning (ML). In general, different measures evaluate different characteristics of the classifier induced by the algorithm. There are many ways to evaluate a classifier, and there are also many measures. The main measure is the classification accuracy, which is the number of correctly classified instances in the test set divided by the total number of instances in the test set (Liu, 2007). Because the text categorization problem is not sufficiently well-defined, classifiers' performance can be evaluated only experimentally. This collection is divided into two areas: the training and test document sets. When there is a need to optimize some classifier parameters experimentally, the training set is further divided into two parts the training set proper and a validation set, which is used for the parameter optimizations

(Feldman, 2007). The inbuilt NLTK metrics are used to measure accuracy, precision-measure, and recall.

1. Accuracy

The accuracy of any classifier on a given test set is the percentage of test set data classified by the classifier. Each test data's given class label is compared with the learned classifier's class prediction for that data. If the classifier's accuracy is acceptable, the classifier can classify future data tuples for which the class label is not known. Such data are also referred to in the machine learning literature as \unknown" or \previously unseen" data (Cios, 2007). Any given classifier's accuracy illustrates a given classifier's ability to correctly predict the class label of new or previously unseen data. A classifier's accuracy defines how well a given predictor can guess the predicted attribute value for new or previously unseen data (Cios, 2007).

True Positives (TP): the sentence that is positive and was estimated as positive.

True Negatives (TN): the sentence that is negative and was estimated as negative. False Positives (FP): the sentence that is negative but estimated as positive.

False Negatives (FN): the sentence/document that is positive but estimated as negative

.

Accuracy **=** **TP** **+** **TN**

(2. 14)

TP+TN+FP+FN

2. Recall

The recall is the number of correct results divided by the number of results that should have been returned. Recall in information retrieval is the fraction of the documents relevant to the query that are successfully retrieved. It is trivial to recall 100% by returning all documents in response to any query. Therefore, recall alone is not enough, but one needs to measure the number of non-relevant documents.

$$\text{Recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}}$$

$$Recall = \frac{TP}{TP+FN} \tag{2. 15}$$

3. Precision

Precision is the percentage of correctly classified documents among all assigned documents to the category by the classifier (Witten, 2005). According to (Swamy, 2014) precision is the percentage of correctly classified documents among all assigned documents to the classifier category.

$$\boldsymbol{Precision = \frac{TP}{TP+FP}} \tag{2. 16}$$

4. F-Measure

It is a measure of a test's accuracy. It considers both the precision P and the recall R of the test to compute the score. The F1 score can be interpreted as a weighted average of precision and recall, where an F1score reaches its best value at one and the worst score at 0. The traditional F- measure or balanced F-Score (F1 score) is the harmonic mean of precision and recall.

The harmonic mean discourages classifiers that sacrifice one measure for another too drastically (Chakrabarti, 2003). Among the four standard measures, accuracy assesses the overall classification correctness, while the others evaluate the correctness regarding each class (Suh,2016).

$$\mathrm{F1} = 2.\frac{\mathrm{presision.recall}}{\mathrm{precision} + \mathrm{recall}} \tag{2.17}$$

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

In this chapter, the methodology is outlined. Data collection process, Text preprocessing, Sentiment detection, sentiment analysis, and model development are done to achieve the thesis objectives of identifying and classifying hate speech.

3.2 Data Collection

This research was based on data from social media and particularly on Twitter. The data was collected between November 2017 to August 2018. Data in the form of raw tweets was acquired using the python library "tweepy," which provides a simple Twitter streaming API package. In particular, Tweepy is one of the most interesting and straightforward to use. The model was implemented on Pycharm IDE using Python as the programming language. JSON (JavaScript Object Notation) is implemented to text and stored in MongoDB.To build our classifier, a library of Python called, Scikit-learn is used. Scikit- learn has tools for classification, clustering, regression, and visualization.

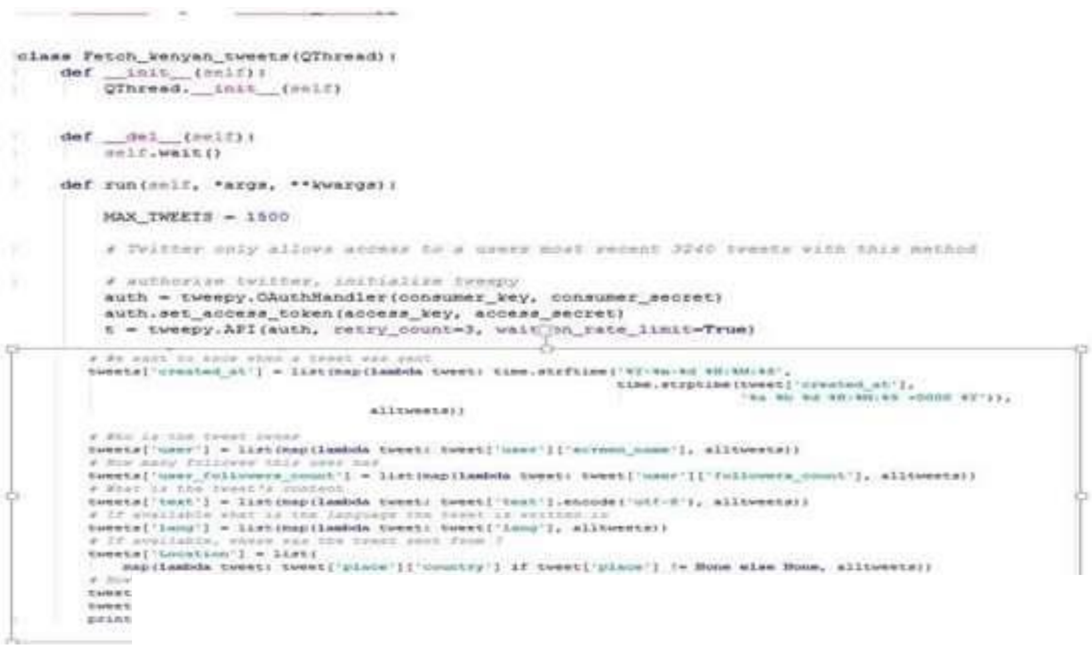
To install Scikit-learn, we simply use an online command in python, 'pip install sci-kit learn.' This section focuses on the process of classifying the data taken from the twitters API. Python is a very powerful language that provides many services with the help of many Python libraries. Tweepy is one of the open-source Python libraries that enables Python to communicate with Twitter and use its API to collect data to use it in our program.

Data collection is a priority as the collected data is vital for the project's overall success.

Data was collected from Twitter using the Twitter API, tweepy. Getting Twitter data is comparably easy, as Twitter offers an easy to use API. Restricted by an hourly limit of API calls, all tweets were accessed via different HTTP endpoints, such as tweets posted by a specific user, tweets containing specific terms, or by the individual ID of each tweet. The tweets and their associated Meta information such as the date, author, language, location, time zone, and more are returned in JSON format, a popular format for data exchange, and is supported by many programming languages.

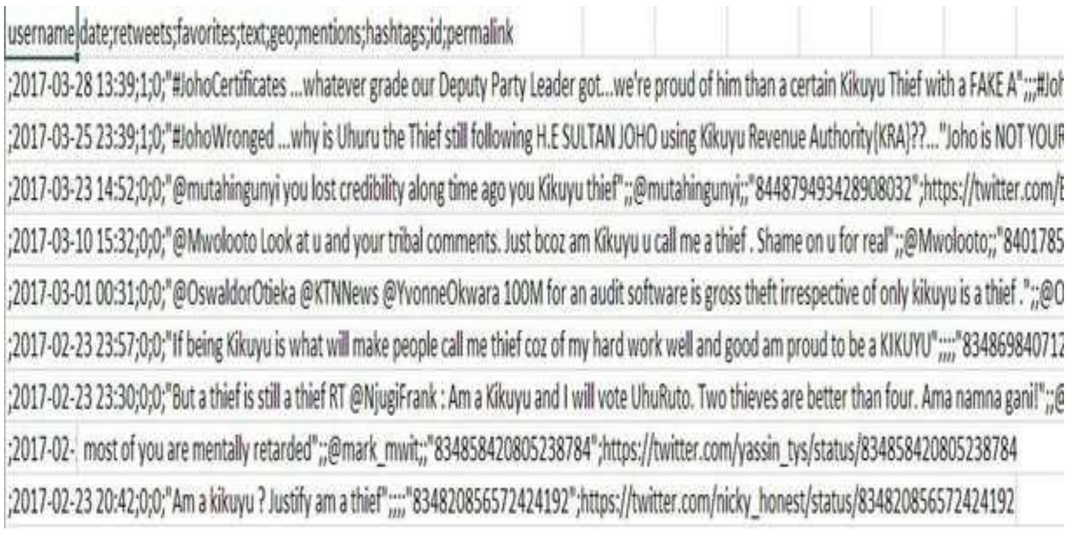
Creating new datasets, the so-called streaming API is most interesting, as it provides unlimited access to the live stream of incoming tweets matching a given query. Typical queries are lists of emoticons, create annotated datasets with noisy labels, frequent words such as stop words, capture a mostly unbiased stream of tweets with a wide variety of different topics, or specific terms like names of entities, to create datasets concerning a certain topic. (George Valkanas, 2013), demonstrated that though we have the free API, which provides a relatively low tweet count compared to the commercial ones, the main difference is the magnitude; otherwise, they both observe identical periodicity and temporal patterns. Twitter API provided by Twitter Inc. is a free and readily available tool for use in accessing the Twitter network platform. Though the content is limited for free version, (George Valkanas, 2013) report showed the difference is only in magnitude. Otherwise, they have relatively identical temporal patterns and periodicity. Also, hashtags are useful as queries, as they can be seen as some sort of label the author attaches to the message.

Twitter API allows sharing tweet IDs, which can be used to download the corresponding tweets via the Twitter API; however, Twitter users can choose to delete or privatize their tweets. The data collection requirement's scope is to collect data (tweets) from Twitter using the Twitter API and then store the collected data for further analysis. For this study, data collection was implemented multiple times to harvest a large amount of data. The data collection from Twitter using python codes is shown in figure 3.1, and the processed is shown in figure 3.2.



```
class Fetch_kenyan_tweets(QThread):
    def __init__(self):
        QThread.__init__(self)


    def __del__(self):
        self.wait()

    def run(self, *args, **kwargs):

        MAX_TWEETS = 1500

        # Twitter only allows access to a users most recent 3240 tweets with this method

        # authorize twitter, initialize tweepy
        auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
        auth.set_access_token(access_key, access_secret)
        t = tweepy.API(auth, retry_count=3, wait_on_rate_limit=True)
        # We want to know when a tweet was sent
        tweets['created_at'] = list(map(lambda tweet: time.strftime('%Y-%m-%d %H:%M:%S',
                                                       time.strptime(tweet['created_at'],
                                                            '%a %b %d %H:%M:%S +0000 %Y')),
                                   alltweets))

        # Who is the tweet owner
        tweets['user'] = list(map(lambda tweet: tweet['user']['screen_name'], alltweets))
        # How many follower this user has
        tweets['user_followers_count'] = list(map(lambda tweet: tweet['user']['followers_count'], alltweets))
        # What is the tweet's content
        tweets['text'] = list(map(lambda tweet: tweet['text'].encode('utf-8'), alltweets))
        # If available what is the language the tweet is written in
        tweets['lang'] = list(map(lambda tweet: tweet['lang'], alltweets))
        # If available, where was the tweet sent from ?
        tweets['Location'] = list(
            map(lambda tweet: tweet['place']['country'] if tweet['place'] != None else None, alltweets))
        # How
        tweet
        tweet
        print
```

Figure 3.1: python code for collecting tweets in Kenya.



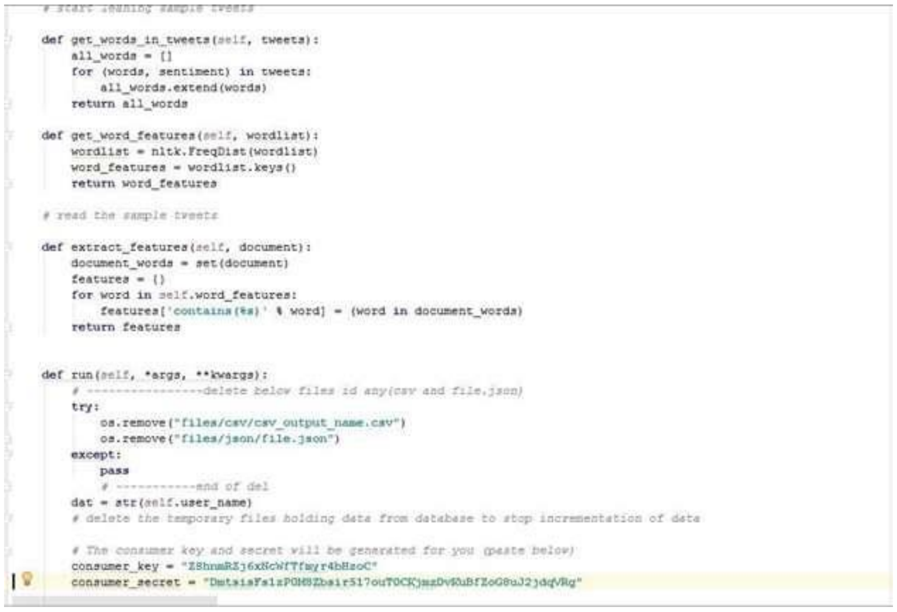
```
username date;retweets;favorites;text;geo;mentions;hashtags;id;permalink
;2017-03-28 13:39;1;0;"#JohoCertificates ...whatever grade our Deputy Party Leader got...we're proud of him than a certain Kikuyu Thief with a FAKE A";;;#Joh
;2017-03-25 23:39;1;0;"#JohoWronged ...why is Uhuru the Thief still following H.E SULTAN JOHO using Kikuyu Revenue Authority(KRA)??..."Joho is NOT YOUR
;2017-03-23 14:52;0;0;"@mutahingunyi you lost credibility along time ago you Kikuyu thief";;@mutahingunyi;;"844879493428908032";https://twitter.com/
;2017-03-10 15:32;0;0;"@Mwolooto Look at u and your tribal comments. Just bcoz am Kikuyu u call me a thief . Shame on u for real";;@Mwolooto;;"8401785
;2017-03-01 00:31;0;0;"@OswaldorOtieka @KTNNews @YvonneOkwara 100M for an audit software is gross theft irrespective of only kikuyu is a thief .";;@O
;2017-02-23 23:57;0;0;"If being Kikuyu is what will make people call me thief coz of my hard work well and good am proud to be a KIKUYU";;;;"83486984071
;2017-02-23 23:30;0;0;"But a thief is still a thief RT @NjugiFrank : Am a Kikuyu and I will vote UhuRuto. Two thieves are better than four. Ama namna gani!";;@
;2017-02-  most of you are mentally retarded";;@mark_mwit;;"834858420805238784";https://twitter.com/yassin_tys/status/834858420805238784
;2017-02-23 20:42;0;0;"Am a kikuyu ? Justify am a thief";;;;"834820856572424192";https://twitter.com/nicky_honest/status/834820856572424192
```

Figure 3.2: Showing data collection from Twitter API.

3.3 Keyword Selection

Twitter produces around 6000 tweets per second (Stats, 2018), thus increasing the dataset's probability of noise. To have the relevant data for the experiment, the keywords/wordlist were selected carefully and were used in analyzing and classifying tweets as either positive or negative. The wordlist was integrated into the classification model, as shown in figure 3.3. The keywords are neither very specific nor too general. If the keywords are very specific to a certain event, it might limit the extracted tweets. For example - #kill very specific. Also, not everybody will tweet using these keywords, and hence informational tweets might not be captured by the classification model.

On the other hand, using very general keywords like #Killall lead to too much noise in the collected dataset as this keyword does not necessarily mean kill all Kenyans. Twitter users can just be talking about a kill a specific person. Therefore, it becomes extremely crucial to choose the correct set of keywords.



```
# start loading sample tweets

def get_words_in_tweets(self, tweets):
    all_words = []
    for (words, sentiment) in tweets:
        all_words.extend(words)
    return all_words

def get_word_features(self, wordlist):
    wordlist = nltk.FreqDist(wordlist)
    word_features = wordlist.keys()
    return word_features

# read the sample tweets

def extract_features(self, document):
    document_words = set(document)
    features = {}
    for word in self.word_features:
        features['contains(%s)' % word] = (word in document_words)
    return features


def run(self, *args, **kwargs):
    # ----------------delete below files id any(csv and file.json)
    try:
        os.remove("files/csv/csv_output_name.csv")
        os.remove("files/json/file.json")
    except:
        pass
        # -----------end of del
    dat = str(self.user_name)
    # delete the temporary files holding data from database to stop incrementation of data

    # The consumer key and secret will be generated for you (paste below)
    consumer_key = "Z8hnmRZj6xNcWfTfmyr4bHzoC"
    consumer_secret = "DmtsisFs1zPOM8Zbsir517ouTOCKjmzDv$uBfZoG8uJ2jdqVRg"
```

Figure 3.3: Python code for extracting tweets using a wordlist

3.4 Data Annotation

Building models to classify data according to a predefined coding scheme is essential in digital social research to understand social interactions, beliefs, emotions, and the like. In this research, once the Twitter data were collected, we built a supervised machine learning classifier to distinguish between hateful or antagonistic responses, focusing on race, ethnicity, religion, and more general responses, following the event. To complete this subjective task using large-scale data analytics, which is necessary for the volumes of data produced, machine classifiers were used to learn the features of tweets that indicate the class they belong to, as shown in Figure 3.4.

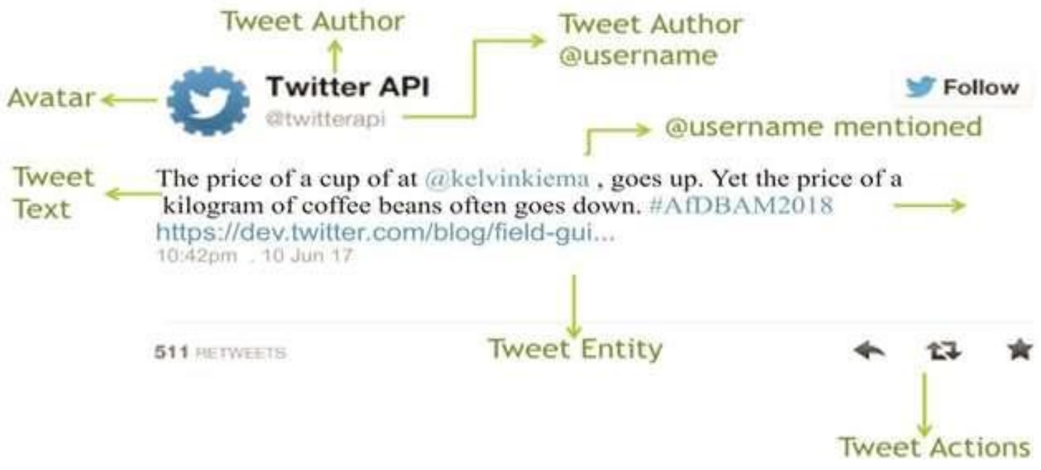

Tweet Author
Tweet Author @username
Avatar
Twitter API
@twitterapi
Follow
@username mentioned
Tweet Text
The price of a cup of at @kelvinkiema , goes up. Yet the price of a kilogram of coffee beans often goes down. #AfDBAM2018
https://dev.twitter.com/blog/field-gui...
10:42pm . 10 Jun 17
511 RETWEETS
Tweet Entity
Tweet Actions

Figure 3.4 Illustrative examples of annotated Tweet

3.5 Twitter Data Analysis Process.

3.5.1 Twitter Mining Application Setup

To initiate authorized calls to Streaming API and collect data for the preprocessing phase, we need to create a Twitter application that will obtain the access token. Open Authentication (OAuth) is standard for authentication that provides applications to access data from other services without revealing credentials. In our solution for data mining, we want to establish a connection to Streaming API and thus Twitter control panel for developers, apps.Twitter.com offers to generate our access token.

3.5.2 Creating Twitter application

Firstly, getting an access token starts with creating new applications from the Twitter Application Management panel by filling up required attributes for the application, such as Name, Description, and Website.

Figure 3.5: Twitter application setup

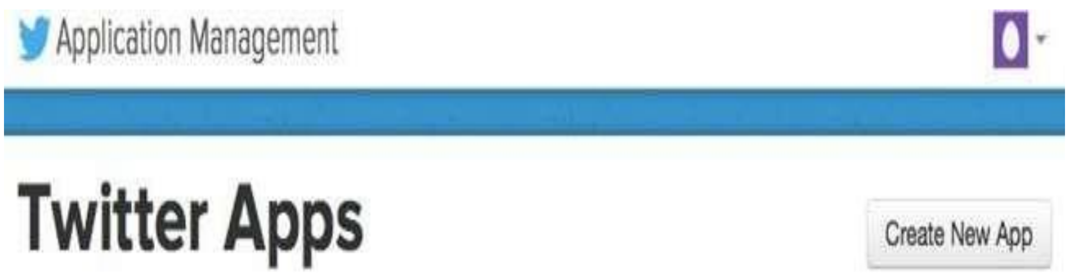
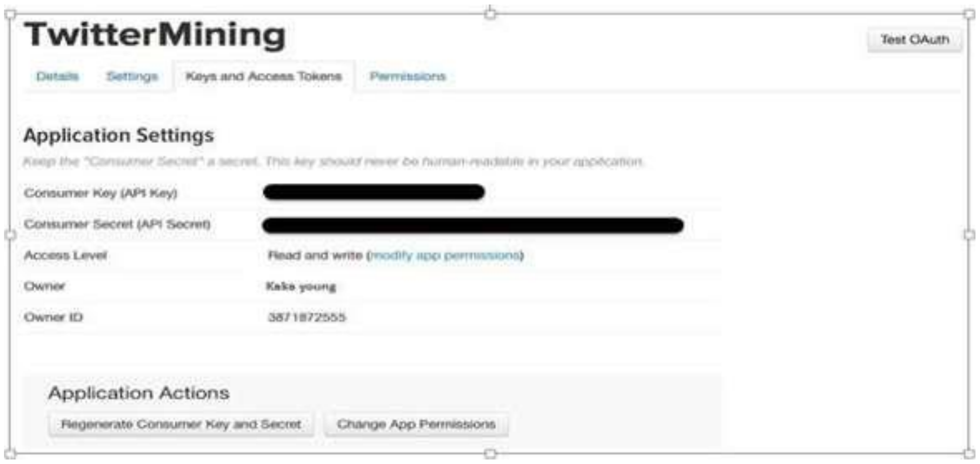


Figure 3.6: Twitter application setup

3.5.3 Obtaining Twitter credentials

For the development and Streaming API, access is important credentials located under the Keys and Access Tokens tab in the newly created Twitter Mining application settings. There are four credentials to note: Consumer Key (API Key) and Consumer Secret (API Secret), Access Token, and Access Token Secret. These credentials provide everything for the Twitter Mining application to authorize itself and make API requests on its Owner (kakayoung) behalf. Owner username – (kaka young) represents the researcher's own Twitter account. Figure 3.6 shows the configuration control panel with Twitter application settings. Credentials are blacked, as they present sensitive information.



Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

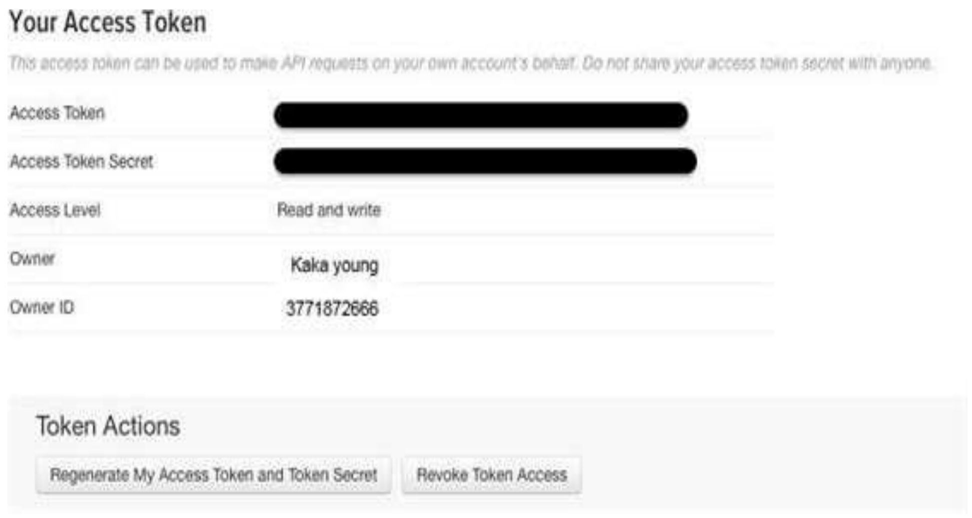
Consumer Key (API Key)	[redacted]
Consumer Secret (API Secret)	[redacted]
Access Level	Read and write (modify app permissions)
Owner	Kaka young
Owner ID	3871872555

Application Actions

Figure 3.7: Twitter Mining Application Settings

3.5.4 Twitter access token

Access tokens are used to make API requests to Twitter service from owners' accounts. Moreover, the access level is set to Read and Write for this project's purpose; however, it can be changed according to the application's permission settings if necessary. Furthermore, tokens can be regenerated or revoked as shown in Figure 3.14



Your Access Token

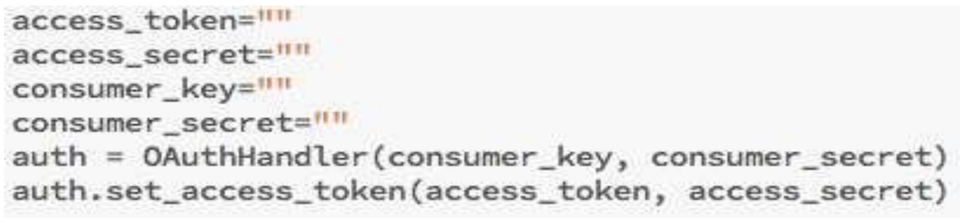
This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	[redacted]
Access Token Secret	[redacted]
Access Level	Read and write
Owner	Kaka young
Owner ID	3771872666

Figure 3.8: Twitter access token credentials settings

3.5.5 Creating Streaming Connection

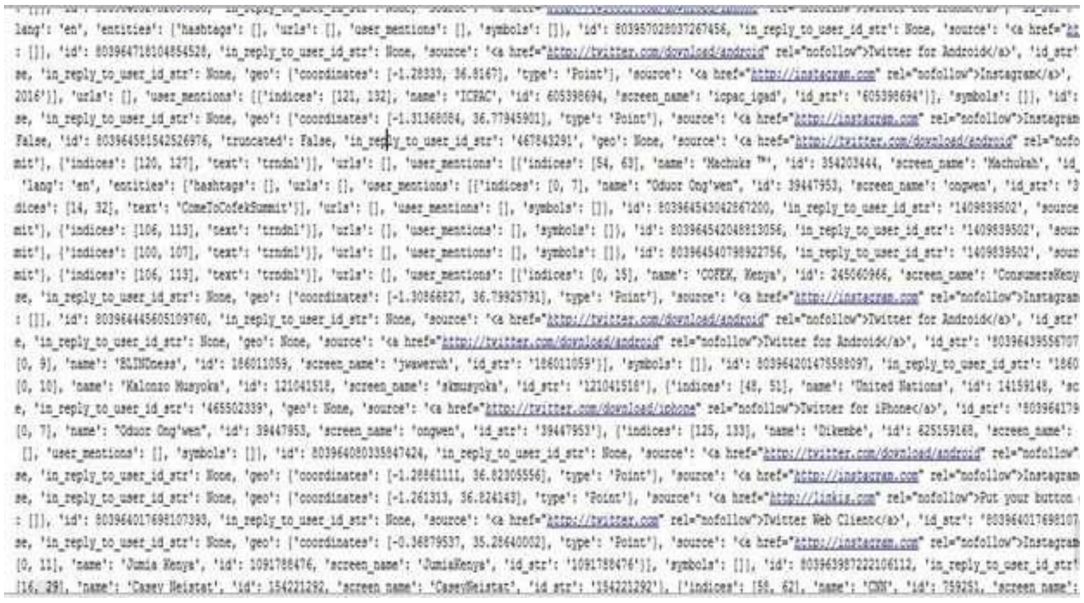
After successfully obtaining credentials for Streaming API, we need to initialize the connection and collect sample tweets to analyze its attribution. API key and API secret have to be passed to OAuthHandler to create object *auth* to setup authentication while function set_access_token will setup Access Token and Access Token Secret. The code is included in the attachment file that comes with this project. Figure 3.8 shows the approach.



```
access_token=""
access_secret=""
consumer_key=""
consumer_secret=""
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)
```

Figure 3.9: Twitter Streaming API authentication

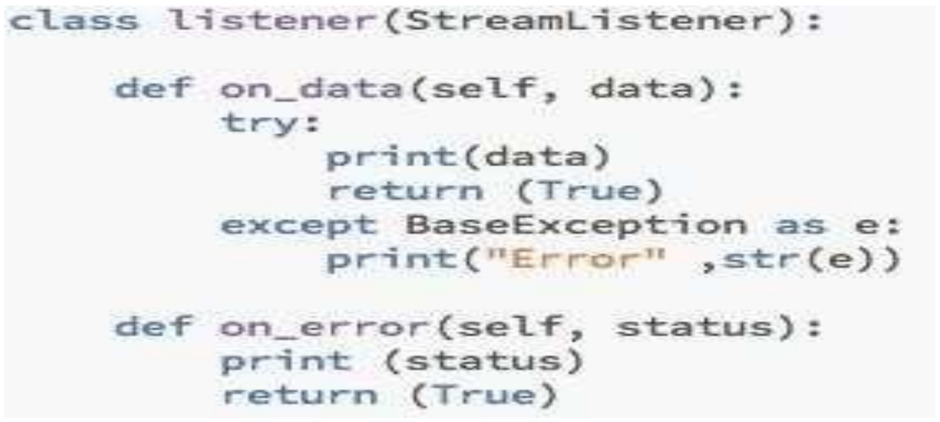
Data analyses require some sample partition of tweets to get familiar with its content; therefore, the following code will stream near real-time tweets based on the user input. All tweets are printed in the console, but they are also piped into text documents for a better overview. For example, for user input: user *3*, *the* tweet stream looks as in Figure 3.9.



```
lang': 'en', 'entities': {'hashtags': [], 'urls': [], 'user_mentions': [], 'symbols': []}, 'id': 803957028037267456, 'in_reply_to_user_id_str': None, 'source': '<a href="ht
: []}, 'id': 803964718104854528, 'in_reply_to_user_id_str': None, 'source': '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>', 'id_str'
se, 'in_reply_to_user_id_str': None, 'geo': {'coordinates': [-1.28333, 36.8167], 'type': 'Point'}, 'source': '<a href="http://instagram.com" rel="nofollow">Instagram</a>',
2016'}], 'urls': [], 'user_mentions': [{'indices': [121, 132], 'name': 'ICPAC', 'id': 605398694, 'screen_name': 'icpac_igad', 'id_str': '605398694'}], 'symbols': []}, 'id':
se, 'in_reply_to_user_id_str': None, 'geo': {'coordinates': [-1.31368084, 36.77945901], 'type': 'Point'}, 'source': '<a href="http://instagram.com" rel="nofollow">Instagram
False, 'id': 803964581542526976, 'truncated': False, 'in_reply_to_user_id_str': '467843291', 'geo': None, 'source': '<a href="http://twitter.com/download/android" rel="nofo
mit'}, {'indices': [120, 127], 'text': 'trndnl'}], 'urls': [], 'user_mentions': [{'indices': [54, 63], 'name': 'Machuks ™', 'id': 354203444, 'screen_name': 'Machukah', 'id_
 'lang': 'en', 'entities': {'hashtags': [], 'urls': [], 'user_mentions': [{'indices': [0, 7], 'name': 'Oduor Ong'wen', 'id': 39447953, 'screen_name': 'ongwen', 'id_str': '3
dices': [14, 32], 'text': 'ComeToCofekSummit'}], 'urls': [], 'user_mentions': [], 'symbols': []}, 'id': 803964543042867200, 'in_reply_to_user_id_str': '1409839502', 'source
mit'}, {'indices': [106, 113], 'text': 'trndnl'}], 'urls': [], 'user_mentions': [], 'symbols': []}, 'id': 803964542048813056, 'in_reply_to_user_id_str': '1409839502', 'sour
mit'}, {'indices': [100, 107], 'text': 'trndnl'}], 'urls': [], 'user_mentions': [], 'symbols': []}, 'id': 803964540798922756, 'in_reply_to_user_id_str': '1409839502', 'sour
mit'}, {'indices': [106, 113], 'text': 'trndnl'}], 'urls': [], 'user_mentions': [{'indices': [0, 15], 'name': 'COFEK, Kenya', 'id': 245060966, 'screen_name': 'ConsumersKeny
se, 'in_reply_to_user_id_str': None, 'geo': {'coordinates': [-1.30866827, 36.79925791], 'type': 'Point'}, 'source': '<a href="http://instagram.com" rel="nofollow">Instagram
: []}, 'id': 803964445605109760, 'in_reply_to_user_id_str': None, 'source': '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>', 'id_str'
e, 'in_reply_to_user_id_str': None, 'geo': None, 'source': '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>', 'id_str': '80396439556707
[0, 9], 'name': 'BLINDness', 'id': 186011059, 'screen_name': 'jwaweruh', 'id_str': '186011059'}], 'symbols': []}, 'id': 803964201475588097, 'in_reply_to_user_id_str': '1860
[0, 10], 'name': 'Kalonzo Musyoka', 'id': 121041518, 'screen_name': 'skmusyoka', 'id_str': '121041518'}, {'indices': [48, 51], 'name': 'United Nations', 'id': 14159148, 'sc
e, 'in_reply_to_user_id_str': '465502339', 'geo': None, 'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>', 'id_str': '80396417
[0, 7], 'name': 'Oduor Ong'wen', 'id': 39447953, 'screen_name': 'ongwen', 'id_str': '39447953'}, {'indices': [125, 133], 'name': 'Dikembe', 'id': 625159168, 'screen_name':
 [], 'user_mentions': [], 'symbols': []}, 'id': 803964080335847424, 'in_reply_to_user_id_str': None, 'source': '<a href="http://twitter.com/download/android" rel="nofollow"
se, 'in_reply_to_user_id_str': None, 'geo': {'coordinates': [-1.28861111, 36.82305556], 'type': 'Point'}, 'source': '<a href="http://instagram.com" rel="nofollow">Instagram
se, 'in_reply_to_user_id_str': None, 'geo': {'coordinates': [-1.261313, 36.824143], 'type': 'Point'}, 'source': '<a href="http://linkis.com" rel="nofollow">Put your button
: []}, 'id': 803964017698107393, 'in_reply_to_user_id_str': None, 'source': '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>', 'id_str': '803964017698107
se, 'in_reply_to_user_id_str': None, 'geo': {'coordinates': [-0.36879537, 35.28640052], 'type': 'Point'}, 'source': '<a href="http://instagram.com" rel="nofollow">Instagram
[0, 11], 'name': 'Jumia Kenya', 'id': 1091788476, 'screen_name': 'JumiaKenya', 'id_str': '1091788476'}], 'symbols': []}, 'id': 803963987222106112, 'in_reply_to_user_id_str'
[16, 29], 'name': 'Casey Neistat', 'id': 154221292, 'screen_name': 'CaseyNeistat', 'id_str': '154221292'}, {'indices': [58, 62], 'name': 'CNN', 'id': 759251, 'screen_name':
```

Figure 3.10: A tweet Metadata saved in JSON format

The idea of analyzing tweets as part of pre-processing before sentiment analyses is to understand attributes within JSON format for each tweet and normalize them according to natural language processing paradigms, as indicated in Figure 3.10. The following class is a listener instance for Twitter streaming that will print tweets and save them into a document – TwitterMining.txt and inform the user with the message if any error occurs with streaming.



```
class listener(StreamListener):

    def on_data(self, data):
        try:
            print(data)
            return (True)
        except BaseException as e:
            print("Error" ,str(e))

    def on_error(self, status):
        print (status)
        return (True)
```

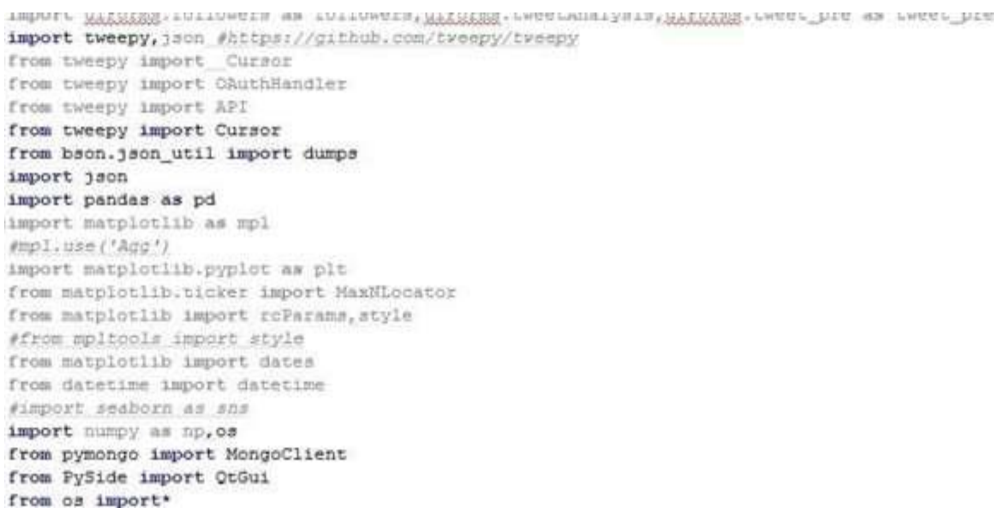
Figure 3.11: Listener class

Twitter streaming is ready after auth and listener () instances are passed to the Stream object that will now contain credentials for authentication and information about tweets, which have to be streamed. Moreover, the filter () method called on Twitter Stream will take an input stream word from the user and collect only tweets in Kenya with various attributes. Collecting and analyzing tweets is an important part of data mining because it prepares data in the processing format. The collected Twitter stream of sample tweet dataset is now capable of storing data in JSON format. JavaScript Object Notation (JSON) is a format for data interchange based on comma-separated key-value pairs. It is a human-readable data format that is easy to parse.

3.5.6 Storing tweets in MongoDB

Once we start getting our data from Twitter API, the next step was to store that sentiment analysis data. The data collection was run for several months for different Twitter users in Kenya. The data was collected from Twitter between November 2017-August, 2018. The data, as described in figure 3.11. CSV (Comma Separated Values) file is generated, which consists of tweets that are extracted from Twitter API. The thesis used a .csv format for collected data files because data consists of many fields. CSV separate each field with a comma, making it easier to access the particular field that consists of text. CSV files also provide faster read/write time as compared to others.

To store the dataset from API, not only in the text document but also into the database, the thesis is implemented in MongoDB. The core of the system is a MongoDB database. MongoDB is suitable for storing data such as tweets in JSON format. The choice of this type of database is supported by the fact that MongoDB supports geo-spatial and temporal indices. Another important feature is that it can easily scale in a number of instances. It is a NoSQL open-source database capable of storing massive volumes of data and supporting effective data integration in dynamic formats such as JSON. Developing in Python enables to implement of a library for MongoDB called pymongo. The middleware between Twitter Streaming API and MongoDB is Jupyter. It is a tool for interactive Python development, which serves in this project for Twitter data analyses.



```
import [illegible].followers as followers,[illegible].tweetAnalysis,[illegible].tweet_pre as tweet_pre
import tweepy,json #https://github.com/tweepy/tweepy
from tweepy import  Cursor
from tweepy import OAuthHandler
from tweepy import API
from tweepy import Cursor
from bson.json_util import dumps
import json
import pandas as pd
import matplotlib as mpl
#mpl.use('Agg')
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator
from matplotlib import rcParams,style
#from mpltools import style
from matplotlib import dates
from datetime import datetime
#import seaborn as sns
import numpy as np,os
from pymongo import MongoClient
from PySide import QtGui
from os import*
```

Figure 3.12: Mongo connection

3.5.7 Tweets Attributes

The anatomy of the tweets stored in the database can be analyzed from the Jupyter. Following are some interesting attributes that contain each tweet:

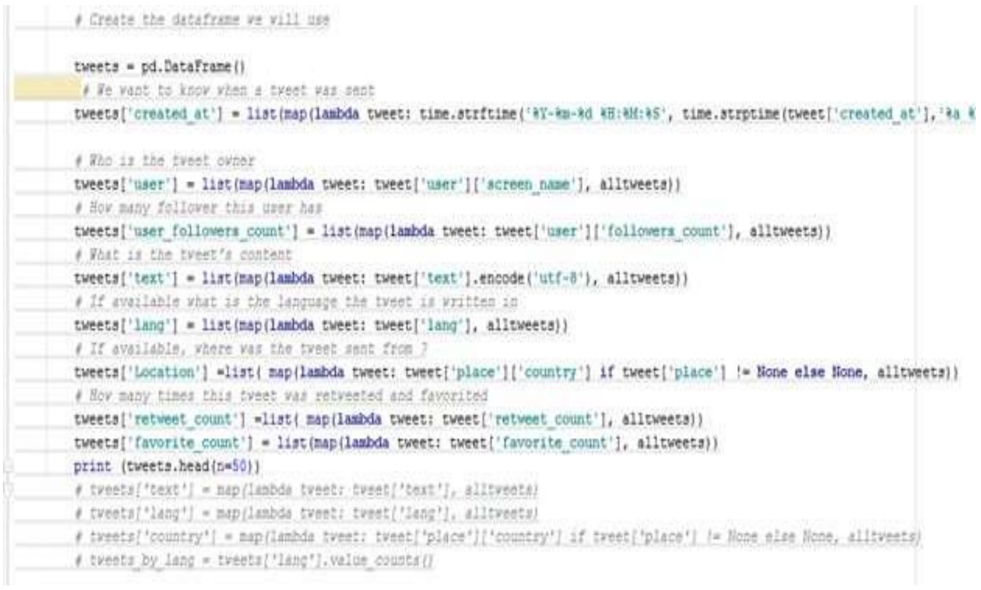
Table 3.1: Main Attributes of Data Set.

Attribute Name	Attribute Description
User ID	The unique ID assigned by Twitter to each user
Created_at	Date and time on which tweet as posted
Screen_name	Account name displayed on twitter for each user
Followers_count	Number of other users following this account
Retweets_count	Total No of retweets posted for this tweet
Mentions	Total No of mentions of this tweet
Following_count	No of the Twitter accounts a user is following
Text	Text/content of the tweet
Lang	The language used in the tweet
entities	Url, hashtags#

From the attributes overview, we can implement different data mining algorithms to perform processing, such as the most favorite tweets compared to the user's profile or analyses towards most used hashtags. However, for the sentiment analyses and classification with machine learning, text attribute matter the most. Based on this, tweet text will perform algorithm implicitly decisions if it contains positive or negative statements. Collecting tweets from Streaming API are stored in the database; however, there is the capability to view near real-time streaming output in the text file used for pre-processing methodology.

3.5.8 Creating a data frame

In data analyses, an important role is a lead towards data selection based on data structures. Effective, organized data into tables gives a better overview of data scientists about its structures. In our tweet data analysis model, we used Data Frames enabled with Python Pandas. The Data frame is capable of storing data into tables and label its rows and columns. Hence, data mining can process data-driven solutions and deliver meaningful insights. Tweets that are stored in the database are organized into Data Frames, As indicated in figure 3.12.



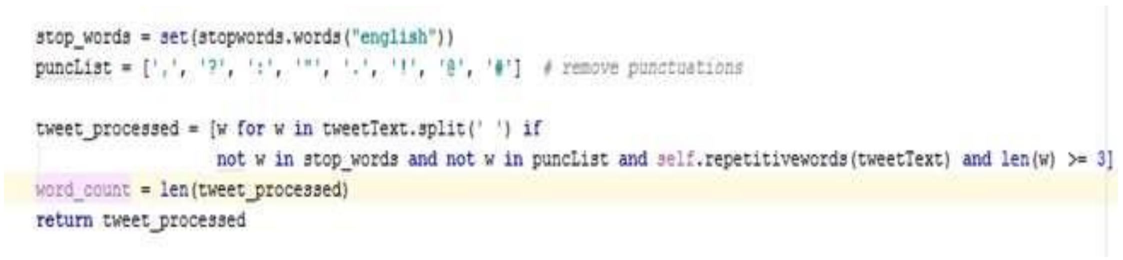
```
# Create the dataframe we will use

tweets = pd.DataFrame()
 # We want to know when a tweet was sent
tweets['created_at'] = list(map(lambda tweet: time.strftime('%Y-%m-%d %H:%M:%S', time.strptime(tweet['created_at'],'%a %

# Who is the tweet owner
tweets['user'] = list(map(lambda tweet: tweet['user']['screen_name'], alltweets))
# How many follower this user has
tweets['user_followers_count'] = list(map(lambda tweet: tweet['user']['followers_count'], alltweets))
# What is the tweet's content
tweets['text'] = list(map(lambda tweet: tweet['text'].encode('utf-8'), alltweets))
# If available what is the language the tweet is written in
tweets['lang'] = list(map(lambda tweet: tweet['lang'], alltweets))
# If available, where was the tweet sent from ?
tweets['Location'] =list( map(lambda tweet: tweet['place']['country'] if tweet['place'] != None else None, alltweets))
# How many times this tweet was retweeted and favorited
tweets['retweet_count'] =list( map(lambda tweet: tweet['retweet_count'], alltweets))
tweets['favorite_count'] = list(map(lambda tweet: tweet['favorite_count'], alltweets))
print (tweets.head(n=50))
# tweets['text'] = map(lambda tweet: tweet['text'], alltweets)
# tweets['lang'] = map(lambda tweet: tweet['lang'], alltweets)
# tweets['country'] = map(lambda tweet: tweet['place']['country'] if tweet['place'] != None else None, alltweets)
# tweets_by_lang = tweets['lang'].value_counts()
```

Figure 3.13: Organizing tweets into Data Frame.

3.5.9 Tweets Pre-processing

Natural Language Tool Kit called NLTK is a platform for building Python programs to work with text data. It provides text processing libraries for tokenization, stemming, tagging, parsing, and classification. NLTK libraries are downloaded and integrated with the Pycharm IDE. The NLTK library is imported to perform the tokenization and parts of speech tagging to achieve user tweets' data preprocessing. The code used to achieve preprocessing is shown in figure 3.13.

```
from nltk.corpus import stopwords

stopwords.words('English')
```



```python
stop_words = set(stopwords.words("english"))
puncList = [',', '?', ':', '"', '.', '!', '@', '#']  # remove punctuations

tweet_processed = [w for w in tweetText.split(' ') if
                   not w in stop_words and not w in puncList and self.repetitivewords(tweetText) and len(w) >= 3]
word_count = len(tweet_processed)
return tweet_processed
```

Figure 3.14: Python code for removing stop words.

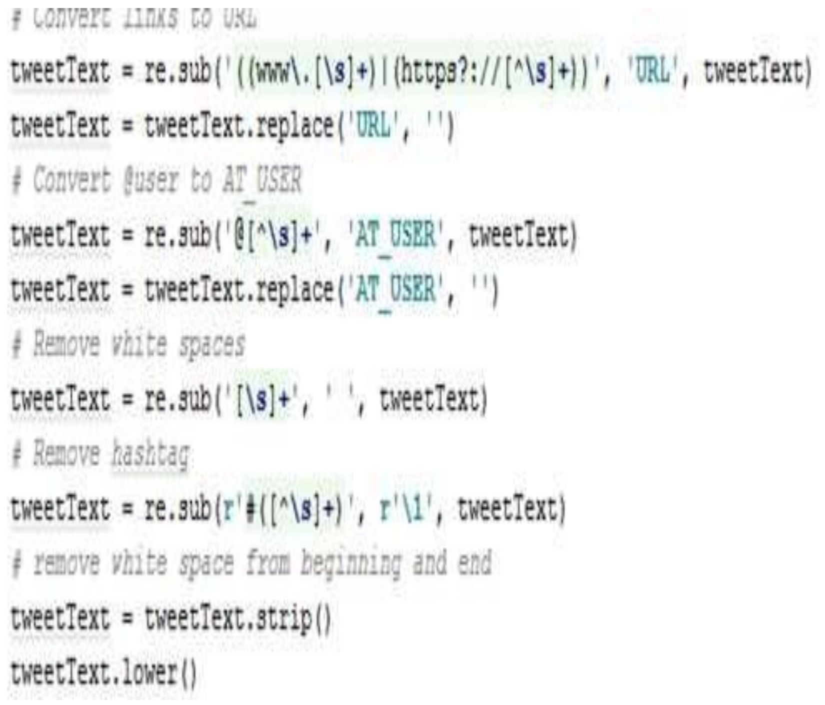
Table 3.2: List of Stopwords of NLTK

Stopwords	‘I’, ‘you (singular), thou’, ‘he’, ‘we', 'you (plural)', 'they', 'this', 'that', 'here', 'there', 'who', 'what', 'where', 'when', 'how', 'not', 'all', 'many', 'some’, ‘few', 'other', 'one', 'two', 'three', 'four', 'five', 'big', 'long', 'wide'

3.5.10 Removing Twitter Symbols

We also found that some features could affect the experiment’s result, which included “http://t.co/kk,” “@,” or “#” in Table 3 so that we have to remove these as well. Many Tweets contain non-alphabetic symbols such as “@” or “#” and active weblinks. The word immediately following the “@” symbol indicates a username, which we filter out entirely. The username symbol is deemed to add no sentiment value to the text but could prove instrumental in performing user activities analysis. Words following “#,” known as the hashtag, are also removed, even if text connected to the hashtag contains information used for categorization. This experiment's focus is a textual analysis, and the hashtag is assumed not to contribute to the text of an individual message. URLs are filtered out entirely, as they add no sentiment value to the text. To eliminate non-alphabetic symbols, we used a regex that matches these symbols. Additionally, any non-word symbols in the bag of words are filtered out as well. Implementing Python code, as shown in figure 3.14

and subsequently, the sampled examples of Tweets cleaned for non-alphabetic symbols, are available in Table 3.3.



```
# Convert links to URL
tweetText = re.sub('((www\.[\s]+)|(https?://[^\s]+))', 'URL', tweetText)
tweetText = tweetText.replace('URL', '')
# Convert @user to AT_USER
tweetText = re.sub('@[^\s]+', 'AT_USER', tweetText)
tweetText = tweetText.replace('AT_USER', '')
# Remove white spaces
tweetText = re.sub('[\s]+', ' ', tweetText)
# Remove hashtag
tweetText = re.sub(r'#([^\s]+)', r'\1', tweetText)
# remove white space from beginning and end
tweetText = tweetText.strip()
tweetText.lower()
```

Figure 3.15: Python Code for Removing Tweet Symbols

Table 3.3: Example of Text after Removing Symbols.

Number of tweets	**Text after removing the symbols**
1	HAS NEVER missed a FUNERALHe LOVES them BUT addressing HOSTILE Kikuyus TODAY was more IMPORTANT than Nkaissery WHY
2	infected NASA with the SLEEPING SICKNESS. They SLEPT through VOTER REGISTRATION
3	No to FAKE Hacking
4	bTo call the MEDIA Watu Wakufunga NYAMA is WRONG It INSULTS the Intelligence of BUTCHERS Our MEDIA falls BELOW the level of NON-SENSE
5	bA WICKED Man will run when NO one is chasing him RIGHTEOUS MAN is as Bold as a LIONBring the STUFF On

3.5.11 Tokenization

This process involves splitting the text by spaces, forming a list of individual words per text. This is also called a bag of words. Tokenizing the text makes it easy to separate other unnecessary symbols and punctuations and alter only those words that add value to the text's sentimental polarity score. The tweet's tokens are matched against three different sentiment lexicons: A version of the data without these replacements is also maintained for comparison purposes. Later the tokens will be used to form a feature extraction approach to train the classifier. The research used the “word_tokenize” method from the NLTK library to process tokenization, and sampled examples are shown in table 3.4.

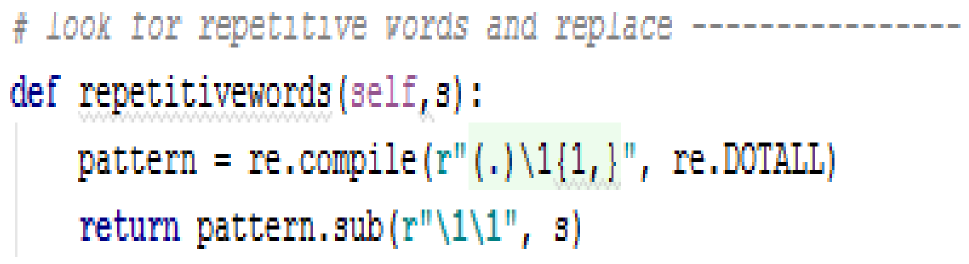
```
from nltk import word_tokenize

 tokens = word_tokenize(raw_data)
```

`Table 3.4: Example of Tokenized Text after Removing Irrelevant Information from Original Tweets

Number of tweets	**Tokenized Text/Status after remove other irrelevant information from original Tweets**
1	['luos', 'fear', 'fight', 'kikuyus?why', 'stupids?when', 'luos', 'change', 'stone-throwing', 'fight', 'guns', 'modern', 'weapons?']
2	['luo', 'nation', 'spell', 'bound', 'raila.', 'period.']
3	['jaluo', 'zote', 'ziko', 'bonde', 'ufa.', 'zitoke', 'polepole.', 'coz', 'lazima', 'mtahiri', 'mabati', 'ata', 'mkose', 'kuanza', 'fujo.', 'iyo', 'kitu', 'imepangwa', 'vizuri.']
4	['mabati', 'ata', 'mkose', 'kuanza', 'fujo.', 'iyo', 'kitu', 'imepangwa', 'vizuri.']
5	['mama', 'yako”', "mother's"]
6	['happy', 'government', 'dogs.', 'waiting', 'killed', 'do.', 'islam', 'terrorist', 'religion']
7	['swoop', 'nairobi', 'kumetulia.', 'kudos', 'gava', 'protecting', 'real', 'kenyans', 'threat', 'refugee', 'terrorists.']
8	['f***', 'matapaka', 'hell', 'alshabaab']
9	['examples', 'online', 'vitriol', 'include', 'calls', "'chinja', 'chinja",', "'butcher', 'butcher"', 'swahili,', 'well', 'beat,', 'loot,', 'riot,', 'kill,', 'drive', 'tribes.']
10	['fact', 'baba', 'nyayo.', 'mental', 'slavery.']

3.5.12 other tools

Removal retweets and duplicates. A retweet is a tweet written by one user and then copied and posted by another user. Retweet contains the “RT” abbreviation. Repeated tweets and retweets are removed (refer to figure 3.15) to exclude putting extra weight on a specific tweet.



```
# look for repetitive words and replace ----------------
def repetitivewords(self,s):
    pattern = re.compile(r"(.)\1{1,}", re.DOTALL)
    return pattern.sub(r"\1\1", s)
```

Figure 3.16: Python code for removal of retweets/duplicate

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.

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 only argument. This is considered as one of the initial pre-processing stages in NLP. The POS Tagging task involves identifying the right part of speech for each token and tagging it accordingly. Tagging on English texts has been carried out extensively using many machine learning techniques. The tagger program was altered to use algorithms from the scikit learn package –NBclassifier, Maximum Entropy Classifier. Finally, the same corpus was tagged by the built-in Perceptron Tagger available in NLTK. 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

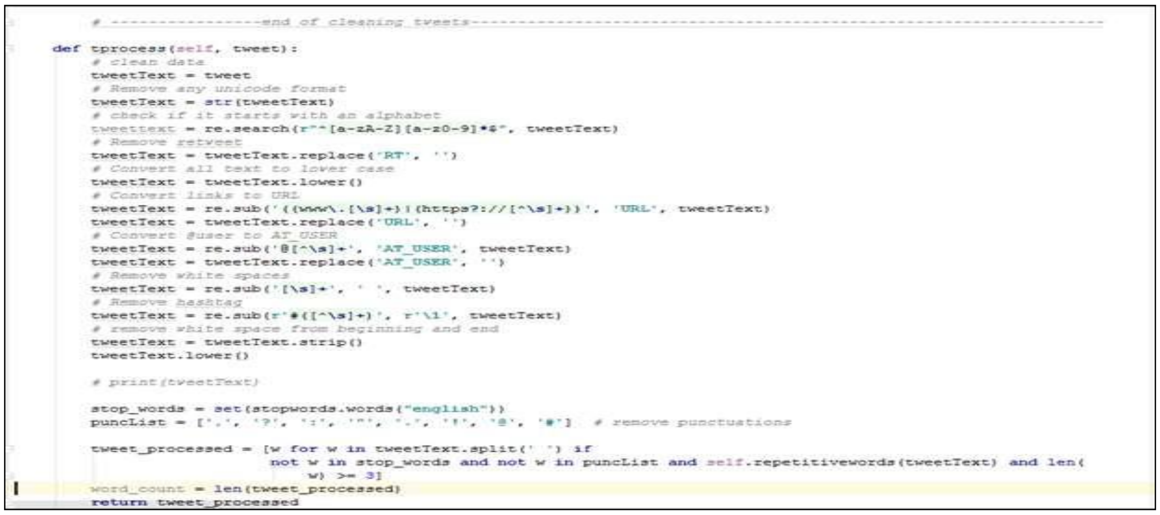
1. Lemmatization and Stemming: To homogenize verbs being in different tenses, we implemented a method that applies lemmatization and stemming techniques to the words of the input tweets. Lemmatization is often used in computational linguistics problems. It is a process that determines the lemma of a word. In English, a word can have different inflected forms. For instance, the word ’walk’ can be used as ’walked,’’ walking,’’ walks.’ The base form of those words is ’walk.’ This base form of the words is called lemma
2. Lowercase Conversion: Tweet may be normalized by converting it to lowercase, making the comparison with an English dictionary easier. Finally, the users mentioned in the tweet, which is marked with “@,” are replaced with “PERSON,” and the topics which the tweet refers to (marked with “#”) are replaced with “TOPIC.”
3. Remove additional white space-Twitter users make many mistakes while writing tweets. Removal of additional white space means the removal of multiple white spaces with a single white space. Data is presented in a single stream of lines without any extra white space, and every word is differentiated from others.
4. 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, this five-gram modeling process's outcome can be seen below.

[‘I,’ ‘really,’ ‘hate,’ ‘the,’ ‘new,’ ‘Firefox’]

)

[((‘I,’ ‘PRP’), (‘really,’ ‘RB’), (‘hate,’ ‘JJ’), (‘the,’ ‘DT’), (‘Luo,’ ‘JJ’)),

1. Removal retweets and duplicates. A retweet is a tweet written by one user and then copied and posted by another user. Retweet contains the “RT” abbreviation. Repeated tweets and retweets are removed to exclude putting extra weight on a specific tweet.
2. Removal of usernames. Post creator can mention another user in the tweet by using the “@” symbol followed by a username, i.e., @Superman. Due to this feature does not provide any relevant information, it was also excluded from the tweet.
3. Removal of hashtags. The hashtag is depicted using the “#” symbol and used before representing a topic name. Topics are not the task to be classified; hence they are omitted.
4. 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 classifying a document. Since the query term itself should not be used to determine the sentiment of the post concerning it, every query term is replaced with a QUERY keyword.
5. Lowercase Conversion: Tweet may be normalized by converting it to lowercase, making it a comparison with an English dictionary.
6. Finally, the users mentioned in the tweet, which is marked with “@,” are replaced with “PERSON,” and the topics which the tweet refers to (marked with “#”) are replaced with “TOPIC.”Tweet preprocessing process, as indicated in figure 3.16.



```
    # ----------------end of cleaning tweets------------------------------------------------------------------------

def tprocess(self, tweet):
    # clean data
    tweetText = tweet
    # Remove any unicode format
    tweetText = str(tweetText)
    # check if it starts with an alphabet
    tweettext = re.search(r"^[a-zA-Z][a-z0-9]*$", tweetText)
    # Remove retweet
    tweetText = tweetText.replace('RT', '')
    # Convert all text to lower case
    tweetText = tweetText.lower()
    # Convert links to URL
    tweetText = re.sub('((www\.[\s]+)|(https?://[^\s]+))', 'URL', tweetText)
    tweetText = tweetText.replace('URL', '')
    # Convert @user to AT_USER
    tweetText = re.sub('@[^\s]+', 'AT_USER', tweetText)
    tweetText = tweetText.replace('AT_USER', '')
    # Remove white spaces
    tweetText = re.sub('[\s]+', ' ', tweetText)
    # Remove hashtag
    tweetText = re.sub(r'#([^\s]+)', r'\1', tweetText)
    # remove white space from beginning and end
    tweetText = tweetText.strip()
    tweetText.lower()

    # print(tweetText)

    stop_words = set(stopwords.words("english"))
    puncList = ['.', '?', ':', '"', '.', '!', '@', '#']  # remove punctuations

    tweet_processed = [w for w in tweetText.split(' ') if
                       not w in stop_words and not w in puncList and self.repetitivewords(tweetText) and len(
                           w) >= 3]
    word_count = len(tweet_processed)
    return tweet_processed
```

Figure 3.17: Tweet preprocessing python code.

3.6 Implementation of Classification classifiers.

We first applied various popular and powerful supervised classification algorithms to the data, namely—Naive Bayes, Support Vector Machines (SVMs), and Maximum Entropy (ME). We used the Python implementations found in the Natural Language Toolkit (NLTK) and Scikit-Learn. Twitter API, Tweepy, was implemented to collect real data set from Twitter public available information. The collected tweets formed a training data set and test data. In our case, Kenya, the geo-coordinates of the selected country, were obtained using Google's geocoding API. The extracted geo-coordinates were fed to Twitter's Search API to find the users who have reported their location (Kenya) in their Twitter profiles. We created an account on Tweepy API linked to my Twitter account. To retrieve the tweets, Tweepy API accepted parameters and provided the Twitter account's data in return. From Twitter accounts, retrieved tweets were saved in the Mongo DB in the following fields: twitter_id, hashtag, tweet created, user_id, screen name, tweet text, retweet count, follower count, and, and favourite_count of each tweet.

For this study, the prototype was developed and implemented using python for both logic application and interfacing. MongoDB was used as the back end for storing application data and the data fetched. Scikit-Learn, a python module integrating classic machine learning algorithms in the tightly-knit scientific python, was used to handle the classification of hate tweets development and evaluation. It the part of the prototype that handles data scraped from social media platforms. MongoDB is utilized. It is a NoSQL document database. This makes it handle unstructured data from the Twitter platform is appropriate. It stores a JSON-style document, which is basically how python handles the data. In case this prototype was used in a distributed environment, then MongoDB provided a method for storing data across multiple machines. The automatic classification of tweets begins with the manual classification of a dataset, which serves as the ground truth for evaluating the classifying algorithms' performance. Hate tweets were detected and classified using the content and user-based attributes.

This thesis classified the hate tweets/speech user twitter data using Twitter API, text mining, and classifiers for tweets classification. The collected data set was stored in Mongo Database. Textual description and Machine learning algorithms on the dictionary for classifying twitter hate dataset. A novel idea of utilizing a TF-TD score of real-time tweets data set and feature selection method to create a dictionary of critical terms and use it to predict hate speech. For this purpose, we used text mining algorithms for preprocessing steps. Preprocessing of tweets is done using steps such as tokenization, stop word removal, and stemming.

Further TF-IDF score was calculated for each term. Then two feature selection methods are used for creating a dictionary of terms. Therefore, Machine learning algorithms such as Naive Bayes algorithms were used as classifiers.

From the collected dataset, tweets were randomly selected. Annotators initially classified the randomly selected tweets as to whether they are encoded in English, a combination of English, or other languages or dialects. Pre-processing procedures were implemented in python, such as tokenization, stemming, and stop word removal were applied to the corpus to create the word vector. The bag of words technique was used to produce the main features in the word vector. The word vector was generated using the binary term occurrence as weights for the features, and the generated word vector was used as the training dataset for machine learning. Supervised learning was used in training the machine to classify hate tweets and identify the propagators.

3.6.1 Implementation of Naïve Bayes in NLP toolkit

A naïve Bayes classifier is important for classification. Naive Bayes Classifier NLTK kit was installed before using this approach. The classifiers have been built by training them on the training set. The naïve Bayes classifier in the NLP toolkit is used ***from*** *nltk. Classify. Naive Bayes* ***import*** *NaiveBayesClassifier.*

```
    NaiveBayesClassifier (training_set) testTweet='Sentimental analysis'


    ProcessedTestTweet=processTweet(testtweet)

  Print                                         NBClassifier.classify
  (extract_features(getFeatureVector(proccessedTestTweet)))
```

Naïve Bayes classifier accepts the training set. In the NLP tool kit, the Naïve Bayes classifier was used directly to import a program for implementation. Test tweets were processed by the function ProcessedTestTweet. Naïve Bayes classifier extracted the features of processed tweets and then classified them.

3.6.2 Implementation of Maximum Entropy in NLP toolkit

Within NLTK, the Maximum Entropy training algorithms. The first two were implemented in NLTK by Python but seemed very slow and costs large memory for the same training data.

Maximum Entropy Classifier Algorithm

```
MaxEntClassifier=nltk.classify.maxent.Maxentclassifier.train(trainset,algorithm="gis")
testTweet='Sentimental Analysis'
```

```
processedtesttweet=processTweet(testTweet)

print
    MaxEntClassifier.classify(extract_feature(getFeatureVector(processedTestTweet))
)
```

The processed Tweet function was used to process the tweets. After training the data set, the classifier accepted the training data and classified the sentiment words.

3.6.3 Implementation of SVM in NLP Toolkit

After training the data set, an SVM classifier was used in the NLP toolkit. SVM used

nltk.classify.sci-kit learn. It consists of class nltk. Classify.SVM.SvmClassifier(training set).

SVM Classifier Algorithm

```
  Classifier=
        nltk.classify.svm.SvmClassifier.train(train_set)
for features in test_data_list:

  print                                       predict=
        classifier.classify(extract_features(features))
```

SVM classifier is fast efficient as compared to others. It accepted the training data and extracted the features, and then built the classifier in a statement.

3.7 Feature Extraction

We evaluated a given classifier's performance when using all our features and then again after removing each one of these features. The difference in the performance is used as a measure of the importance of the feature. We chose to use the difference in the F_1 metric over F_2 in this analysis because we wanted to convey how the features performed in the general task of tweet classification. We also performed some analysis on the word (i.e., *n*-gram) features to learn which words in our vocabulary were the best indicators of relevant tweets. We analyzed the *n*-gram component of our compound feature vectors to calculate each word's information unigram.

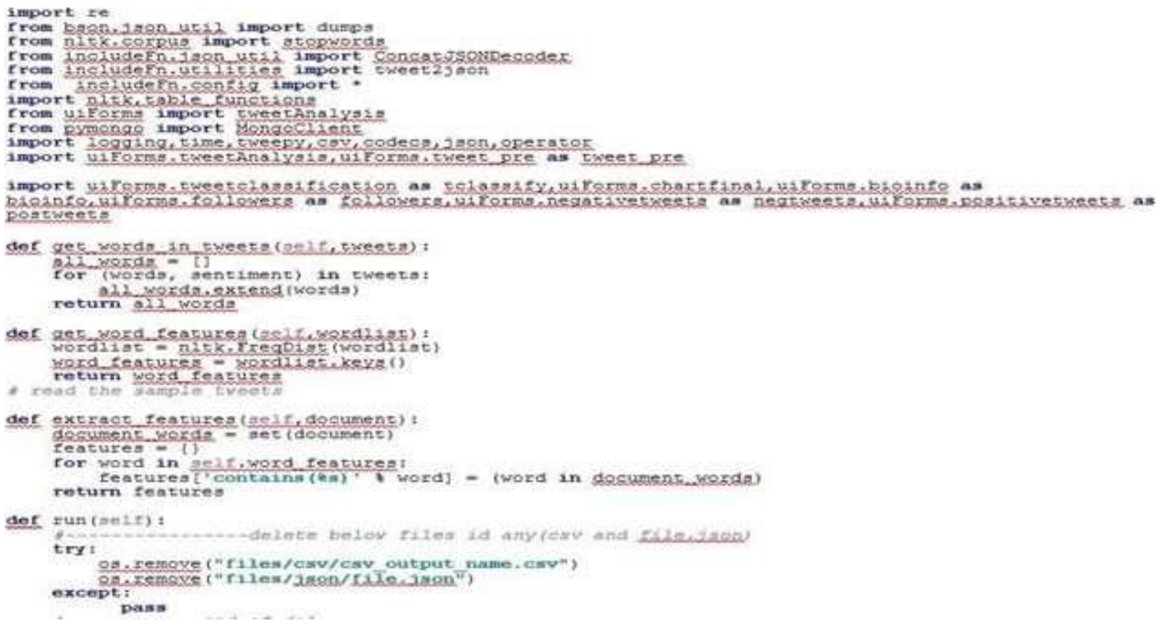
Each feature pair's information gain is based on the prior probability of the feature pair occurring for each class label. A higher information gain (hence, a more informative feature) is a feature that occurs primarily in one class and not in the other. Similarly, less informative features are features that occur evenly in both classes. The information gain idea is pivotal to the decision tree algorithm but generalizes to others and was adapted in the NLTK package for use in a broader sense.

> In NLTK, informativeness of a word w was calculated as the highest value of $P(w = feature\_value|class)$ for any class, divided by the lowest value of $P(w = feature\_value|class)$ (Hardeniya, 2015). This informativeness I is summarized below:
>
> $$I = \frac{\forall c € C\ ;\max\ (p(feature = feature_{value\backslash c}))}{\forall c € C\ ;\min\ (p(feature = feature_{value\backslash c}))}$$
>
> (3.1)

Recall that to collect tweets; we used Twitter’s streaming API to specify keywords to restrict the data collection to tweets containing those specific terms. We measured the usefulness of the keywords we selected. To do this, we assessed their information retrieval performance. Specifically, we used the precision-recall metric. In an information retrieval context, precision and recall are defined according to a set of retrieved documents and their relevance. We use our original set of labeled tweets for this assessment (i.e., the set of 45,645 tweets).

In our scenario, the labeled tweets make up the set of retrieved documents, and the tweets labeled as belonging to the “relevant” class make up the set of relevant documents. In this context, recall measures the fraction of relevant tweets that are successfully retrieved, while precision measures the fraction of retrieved tweets relevant to the query. Training and testing data is collected from the NLTK corpus. We have around 68,465 tweet datasets each for positive and negative classes. We take the first 45,645 datasets as a training set and the remaining 22,820 as testing sets. Both the training and testing data must be represented in the same order for learning. One of the ways that data can be represented is feature-based.

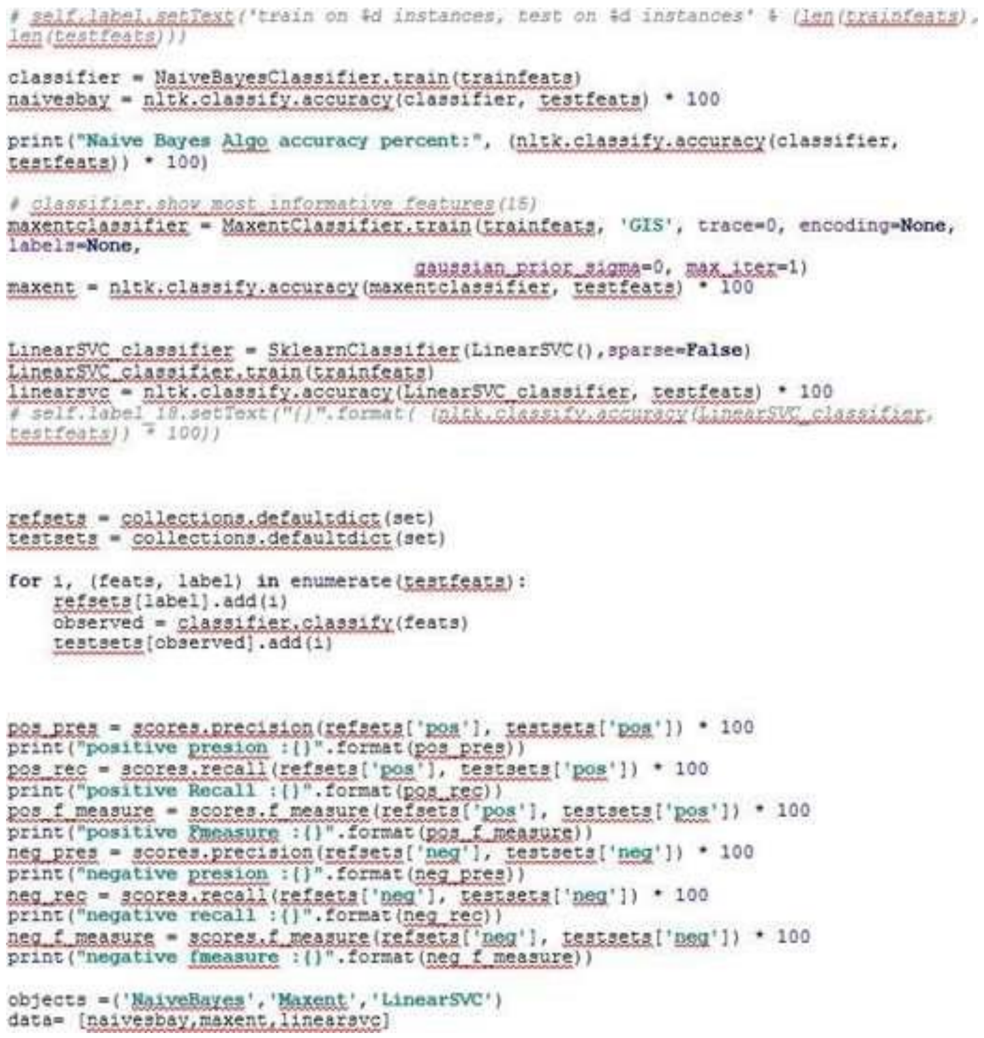
By features, it is meant that some attributes that are thought to capture the pattern of the data are first selected, and the entire dataset must be represented in terms of them before it is fed to a machine-learning algorithm. Attribute selection is the process of extracting features by which the data will be represented before any machine learning training takes place. Attribute selection is the first task when one intends to represent instances for machine learning. Once the attributes are selected, the data will be represented using the attributes. Therefore, attributes are features. Although we used the entire data set in our selection of attributes, the data representation must be done on a per instance (Twitter post) basis. We created a python script to extract the features from the training data. The code snippet for extracting features is shown in Figure 3.17.



```
import re
from bson.json_util import dumps
from nltk.corpus import stopwords
from includeFn.json_util import ConcatJSONDecoder
from includeFn.utilities import tweet2json
from  includeFn.config import *
import nltk,table_functions
from uiForms import tweetAnalysis
from pymongo import MongoClient
import logging,time,tweepy,csv,codecs,json,operator
import uiForms.tweetAnalysis,uiForms.tweet_pre as tweet_pre

import uiForms.tweetclassification as tclassify,uiForms.chartfinal,uiForms.bioinfo as
bioinfo,uiForms.followers as followers,uiForms.negativetweets as negtweets,uiForms.positivetweets as
postweets

def get_words_in_tweets(self,tweets):
    all_words = []
    for (words, sentiment) in tweets:
        all_words.extend(words)
    return all_words

def get_word_features(self,wordlist):
    wordlist = nltk.FreqDist(wordlist)
    word_features = wordlist.keys()
    return word_features
# read the sample tweets

def extract_features(self,document):
    document_words = set(document)
    features = {}
    for word in self.word_features:
        features['contains(%s)' % word] = (word in document_words)
    return features

def run(self):
    #-----------------delete below files id any(csv and file.json)
    try:
        os.remove("files/csv/csv_output_name.csv")
        os.remove("files/json/file.json")
    except:
            pass
```

Figure 3.18: Code for extracting features from tweet

Once we extract the features from training data, we will pass these in our build classifiers. A script is written in python, which is used to pass training sets in classifiers. Once the classifier is trained, we can also check each classifier's accuracy bypassing the testing set. A sample script of training and testing of the classifier is shown in Figure 3.1.8



```
# self.label.setText('train on %d instances, test on %d instances' % (len(trainfeats),
len(testfeats)))

classifier = NaiveBayesClassifier.train(trainfeats)
naivesbay = nltk.classify.accuracy(classifier, testfeats) * 100

print("Naive Bayes Algo accuracy percent:", (nltk.classify.accuracy(classifier,
testfeats)) * 100)

# classifier.show_most_informative_features(15)
maxentclassifier = MaxentClassifier.train(trainfeats, 'GIS', trace=0, encoding=None,
labels=None,
                                          gaussian_prior_sigma=0, max_iter=1)
maxent = nltk.classify.accuracy(maxentclassifier, testfeats) * 100


LinearSVC_classifier = SklearnClassifier(LinearSVC(),sparse=False)
LinearSVC_classifier.train(trainfeats)
linearsvc = nltk.classify.accuracy(LinearSVC_classifier, testfeats) * 100
# self.label_18.setText("{}".format( (nltk.classify.accuracy(LinearSVC_classifier,
testfeats)) * 100))


refsets = collections.defaultdict(set)
testsets = collections.defaultdict(set)

for i, (feats, label) in enumerate(testfeats):
    refsets[label].add(i)
    observed = classifier.classify(feats)
    testsets[observed].add(i)


pos_pres = scores.precision(refsets['pos'], testsets['pos']) * 100
print("positive presion :{}".format(pos_pres))
pos_rec = scores.recall(refsets['pos'], testsets['pos']) * 100
print("positive Recall :{}".format(pos_rec))
pos_f_measure = scores.f_measure(refsets['pos'], testsets['pos']) * 100
print("positive Fmeasure :{}".format(pos_f_measure))
neg_pres = scores.precision(refsets['neg'], testsets['neg']) * 100
print("negative presion :{}".format(neg_pres))
neg_rec = scores.recall(refsets['neg'], testsets['neg']) * 100
print("negative recall :{}".format(neg_rec))
neg_f_measure = scores.f_measure(refsets['neg'], testsets['neg']) * 100
print("negative fmeasure :{}".format(neg_f_measure))

objects =('NaiveBayes','Maxent','LinearSVC')
data= [naivesbay,maxent,linearsvc]
```

Figure 3.19: Sample code for training and testing build classifier.

3.8 Experimental Setup

A few steps did data collection, do login twitter, register on API Twitter to get the access token, and then create scripts for crawling data and input access token obtained in API twitter, then save the log data database the form of JSON files. Second, doing the analysis preprocessing and data cleansing with the method described previously to get structured data. Third, classification was done using various classifiers, including Naïve Bayes, Maximum entropy Support Vector Machine, and manual classification, which was performed on the data that has been cleaned. Since our model, it has been implemented on Pycharm IDE using Python as the programming language and Mongo DB for the database. Python provides a popular machine learning library, scikit learn, which provides a handful of algorithms and resources for data scientists. The data in this paper was collected using Twitter API (Tweepy). The API helps to retrieves tweets for any user or hashtags from the Twitter platform

With the API's help, the data containing hashtag hate tweets are collected and saved in the Mongo database. The data collected consisted of 45000 raw tweets in JSON format. This data contains hashtag, tweet_created, user_id, screen_name Tweet text, Tweet ID, date, retweet_count, follower_count, and favourite_count of each tweet 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, the user's location, and whether the user is verified or not. All these functionalities were achieved through a GUI interface. 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 the constraint of the python libraries and codes used during the Data preprocessing step.

3.9 Training Data

Other data that was collected for this thesis was training data. This data was used to train the classifier. To collect this data, we used the NLTK library of Python. NLTK consists of corpora, which is very large and consists of a structured set of text files used to perform analysis. There are various types of text files in these corpora, like quotes, reviews, chat, history, etc. From these corpora, we will select files of tweets for our training purpose. To precisely label the text into their respective classes and thus achieve the highest possible accuracy, we trained the classier using pre-labeled Twitter data it-self. The training data has 45645 total tweets, and the test data has 22820 total tweets. This training data was obtained from sample sentences, and word from a text file classified manually. They were loaded to the classifier for training. NLTK library was used partially to preprocess and classify the tweets to the training set using Naive Bayes. In addition, the sample data was preprocessed before it is loaded to the classifier. The Test data was fetched from Twitter using tweeter API. These data is loaded to the trained classifier for sentiment analysis as shown in table 3.5

Table 3.5: Sample tweets dataset in NLTK Corpus

Tweets	CLASS
I don't think am the only Luo who is against......Most educated and enlightened luos are against this mongrel, too... they know his devilish plans of purportedly killing his supporters for blood to feed his jochiende!!!! Ask	NEGATIVE
the rock is destined to be the 21st century's new Conan, and that he's going to make a splash even greater than Arnold Schwarzenegger	POSITIVE
the beginning of the downfall is now...then can be deported with his Kikuyu concubine	NEGATIVE
the seaside splendor and shallow, beautiful people are nice to look at while you wait for the story to get going	POSITIVE
Most Kisii'z are either thieves or conmen. ....munatua bisha sana	NEGATIVE
How Prostitutes In Parliament Cleared St. Mary’s Alumni & Petroleum PS Andrew Kamau	NEGATIVE

Source: Research Data.

3.9.1 Development Technologies

Research on the development technologies used for Twitter sentiment analyses implement- tation is discussed within this section.

1. **Python-**Python is a programming language developed under an open-source license. It is an interpreted language applied within various development tasks, such as web applications, data analyses, and data visualizations.
2. **SQL-**There were many storage options such as a comma-separated values (CSV)

file, a text file, or a database. SQLite was used as the database management system as it is an open-source application that is easily integrated with the Python programming language Scikit-learn.

3. **Natural Language Processing-**Natural Language Tool Kit (NLTK) is an open-source library developed in Python. This technology contains the necessary tools for text processing and sentiment analyses on Twitter. Text processing belongs to the pre-processing data analysis phase in methodology because text as an attribute in a tweet has to be processed into a suitable form before actual data mining.
4. **Scikit-learn -**Scikit-learn is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. This package focuses on bringing machine learning to non-specialists using a general-purpose, high-level language. The requirements that we'll need to install several libraries;
5. **NumP**y: This is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.
6. **Pandas**: This is an open-source library providing high-performance, easy-to-use data structures, and data analysis tools.
7. **Tweepy**: This is an easy-to-use Python library for accessing the Twitter API.
8. **Matplotlib**: This is a Python 2D plotting library that produces publication quality figures in various hardcopy formats and interactive environments across platforms.
9. **Seaborn**: This is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.
10. **SentiWordNet-** SentiWordNet is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity.

3.9.2 Presentation of Results.

This is the side that provides for the user interface. It is built on the Django framework.

Django is a high-level python web framework that encourages rapid development and clean, pragmatic design. It is created inside a python virtual environment containing all the required dependency, including NLTK and scikit-learn. The general purpose of the analysis is to transfigure the harvested data into meaningful and useful information. Once the analysis is completed, many conventional options are used to display the result of text analysis (Rambocas, 2013). This study utilizes a graphical presentation in a web-based tool.

CHAPTER FOUR

EXPERIMENTS, ANALYSIS, AND RESULTS

4.1 Introduction

This chapter deals with a discussion of the results and findings described in the previous chapter.

4.2 Naive Bayes Results

Approximately 66.7% of the tweets were retweets (as shown in table 4.1), so many of the activities around the event shown in figure 4.2 are based on re-posting information for followers. This makes sense in a news-based event.

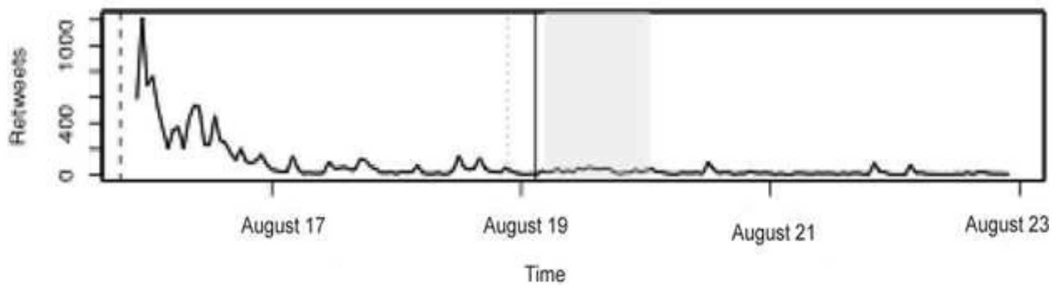


Figure 4.1: Retweet analysis for a selected period

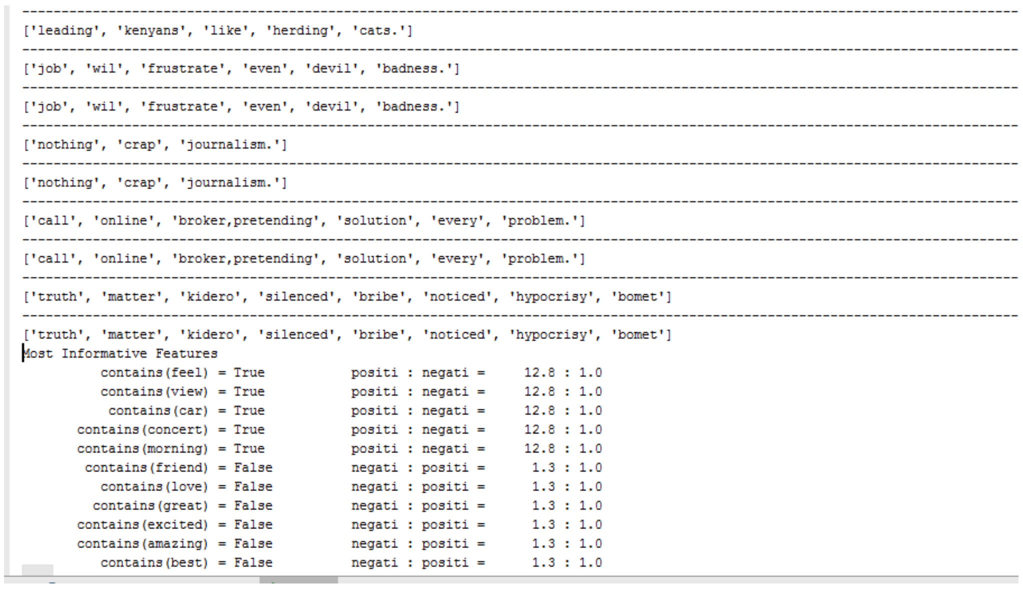
4.3 Quantitative Characteristics of Twitter Data

Table 4.1 Tweets Statistics

Type of tweet	No. of Tweets	Relative
Tweets	65,043	
No-retweets	28,175	33.3%
Retweets	36,868	66.7%
Mentions	56,713	75.24%
Replies	9,732	52.64%
Users	45,000	

Source: Research Data.

A brief overview of the quantitative characteristics of this dataset is shown in Table 4.1. Approximately 57% of the tweets were retweets. Implying Twitter users were disseminating most of the hate messages posted by specific user(s). There is an unusually large number of mentions, but this could be explained by the fact that mentions were tracked, so most tweets mention the tracked accounts. Just 14% of these mentions are replies, so most tweets mentioned users, not receiving a reply. In addition, explaining either the replies came from those who felt the hate speech target them positively or negatively. The extraction of tweets is a process that takes the feature vector, and it checks the presence of words in the feature vector of the given tweet and present in a feature list formed by combining the feature vectors of the tweets. The result of feature extraction is described in Figure 4.2



```
----------------------------------------------------------------------------------------------------
['leading', 'kenyans', 'like', 'herding', 'cats.']
----------------------------------------------------------------------------------------------------
['job', 'wil', 'frustrate', 'even', 'devil', 'badness.']
----------------------------------------------------------------------------------------------------
['job', 'wil', 'frustrate', 'even', 'devil', 'badness.']
----------------------------------------------------------------------------------------------------
['nothing', 'crap', 'journalism.']
----------------------------------------------------------------------------------------------------
['nothing', 'crap', 'journalism.']
----------------------------------------------------------------------------------------------------
['call', 'online', 'broker,pretending', 'solution', 'every', 'problem.']
----------------------------------------------------------------------------------------------------
['call', 'online', 'broker,pretending', 'solution', 'every', 'problem.']
----------------------------------------------------------------------------------------------------
['truth', 'matter', 'kidero', 'silenced', 'bribe', 'noticed', 'hypocrisy', 'bomet']
----------------------------------------------------------------------------------------------------
['truth', 'matter', 'kidero', 'silenced', 'bribe', 'noticed', 'hypocrisy', 'bomet']
Most Informative Features
          contains(feel) = True           positi : negati =     12.8 : 1.0
          contains(view) = True           positi : negati =     12.8 : 1.0
           contains(car) = True           positi : negati =     12.8 : 1.0
       contains(concert) = True           positi : negati =     12.8 : 1.0
       contains(morning) = True           positi : negati =     12.8 : 1.0
        contains(friend) = False          negati : positi =      1.3 : 1.0
          contains(love) = False          negati : positi =      1.3 : 1.0
         contains(great) = False          negati : positi =      1.3 : 1.0
       contains(excited) = False          negati : positi =      1.3 : 1.0
       contains(amazing) = False          negati : positi =      1.3 : 1.0
          contains(best) = False          negati : positi =      1.3 : 1.0
```

Figure 4.2: Result of Feature Extraction

4.4 Performance Comparison of Different Classifiers

This experiment aimed to compare the Naïve Bayes classifier's performance with two other machine learning classifiers. The results of the experiment are summarized in Table 4.2

Table 4.2: The results of the Classifiers using term occurrence

Algorithm	**Accuracy**	**Precision**	**Recall**	**F-Measure**
Naïve Bayes	0.83	0.80	0.74	0.76
Maximum Entropy	0.65	0.43	0.59	0.55
SVM	0.67	0.60	0.66	0.60

The classifiers' results have been recorded, and Table 4.2 illustrates how each of the three classifiers performed against the four evaluation metrics. Naïve Bayes returned with the highest values in three of the four metrics – accuracy, recall, and F-measure. Naïve Bayes is the best performing method. The Classifiers used in table 4.2 to perform sentiment analysis on the given tweets. Unigrams, bigrams, and a combination of both were given as input features to the classifier.

The findings also exhibited that TF-IDF had better results with 81% accuracy compared to TF (68%), as shown in table 4.3. In addition, TF-IDF had better performance with or without N-grams. However, TF-IDF performed better when combined with the N-gram method.

Table 4.3: Accuracy of TF and TF-IDF with N-Grams

Feature selection method	NB	SVM	ME
TF	0.72	0.76	0.60
TF-IDF	0.80	0.79	0.68

Effect of n-grams

Bigram uses a combination of two words as a feature. Bigram effectively captures some features in the data that unigram fails to capture. This effect can be seen from the increase in accuracy from 74.1 (UniUnigram) to 83.1 percent, almost a 9% increase.

In addition, with unigrams, we studied the effect of bigrams to predict sentiments. We extracted bigrams with a technique that yielded extremely good results. Bigrams and unigrams applied together yielded 83.1% accuracy with the Naïve Bayes method. When extracted naively, bigrams, i.e., when formed by adding intensifiers to unigrams, perform poorly on trained classifiers. But, when bigrams are extracted with the technique mentioned above in the paper and used with unigrams, they give acceptable results. In our experiments, as in previous work, character n-grams were the most useful features for classification (outperforming word-based lexical features but also manually specified features).

Table 4.4: Table show combined results of n-gram features using various classifiers

Method	Accuracy(Unigram) %	Accuracy(Bigram) %	Precision	Recall
Naïve Bayes	74.1	83.1	77.1	85.2
Maximum Entropy	65%	67.2	69.2	73.1
Support Vector Machine	67.6	73.1	75.1	78.2

We conjecture that Naıve Bayes tends to select fewer features for training, which allows it to achieve higher precision accuracy and recall over other classifiers. For the Naive Bayes classifier, the results obtained are illustrated in figure 4.4

Table 4.5 shows the words found to be most informative. For example, the table shows that, of the tweets containing the word *kill*, 96% are relevant, and only 4% are irrelevant. The training data is used for this calculation. A surprising negative predictor was the word *poor*. When the *poor* appeared in a tweet, the tweet was irrelevant 94% of the time. The word *stupidity* shows a similar trend. This suggests that when Twitter users use the poor, they may not use precise or formal terms, opting for simple symptomatic such as *jinga*. The more formal terms may be more often associated with news items or general chat or discussion.

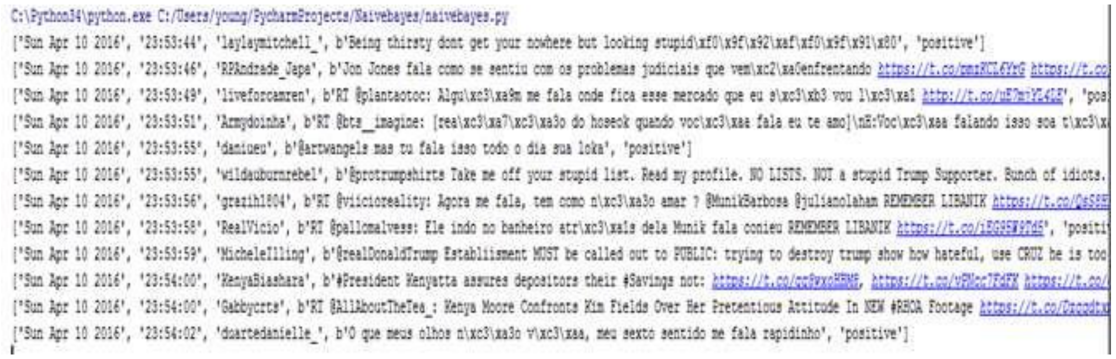
Table 4.5: Most informative words.

Word	Relevant: Irrelevant	Relevant Prior Probability	Irrelevant Prior Probability
kill	22/4	0.96	0.04
Luo	17/1	0.95	0.05
Fuck	17/1	0.95	0.05
poor	1/17	0.06	0.94
murder	16/1	0.94	0.06
tribe	15/1	0.94	0.06
Tahiri	17/1	0.93	0.07
Poverty	16/1	0.95	0.05
Stupidy	1/12	0.08	0.92
Jinga	16/1	0.94	0.06

Table 4.6 below indicates the different SVM kernels that are used to classify text without N-grams. For the analysis using python, Linear SVM performed the best at 67%.

Table 4.6: Accuracy of kernel methods without n-grams

Kernel Min	Min Range (*100%)	Max Range (*100%)
Linear	0.78	0.67
Poly	0.57	0.57
RBF	0.57	0.57



```
C:\Python34\python.exe C:/Users/young/PycharmProjects/Naivebayes/naivebayes.py
['Sun Apr 10 2016', '23:53:44', 'laylaymitchell_', b'Being thirsty dont get your nowhere but looking stupid\xf0\x9f\x92\xaf\xf0\x9f\x91\x80', 'positive']
['Sun Apr 10 2016', '23:53:46', 'RPAndrade_Japa', b'Jon Jones fala como se sentiu com os problemas judiciais que vem\xc2\xa0enfrentando https://t.co/pmzKCL6YrG https://t.co
['Sun Apr 10 2016', '23:53:49', 'liveforcamren', b'RT @plantaotoc: Algu\xc3\xa9m me fala onde fica esse mercado que eu s\xc3\xb3 vou l\xc3\xa1 http://t.co/uE7mzYL4LE', 'pos
['Sun Apr 10 2016', '23:53:51', 'Armydoinha', b'RT @bts_imagine: [rea\xc3\xa7\xc3\xa3o do hoseok quando voc\xc3\xaa fala eu te amo]\nE:Voc\xc3\xaa falando isso soa t\xc3\x
['Sun Apr 10 2016', '23:53:55', 'daniueu', b'@artwangels mas tu fala isso todo o dia sua loka', 'positive']
['Sun Apr 10 2016', '23:53:55', 'wildauburnrebel', b'@protrumpshirts Take me off your stupid list. Read my profile. NO LISTS. NOT a stupid Trump Supporter. Bunch of idiots.
['Sun Apr 10 2016', '23:53:56', 'grazih1804', b'RT @viicioreality: Agora me fala, tem como n\xc3\xa3o amar ? @MunikBarbosa @julianolaham REMEMBER LIBANIK https://t.co/QsS8H
['Sun Apr 10 2016', '23:53:58', 'RealVicio', b'RT @pallomalvess: Ele indo no banheiro atr\xc3\xa1s dela Munik fala conieu REMEMBER LIBANIK https://t.co/iEG9EW9Td5', 'positi
['Sun Apr 10 2016', '23:53:59', 'MicheleIlling', b'@realDonaldTrump Establiisment MUST be called out to PUBLIC: trying to destroy trump show how hateful, use CRUZ he is too
['Sun Apr 10 2016', '23:54:00', 'KenyaBiashara', b'#President Kenyatta assures depositors their #Savings not: https://t.co/qq9wxoHEMS, https://t.co/vPNocTFdFX https://t.co/
['Sun Apr 10 2016', '23:54:00', 'Gabbycrts', b'RT @AllAboutTheTea_: Kenya Moore Confronts Kim Fields Over Her Pretentious Attitude In NEW #RHOA Footage https://t.co/Dzqqdtx
['Sun Apr 10 2016', '23:54:02', 'duartedanielle_', b'O que meus olhos n\xc3\xa3o v\xc3\xaa, meu sexto sentido me fala rapidinho', 'positive']
```

Figure 4.3: Showing Classification process of tweets using Naive Bayes classifier results

To evaluate our proposed solution approach's performance, we use basic relevance measures in information retrieval and machine learning. We manually and automatically annotate each tweet in the dataset and validate our results based upon our decisions. We compute the accuracy of our classifier in terms of precision, recall, and f-score. Figure 4.4, figure 4.5, and figure 4.6 show various evaluation metrics of the various classified tweets.

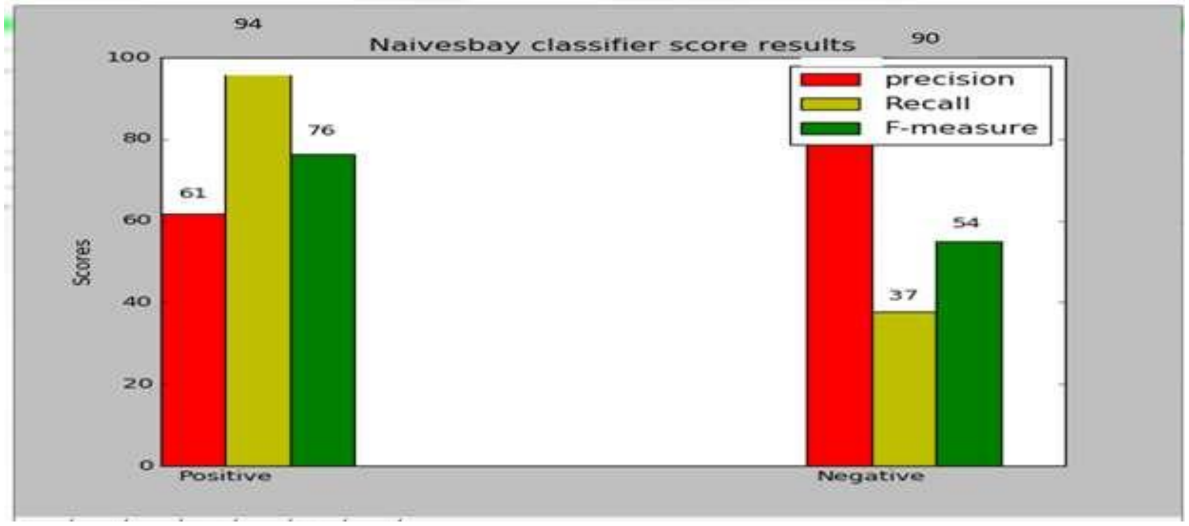


Figure 4.4: Precision, Recall & F-Measure for naïve Bayes classifier

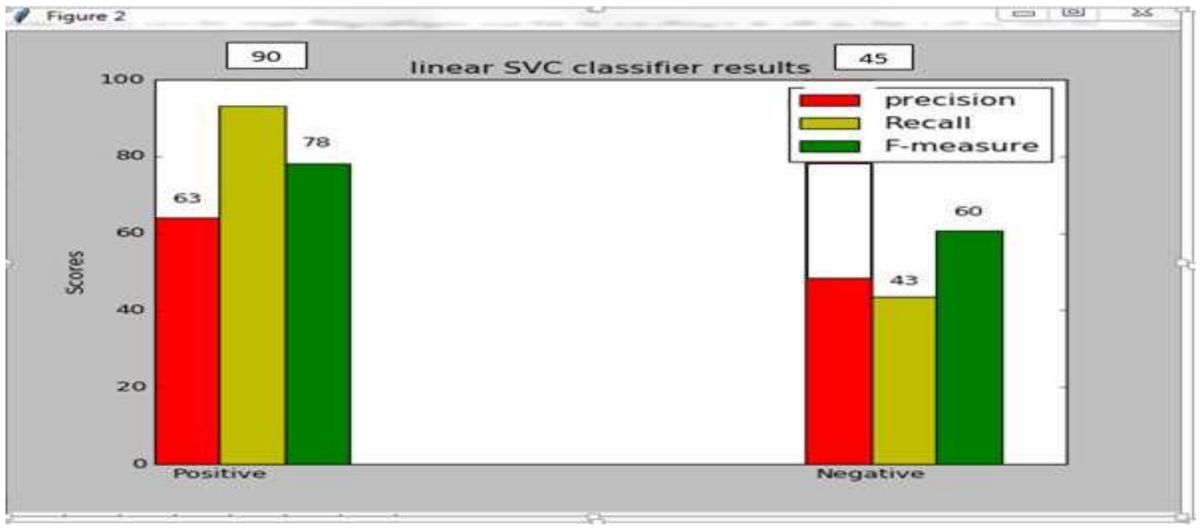


Figure 4.5: Precision, Recall & F-Measure for SVM classifier

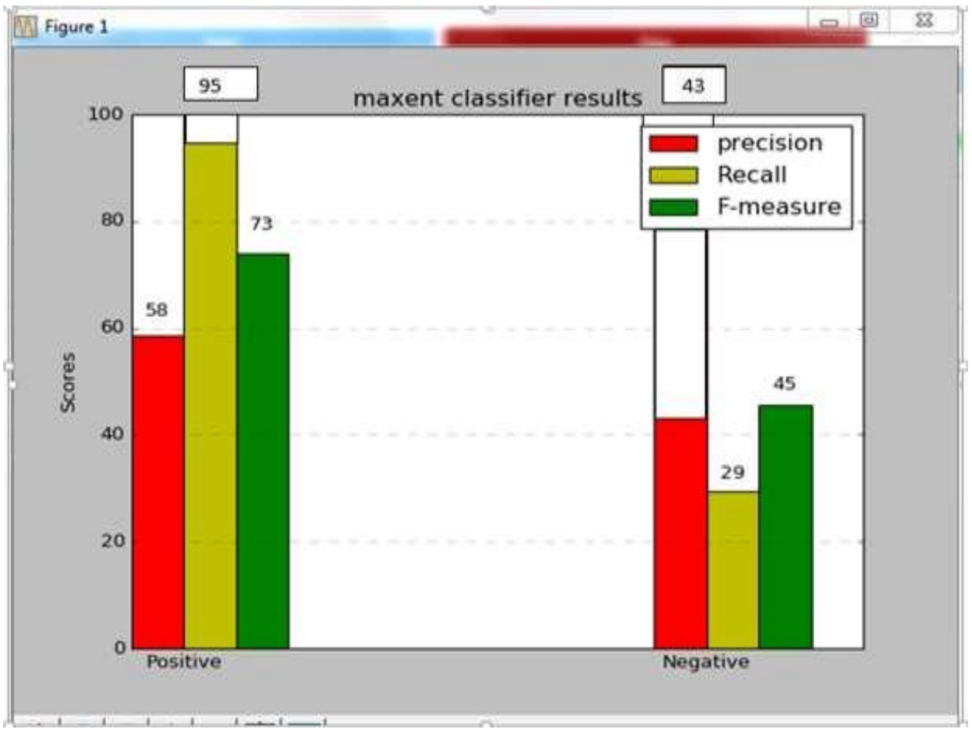


Figure 4.3: Precision, Recall & F-Measure for Maximum Entropy classifier.

A classifier's accuracy is vital to understand how effective it is in predicting sentiment to classify hate-related tweets. Accuracy gives us the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage. We measured the accuracy of each of the machine learning techniques used to identify the most effective one. For Naive Bayes, Maximum entropy, and Linear SVC, we used NLTK's to classify accuracy and scikit's sklearn.metrics.accuracy_score to determine the classifier accuracy.

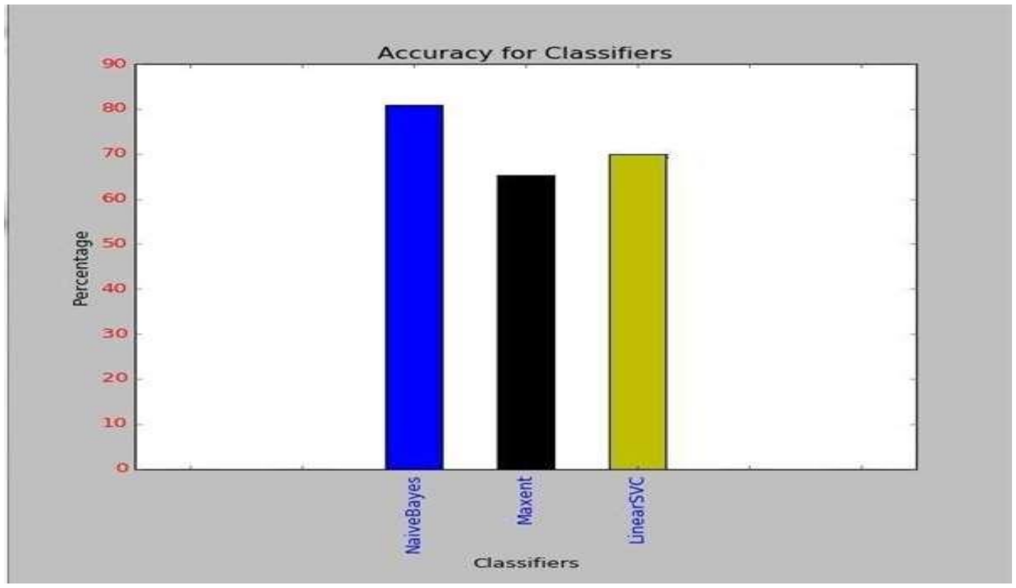
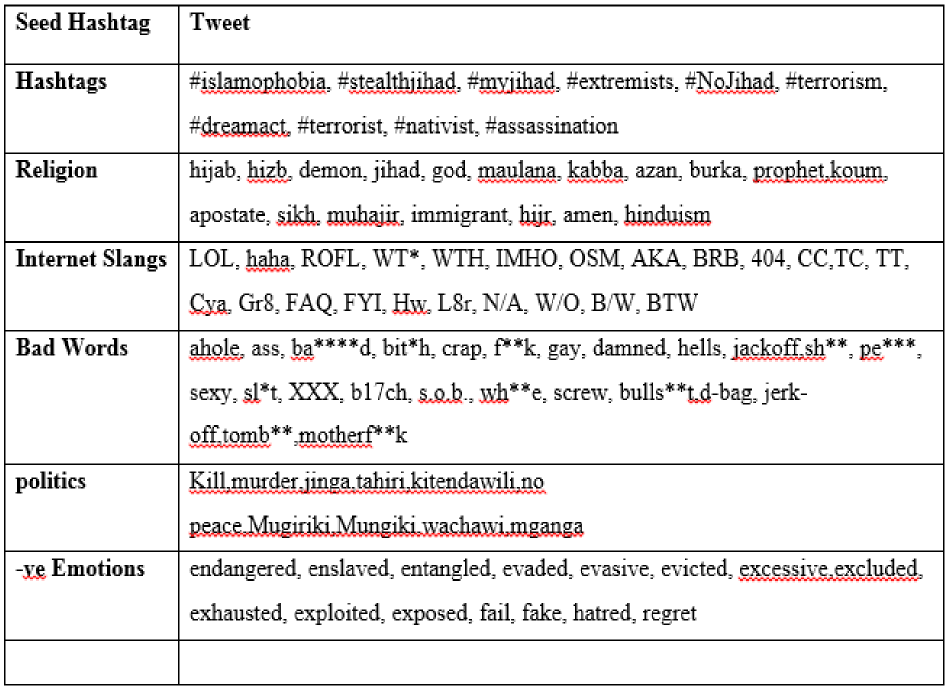


Figure 4.7: Showing Accuracy using various classifiers

As shown in Figure 4.7, the accuracy of Naive Bayes was the best when compared to Maximum entropy and SVM, which came second and third, respectively. Naive Bayes was correctly able to classify hate dataset sentiment with an accuracy of 83.1%. The lowest performance was 50.6% accuracy, which was by linear SVM. Figure 4.4 reveals that internet slangs and punctuations (! and ? marks) are less important features and doesn't affect the accuracy by a major difference. Still, we cannot neglect them completely because they affect the overall accuracy. The Naive Bayes classifier method provided satisfactory preliminary results compared to SVM and maximum entropy. The kind of tweets that were filtered were negative related messages posted by Twitter users. Some of the keywords selected from the raw tweets being fetched from the Tweepy API, as shown in Table 4.7. Most of the words are used in disseminating hate tweets to a specific target group(s)

Table 4.7: **Sample of Keywords present in Hate promoting Tweets**



Seed Hashtag	Tweet
Hashtags	#islamophobia, #stealthjihad, #myjihad, #extremists, #NoJihad, #terrorism, #dreamact, #terrorist, #nativist, #assassination
Religion	hijab, hizb, demon, jihad, god, maulana, kabba, azan, burka, prophet,koum, apostate, sikh, muhajir, immigrant, hijr, amen, hinduism
Internet Slangs	LOL, haha, ROFL, WT*, WTH, IMHO, OSM, AKA, BRB, 404, CC,TC, TT, Cya, Gr8, FAQ, FYI, Hw, L8r, N/A, W/O, B/W, BTW
Bad Words	ahole, ass, ba****d, bit*h, crap, f**k, gay, damned, hells, jackoff,sh**, pe***, sexy, sl*t, XXX, b17ch, s.o.b., wh**e, screw, bulls**t,d-bag, jerk-off,tomb**,motherf**k
politics	Kill,murder,jinga,tahiri,kitendawili,no peace,Mugiriki,Mungiki,wachawi,mganga
-ye Emotions	endangered, enslaved, entangled, evaded, evasive, evicted, excessive,excluded, exhausted, exploited, exposed, fail, fake, hatred, regret

Source: Research Data.

Tweets often contain various types of information, including usernames, date, retweets, favorites, text, geo, mentions, hashtags, id, and permalink. The research was only interested in the next part of the tweet to build the corpora. The collected tweets were, therefore, processed to separate the various parts. In addition, as shown in table 4.7, a sample of these terms present in hate and extremism promoting tweets as used by user 1 & user 2 in their handles.

Dataset Description

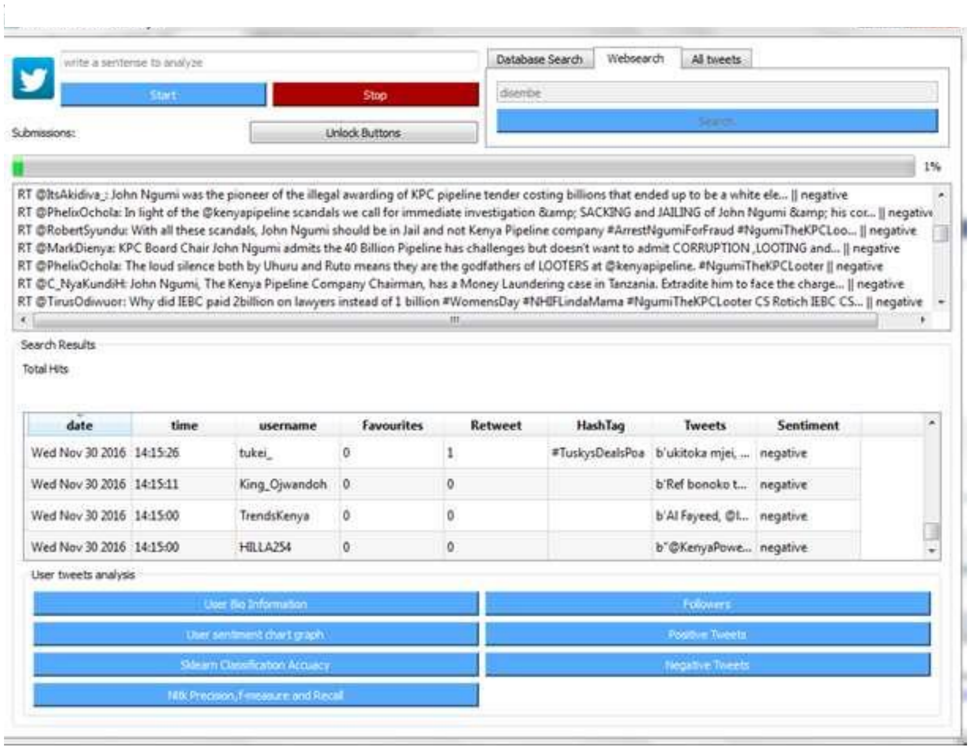
Analyses were done on these labeled datasets using various feature extraction techniques. We used the framework where the preprocessor is applied to the raw sentences that make it more appropriate to understand. Further, the different machine learning techniques train the dataset with feature vectors. The semantic analysis offers a large set of synonyms and similarity, which provides the content's polarity, as shown in table 4.8.

Table 4.8: Dataset Description

category	Total	Negative	Positive
Train Data	45,645	23514	22,131
Test Data	22,820	12235	10,585

Retrieving Tweets

A user interacts with the application by entering the keywords to retrieve tweets from Twitter or Mongo database. Once the user enters the keywords, they are passed to collect tweets. Figure 4.8 illustrates the web platform's user interface to view user analyzed tweets and biodata as analyzed.



```
RT @ItsAkidiva_: John Ngumi was the pioneer of the illegal awarding of KPC pipeline tender costing billions that ended up to be a white ele... || negative
RT @PhelixOchola: In light of the @kenyapipeline scandals we call for immediate investigation &amp; SACKING and JAILING of John Ngumi &amp; his cor... || negative
RT @RobertSyundu: With all these scandals, John Ngumi should be in Jail and not Kenya Pipeline company #ArrestNgumiForFraud #NgumiTheKPCLoo... || negative
RT @MarkDienya: KPC Board Chair John Ngumi admits the 40 Billion Pipeline has challenges but doesn't want to admit CORRUPTION ,LOOTING and... || negative
RT @PhelixOchola: The loud silence both by Uhuru and Ruto means they are the godfathers of LOOTERS at @kenyapipeline. #NgumiTheKPCLooter || negative
RT @C_NyaKundiH: John Ngumi, The Kenya Pipeline Company Chairman, has a Money Laundering case in Tanzania. Extradite him to face the charge... || negative
RT @TirusOdiwuor: Why did IEBC paid 2billion on lawyers instead of 1 billion #WomensDay #NHIFLindaMama #NgumiTheKPCLooter CS Rotich IEBC CS... || negative
```

date	time	username	Favourites	Retweet	HashTag	Tweets	Sentiment
Wed Nov 30 2016	14:15:26	tukei_	0	1	#TuskysDealsPoa	b'ukitoka mjei, ...	negative
Wed Nov 30 2016	14:15:11	King_Ojwandoh	0	0		b'Ref bonoko t...	negative
Wed Nov 30 2016	14:15:00	TrendsKenya	0	0		b'Al Fayeed, @I...	negative
Wed Nov 30 2016	14:15:00	HILLA254	0	0		b"@KenyaPowe...	negative

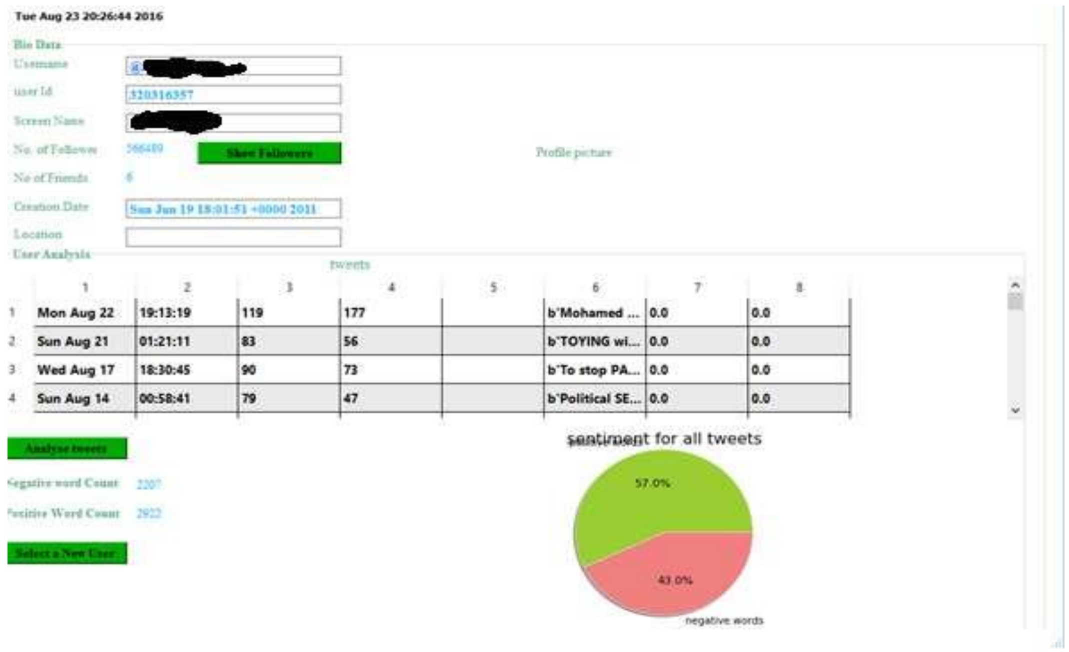
Figure 4.8: User interface to analyze user tweets

Retweets are driven by the content value, while the name value of the user drives mentions. Thus, most Twitter users continued propagating hate messages to gain more audience. Retweet, mention, and follower features enable the Twitter user to propagate their tweet or other user tweets; this mechanism is also known as information diffusion between users. Most users tend to share their favorite tweets with their followers; therefore, retweets can also be viewed as an important signal of user interest and needs.

Each Twitter user is given a unique ID. In this research, the retweet number will be analyzed because users retweet a tweet to improve the traditional classification algorithm. Unlike retweets, like feature, show the agreement about the content and see the list of tweets we like before. The number of like which symbolized by heart-shaped icon means that some Twitter users like the tweet. Figure 4.10 represent a tweet from user 1 has 685 of retweet and 2.0 K number of user that like this tweet. This research will use these two features to improve sentiment analysis classification due to a limited number of labeled sentiment tweets.

a) User 1

For instance, we have noted (User 1 (@user1) who has been posting hate tweets on his handle. Some of the Twitter users who has been posting hate tweets include but not limited to as illustrated in figure 20.User 1 (@User1)) after analyzing his tweets l discovered he had posted 2207 negative tweets and 2922 positive tweets. User 1 has 566,489 followers and six friends. The following are some of his tweets. After analyzing his 453 tweets, we can say that he propagates hate tweets in his handle, as illustrated in figure 4.9.



Tue Aug 23 20:26:44 2016

Bio Data

Username	@[redacted]
user Id	320316357
Screen Name	[redacted]
No. of Followers	566489
No of Friends	6
Creation Date	Sun Jun 19 18:01:51 +0000 2011
Location	

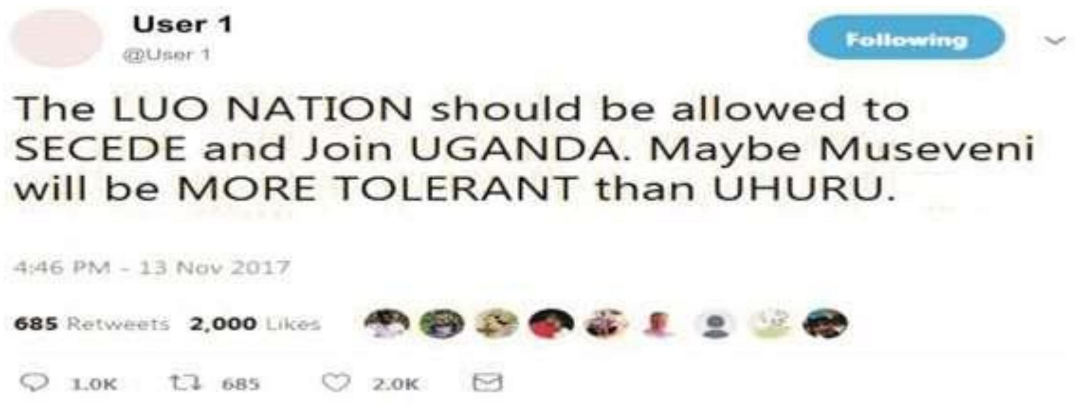
User Analysis

tweets

	1	2	3	4	5	6	7	8
1	Mon Aug 22	19:13:19	119	177		b'Mohamed ...	0.0	0.0
2	Sun Aug 21	01:21:11	83	56		b'TOYING wi...	0.0	0.0
3	Wed Aug 17	18:30:45	90	73		b'To stop PA...	0.0	0.0
4	Sun Aug 14	00:58:41	79	47		b'Political SE...	0.0	0.0

Negative word Count	2207
Positive Word Count	2922

Figure 4.9: user1 sentiment analysis



User 1
@User 1

The LUO NATION should be allowed to SECEDE and Join UGANDA. Maybe Museveni will be MORE TOLERANT than UHURU.

4:46 PM - 13 Nov 2017

685 Retweets 2,000 Likes

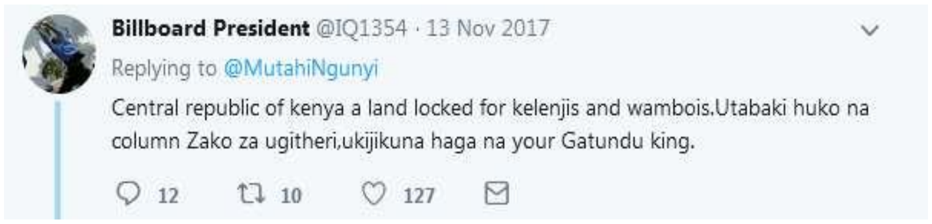
Figure 4.4: Sampled tweet of user 1

Figure 4.10 represent a sampled hate tweet from user1 @user1 posted on November 13, 2017, at4.46 PM, which has 685 retweets and 2.0 K number of users that like this tweet.

Our analysis of Twitter messages showed that prominent personalities sent the most influential disseminators of hate speech messages in terms of numbers of likes and retweets in society. Based on their profiles and the ideology they promote via their accounts, they have a massive following (for instance, from Figure 4.10, user 1 has 566,489 at the time of conducting the study) support gaining traction across the country. Many of the users who reacted had comparatively newer Twitter accounts, having joined in 2016 or 2017. The reactions were aggressive and seemed designed to evoke an emotional response from us. Responses came almost immediately, and users were unwilling to change their opinions.

We analyze the number of tweets, followers, favorite tweets a user has, and the interval in seconds between their tweets. We show these statistics in Figure 4.10. We normalize the number of tweets and followers by the number of days the users have since their account creation date. Our results suggest that hateful users are “power users” in the sense that they tweet more, in shorter intervals, have more retweets by other people, and follow other users with more influence.

For instance, “Your argument is pathetic. What happens in a corrupt marginalized banana republic is secession. When some part of the community decides to rule forever, then be ready to witness ultimate secession. A few years ago, Sudan & Yugoslavia were intact. Today...? Open your eyes”. (Saxo™ Movich @**saxopolis**). Some Twitter users also set up accounts with pseudonyms. The user utilized bots to reply to hate tweets/ posts. When a hate tweet was posted, the bot would respond with a pre-written answer. Their strategy was to respond either with memes, a declarative statement, or articles and data-heavy responses. As shown in figure 4.11, Billboard President @IQ1354 replying to user 1, posted a hate message. One of our Twitter accounts had 156 interactions. The account saw 7,500 impressions within a month, meaning that 7,500active Twitter users saw this account’s posts attract 127 likes, ten retweets, and twelve replies.



Billboard President @IQ1354 · 13 Nov 2017

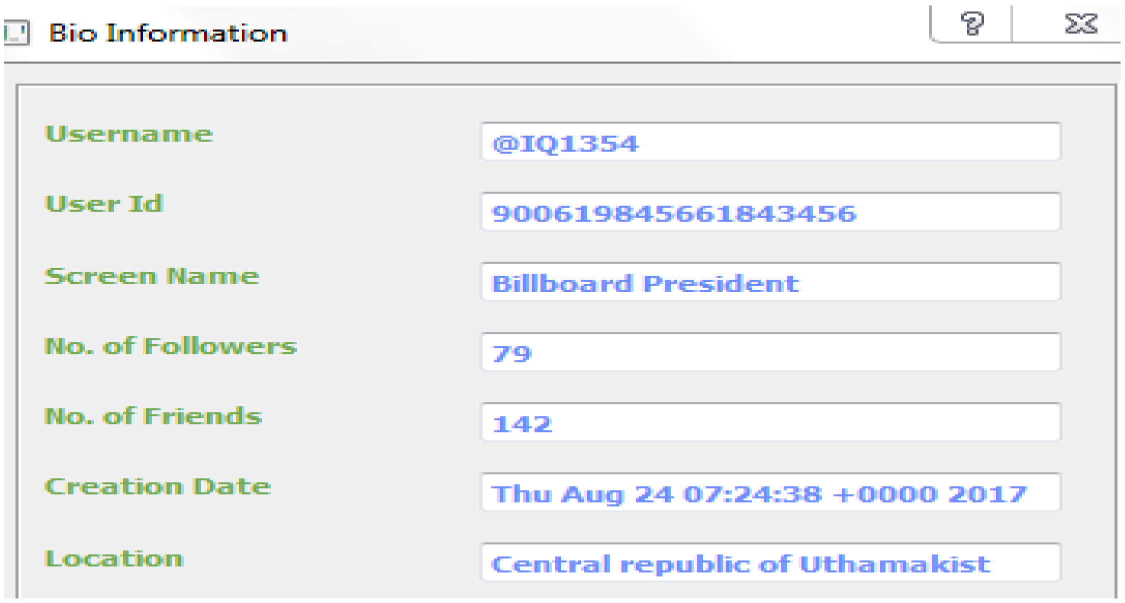
Replying to @MutahiNgunyi

Central republic of kenya a land locked for kelenjis and wambois.Utabaki huko na column Zako za ugitheri,ukijikuna haga na your Gatundu king.

12 10 127

Figure 4.5: Sampled Twitter user replying to user 1

Many of the users who reacted had comparatively newer Twitter accounts, having joined in 2016 or 2017. For instance, Billboard President @IQ1354 joined the platform in August 2017, as shown in figure 4.12.



Bio Information

Username	@IQ1354
User Id	900619845661843456
Screen Name	Billboard President
No. of Followers	79
No. of Friends	142
Creation Date	Thu Aug 24 07:24:38 +0000 2017
Location	Central republic of Uthamakist

Figure 4.12: Sampled new user replying to user 1

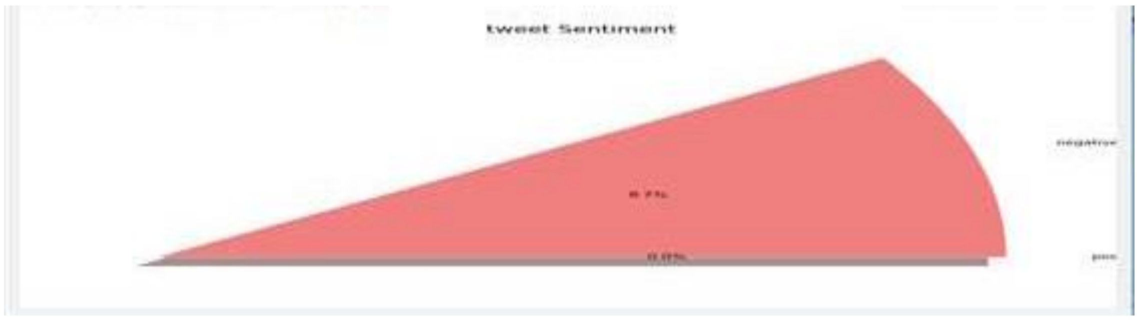
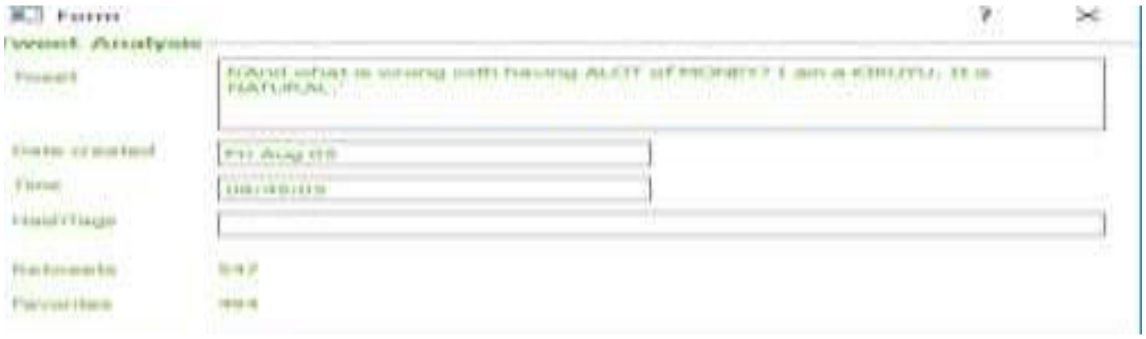


Figure 4.6: Tweet sentiment analysis for user 1

1. Example1: of user one hate tweets

And what is wrong with having A LOT of MONEY? I am a KIKUYU.It is NATURAL. ? — User 1 (@User1) posted on August 05, 2016,08:45:05Pm,as shown in figure 4.14.



Form

Tweet Analysis

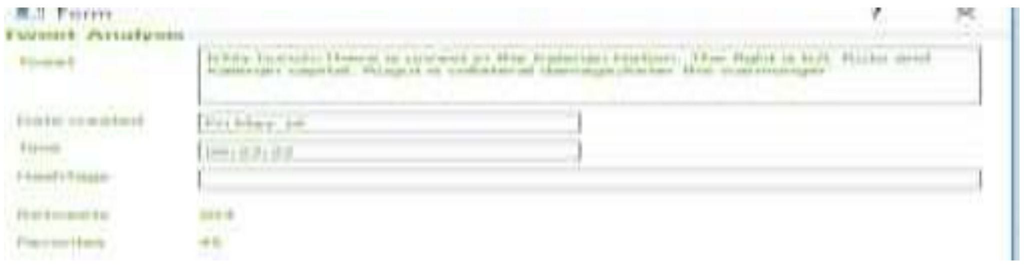
Tweet	b'And what is wrong with having ALOT of MONEY? I am a KIKUYU. It is NATURAL.
Date created	Fri Aug 05
Time	08:45:05
HashTags	
Retweets	547
Favorites	494

Figure 4.7: Sampled user one hate message

The tweet had 547 retweets and 494 likes. The tweets used tribal stereotypic derogations to pass the hate speech to the public, as shown in figure 4.14.

2. Example2: of user one hate tweets

My Hunch :There is unrest in the Kalenjin Nation.The fight is b/t Ruto and Kalenjin capital.Rugut is collateral damage;Keter the Warmonger May 16, 2016. — User 1 (@User1) May 16, 2016 (refer figure 4.15)



Form

Tweet Analysis

Tweet	[illegible]
Date created	Fri May 16
Time	[illegible]
HashTags	
Retweets	204
Favorites	45

Figure 4.8: Sampled user one hate message

The tweet posted on 16/05/2016 was retweeted 204 times and liked 45 times. From the tweet, warmonger and fight have been highlighted as a negative tweet word. In addition, his tweet is targeting the entire community. Using sentiment analysis, the tweet scored a high level of hatred, as shown in figure 4.15.

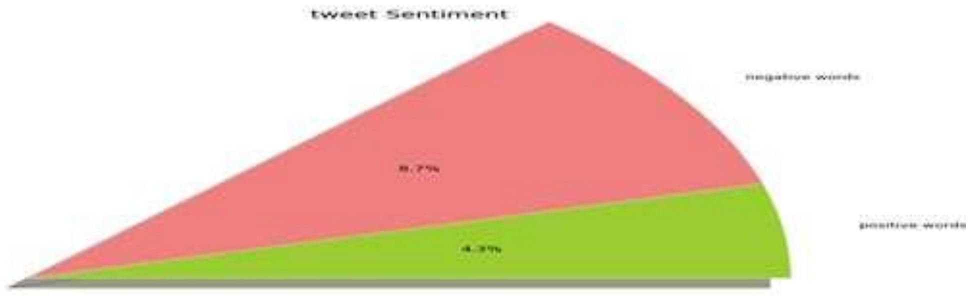


Figure 4.9: Tweet sentiment score

Example3: of user one hate tweets

As Kibera youth wake up to their shame tomorrow, remember this:'…User 2 is like a ROCKING CHAIR: keeps you busy, takes you NOWHERE User 1(one) (@User1) June 22, 2016(refer figure 4.17)

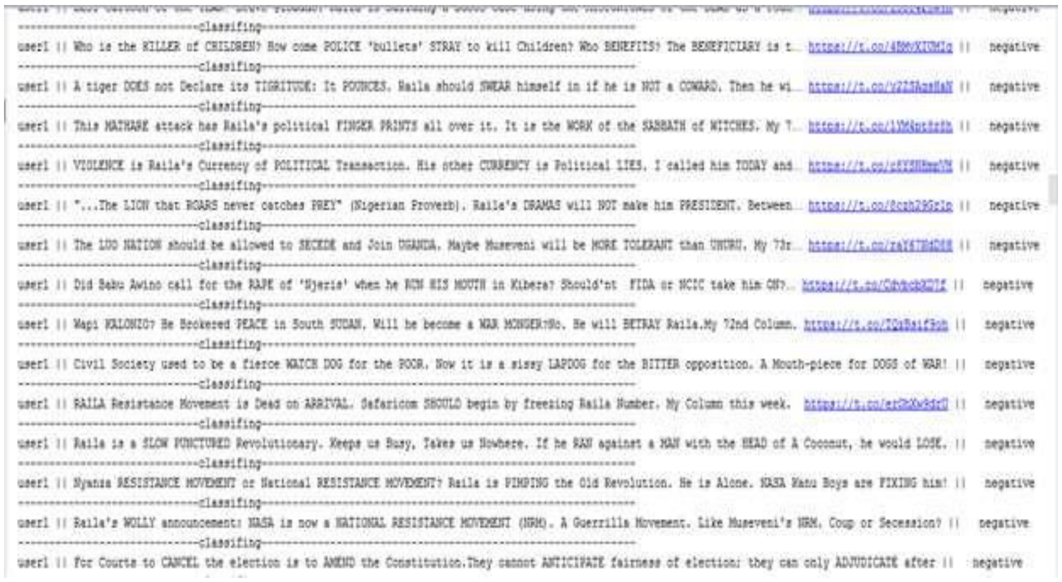
Figure 4 10: Sampled user one hate message

The tweet targeted Kibera's youth, which User 2 has been their MP for over ten years. He insulted them by telling them to wake up to their shame tomorrow. This tweet targets Kibera in general, and User 2(two) has an individual.

Are LUOs poor? No idea. THEY SHOULD TELL US. Are there POVERTY STRICKEN LUOs. YES. Statistics: 82% of people in BONDO live below poverty line.(User 1 (@User1) August 21, 2015) In his tweets, he suggests that Luos are poverty-stricken. According to analysis, this is a hate speech targeting one of the Kenya community.

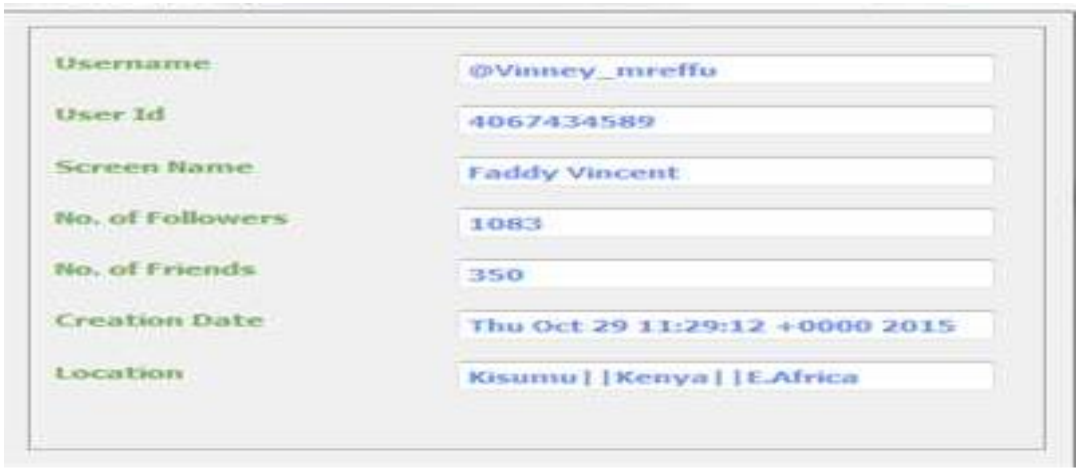
My KENYAN DREAM is to RETIRE in Kisumu City. I want the intellectual conversation in old age. But one cannot PONTIFICATE in POLITICAL BONDAGE.— User 1 (@User1) August 19, 2015)

In addition, User 1 chooses Kisumu, the third-largest city in Kenya, this is predominantly Luo, as his city for retirement. He continues to pose hatred to the entire community. Furthermore, in three days, He tweets very dangerous tweets targeting a particular community. Further, having huge followers User1 hate tweet has the potential of causing more harm. Having examined his hate tweets, it, therefore, follows that he propagates hate tweets via his Twitter handle and is a threat to a civilized society. The National Cohesion and Integration Act 2008, for example, states explicitly in Section 62 (1): On User 1 (@User1) Tweets in Twitter after analyzing his tweets, we have discovered that most of his tweets contain hatred and most target one of Kenyan community (Luo), political individuals and the general public. Most of his tweets are tweeted on average 600 times and liked more than 500 times on average. After collection, preprocessing, and classification of tweets of user 1, the classification can be represented as indicated in figure 4.18.



```
[illegible]
------------------------------classifing------------------------------
user1 || Who is the KILLER of CHILDREN? How come POLICE 'bullets' STRAY to kill Children? Who BENEFITS? The BENEFICIARY is t… https://t.co/4B6vXIUMIq ||   negative
------------------------------classifing------------------------------
user1 || A tiger DOES not Declare its TIGRITUDE: It POUNCES. Raila should SWEAR himself in if he is NOT a COWARD. Then he wi… https://t.co/V22SAgsHaN ||   negative
------------------------------classifing------------------------------
user1 || This MATHARE attack has Raila's political FINGER PRINTS all over it. It is the WORK of the SABBATH of WITCHES. My 7… https://t.co/1YM4pt8rfh ||   negative
------------------------------classifing------------------------------
user1 || VIOLENCE is Raila's Currency of POLITICAL Transaction. His other CURRENCY is Political LIES. I called him TODAY and… https://t.co/z8YSHBmgVN ||   negative
------------------------------classifing------------------------------
user1 || "...The LION that ROARS never catches PREY" (Nigerian Proverb). Raila's DRAMAS will NOT make him PRESIDENT. Between… https://t.co/8czh29Gz1p ||   negative
------------------------------classifing------------------------------
user1 || The LUO NATION should be allowed to SECEDE and Join UGANDA. Maybe Museveni will be MORE TOLERANT than UHURU. My 73r… https://t.co/zaY6T8dD88 ||   negative
------------------------------classifing------------------------------
user1 || Did Babu Awino call for the RAPE of 'Njeris' when he RUN HIS MOUTH in Kibera? Should'nt  FIDA or NCIC take him ON?… https://t.co/CdvbcbXD7f ||   negative
------------------------------classifing------------------------------
user1 || Wapi KALONZO? He Brokered PEACE in South SUDAN. Will he become a WAR MONGER?No. He will BETRAY Raila.My 72nd Column. https://t.co/TQzBaif9oh ||   negative
------------------------------classifing------------------------------
user1 || Civil Society used to be a fierce WATCH DOG for the POOR. Now it is a sissy LAPDOG for the BITTER opposition. A Mouth-piece for DOGS of WAR! ||   negative
------------------------------classifing------------------------------
user1 || RAILA Resistance Movement is Dead on ARRIVAL. Safaricom SHOULD begin by freezing Raila Number. My Column this week.  https://t.co/er2hXw8drD ||   negative
------------------------------classifing------------------------------
user1 || Raila is a SLOW PUNCTURED Revolutionary. Keeps us Busy, Takes us Nowhere. If he RAN against a MAN with the HEAD of A Coconut, he would LOSE. ||   negative
------------------------------classifing------------------------------
user1 || Nyanza RESISTANCE MOVEMENT or National RESISTANCE MOVEMENT? Raila is PIMPING the Old Revolution. He is Alone. NASA Nanu Boys are FIXING him! ||   negative
------------------------------classifing------------------------------
user1 || Raila's WOLLY announcement: NASA is now a NATIONAL RESISTANCE MOVEMENT (NRM). A Guerrilla Movement. Like Museveni's NRM. Coup or Secession? ||   negative
------------------------------classifing------------------------------
user1 || For Courts to CANCEL the election is to AMEND the Constitution.They cannot ANTICIPATE fairness of election; they can only ADJUDICATE after ||   negative
```

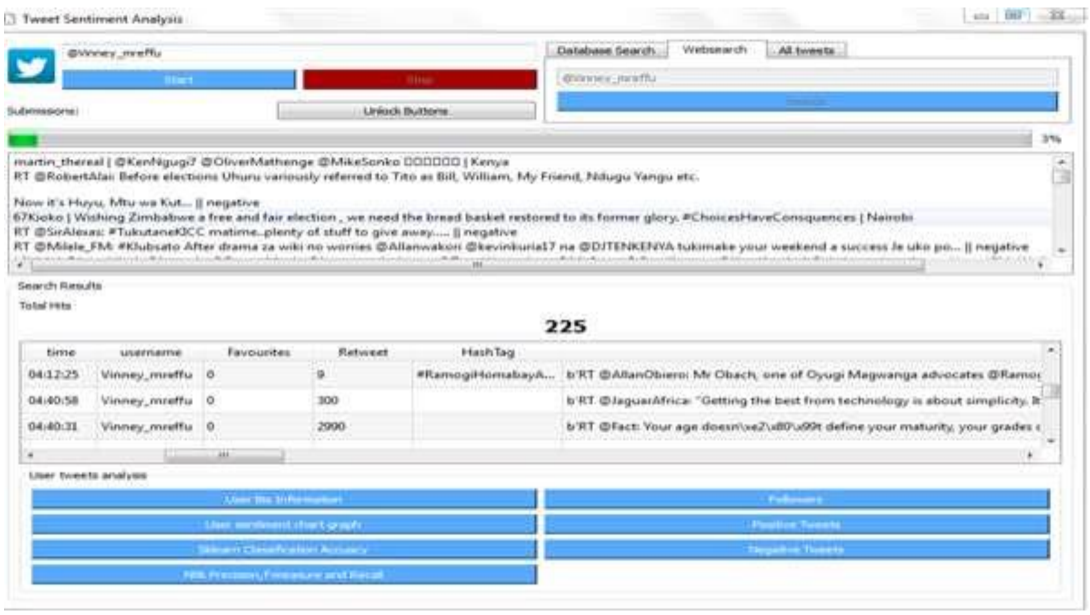
Figure 4.11: Tweets classification process of user 1User 4



Username	@Vinney_mreffu
User Id	4067434589
Screen Name	Faddy Vincent
No. of Followers	1083
No. of Friends	350
Creation Date	Thu Oct 29 11:29:12 +0000 2015
Location	Kisumu		Kenya		E.Africa

Figure 4.12: User 4 Bio Data

User 4; Another Twitter user who is leading in disseminators hate speech on the platform.As shown in figure 4.19. From the biodata of user 4, he can be identified, for instance, the location (Kisumu) and Twitter's unique user id.

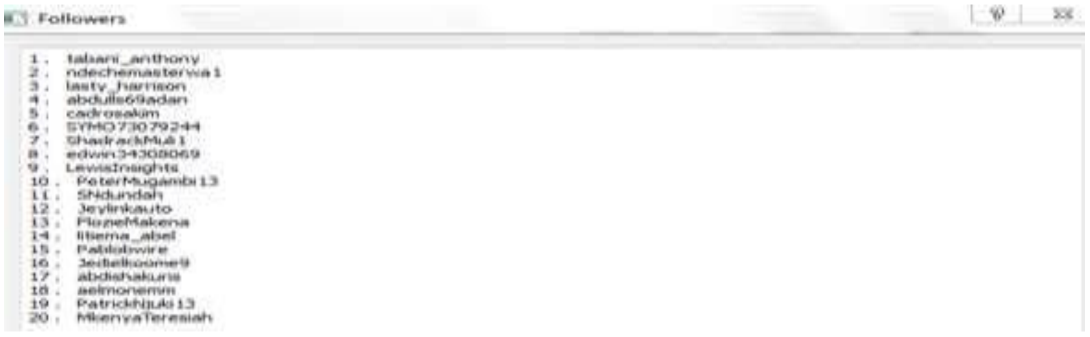


```
martin_thereal | @KenNgugi7 @OliverMathenge @MikeSonko □□□□□□ | Kenya
RT @RobertAlai: Before elections Uhuru variously referred to Tito as Bill, William, My Friend, Ndugu Yangu etc.

Now it's Huyu, Mtu wa Kut... || negative
67Kioko | Wishing Zimbabwe a free and fair election , we need the bread basket restored to its former glory. #ChoicesHaveConsquences | Nairobi
RT @SirAlexas: #TukutaneKICC matime..plenty of stuff to give away..... || negative
RT @Milele_FM: #Klubsato After drama za wiki no worries @Allanwakori @kevinkuria17 na @DJTENKENYA tukimake your weekend a success Je uko po... || negative
```

time	username	Favourites	Retweet	HashTag	
04:12:25	Vinney_mreffu	0	9	#RamogiHomabayA...	b'RT @AllanObiero: Mr Obach, one of Oyugi Magwanga advocates @Ramo
04:40:58	Vinney_mreffu	0	300		b'RT @JaguarAfrica: "Getting the best from technology is about simplicity. It
04:40:31	Vinney_mreffu	0	2900		b'RT @Fact: Your age doesn\xe2\x80\x99t define your maturity, your grades

Figure 4.13: Tweet Sentiment analysis for user 4

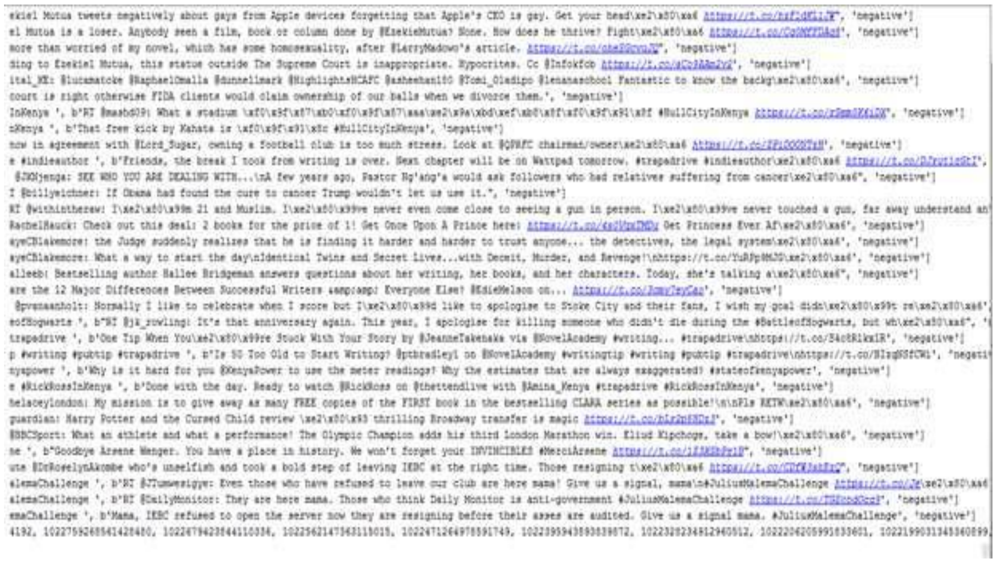
From user sentiment analysis, It is evident that most of his hate tweets get massive retweets (Refer Figure 4.20). A total of 225 tweets have positively been identified to contain hate messages/words. In addition, from figure 4.20, the period of posting the hate message can be traced.



```
1 .  tabani_anthony
2 .  ndechemasterwa1
3 .  lasty_harrison
4 .  abdulle69adan
5 .  cadrosalum
6 .  SYMO73079244
7 .  ShadrackMuli1
8 .  edwin34308069
9 .  LewisInsights
10 .  PeterMugambi13
11 .  SNdundah
12 .  Jeylinkauto
13 .  FlozeMakena
14 .  lttema_abel
15 .  Pablobwire
16 .  Jedbelkoomer9
17 .  abdishakuris
18 .  aelmonemm
19 .  PatrickNjuki13
20 .  MkenyaTeresiah
```

Figure 4.14: User 4 Followers

Figure 4.21: We can positively identify and analyze four followers who help promote and propagate Twitter's hatred. Furthermore, from figure 4.1.8, identifying followers who promote and propagate hate tweets can be viewed.



```
ekiel Mutua tweets negatively about gays from Apple devices forgetting that Apple's CEO is gay. Get your head\xe2\x80\xa6 https://t.co/bzf1dKl1JW", 'negative']
el Mutua is a loser. Anybody seen a film, book or column done by @EzekieMutua? None. How does he thrive? Fight\xe2\x80\xa6 https://t.co/Co0MYTDAqd', 'negative']
more than worried of my novel, which has some homosexuality, after @LarryMadowo's article. https://t.co/oke8GvquJU", 'negative']
ding to Ezekiel Mutua, this statue outside The Supreme Court is inappropriate. Hypocrites. Cc @Infokfcb https://t.co/aCo9AAm2v2', 'negative']
ital_ME: @lucamatoke @RaphaelOmalla @dunnelimark @HighlightsHCAFC @asheehan100 @Tomi_Oladipo @lenanaschool Fantastic to know the backg\xe2\x80\xa6", 'negative']
court is right otherwise FIDA clients would claim ownership of our balls when we divorce them.', 'negative']
InKenya ', b'RT @masbd09: What a stadium \xf0\x9f\x87\xb0\xf0\x9f\x87\xaa\xe2\x9a\xbd\xef\xb8\x8f\xf0\x9f\x91\x8f #HullCityInKenya https://t.co/r9em0K4iDX", 'negative']
nKenya ', b'That free kick by Kahata is \xf0\x9f\x91\x8c #HullCityInKenya', 'negative']
now in agreement with @Lord_Sugar, owning a football club is too much stress. Look at @QPRFC chairman/owner\xe2\x80\xa6 https://t.co/ZPiOOOOTxH', 'negative']
e #indieauthor ', b"Friends, the break I took from writing is over. Next chapter will be on Wattpad tomorrow. #trapadrive #indieauthor\xe2\x80\xa6 https://t.co/DJrutizGtI",
 @JKNjenga: SEE WHO YOU ARE DEALING WITH...\nA few years ago, Pastor Ng'ang'a would ask followers who had relatives suffering from cancer\xe2\x80\xa6", 'negative']
I @billyeichner: If Obama had found the cure to cancer Trump wouldn't let us use it.", 'negative']
RT @withintherew: I\xe2\x80\x99m 21 and Muslim. I\xe2\x80\x99ve never even come close to seeing a gun in person. I\xe2\x80\x99ve never touched a gun, far away understand an
RachelHauck: Check out this deal: 2 books for the price of 1! Get Once Upon A Prince here: https://t.co/4ed0pxTMDy Get Princess Ever Af\xe2\x80\xa6', 'negative']
ayeCBlakemore: the Judge suddenly realizes that he is finding it harder and harder to trust anyone... the detectives, the legal system\xe2\x80\xa6', 'negative']
ayeCBlakemore: What a way to start the day\nIdentical Twins and Secret Lives...with Deceit, Murder, and Revenge!\nhttps://t.co/YuRFp9MJ0\xe2\x80\xa6', 'negative']
alleeb: Bestselling author Hallee Bridgeman answers questions about her writing, her books, and her characters. Today, she's talking a\xe2\x80\xa6", 'negative']
are the 12 Major Differences Between Successful Writers &amp;amp; Everyone Else? @EdieMelson on... https://t.co/3omyTeyCas', 'negative']
 @pvenaanholt: Normally I like to celebrate when I score but I\xe2\x80\x99d like to apologise to Stoke City and their fans, I wish my goal didn\xe2\x80\x99t re\xe2\x80\xa6',
eofHogwarts ', b"RT @jk_rowling: It's that anniversary again. This year, I apologise for killing someone who didn't die during the #BattleofHogwarts, but wh\xe2\x80\xa6", '
trapadrive ', b'One Tip When You\xe2\x80\x99re Stuck With Your Story by @JeanneTakenaka via @NovelAcademy #writing... #trapadrive\nhttps://t.co/S4c8RlkxIR', 'negative']
p #writing #pubtip #trapadrive ', b'Is 50 Too Old to Start Writing? @ptbradley1 on @NovelAcademy #writingtip #writing #pubtip #trapadrive\nhttps://t.co/BIzqNSfCWi', 'negati
nyapower ', b'Why is it hard for you @KenyaPower to use the meter readings? Why the estimates that are always exaggerated? #stateofkenyapower', 'negative']
e #RickRossInKenya ', b'Done with the day. Ready to watch @RickRoss on @thettendlive with @Amina_Kenya #trapadrive #RickRossInKenya', 'negative']
helaceylondon: My mission is to give away as many FREE copies of the FIRST book in the bestselling CLARA series as possible!\n\nPls RETW\xe2\x80\xa6', 'negative']
guardian: Harry Potter and the Cursed Child review \xe2\x80\x93 thrilling Broadway transfer is magic https://t.co/bIz2n8NDzJ', 'negative']
@BBCSport: What an athlete and what a performance! The Olympic Champion adds his third London Marathon win. Eliud Kipchoge, take a bow!\xe2\x80\xa6', 'negative']
ne ', b"Goodbye Arsene Wenger. You have a place in history. We won't forget your INVINCIBLES #MerciArsene https://t.co/1fJNSbPr1P", 'negative']
ute @DrRoselynAkombe who's unselfish and took a bold step of leaving IEBC at the right time. Those resigning t\xe2\x80\xa6 https://t.co/CDfWJabEzQ", 'negative']
alemaChallenge ', b'RT @JTumwesigye: Even those who have refused to leave our club are here mama! Give us a signal, mama\n#JuliusMalemaChallenge https://t.co/Je\xe2\x80\xa6
alemaChallenge ', b'RT @DailyMonitor: They are here mama. Those who think Daily Monitor is anti-government #JuliusMalemaChallenge https://t.co/TSEcbdGcs9', 'negative']
emaChallenge ', b'Mama, IEBC refused to open the server now they are resigning before their asses are audited. Give us a signal mama. #JuliusMalemaChallenge', 'negative']
4192, 1022759268541428480, 1022479423844110336, 1022562147363115015, 1022471264978591749, 1022395943893839872, 1022328234812940512, 1022206205991833601, 1022199031345340899,
```

Figure 4.15: User 4, Hate tweets being classified.

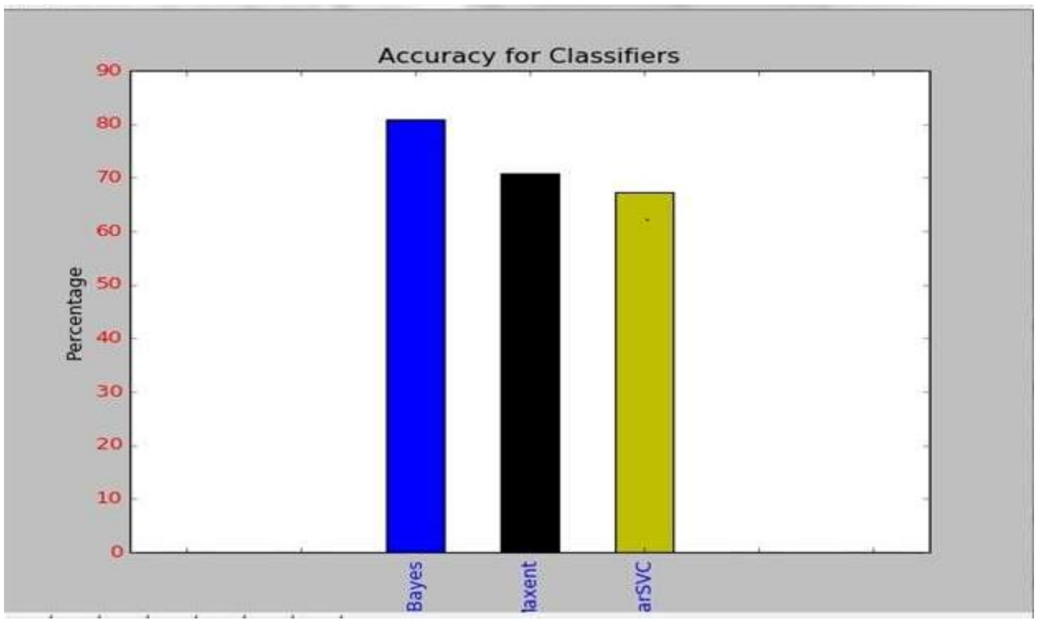


Figure 4.16: Accuracy analysis of user 4

Figure 4.23 shows that the model's accuracy in identifying and classifying hate tweets posted by individual users can be positively analyzed. For user 4, he scores 65% on disseminating hate speech on Twitter. Any person who utters words intended to incite feelings of contempt, hatred, hostility, violence, or discrimination against any person, group, or community-based on ethnicity or race, commits an offense and shall be liable on conviction to a fine not exceeding one million shillings, or to imprisonment for a term not exceeding five years, or both. As mentioned earlier, we decided to include in this category tweets in which the word choices consisted of ‘stereotypic derogations.’ (Burgers, 2014), conclude that verbal irony contributes to the communication and maintenance of stereotypes. Stereotypic derogations refer to expressions of discriminatory epithets and offensive words, expressions of lessening, or detraction, especially of power, reputation, and value-based on stereotypical beliefs. Stereotypical beliefs that target the Kenya community's individual members are the perceptions of them being evolved in criminal activities, being uneducated, and uncivilized. Tweets that include phrases or words that indicate the aforementioned stereotypical characteristics were coded as hateful tweets. According to our results, the hateful tweets' content was in English, along with the locally spoken vernacular languages. Hateful tweets noted was predominantly based on political, ethnicity, and religious affiliation, and much online hate speech comes in reaction to events that transpire or are witnessed offline. Online hate speech disseminators largely identify themselves with a real or fake name and use languages widely understood in Kenya (English, Swahili, and Sheng). In the thesis analysis, we have identified factors for identifying online inflammatory statements on Kenya's Twitter. Comparing a group of people with animals, insects or vermin, sorcerers, wezi (thieves), stone-throwers, or poverty-stricken. Encouraging a particular audience to commit acts of violence towards a group of people

4.5 Discussion.

To study profile presentation, we analyze whether the user provides profile image, location, and timezone; whether the user has enabled the geo-location to be posted along with their tweets; whether Twitter verifies the account; and the length of profile description in characters. We analyze the number of tweets, friends, followers, lists, and retweets for user activity levels. The last three of these indicate how Twitter users interact with an account and are used as visibility measures (Nilizadeh 2016). Through a multi-step classification process, we curate a comprehensive hate speech dataset capturing various types of hate. We study the distinctive characteristics of hate instigators and targets in terms of their profile self-presentation, activities, and online visibility, as shown in figure 4.10.

We find that hate instigators target more popular and high profile Twitter users and that participating in hate speech can result in greater online visibility. We conduct a personality analysis of hate instigators and targets and show that both groups have eccentric personality facets that differ from the general Twitter population. All characteristics can be extracted from the meta-data provided with the tweets, except the retweet count. We count the number of times the user's tweets are reposted in our 1% dataset for every user. Although the obtained retweet counts only represent a subset of the actual

Retweets provide useful insight when comparing different samples. Twitter verifies accounts that are of public interest. When accounts are verified, a blue badge appears next to the user's name on their profile. Interestingly, when comparing HIs and HTs, we observe that HTs include significantly more high profile and established users; 12% belong to verified accounts. However, HIs themselves are less likely to have verified accounts, even compared to random general users. Interestingly, HTs have more friends and post more tweets than both HIs and general users. They also have higher visibility and influence; their median numbers of followers and retweets are larger than those of both HIs and general users.

We examine the visibility of HIs and HTs by controlling for variables that can impact the visibility measures. For example, older accounts have had more time to accumulate followers; following many others usually yields more followers by sheer reciprocity; and posting many tweets can increase the chances to be noticed. These results suggest that regardless of user activity level, profile self-presentation, and gender, more visible Twitter users (with more followers, lists, and retweets) are more likely to become the target of hate.

We have used a few open-source machine learning library during the performance evaluation. Here we give the accuracy of all algorithms over 45000 tweets. From the result, there are four attributes whose are precession, recall, F-Score, and Accuracy. If we only consider the accuracy, then it may be misleading us. Sometimes classifiers have greater predictive power on the problem with lower accuracy may be desirable to select. If we consider a problem where there is a large class imbalance, and there may be classifiers that can predict the majority's value and high classification accuracy achieved by it. Still, this kind of classifier is not useful in the problem domain. The results of the classifiers have been recorded, and Table 4.2 illustrates how each of the three classifiers performed against the four evaluation metrics. Naïve Bayes returned with the highest values in three of the four metrics – accuracy, recall, and F-measure. This indicates that it is a suitable classifier for a balanced dataset. When evaluating classifiers, it is also necessary to consider the precession because precession can measure classifier exactness. A low precession indicates a large number of false positives. Sometimes a classifier may have low accuracy, but it gives high precision. In a problem where exactness is more important than high accuracy than the Precession evaluation is critical. From Fig. 4.3, we see that the precision of Naïve.Bayes classifier is high on 90% for negative tweets compared to 60% for positive tweets. They are thus implying that the classifier's exactness is perfect. Figure 4.4 and figure 4.5 for Linear SVC classifier and Maximum Entropy classifier show their precision is 45% and 43%, respectively, implying they are not good for classification of hate tweets. The results obtained from the three machine learning approaches are based on their precision, recall, accuracy, and F score. Figure 4.3 for Naïve Bayes classifier for positive precision of 61% and a recall of 94%. These are shallow values for precision and recall. It means that only 61% 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 may be that the context of training tweets was mostly hated messages, which means they had many slang and hate words on the negative results since our context is hate speech.

The negative precision came out approximately 90%, and recall was 37%. This means that most of its predicted sentiment was accurate compared to its training set by a small amount. Hence 90% of the negative tweets were relevant, and 37% of the relevant negative tweets were retrieved. This means very few *false positives* were found for the negative class. However, many negative tweets are incorrectly classified. Table 4.1, an unusually large number of mentions, could be explained by the fact that mentions were tracked, so most tweets mention the tracked accounts. Just 14% of these mentions are replies, so most tweets mentioned users, not receiving a reply.

Also, explaining either the replies came from those who felt the hate speech target them positively or negatively. The Naive Bayes classifier is a simple probabilistic classifier based on Bayes theorem with strong and naive independence assumptions. It is one of the most basic text classification techniques. Naive Bayes is convenient since it can be trained very fast. But we can note that the Naive Bayes method results increase when we increase the training dataset's size.

4.5.1 Summary

Micro-blogs' evolution, which has resulted in user enriched information and highly opinionated personal commentary, offers a unique look into people's opinion. Thus, the results of the proposed framework can be beneficial in different aspects. In the study, Tweepy was used as a python library to access and classify Tweets using Naïve Bayes, a Machine Learning technique. The Technique was meant to ease out the process of analysis, summarization, and classification. Additionally, we applied the extracted Twitter sentiment to accomplish two tasks. First, how Twitter users use the platform to disseminate and promote hate speech. Secondly, we determined which words/tweets promote violence and hatred in the site by posting the dataset and finally classifying hate tweets. The study accomplished this by text mining tweets using Twitter's search API and subsequently processing them for analysis and classification. In this thesis, technological tools have been Reviewed, which reports programming languages, development packages, and libraries were meant for implementing machine learning algorithms and applying Natural Language Processing techniques. Python (Python) was chosen as the programming language of preference because of its easy syntax and the larger variety of machine learning and NLP libraries that offer. Moreover, the thesis has provided step by step the methodology and the approach that has been used to:

1. Collect data from Twitter.

2. Analyze the data set.

3. Preprocessing and processing the data with NLP techniques.

4. The features extraction process.

5. The preparation of the train and test datasets.

6. Evaluation of the results of experimentation with Naïve Bayes, Maximum Entropy, and Support Vector Machines algorithms.

As mentioned earlier, we used Twitter API, Tweepy to retrieve tweets from the Twitter Search API. We built a hate classification algorithm. The classifiers were implemented on Pycharm IDE using Python as the programming language. Mongo DB was used for storage. The experimentation results prove that the proposed system manages to classify hate tweets efficiently. The classifier was built using a Machine Learning approach. For the task of determining sentiment, we tested the effectiveness of three machine learning classifiers, Naïve-Bayes Classifier, Maximum Entropy (ME), and Support Vector Machine (SVM). The study can inform educational administrators, law enforcers, and other relevant decision-makers to understand sentiment analysis in Twitter to classify hate tweets. The results provide a workflow for analyzing social media data for educational purposes that will overcome the significant limitations of manual qualitative analysis and large-scale computational analysis of the user-generated textual content, further understanding engineering students' college experiences. It can be concluded that the classification classifier was able to classify data within our domain correctly with a reasonable level of accuracy.

The research has provided an innovative framework for fine-grained sentiment analysis at different levels. The framework has been tested and evaluated in different domains for its accuracy and reliability. The promising results bring us a lot of potential in terms of applying this framework in other areas.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1 Introduction

This chapter analyzes the work conducted in this thesis regarding our goals, how we accomplish them, and the obtained results. After that, we regard the open questions and possible future work in the area.

5.2 Knowledge Contributions

This thesis's work will contribute to a deeper insight into a field that needs more research and can build upon other researchers. An increased amount of research in this area will hopefully improve the methods for removing hateful content currently adopted by online communities. More specifically, the research conducted in this thesis will contribute to the following:

1. These are the determination and classification of the expression of features and identifying the other relevant factors. The purpose of these contributions is to investigate the features within preprocessing data by showing them as feature matrixes and investigating them through factorial experiments concerning the feature's effectiveness in sentiment analysis performance. Further, we tried to identify which factor(s) brought the most significant improvement to the thesis objective's model performance and attainment. To determine and classify the expression of features, eight features were used: emoticons, n-gram, negations, Twitter features, repeated letters, special characters, slang, and stop words.
2. A quantitative analysis of Twitter users' characteristics, comparing users in different target classes based on their tweets.
3. The research and implementation of a baseline hate speech classification model with textual and user-related features classify hate tweets.
4. A thorough comparison of text classification methodologies on datasets from the

literature and a new real-world corpus. The results show that the model proposed in this study provides better performance than previous models in all the datasets considered.

5. Experiments with textual and user-related features in hate speech classification to investigate the effects of individual features and feature subsets.
6. We propose the novel framework that leverages sentiment analysis of hate speech detection and classification in social media, particularly Twitter, using various features.
7. Provided a comprehensive overview of the sentiment analysis by defining hate speech, distinguishing it from related terms, and explaining how previous research work has handled this phenomenon's detection and classification and further build a hybrid supervised learning model.

5.2 Conclusion.

Conclusion and Discussion Mitigating online hate speech is important for reducing its harmful effects on society. This purpose represents one of the major impacts of computational social science on society. The purpose of this research is to help the community manager's spot and remove malicious content by paving ways for automated or computer-aided moderation with the help of machine learning. Earlier efforts in this field have been myriad. Still, they tend to rely heavily on using a dictionary of hateful words, which has been found inadequate and relies on coarse categorizations providing little detailed information on hate targets.

To address these issues, we collected tweets from Twitter from a given period and created training data by identifying hateful comments using human judgment, as encouraged by prior research (Sood S, 2012a). We find that this annotation process is time-consuming but better captures the linguistical diversity of hateful comments than dictionary-based techniques. Applying the classification to the full dataset, we find that the media and the authorities (the police) are highly targeted among the dataset's commenters. In conjunction, we found that surprisingly few comments were targeting other discussants.

Most comments focused on outside targets – people were not arguing amongst themselves, but almost in isolation, or jointly at times, toward external targets. This indicates that isolated social media communities could become powerful catalysts of hate. Our results can be explained in two ways: firstly, by considering recent media controversy and people's political polarization into different camps. Secondly, by the research context: online news media is likely to receive relatively more hate when reporting on political issues than when the topic is entertainment, for example. This suggests that online hate could be different in other contexts and that further research is needed to tie the topics of reportage with the types of hate in the related comments. To identify and analyze techniques used in hate speech monitoring and select the best suitable technique for creating a customized hate speech application. Different hate speech monitoring techniques were analyzed to achieve this objective, including hate speech using machine language, natural processing algorithms, and classifiers.

We have presented the first comparative study of hate speech instigators, targets, and general Twitter users. We have outlined a semi-automated classification approach for the duration of directed explicit hate speech. Our analysis yields several interesting and unexpected findings of actors of hate speech. For example, we found that hate instigators target more visible users and that participating in hate commentary is associated with higher visibility. Additionally, apart from the prototype model's ability to predict a given tweet, whether it is hateful or not, the classifier 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 regarding how they express bigotry, hate, and propaganda.

In this thesis, we set out a hate speech classification framework in social media, especially Twitter. We were successfully able to project the hate content problem on Twitter into a classification problem, which we solved to an extent using a naïve Bayes classifier. Additionally, apart from the system's ability to predict 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 regarding how they express bigotry and propaganda.

Also, we presented an overview of the most recent techniques for detecting and classifying hate speech in social media, specifically Twitter. This research area emerged during the last years, parallel with user participation in social networks. In this overview, we made an effort to organize the most important research lines and their results. Furthermore, we focused on the architecture element of such hate classifier. Due to large volumes of data, state-of-the-art data stream and database frameworks had to be utilized. Finally, we discussed how the evaluation is being executed in hate tweet(s) from data collection, preprocessing, detection, and classifying the tweets using naïve Bayes classifier.

5.3 Future Work

Finally, during the conducted research, the study spotted some opportunities in the field. Future direction is to consider more features in event discovery, such as social network features (community influence detection), visual features (images and video), and semantic features. Finally, another possible direction is to identify some network structure between the users, i.e., if the users are connected, follow the same personalities, etc. The hypothesis is there indeed existing some nexus, although this needs to be verified. In this manner, community detection would provide an extra boost to identifying more aggressors and their usage patterns

REFERENCES

Abbasi, H. C. (2008). A. AbbasSentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *A. Abbasi, H. Chen and A. Salem, "Sentiment analysis in multiple languages: Feature select ACM Transactions on Information Systems.*, 1-34.

Agarwal S, S. A. (2015). Using KNN and SVM Based One-Class Classifier for Detecting Online Radicalization on Twitter. *11th International Conference, ICDCIT 2015, Bhubaneswar, India, February 5-8, 2015. Proceedings* (pp. pp 431–442.). Springer, Cham.

Aggarwal, C. C. (2011). *Social Network Data Analytics.* US: Springer.

Ahktar, J. a., (2009). *Sentiment Analysis: Facebook Status Messages.* Stanford: Stanford University Technical Report.

Akshay Java, X. S. (2007). Why we Twitter: understanding microblogging usage and communities. *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis* (pp. 56-65). San Jose, California: ACM, Digital library.

Akshay, B. A. (2016). Predictive and Corrective Text Input for desktop editor using n-grams and suffix trees. *International Conference on Advances in Human-Machine Interaction.* Bangalore, India.

Alfredo Cobo, D. P. (2015). Identifying relevant messages in a twitter-based citizen channel for natural disaster situations. *In Proceedings of the 24th International Conference on World Wide Web Companion,* (pp. 1189–1194).

Alizadeh, S., Groggel, A., Lista, P., Das, S., Ahn, Y.-Y., KapadiaA., & and Rojas, F. (2016). Twitter's Glass Ceiling: The Effect of Perceived Gender on Online

Visibility. *In ICWSM'16.*

ÁlvaroCuesta, D. F.-M. (2014). A Framework for Massive Twitter Data Extraction and Analysis. *Malaysian Journal of Computer Science*.

Anbananthen, K. S. (2013). 'Evolution of Opinion Mining.' *Australian Journal of Basic and Applied Sciences*, *7(6),* pp.359-370.

Anwar, T., & Abulaish, M. (2012, June). Identifying cliques in dark web forums-an agglomerative clustering approach. In *2012 IEEE International Conference on Intelligence and Security Informatics* (pp. 171-173). IEEE.

Balachander Krishnamurthy, P. G. (2008). A few chirps about Twitter: *In Proceedings of the first workshop on Online social networks* (pp. 19-24). New York: ACM Press.

Banfield, A. (1982). *Unspeakable Sentences: Narration and Representation in the Language of Fiction.* Boston: Routledge & Paul.

Berry, M. W. (2008). *Survey of Text Mining: Clustering, Classification, and Retrieval.*Springer.

Bifet, A. &. (2005). Sentiment knowledge discovery in Twitter streaming data. *In Proc 13th International Conference on Discovery Science* (pp. 1-15). Berlin: Springer.

Bifet, A. &. (2010). Sentiment knowledge discovery in Twitter streaming data. *Proceedings of 13th International Conference on Discovery Science* (pp. 1-15). Berlin, Germany: Springer.

Biz Stone, J. D. (2006). *The team behind Twitter.*

Bjorn Ross, M. R. (2017). Measuring the reliability of hate speech annotations: The Case of the European Refugee Crisis. *Proceedings of NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer-Mediated Communication (Bochum), Bochumer Linguistische Arbeitsberichte, vol. 17*, (pp. pp. 6-9).

Bollen, J. B. (2011). *Happiness Is Assortative in Online Social Networks." Artificial Life.*

Bouras, &. T. (2016). Assisting Cluster Coherency via n-grams and clustering as a tool to deal with the new user problem. *International Journal of Machine Learning and Cybernetics*, 1-14.

Burnap, P. O. (2013). *"Detecting Tension in Online Communities With Computational Twitter Analysis." Technological Forecasting and Social Change.*

Cambria, E. (2013). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*.

Cammaerts, B. (2009). Radical pluralism and free speech in online public spaces: The case of North Belgian extreme right discourses. *International Journal of Cultural Studies.*

Carbonell, J. (1979). *'Subjective Understanding: Computer Models of Belief Systems,' Ph.D. thesis, and Technical Report (150), Department of Computer Science.*

Chalothorn, T. (2012). Using SentiWordNet and Sentiment Analysis for Detecting Radical Content on Web Forums. *In: 6th Conference on Software, Knowledge, Information Management, and Applications (SKIMA 2012),* (pp. 9-11). Chengdu University.

Chen, Y. Y. (2012). Detecting Offensive Language in Social Media to Protect Adolescent Online Safety. *Chen, Y., Y. Zhou, S. Zhu, and H. Xu. 2012. "Detecting Offensive Language in Social MediaIn Proceedings of the Fourth ASE/IEEE International Conference on Social Computing.*

Chikashi Nobata, J. T. (2016). Abusive Language Detection in Online User Content. *In the Proceedings of the 25th International Conference on World Wide Web* (pp. 145-153). Montréal, Québec, Canada: ACM, Press.

Chikashi Nobata, J. T. (2016). Abusive language detection in online user content. *In Proceedings of the 25th International Conference on World Wide Web,* (pp. pages 145–153). , Montréal, Québec, Canada.

Cios, K. J. (2007). *Data Mining: A Knowledge Discovery Approach.* New York: Springer Science and Business Media.

Collie, T. M. (2004). Sentiment Analysis using Support Vector Machines with Diverse I n- formation Sources. *Proceedings of EMNLP-04, 9th Conference on Empirical Methods in Natural Language Processing*, (pp. 412-418).

Commission., N. C. (2013). *The Use of Coded Language and Stereotypes among Kenyan Ethnic Communities.* Nairobi: NCIC.

Coppin, B. (n.d.). *Artificial intelligence is illuminated.* Jones and Bartlett.

Courtenay Honeycutt, S. C. (2009). Beyond Microblogging: Conversation and Collaboration via Twitter. *Hawaii International Conference on System Sciences* (pp. 1- 10). Los Alamitos: IEEE.

Dan Jurafsky, J. H. (2014). *Speech and Language Processing.* Springer Science and Business Media.

Danah Boyd, S. G. (2010). Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. *HICSS '10 Proceedings of the 2010 43rd Hawaii International Conference on System Sciences* (pp. 1-10). USA: ACM Digital.

Daniel Jurafsky, J. H. (2009). *Speech and Language Processing: An introduction to natural language processing, Computational Linguistics, and Speech*

Recognition. United States of America: Prentice-Hall.

Dave K, L. S. (2003). 'Mining the peanut gallery: Opinion extraction and semantic classification of product reviews.' *In Proceedings of WWW,* (pp. pp. 519–528).

Decker (2011). Gangs, Terrorism, and Radicalization. *Journal of Strategic Security, vol. 4, no. 4,* 151-166.

Dennis Njagi, D. H. (2015). A Lexicon-based Approach for Hate Speech Detection.*International Journal of Multimedia and Ubiquitous Engineering* (pp. 215-230).

Denzil Correa, L. (2015). The Many Shades of Anonymity: Characterizing Anonymous Social Media Content. *Proceedings of the Ninth International AAAI Conference on Web and Social Media(ICWSM).*

Dhande, L. &. (2014). Analyzing Sentiment of Movie Review Data using Naive Bayes Neural Classifier. *International Journal of Emerging Trends and Technology in Computer Science,3,* 313-320.

Diana Maynard. (2012). Diana Maynard, Kalina Bontcheva, and Challenges in developing opinion mining tools for social media. . *Diana Maynard, Challenges in developing opinion mining to In Proceedings of @NLP can u tag #usergeneratedcontent?* Turkey.

Dietterich (2000). *"Ensemble methods in machine learning." Multiple classifiers.* r Berlin Heidelberg: Springer.

Ding, X., Liu, B., & Yu, P. S. (2008, February). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 231-240).

Djuric N, Z. J. (2015). Hate Speech Detection with Comment Embeddings. *In:*

Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion (pp. 29–30.). New York, NY, USA: ACM.

Douglass S, M. S. (2016)." They were just making jokes": Ethnic/racial teasing and discrimination among adolescents. *Cultural Diversity and Ethnic Minority Psychology*, 69–82.

Dulac-Arnold, L. &. (2011). Dulac-Arnold, G., Ludovic, D., L., & Gallinari, P. (2011). Text Classification: A Sequential Reading Approach. *Advances in Information*, 411–423.

Efthimiadis, E. N. (2010). conversational tagging on Twitter. *Proceedings of the 21st ACM conference on Hypertext and hypermedia* (pp. 173-178). New York: ACM Press.

Erik, e. a. (2013). "New avenues in opinion mining and sentiment analysis.*IEEEIntelligent Systems*.

Feldman, R. &. (2007). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* Cambridge, Cambridge University.

Fersini, E. E. (2004). *Sentiment analysis: Bayesian Ensemble Learning." Decision Support Systems 68.*

Firat Tekiner, Y. T. (2009). Highly scalable Text Mining - parallel tagging application. *In Proceedings of IEEE 5th International Conference On Soft Computing With Words and Perception In System Analysis Decision and Control (ICSCCW).* China: IEEE.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. . *Journal of Machine Learning Research 3*, 1289-1305.

Fortuna P, N. S. (2018, July). A survey on automatic detection of hate speech in the text. *ACM Computing Surveys*.

Gamallo, P. a., (2014). "Citius: A Naive - Bayes Strategy for Sentiment Analysis on English Tweets. . *SemEval 20 14*, 171.

Gamon, M. A.-O. (2005). *Pulse: Mining customer opinions from free text.*

Gamon, M. C.-O. (2005). Mining customer opinions from free text. *Proceedings of IDA-05, the 6th International Symposium on Intelligent Data Analysis* (pp. 121–132). Springer- Verlag.

George Valkanas, D. G. (2013). *Event Detection from Social Media Data.*

Giannasi, P. (2014). *Hate on the Internet: Progress on the UK Government's Hate Crime Action Plan, presented at the All Wales Hate Crime Criminal Justice Board.* Cardiff.

Go, A. R. (2009). *Twitter sentiment classification using distant supervision.* Stanford: Stanford.

Go, R. B. (2009). *Twitter Sentiment Classification using Distant Supervision CS224N Project Report, Stanford, vol. 1, p. 12.*

Guyon, I. a. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research 3*, 1157–1182.

Hardeniya, N. (2015). *NLTK Essentials.* Birmingham, UK: Packt Publishing Ltd.

Harris, Z. S. (n.d.). *Distributional Structure Synthese Language Library*. Netherlands, Dordrecht: Springer.

Hassan Saif, Y. H. (2012). Alleviating data sparsity for Twitter sentiment analysis. . *In Proceedings, 2nd Workshop on Making Sense of Microposts (#MSM2012) in* *conjunction with WWW 2012*, Lyon, France.

Hearst, M. a., (1992). 'Direction-Based Text Interpretation as an Information Access Refinement'. . *Text-Based Intelligent Systems.*

Hernández, R. L. (2012). Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers. *Neurocomputing*, 98-115.

Hinrich Schütze, P. R. (2008). *Introduction to Information Retrieval.* England: Cambridge University Press.

Hughey, M. W., & Daniels, J. (2013). Racist comments at online news sites: a methodological dilemma for discourse analysis. *Media, Culture & Society*, *35*(3), 332-347.

Jacobs, J. B. (2000). *Hate crimes: Criminal Law and Identity Politics.* . Oxford University Press.

Jansen, B. J., Zhang, M., Sobel, K., & Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, *60*(11), 2169-2188.

Jebaseeli, A. N. (2012). "A Survey on Sentiment Analysis of (Product) Reviews.*International Journal of Computer Applications*.

John Hunter, D. D. (2017). *matplotlib.* Retrieved from Python 2D plotting library, [Online].: http://matplotlib.org.

Johnson, S. (2010, 9 August). *How Twitter Will Change the Way We Live.* Retrieved from http://www.time.com/time/printout/0,8816,1902604,00.html

Julian Fierrez, A. M.-R. (2017). *Multiple classifiers in biometrics.Trends and challenges.*Information fusion.

Jurafsky, D. a., (2008). .*Speech and language processing (2nd edition) (prentice hall* *series in artificial intelligence) (2 ed.).* Prentice-Hall.

Justin Chang, C. D.-N.-M. (2015). Antisocial Behavior in Online Discussion Communities.*Proceedings of the 9th International AAAI Conference on Web and Social Media.*

Kalchbrenner N, G. E. (2014). A Convolutional Neural Network for Modelling Sentences. *In Proceedings of ACL,* (pp. 655–665).

Kešelj, P. C. (2003). N-gram-based author profiles for authorship attribution. *in Pacific Association for ComputationalLINGuistics(PACLING'03), (pp. 25-264).* Canada.

Khan, F. H. (2016). 'SentiMI: Introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection'. . *Applied Soft Computing, 39,* pp.140-153.

Korde, V. &. (2012). Text Classification and classifiers. *International Journal of Artificial intelligence and applications,* 85.

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. Handling imbalanced datasets: A review, GESTS *International Transactions on Computer Science and Engineering 30* (2006) 25–36. Synthetic Oversampling of Instances Using Clustering.

Kouloumpis, E., Wilson, T., & Moore, J. (2011, July). Twitter sentiment analysis: The good the bad and the omg!. In *Fifth International AAAI conference on weblogs and social media*.

Kwok and Y. Wang. (2013). Locate the Hate: Detecting Tweets against Blacks. *Proceedings of the 27thI. Kwok and Y. Wang, "Locate the Hate: DeteProceedings of the 27th National Conference on Artificial Intelligence (AAAI),* (pp. 1621-162).

Lam Hong, L. C. (2010). A Review of Nearest Neighbor-Support Vector Machines Hybrid Classification Models. *Journal of Applied Sciences*, 1841-1858.

Leandro Araújo Silva, M. M. (2016). Analyzing the Targets of Hate in Online Social Media. *In Proceedings of the 10th international conference on web and social media. (ICWSM)*, (pp. 687-690).

Lee, P. &. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *Proceedings of the 43rd Annual Meeting of the ACL* (pp. 115- 124.). ACL.

Levy, L. W., Karst, K. L., & Winkler, A. (2000). *Encyclopedia of the American Constitution,2nd edition.* New York: Macmillan.

Leysia Palen, A. L. (2009). Twitter Adoption and Use in Mass Convergence and emergency events. *International Journal of Emergency Management,.*, 248 - 260.

Liu, B. (2012). 'Sentiment analysis and opinion mining.' *Synthesis lectures on human language technologies, 5(1),* pp.1-167.

Liu, B., H. M. (2005). 'Opinion observer: analyzing and comparing opinions on the web.' *Proceedings of the 14th International World Wide Web Conference,* (pp. pp. 342-51.). New York, NY, ACM Press.

Majumder, P. M. (2002). N-gram: A language-independent approach to IR and NLP. *In International Conference on universal knowledge and language.*

Maks Isa, V. P. (2012). A lexicon model for deep sentiment analysis and opinion mining applications. *Decis Support Syst*, 53,680-688.

Maynard, D. G. (2016). Challenges of Evaluating Sentiment Analysis Tools on Social Media', Proceedings of LREC. *Proceedings of the Tenth International*

Conference on Language Resources and Evaluation (LREC).

McNamee L. G, P. B. (2010). A Call to Educate, Participate, Invoke, and Indict: Understanding the Communication of Online Hate Groups. *Communication Monographs, Vol. 77, No. 2,* 257-280.

Medhat, W. H. (2004). 'Sentiment analysis algorithms and applications: A survey. *' Ain Shams Engineering Journal, 5(4),* pp.1093-1113.

Mehdi Allahyari, K. J. (2014). Ontology-Based Text Classification into Dynamically Defined Topics. *Semantic Computing (ICSC), 2014 IEEE International Conference.* Newport Beach, CA, USA: IEEE.

Michael Haenlein, A. M. (2010). Users of the world, unite! The challenges and opportunities of social media. *Business Horizons-Elsevier*, 59-68.

Mike Thelwall, P. W. (2008). Information-Centred Research for Large-Scale Analysis of New Information Sources. *Journal of the American Society for Information Science and Technology*, 1523-1527.

Mondal, M, S. L. (2017). A Measurement Study of Hate Speech in Social Media, HT '17. *Of the 28th ACM Conference on Hypertext and Social Media* (pp. pp. 85–94.). New York, NY, USA: ACM.

Mor Naaman, J. B.-H. (2010). Is it Really About Me? Message Content Social in social awareness streams. *Proceedings of the 2010 ACM conference on Computer supported cooperative work* (pp. 189-192). New York: ACM, Press.

Muthiah S, H. B. (2015). *Planned Protest Modeling in News and Social Media.*

Najaf Ali Shah, E. M. (April 02 - 03, 2004). Topic-based clustering of news articles.*ACM-SE 42 Proceedings of the 42nd annual Southeast regional conference* (pp. 412-413). Huntsville, Alabama: ACM Digital Library.

Nemes, I. (2002). Regulating Hate Speech in Cyberspace: Issues of Desirability and Efficacy, Information & Communications Technology Law. *Information & Communications Technology Law 11(3)*, 193-220.

Neviarouskaya, P. &. (2007). *Textual affect sensing for sociable and expressive online communication.*

Nielsen, L. (2002). Subtle, Pervasive, Harmful: Racist and Sexist Remarks in Public as Hate Speech. *Journal of Social Issues, Vol. 58, No. 2,* 265—280.

Nobata, C, T. J. (2016). Abusive Language Detection in Online User Content. *In Proceedings of the 25th International Conference on World Wide Web, WWW '16, Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee,* (pp. pp. 145–153.). Canton of Geneva, Switzerland.

Pak, A., & Paroubek, P. (2010, May). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc* (Vol. 10, No. 2010, pp. 1320-1326).

Pang and L. Lee. (2004). "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," *in Proceedings of the 42nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2004,* (pp. 271–278). Barcelona, Spain.

Pang, B. &. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *In Proceedings of the 42nd Annual Association for Computational Linguistics Conference* (pp. 271–278). Morristown: Association for Computational.

Pang, B. &. (2008). Foundations and Trends in Information Retrieval. *Opinion mining and sentiment analysis* (pp. 1-135).

Pang, B. a., (2008). 'Opinion mining and sentiment analysis,' 2(1-2). *Foundations and trends in information retrieval*, pp.1-135.

Pang, B. L. (2002). "Thumbs up? Sentiment classification using machine learning techniques". *Proceedings of the ACL - 02 Conference on Empirical Methods in Natural Language Processing - Vol 10* (pp. 79-86). ACL.

Parekh, B. (2006). *Hate Speech: Is there a case for banning?, Public Policy Research*, Cambridge: Cambridge University Press.

Parikh, R., & Movassate, M. (2009). Sentiment analysis of user-generated twitter updates using various classification techniques. *CS224N Final Report*, *118*.

Paroubek, A. P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Conference: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010,* Valletta, Malta.

Perkins, J. (2014). *Python 3 Text Processing with NLTK 3 Cookbook*. Birmingham, UK: Packt Publishing Ltd.

Po-Wei Liang, B.-R. D. (2013). Opinion Mining on Social Media Data. *IEEE 14th International Conference on Mobile Data Management - Volume 02* (pp. 91-96). Washington, DC: ACM, Press.

Prentice S, T. P. (2011). Analyzing the semantic content and persuasive composition of extremist media: A case study of texts produced during the Gaza conflict. *Information Systems Frontiers*, 61–73.

Qin, J., Zhou, Y., & Chen, H. (2011). A multi-region empirical study on the internet presence of global extremist organizations. *Information Systems Frontiers*, *13*(1), 75-88.

Rajadesingan A, Z. R. (2015). Sarcasm detection on Twitter: A behavioral modeling approach. *In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining* (pp. 97–106.). ACM.

Rambocas, M. a. (2013, April). Marketing Research: The Role of Sentiment Analysis. *Universidade do Porto; Apr2013, Issue 489, preceding p1*.

Raphael, C. A. (2011). Fighting Hate and Bigotry on the Internet", Policy and Internet.

Rebecca, B. W. (1999). 'Development and Use of a Gold-Standard Data Set for.' *Proceedings of the Association for Computational Linguistics (ACL).*

Riloff, E. P. (2006). Feature subsumption for opinion analysis. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (pp. 440–448). Morristown: Association for Computational Linguistics.

Robertson, A. M. (1998). Applications of N-Grams In TextualInformation Systems.*Journal of Documentation, 54 (1), 48-67.*, 48-67.

Rosti, A. V., Matsoukas, S., & Schwartz, R. (2007, June). Improved word-level system combination for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (pp. 312-319).

Rowe, M, S. M. (2015). *A Topical Crawler for Uncovering Hidden Communities of Extremist Micro-Bloggers on Tumblr.*

Saif M. Mohammad, P. S. (2016). Stance and Sentiment in Tweets. *Special Section of the ACM Transactions on Internet Technology (TOIT) on Argumentation in Social Media, in press*.

Saif, H. H. (2012). Semantic Smoothing for Twitter Sentiment Analysis. *The 11th International Semantic Web Conference ISWC 2012.* Boston- USA.

Saleem HM, D. K. (2017).) A Web of Hate: Tackling Hateful Speech in Online Social

Spaces,arXiv:1709.10159 [cs.CL]. *Computation and Language*.

Samani, M. H. (2015). A Content-based Method for Persian Real-Word Spell checking. *In Information and Knowledge Technology (IKT), 2015 7th Conference* (pp. 1-5). IEEE.

Shabajee, P. a., (2012). *Shabajee' What is Annotation? A Short Review of Annotation and Annotation Systems, ILRT Research Report, No. 1053. Institute for Learning & Research Technology*.

Shang, W. H. (2006). "A Noval Feature Selection Algorithm for text categorization.". *Elsevier, Science Direct Expert system with the application*, pp.1-5.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *BellSystem Technical Journal*, 379-423.

Siqueira, H. a., (2010). A feature extraction process for sentiment A feature extraction process for sentiment analysis of opinions on services. *A feature extraction process for sentiment anaProceedings of International Workshop on Web and Text Intelligence.*

Sitaram Asur, B. A. (2010). *Predicting the Future With Social Media.* Palo Alto, California: Social Computing Lab, HP Labs.

Songdo, a. J. (2008). An empirical study of sentiment analysis for Chinese documents.*Expert Systems with Applications*, 2622-2629.

Sood S, A. J. (2012a). Profanity Use in Online Communities. *In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12* (pp. 1481–1490.). New York, NY, USA: ACM.

Stats, I. L. (2018). *Twitter Usage Statistics.* Retrieved from Internet Live

Stats: http://www.internetlivestats.com/twitter-statistics/

Strapparava, C. &. (2008). Learning to identify emotions in text. *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 1556-1560). New York: ACM.

Swati Agarwal, A. S. (2015). Using kNN and SVM Based One-Class Classifier for Detecting Online Radicalization on Twitter *in the 11th International Conference on Distributed Computing and Internet Technology.*

Thelwall, M. K. (2010). Sentiment Strength Detection in Short Informal Text. *Thelwall, M., K. Buckley, G. Paltogou, D. Cai, and A. Kappas. 2010a. "Sentiment Strength Detection Journal of the American Society for Information Science and Technology 61*, 2544– 58.

Thelwall, M. K. (2011). Sentiment in Twitter events. *Thelwall, M., K. Buckley, and. Paltogou. 2011. "SentimeJournal of the American Society for Information Science and Technology*, 406-418.

Thilagavathi, N. &. (2014). Content-Based Filtering in Online Social Network using Inference Algorithm. *In-Circuit, Power, and Computing Technologies (ICCPCT), 2014 International Conference* (pp. 1416 - 1420). IEEE.

Turney, P. D. (2002). "Thumbs up or thumbs down? : semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th annual meeting on association for computational linguistics.* Association for Computational Linguistics.

Varghese, R. a., (2013). *A Survey On Sentiment Analysis And Opinion Mining.*

Vinodhini, G. &. (2012). Sentiment Analysis and Opinion Mining: *A Survey. International Journal of Advanced Research in Computer Science and Software Engineering*, 2-6.

Vosoughi, S., Zhou, H., & Roy, D. (2016). Enhanced twitter sentiment classification using contextual information. arXiv preprint arXiv:1605.05195.

Walker, S. (1994). *Hate Speech: The History of an American Controversy.* U of Nebraska Press.

Wang, I. K. (2013). Locate the Hate: Detecting Tweets against Blacks. *Proceedings of the 27th National Conference on Artificial Intelligence (AAAI,* (pp. 1621-162).

Wang, X. F. (2011). Topic sentiment analysis in Twitter: a graph-based hashtag sentiment classification approach. *In Proceeding of the ACM conference on Information and knowledge management (CIKM-2011).*

Warner, W., & Hirschberg, J. (2012, June). Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media* (pp. 19-26).

Wiebe, J. a., (2005). Creating subjective and objective sentence classifiers from unannotated texts. *Computational Linguistics and Intelligent Text Processing* (pp. 486- 497). Berlin Heidelberg: Springer.

Wiebe, J. T. (2005). Annotating Expressions of Opinions and Emotions in Language.*AnnotatinLanguage Resources and Evaluation 39,* 165-210.

Wiebe, J. W. (2004). 'Learning a subjective language.' *Computational Linguistics, 30(3),* pp.277–308.

Wiegland, A. S. (2017). A survey on hate speech detection using natural language processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media* (pp. 1–10).

Williams, P. B. (2015). Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modeling for Policy and Decision Making. *Policy*

& Internet, ,pp. 223–242.

Williams, P. B. (2016). "Us and them: identifying cyber hate on Twitter across multiple protected characteristics." *EPJ Data Science*.

Wilson, T. W. (2009). Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level. Sentiment Analysis. *Computational Linguistics,* 399–433.

Wilson, T. W. (2009). Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. . *Computational linguistics*, 399-433.

Wilson, W. &. (2006). Recognizing strong and weak opinion clauses. *Computational Intelligence, 22(2)*, 73-99.

Witten, I. &. (2005). *Data mining: Practical machine learning tools and techniques.* San Francisco: San Francisco: Morgan Kaufmann Publishers.

Wollmer, M. ((2013). YouTube Movie Reviews: Sentiment Analysis in an Audio- Visual Context. *Intelligent Systems, IEEE 28* (pp. 46-53.). IEEE.

Wright, L, R. D. (2017). Vectors for the Counter speech on Twitter. *In: Proceedings of the First Workshop on Abusive Language Online,* (pp. 57–62.).

Wu, Y. Z. (2009). Phrase dependency parsing for opinion mining. *Wu, Y., Zhang, Q., Huang, X., and Wu, L. (2009). Phrase dependency In 14th Conference on Empirical Methods in Natural Language Processing.* (pp., 1533-1541). Singapore: Association for Computational Linguistics.

Yang, P. (1997). A Comparative Study on Feature Selection in TextCategorization.*In Proceedings of the 14th ICML97* (pp. 412-420). ICML.

Yasotha, C. &. (2015). Automated Text Document Categorization. *In 2015 IEEE Seventh International Conference on IntelligentComputing and Information*

Systems (pp. 522-528). IEEE.

Yi, J. T. (2003). 'Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques',. *Proceedings of the IEEE International Conference on Data Mining (ICDM).* Zhai.

Yousef, A. H. (2014). *Sentiment Analysis Algorithms and Applications: A Survey.*

Yu-hwan Kim, S.-Y. H.-T. (2000). *Text filtering by boosting naive Bayes classifiers.* Sigir: ACM Press.

Yung-Ming Li, T.-Y. L. (2013). Deriving market intelligence from microblogs.*Decision Support Systems*, 206-217.

Waseem, Z., & Hovy, D. (2016, June). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop* (pp. 88-93).

Zeerak, W. (2016). Are you a racist, or am I seeing things? Annotator influence on hate speech detection on Twitter. *In Proceedings of the 1st Workshop on Natural Language Processing and Computational Social Science*, (pp. 138–142.).

APPENDICES

Appendix I: Sample codes Twitter Collecting code.

```
from PySide import QtGui

from PySide.QtCore import QThread, SIGNAL

import design

import tweepy,json #https://github.com/tweepy/tweepy

from tweepy import Cursor

from tweepy import OAuthHandler

from tweepy import API

from tweepy import Cursor

from bson.json_util import dumps # from includeFn.config import * import pandas as
pd

import matplotlib as mpl

#mpl.use('Agg')

import matplotlib.pyplot as plt

from matplotlib.ticker import MaxNLocator from matplotlib import rcParams,style
#from mpltools import style

from matplotlib import dates from datetime import datetime #import seaborn as sns

import numpy as np

from pymongo import MongoClient

import os

# The consumer key and secret will be generated for you (paste below)

consumer_key = "Z8hnmRZj6xNcWfTfmyr4bHzoC"

consumer_secret =
"DmtsisFs1zP0M8ZbsirS17ouT0CKjmzDvKuBfZoG8uJ2jdqVRg"


# Create an access token under the the "Your access token" section. # Refresh the app
page to get the access token values (paste below)
```

```
access_key = "719108715182493696-RBPwjK8D3uS6x0zzMoqp92BbGQLF7cD"

access_secret = "Px1eM7bVZ01m4yRyyG0qddWOTsgWcPnL8sFRH6sBDPIDY"


import re

from bson.json_util import dumps

from nltk.corpus import stopwords

from includeFn.json_util import ConcatJSONDecoder

from includeFn.utilities import tweet2json

from includeFn.config import *

import nltk,table_functions

from uiForms import tweetAnalysis

from pymongo import MongoClient

import logging,time,tweepy,csv,codecs,json,operator

import uiForms.tweetAnalysis,uiForms.tweet_pre as tweet_pre

import uiForms.svm as svm_classifier,uiForms.tweetclassification as
tclassify,uiForms.chartfinal,uiForms.bioinfo as bioinfo,uiForms.followers as
followers,uiForms.negativetweets as negtweets,uiForms.positivetweets as postweets
from uiForms import maximum_entropy

class Fetch_kenyan_tweets(QThread):

def __init__(self): QThread.__init__(self)
def del (self):

self.wait()


def run(self, *args, **kwargs):


MAX_TWEETS = 1500

# Twitter only allows access to a users most recent 3240 tweets with this method #
authorize twitter, initialize tweepy

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_key, access_secret)

t = tweepy.API(auth, retry_count=3, wait_on_rate_limit=True)


# initialize a list to hold all the tweepy Tweets

alltweets = []

#
keywords=['karau','sanse','madingo','police','circumcised','mtahiri','terririst','dogs','isla
m terrorist','muslim dogs','chinja cninja','beat','loot','riot','kill ','drive out
tribes','mawe','stupid','stupids','diots','wajinga','jinga','bure','mjinga','mjinga

wee','nyiny','hynna','chuki','fifina','hater','hatred','hate
monger','tribal','raila','kibera','raila
baba','jakom','tinga','rao','masa','mathee','madhaa','mthe','mama','ushango','ocha','mosh
atha','kill','shoot','shot','shut','kil','murder
','sex','fuck','fuuck','s3xseex','tobwa','najisi','mjinga','faala','jinga','jiga']



# make initial request for most recent tweets (200 is the maximum allowed count) #
new_tweets = api.geo_search(query="KENYA",granularity="country")

places = t.geo_search(query="KENYA", granularity="country") place_id =
places[0].id


tweetss = Cursor(t.search, q="place:%s" % place_id).items(MAX_TWEETS)
```

```
# Use tweetdetails database

client = MongoClient('localhost', 27017) db = client.tweetdetails


# try:

# c = MongoClient() # db = c.tweetdetails

# except Exception as e:

# print(e)

# cs = open("files/text/tweets.txt", "w") # cs = open(file, "w", encoding="utf8") for tweet
in tweetss:

alltweets.append(json.loads(json.dumps(tweet._json)))

# cs.write(dumps(tweet._json) + '\n')


# Decode JSON

datajson = json.loads(json.dumps(alltweets)) db.tweets.insert(datajson)

self.output = tweet.text + " | " + tweet.place.name

print(tweet.user.screen_name +" | "+tweet.text + " | " + tweet.place.name if tweet.place
else "Undefined place") self.output=tweet.user.screen_name +" | "+ tweet.text + " | "
+ tweet.place.name self.emit(SIGNAL('add_post(QString)'), self.output) # send data
to textield

self.sleep(2)
```

Appendix II: Train model code.

```
# read the sample tweets

tweets = [] negativelist = [] positivelist = []

for line in tweetsample: sentiment = line[-1] tweet = line[0]

# print(tweet) tweets.append((processsampletweet(tweet), sentiment)) if sentiment ==
"negative":

for word in processsampletweet(tweet): negativelist.append(word)

elif sentiment == "positive":

for word in processsampletweet(tweet): positivelist.append(word)

# ------all words from sampletweet which are manually classifed put to a list----

def get_words_in_tweets(tweets): all_words = []

for (words, sentiment) in tweets: all_words.extend(words)

return all_words

def get_word_features(wordlist): wordlist = nltk.FreqDist(wordlist) word_features =
wordlist.keys() return word_features

word_features = get_word_features(get_words_in_tweets(tweets)) featureList =
get_words_in_tweets(tweets)

# print(word_features)

def extract_features(document): document_words = set(document) features = {}

for word in word_features:

features['contains(%s)' % word] = (word in document_words) return features

# read and convert text to json

tweetFile = open('files/text/{0}.txt'.format(self.name))

tweet_list = json.loads(tweetFile.read(), cls=ConcatJSONDecoder)
tweet2json(tweet_list, "files/json/file.json")

tweet_file = open("files/json/file.json")

tweets_list = json.loads(tweet_file.read(), cls=ConcatJSONDecoder)

# print(processedtweet)

# classify tweets
```

```
training_set = nltk.classify.util.apply_features(extract_features, tweets) test_set =
nltk.classify.util.apply_features(extract_features, tweets)

# print(training_set)

NBClassifier = nltk.NaiveBayesClassifier.train(training_set)
print(NBClassifier.show_most_informative_features(n=10))

posfeats = [] negfeats = []
```

```
tweet_no=len(tweet_list)#total number of tweets in the tweet_list print(tweet_no)
for line in tweets_list[0:tweet_no]:
processedtweet = tweetprocess(line) print()
```

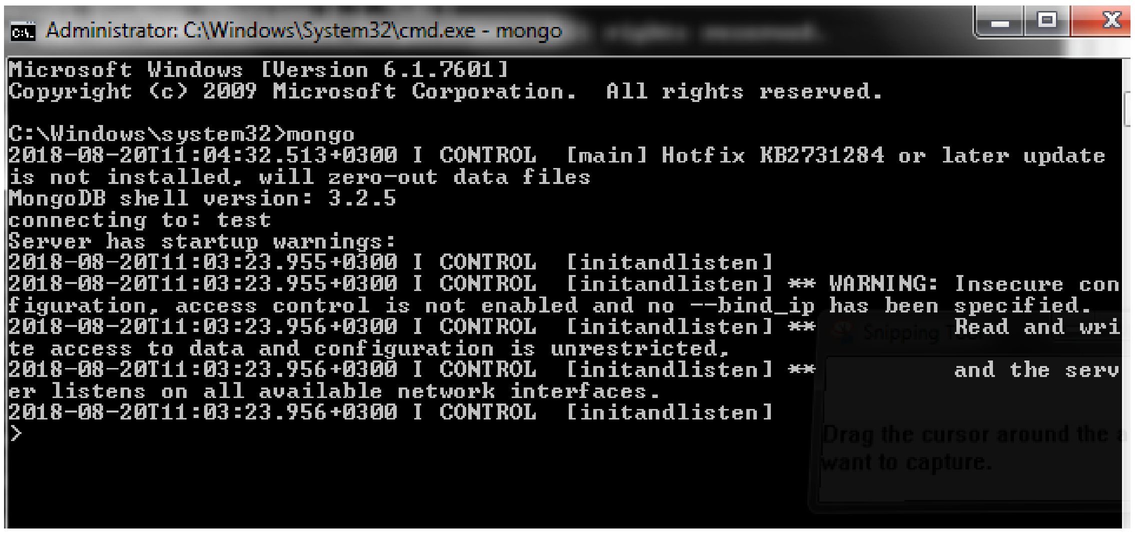
Appendix III: Installation instructions.

a) Python and Virtual environment installation.

1. Python comes preinstalled in Linux. If you are using windows, search, download, and install python 2.7.
2. Virtual environment a tool to keep the dependencies required by different projects in separate places. It is installed by either using the setup tools or git cloning using the virtual-clone script https://pypi.python.org/pypi/virtualenv-clone.
3. After python and the virtual environment are set up, create an environment by entering the command: virtualenv virtual_environment_name. Activating the virtual
4. the environment will be by the command: source virualenvname/bin/activate or C:\windows\path\to\created\env\activate for Linux or windows, respectively.

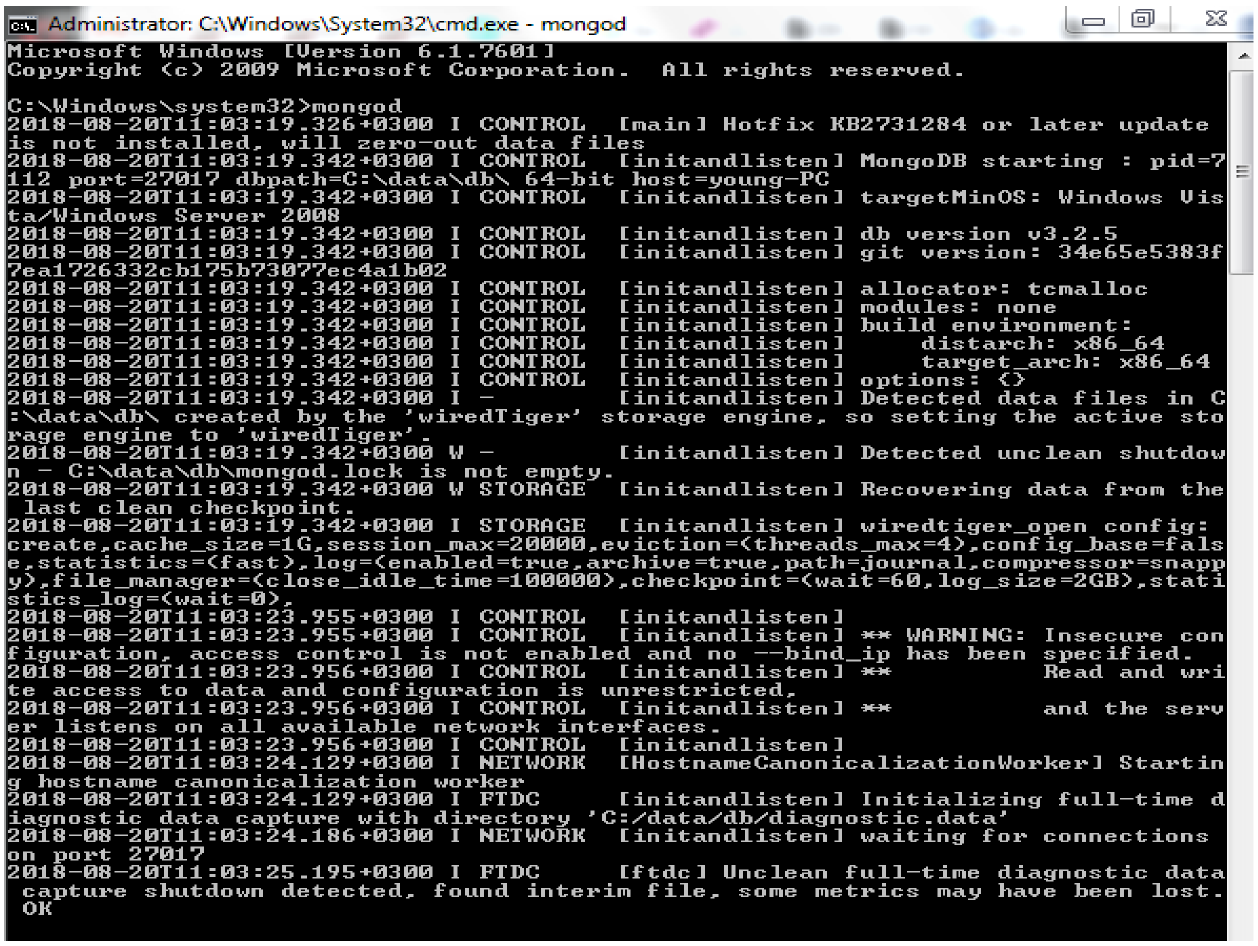
b) MongoDB Installation.

1. Download and install the latest version of MongoDB http://www.mongodb.org/. N/B is an open-source NoSQL document database.(**C:\Program Files\MongoDB\Server\3.4\**
2. To install Python MongoDB on Linux or Windows, do the following: use $ pip install pymongo
3. **Create a data directory where all data is stored.** On C: drive create a folder data inside it create a folder DB or Run.
4. **To start MongoDB.Click Run.**
5. **To connect to MongoDB.** Open another command prompt as administrator and run->



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>mongo
2018-08-20T11:04:32.513+0300 I CONTROL  [main] Hotfix KB2731284 or later update
is not installed, will zero-out data files
MongoDB shell version: 3.2.5
connecting to: test
Server has startup warnings:
2018-08-20T11:03:23.955+0300 I CONTROL  [initandlisten]
2018-08-20T11:03:23.955+0300 I CONTROL  [initandlisten] ** WARNING: Insecure con
figuration, access control is not enabled and no --bind_ip has been specified.
2018-08-20T11:03:23.956+0300 I CONTROL  [initandlisten] **          Read and wri
te access to data and configuration is unrestricted,
2018-08-20T11:03:23.956+0300 I CONTROL  [initandlisten] **          and the serv
er listens on all available network interfaces.
2018-08-20T11:03:23.956+0300 I CONTROL  [initandlisten]
>
```

Ready MongoDB Mongod



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>mongod
2018-08-20T11:03:19.326+0300 I CONTROL  [main] Hotfix KB2731284 or later update
is not installed, will zero-out data files
2018-08-20T11:03:19.342+0300 I CONTROL  [initandlisten] MongoDB starting : pid=7
112 port=27017 dbpath=C:\data\db\ 64-bit host=young-PC
2018-08-20T11:03:19.342+0300 I CONTROL  [initandlisten] targetMinOS: Windows Vis
ta/Windows Server 2008
2018-08-20T11:03:19.342+0300 I CONTROL  [initandlisten] db version v3.2.5
2018-08-20T11:03:19.342+0300 I CONTROL  [initandlisten] git version: 34e65e5383f
7ea1726332cb175b73077ec4a1b02
2018-08-20T11:03:19.342+0300 I CONTROL  [initandlisten] allocator: tcmalloc
2018-08-20T11:03:19.342+0300 I CONTROL  [initandlisten] modules: none
2018-08-20T11:03:19.342+0300 I CONTROL  [initandlisten] build environment:
2018-08-20T11:03:19.342+0300 I CONTROL  [initandlisten]     distarch: x86_64
2018-08-20T11:03:19.342+0300 I CONTROL  [initandlisten]     target_arch: x86_64
2018-08-20T11:03:19.342+0300 I CONTROL  [initandlisten] options: {}
2018-08-20T11:03:19.342+0300 I -        [initandlisten] Detected data files in C
:\data\db\ created by the 'wiredTiger' storage engine, so setting the active sto
rage engine to 'wiredTiger'.
2018-08-20T11:03:19.342+0300 W -        [initandlisten] Detected unclean shutdow
n - C:\data\db\mongod.lock is not empty.
2018-08-20T11:03:19.342+0300 W STORAGE  [initandlisten] Recovering data from the
 last clean checkpoint.
2018-08-20T11:03:19.342+0300 I STORAGE  [initandlisten] wiredtiger_open config:
create,cache_size=1G,session_max=20000,eviction=(threads_max=4),config_base=fals
e,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snapp
y),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),stati
stics_log=(wait=0),
2018-08-20T11:03:23.955+0300 I CONTROL  [initandlisten]
2018-08-20T11:03:23.955+0300 I CONTROL  [initandlisten] ** WARNING: Insecure con
figuration, access control is not enabled and no --bind_ip has been specified.
2018-08-20T11:03:23.956+0300 I CONTROL  [initandlisten] **          Read and wri
te access to data and configuration is unrestricted,
2018-08-20T11:03:23.956+0300 I CONTROL  [initandlisten] **          and the serv
er listens on all available network interfaces.
2018-08-20T11:03:23.956+0300 I CONTROL  [initandlisten]
2018-08-20T11:03:24.129+0300 I NETWORK  [HostnameCanonicalizationWorker] Startin
g hostname canonicalization worker
2018-08-20T11:03:24.129+0300 I FTDC     [initandlisten] Initializing full-time d
iagnostic data capture with directory 'C:/data/db/diagnostic.data'
2018-08-20T11:03:24.186+0300 I NETWORK  [initandlisten] waiting for connections
on port 27017
2018-08-20T11:03:25.195+0300 I FTDC     [ftdc] Unclean full-time diagnostic data
 capture shutdown detected, found interim file, some metrics may have been lost.
 OK
```

To start & stop MongoDB run Type to start: Type Mongodb

c) Twitter Scraper installation.

1. Log in to your Twitter account using your credentials. Redirect to the Twitter developer's site and create an app. You will be issued a set of the App’s keys useful for authentication to the Twitter platform.
2. Copy the keys and fill them appropriately in the twitter scraping script. The keys should remain private to you.
3. Select the phrase you would like to consider and enter it in the track field within the script.
4. Ensuring the database server is running, run the script either in your local machine or in a server similar to the Facebook script.