Identification of Hate Speech in Social Media

N.D.T. Ruwandika

Index No: 13001051

Supervisor: Dr. A. R. Weerasinghe

December 2017

Submitted in partial fulfillment of the requirements of the

B.Sc in Computer Science Final Year Project (SCS4124)

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: N.D.T. Ruwandika

……

Signature of Candidate Date:

This is to certify that this dissertation is based on the work of

Ms. N.D.T. Ruwandika

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Supervisor Name: Dr. A. R. Weerasinghe

……

Signature of Supervisor Date:

Abstract

Hate speech on social media is a common issue seen at present which is growing really fast. Due to the growth of online hate content there's a huge influence for the increase of hate crimes in the society. So, if an accurate efficient methodology can be found to control the online hate content, it will be a great relief to the society. This research represents a study carried out to compare different techniques for the task of hate speech identification of a local English dataset.

A new dataset was created using comments published in a news site of Sri Lanka. We have achieved a dataset of 1500 comments which includes 421 hate comments and 579 comments without hate speech. Totally 1000 comments are annotated out of 1500 comments. We have evaluated and compared different classifiers with different features. At the same time an investigation was done to evaluate the accuracy of models when increasing the size of the dataset.

Five machine learning models were implemented in order to accomplish the task. Since, this task is framed as a supervised learning task in current literature; an unsupervised learning model was also among the five models. Support vector machine, Logistic Regression algorithm, Naïve Bayes algorithm, Decision Tree algorithm and KMeans clustering algorithm were used to build the five classifier models. Bag of words, Tfidf and two more feature types were used as features. Google bad word list was used as hate lexicon to extract features from data.

Naive Bayes classifier with Tfidf features was the best performing model with an F-score value of 0.719. It was observed that in almost all the considered scenarios supervised learning models performed better than unsupervised learning model.

Since the amount of local data available for experiment is really law, extending the current data set is suggested as a future work. Mean time combining different feature types and evaluating the classifier models can also be done. And also, it is really important to create a lexicon relevant to English words used in Sri Lanka.

Preface

Identification of hate Speech in Social Media is a topic which has considerable amount of research recently. Extraction of some basic information from the online text content and utilizes them for hate speech detection purposes play a major role under this research domain. There are several different approaches for the hate speech identification. The basic flow in most of them is to extract some basic text features and come up with a classification model for the separation of hate content between classes.

Different techniques and approaches have been applied to Sri Lankan social media web content in order to identify online hate content. Different approaches for hate speech identification have been explored through the research. Preparation of the dataset by collecting and annotating comments in Sri Lankan news sites was solely my own work. To examine the behavior of unsupervised learning for hate speech identification, a machine learning model using an unsupervised algorithm was implemented and the idea behind this approach is also my own work.

Acknowledgement

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Ruwan Weerasinghe, senior lecturer of University of Colombo School of Computing for the continuous support of my research work, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

My sincere thanks also go to Ms. M.W.A.C.R. Wijesinghe, senior lecturer of University of Colombo School of Computing and Mr. M.D.R.N. Dayarathne, senior lecturer of University of Colombo School of Computing for their feedback provided on research proposal and interim report which helped me to look at different perspectives of my research and improve the study.

I would also like to thank our research project coordinator Dr. H.E.M.H.B. Ekanayake for his guidance given throughout the year. Also, I want to show my gratitude for the university staff and all the lecturers for the support given to successfully complete this research.

Last but not the least; I would like to thank my family and my friends for supporting me throughout the research work and my life in general. It's a great pleasure to acknowledge the assistance and contribution of all people who helped me to successfully complete my research.

Table of Contents

List of Figures

List of Tables

List of Acronyms

NLP	Natural Language Processing
CT	Colombo Telegraph
NLTK	Natural Language Processing Tool Kit
TFIDF	Term Frequency Inverse Term Frequency
POS	Part of Speech
SVM	Support Vector Machine
BoW	Bag of Words
BoF	Bag of Features
TN	True Negatives
TP	True Positives
FP	False Positives
FN	False Negatives
FRE	Flesch Reading Ease Score
FKRA	Flesh-Kincaid Grade Level

DS500 Dataset with 500 Comments

DS1000 Dataset with 1000 Comments

Scikit-learn A Python Library

Chapter 1 - Introduction

1.1 Background to the Research

Social media can be defined as computer-mediated technologies that facilitate the creation and sharing of information, ideas, career interests and other expressions through virtual communities and networks. Social media gives a kind of virtual life for people and a place to openly express feelings, opinions and beliefs. Websites dedicated to forums, micro blogging, social networking and wikis are example for different types of social media. Examples of social media organization include Facebook, YouTube, Twitter …. etc.

It is really difficult to find a single internationally accepted definition for hate speech. Hate speech is highly co-related with freedom of expression, individual, group and minority rights and concepts of dignity, liberty and equality. According to most of national and international legislations, hate speech refers to expressions that prompt to harm, discrimination, hostility and violence based on an identified social group or demographic group. In some cases, it is mentioned that hate speech is a communication that denigrates people on the basis of their membership to a particular group. Hate speech can include any form of expression such as images, videos, songs as well as speech. Hate speech attacks a person or a group on the basis of race, religion, sexual orientation or gender.

According to many rules and legislations in many countries hate speech is illegal. But it depends on the definition of hate speech given by that particular country. Hate speech often shows up online especially on social media at present. Meaning and content of hate speech remains quite similar in both online and offline media. But online hate speech renders current laws and gender regulations in an ineffective manner in many cases when it is compared with offline media. As a positive impact of social media, we can consider social media as an asset in terms of democratic, dialogic expressions. But it can be used by extremist groups as an advantage for them to disseminate hateful content. However, the impact of online hate speech is intensified in comparison to offline hate speech.

Social media for example Facebook, YouTube, Twitter has different policies to handle hate speech. According to YouTube community guidelines [1], they encourage free speech and to defend our right to express unpopular points of view but they don't permit hate speech. They consider hate speech as content that promotes violence, hatred against individuals or groups based on attributes like race, religion, disability, gender, age, veteran status, sexual orientation. According to their guidelines they have drawn a fine line between what is and what is not hate speech. For example, in YouTube it is generally fine to criticize a nation state, but not good to post malicious hateful comments about a group of people solely based on their ethnicity. YouTube has given users few options to report about content which we feel that violate their hate speech policy. We can flag the particular video or we can file an abuse report on particular content.

In Twitter policies [2] they have mentioned that they strictly prohibit the promotion of hate content, sensitive topics and violence globally. They also consider content including attributes like race, religion, disability, gender, sexual orientation, age, veteran status in a violence promoting manner as hate speech. Organizations or individuals associated with promoting hate, criminal or terrorist related content, Inflammatory content which is likely to evoke a strong negative reaction or cause harm, offensive, vulgar, abusive or obscene content are subjected to their policies. News and information that calls attention to hate, sensitive topics or violence but does not advocate for it, commentary about products, services, companies or brands including potentially negative commentary are not subjected to their policies.

Facebook [3] also consider content including race, religion, disability, gender, age, veteran status, sexual status in hatred promoting manner as hate. Facebook also has given few options to report any policy abuse. We can send a message to the person responsible for posting; we can un-friend the person to remove them from our friend list; can block the person from contacting you; report the person if their behavior is abusive or use privacy settings.

According to all the mentioned policies and regulations by different social media organizations, it is clear that there is significant need of removing hate content from the social media sites.

1.2. Research Problem and Research Questions

Online hate speech detection is currently a hot research domain where many researches are conducted in passed two three years. Finding a generalized mechanism for automatic hate speech detection is really difficult since hate speech is context dependent and language dependent. Currently there are no researches conducted using Sri Lankan web contexts. Although different techniques used in other researches done for English language for the purpose can be applied, since the English used by Sri Lankans is simple and different from English used in other countries there can be a loss of accuracy with the same methodology. There for our attempt is to collect, annotate English texts from Sri Lankan News sites and prepare a dataset and then apply different techniques to that data set to identify a mechanism for online hate detection.

Meantime according to the current literature it was identified that this task is usually classified as a supervised learning problem and no considerable number of researches are conducted with unsupervised learning approaches. So, applying an unsupervised learning algorithm and exploring the results were one of our intentions.

1.2.1. Research Question

Is it possible to identify hate speech in social media automatically?

- How to use a lexicon based approach for hate speech identification?
- How to use a machine learning approach for hate speech identification?
- Can we find a combined solution by combining lexicon based approaches and machine learning approaches?

1.3. Goal and Objectives

1.3.1. Goal

Freedom of expression is recognized as a human right under universal declaration of human rights. It is one of the basic pillars of every democratic society. But the presence of hate speech in general public discussions is one of the most direct indicators of a democratically weak society. It is clear that there is a big need to identify online hate speech so that it can lead the path to decrease

cybercrime and reduce the spread of hatred in the society. The aim of this project is to overcome the problem of identification of hate speech in social media using machine learning techniques.

1.3.2. Objectives

- Collect reader responses of Sri Lankan articles on web
- Manually annotate responses as hate speech or not
- Identify appropriate lexicon based method for hate speech identification
- Develop a machine learning approach for hate speech identification
- Compare the effectiveness of the different methods

1.4. Justification for the Research

It is clear that number of social media websites get increased day by day. Meantime number of registered users gets increased day by day and the amount of online user generated content quickly grows. It is really difficult to do manual flagging to remove hateful content in online media. So, 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 also we feel that there's a big need in identifying hate speech automatically. Automatic identification of online hate will led the individuals to engage in more online discussions without any fear and depression while minimizing the impact on different communities as well as individuals. At the same time it will be helpful to decrease the spreading of bad feelings like terrorism and to reduce hate crime [4].

1.5. Research Methodology

Research methodology follows quantitative approach. Quantitative researches are researches that deal with numbers, logic and conduct statistical and mathematical analysis on collected data. Few main characteristics of a quantitative research and how those characteristics are accomplished through our research are explained below briefly.

Data is usually gathered using structured research instruments. Here in the research text data will be collected from websites and then manually annotated and a dataset will be created. This dataset can be considered as a sample of the representative population which is Sri Lankan web

context. This research can be repeated giving the same results for the dataset. At the same time there are clearly defined research questions and objectives for the research as mentioned previously. All aspects of the study are carefully designed before data is collected and data which are fed to the experimental models are in the form of numerical data. We can use the conclusions of the research to generalize concepts more widely and do predictions.

Data collection, data cleaning, model building, testing using models, evaluating models are done using different tools and measures. Throughout the next chapters of the thesis these different steps will be discussed in detail justifying that the research follows a quantitative approach.

1.6. Scope and Limitations

Hate speech is context dependent and language dependent. It is really difficult to identify hate of another culture. So, that a Sri Lankan website is used to identify hate. This study is not focused on Sinhala language since the volume of Sinhala reader responses is not sufficient enough to carry on the research. Colombo Telegraph website [5] was selected for reader response (comments) collection since there are sufficient feedbacks for Sri Lankan articles. There are many registered users in Colombo Telegraph website and it is also an advantage. The research is conducted in the domain of Natural Language Processing considering the task as a text analytic problem. Then the research will be focused on a lexicon based approach for classification of words with the combination of a machine learning technique.

1.7. Outline of the Dissertation

The thesis is organized as follows. Chapter 2 includes the literature review explaining related work and the uniqueness of our work. The planned experimental set up and design is described in Chapter 3, followed by the implementation details in Chapter 4. Then in Chapter 5 results and evaluation criteria is presented. Finally, the Chapter 6 will conclude our study and provide guidance to the future work.

Chapter 2 - Literature Review

This chapter provides detailed description on previously used techniques for automatic hate speech detection. Although it is difficult to do a direct comparison between different methodologies used in different studies this chapter includes few brief explanations on different data sets used, preprocessing protocols used and experimental set ups built. During past recent years there have been many researches done on automatic detection of hate speech on social media. According to the number of researches emerging recently, it is clear that there is a significant attention towards online hate speech detection.

Mainly two main approaches were identified for this task as lexicon based approaches and machine learning based approaches. And in most of the cases researchers have come up with a hybrid approach combining both lexicons based and machine learning approaches. Currently very few researchers have focused on deep learning approaches.

2.1. Lexical based / Rule based Approaches

Lexical based approaches rest on the idea that most important part of a text classification task is being able to understand lexical phrases. Machine is fed with patterns of language, grammar, manually created rules describing certain type of texts or else domain base knowledge describing certain type of texts. Vocabulary plays a major role over grammar in this approach.

For domains like sentiment analysis, there are inbuilt lexicons which are widely used. Those lexicons are comprised of different words and the polarity rates which indicate whether that word gives a feeling of negative or positive. Since, hate speech detection is a currently emerging research area in past few years still there are no such lexicons built for the task of hate speech detection. There are only collections of words which are banned or recognized as hate words. But there are no rates given for the words indicating how much hate is expressed through that word. Google bad word list is the most widely used hate lexicon which is built by Google collecting the banned words by Google.

2.2. Machine Learning Approaches

Field of computer science which includes the topics of the computer's ability to learn without explicitly programmed is known as machine learning. In machine learning algorithms instead of programmer defining rules for particular task, data is fed to the machine and algorithm is adjusted in order to perform the task. So, machine learning is basically a data driven approach. Currently machine learning algorithms and techniques are widely used in different domains of computer science. In text classification tasks also, machine learning plays a major role.

Supervised learning and unsupervised learning are two main strategies of machine learning. When the input data is labeled it is called supervised learning and when input data is given without the label, it is called unsupervised learning. Supervised learning algorithms try to fit its inner machinery to match the mapping function of the labeled data. The data set is split into two as training set and testing set. Algorithm tries to make predictions on training data until a considerable level of performance is achieved. This is known as learning phase and then it's going to be the testing phase. What happens in testing phase is the creation of predictions on the testing set and calculating the performance evaluation matrices to compare the predicted label and actual label.

Support vector machine, Naïve Bayes classifiers, Decision tree classifiers, Logistic regression models are few examples of supervised machine learning algorithms while kmeans clustering, self-organizing maps are grouped as unsupervised learning algorithms. Support vector machine is the most widely used supervised algorithm for the task of hate speech detection. Meantime hate speech detection has been framed as a supervised learning task since the number of researchers who have tried out unsupervised learning for hate speech detection is relatively very law.

2.3. Hybrid Approaches

Hybrid approaches are used by many researchers for the task of hate speech detection. Combination of learning based approaches with lexical based approaches is done in here. In some scenarios first, the lexical based approach is used and data is filtered and then those filtered data is fed in to a machine learning model. Meantime in some scenarios lexical resources are used to extract features from text data and those features are fed to the machine learning model.

2.4. Related Work

2.4.1. Lexical based / Rule based Approaches

A Lexicon based Approach for hate speech detection.

A classifier model for hate speech detection using a lexicon is proposed by N. D. Gitari et al [6]. There they have created a model classifier which uses sentiment analysis with subjectivity detection to detect the subjectivity of a sentence and polarity of the sentiment expression. Then they have used a lexicon build for hate speech detection with classifier model to detect hate.

Identifying hate in online forums, blogs and comment section in news reviews had been their main concern. They have defined hate speech as speech that uses offensive and threatening language targeting a particular group of people or an individual based on their religion, nationality, ethnicity, gender or color. But for the research they have abstract hate speech detection into 3 categories race, nationality and religion. They have used two different sources. 100 blog postings from 10 different websites are collected as the main source. They have selected the websites from Hate Directory which is composed of sites to be generally offensive. Other corpus is created using 150-page document websites. 30% of each corpus has been annotated manually.

The approach proposed in this research consists of three main steps. As the first step subjectivity detection is done. A rule based and learning based approach is used in the subjectivity classifier. Sentiment lexicon resources of Wilson et al. [7] and SentiWordNet [8] are used for this purpose. A list of over 800 subjective clues with positive, negative, neutral and both tags are included into that sentiment lexicon. If a sentence contains two or more strongly subjective clues the sentence is classified as a strong subjective clue. Beside this they have calculated negative and positive scores for each sentiment word in the sentence and subtract the negative score from positive score to obtain the synset score. If synset score is greater than 0.5 or if the score is lesser than -0.5 the sentence is positive or negative. That means subjective, if not objective. According to the two datasets used in the research, 56% of first corpus and 75% of second corpus are subjective.

As the second step of the proposed approach a lexicon for hate speech is built. Here they have built a rule based hate speech classifier which relies on three different sets of features. They are negative polarity words, hate verbs and theme based grammatical patterns. To create negative polarity features they have used subjective sentences with negative semantic orientation identified in

step 1. They have extracted all verbs which have a relation with hate verbs in their hate corpora to prepare the 2nd feature set hate verbs. To build the 3rd set of features all nouns related to 3 domains race, nationality and religion are extracted. Name Entity Recognition software has been used to identify sources, recipients of opinions that include in list of nouns. To classify a sentence into hate categories a set of rules are generated considering the three set of features.

According to their results they have got the best results with an F-score of 70.83 for the first corpus for the combination of three features sets Semantic, hate verbs and theme based.

2.4.2. Machine Learning Approaches

Automated Hate Speech Detection and the Problem of Offensive Language

A multi-class classifier to distinguish between hate speech, offensive language and none of them is presented by Davidson et al [9]. It is emphasized that separating hate speech from other instances of offensive language as a key challenge in automatic hate speech detection. Throughout the research they have tried to separate hate speech from other offensive language and tried to find the instances when this task is more difficult to be done.

By examining laws and regulations and different policies used by different social media organizations they have come up with a definition for hate speech as follows: *"language that is used to expresses hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group*". A hate speech lexicon prepared by hatebase.org [10] is used in this research. They have collected 85.4 million tweets from 33,448 twitter users using twitter API and terms of the hate speech lexicon. From them 25k tweets have been manually annotated into 3 categories as hate speech, offensive but not hate and neither of them. Finally, they have prepared a data set with 24,802 tweets.

Each tweet is lowercased and stemmed using Porter stemmer. Then TF-IDF vectorizer is used to weight each bigram, unigram and trigram features. NLTK library is used to capture syntactic structure information like POS tags. Flesch-Kincaid Grade Level and Flesch Reading Ease scores are used to capture the quality of each tweet. Sentiment score, number of characters, words and syllables are also used as a feature.

Logistic regression with L1 regularization has been used to reduce the dimensionality of the data. They have tested different models used in prior work. For example, logistic regression, naïve

bayes, decision trees, random forests, and linear SVMs. 5-fold cross validation has been used to test each model. Through that they have discovered logistic regression and linear SVM as best performing models and they have selected logistic regression with L2 regularization to build their final model. A separate classifier is used to train each class and class label with highest predicted probability across all classifiers is assigned to each tweet. Scikit – learn toolkit is used for modeling.

They have noted that the best performing model has an overall precision of 0.91, recall of 0.90 and F1 score of 0.90. Then they have analyzed their results showing reasons for misclassification of the tweets. According to their observations model is biased towards classifying tweets as less hateful or offensive than human coders. At the same time, they say that some tweets which have fitted their definition of hate speech have been misclassified since they did not contain any term strongly associated with hate speech. When a much broader definition is used for hate speech, offensive language is tending to be misclassified as hate speech. Through this research they have reduce this misclassification up to some extent.

Hateful Symbols or Hateful People? Predictive Features of hate speech detection on Twitter.

Z. Waseem et al [16] have evaluated the influence of different features for prediction of hate. Their data set consist of tweets collected over 2 months' time period. They have come up with a data set including totally 136,052 tweets where 16,914 tweets are annotated as sexist, racist or neither of them.

A logistic regression classifier and 10-fold cross validation was used to test the influence of various features on prediction performance. They have found that character n-gram is better than word n-gram in accordance with their features. They have used gender, location and length of the tweet as additional features mainly. Best performance has been achieved with character n-grams of lengths up to 4 with the additional feature gender with an F-score 73.93%. Usage of additional features location and length hasn't given improvements to F1-score. What they have concluded is their solution can be useful in some cases but not for all and the problem can be partially solved using a character n-gram based approach.

2.4.3. Hybrid Approaches

Threat Detection in Online discussions

Results of the research conducted by A. Wester et al. [11] shows that combination of lexical features outperforms the use of more complex syntactic and semantic features for the task of detecting online hate. They have used YouTube threat corpus presented by Hammer et al. [12]. The corpus includes 9845 comments from eight different YouTube videos. According to Hammer et al. [12] the inter-annotator agreement of this corpus is 98%. A comment consists of a set of sentences which were manually annotated to be either a threat of violence or not.

In the preprocessing stage first the comments are split into sentences manually as part of the annotation process. Then all the basic preprocessing activities like tokenization, lemmatization, POS-tagging and dependency parsing are done using spaCy NLP toolkit. Then the corpus has been further enriched with cluster labels using brown clustering algorithm. At the same time WordNet resources are used to include synset information of a word as well as its parent and grandparent synset.

Maximum Entropy, SVM and Random Forest are the 3 different classification frameworks used in the research. Task was considered as a binary classification task, using the implementations found at scikit learn toolkit [13]. They have tuned each model with aim of maximizing the F-score. They have selected an initial set of features and a classification framework to use as a basis. Basic lexical features, word forms and lemmas and n-grams are used as initial set of features. In the second round of tuning they have used combinations of features used in first round. Then they have analyzed the different F-score values obtained with different features and different classifiers. According to their analysis they have selected Bag-of-Word model and lexical n-gram model with both MaxEnt and SVM classifiers for final testing since best performance was observed with them.

According to their results n-gram model outperforms BoW model with an F-score of 0.6885 (SVM) and 0.6860 (MaxEnt) while 0.6562 F-score for BoW. They also have concluded that introduction of more complex features like WordNet synset, Brown clusters, POS tags and dependency parsers did not improve on the simpler surface-based feature set.

Detecting hate on World Wide Web

The way the definition is given to hate speech in the research by Warner et al. [14] is quite different. If the identity of the speaker cannot be determined, and if no contextual cues are present, such terms are categorized as hateful. They have used the hypothesis that hate speech resembles a word sense disambiguation task, since a single word may appear quite frequently in hate and non-hate speech texts. The data set is created by using the data received from Yahoo! and American Jewish Congress (AJC). There were about 9000 paragraphs in the data set.

They have decided to attempt at paragraph level for the first classification experiment to make used of contextual features. To annotate the data set they have got used of their hypothesis, *"hate speech employs well known stereotypes to disparage an individual or group"*. So that they have subdivide speech by stereotype, which can distinguish on what form of hate speech from another by identifying the stereotype in the particular context. Seven categories have been used for labeling as anti-Semitic, anti-black, anti-Asian, anti-woman, anti-Muslim, anti-immigrant or other-hate. Annotators can assign one or more of the 7 labels for each paragraph. Next a language model for each stereotype is created.

A template based strategy presented in Yarowsky et al. [15] has been used to generate features from the corpus. Each template was centered around a single word. Literal words in an ordered two-word window on either side of a given word have been used. Then POS-tagging of each sentence provided the similar POS windows as features. Brown clusters were also assigned in the same window. Finally, they have associated each word with the other labels that might have been applied to the paragraph. Features for every paragraph in the corpus were generated using all templates. Using that a count for positive and negative occurrences of each template was maintained. By considering the ratio of positive to negative occurrences they have come up with 4379 features for all templates. Then they have selected a threshold value for that ratio and eliminated some features and ended up with 3537 features. For this task they have used SVM classifier. Besides these features they have come up with two more additional feature sets.

Six classifiers were built for each type of feature template strategy. SVM light with linear kernel function was used as the classifier. 10-fold cross validation has been performed for every classifier. A special point they have come up in this research is *"Specialized knowledge of stereotypical language and the various ways that its authors mask it could make a classifier's*

performance superior to that of the average human reader". They have compared the classifier performance of 6 different classifiers built. According to their comparison "Gold Positive Unigram" classifier has the best F1-score of 0.63 with accuracy of 0.94. In terms of F-score, the best performing classifier was equal to the performance of annotators.

Are You a Racist or Am I seeing Things? Annotator Influence on Hate Speech Detection on Twitter

An examination of influence of annotator knowledge of hate speech on classification models by comparing classification results obtained from training on expert and amateur annotators proposed by Z. Waseem et al [17]. They have used 6909 tweets in their data set and they have used annotators from Crowdflower and annotators that have theoretical and applied knowledge of hate speech. They have used 5-fold cross validation to assess the influence of the features they selected for each annotator group. To create the feature set they have focused on textual and extra-linguistic features like POS tags. In-house mapping of Brown clusters was used to replace unigrams with cluster identifications emphasizing that they used lexical based method to extract features. Since they have used multiple people for annotation they have mitigated the personal bias in annotating. It is mentioned that hate speech is hard to annotate without intimate knowledge of hate speech.

Hate Speech, Machine Classification and Statistical Modeling of Information Flows on Twitter: Interpretation and Communication for Policy Decision Making

P. Burnap et al [18] has proposed a machine learning text classifier trained and tested to identify hate speech using a data set collected from twitter. They have collected around 450 000 tweets and annotated those using human coders with crowd-sourcing. In this study they have implemented probabilistic, rule-based and spatial classifiers which perform similarly across most of their feature sets. Then they have combined the classification output of base classifiers using a voted meta-classifier based on maximum probability matched in every experiment. They have achieved an overall F-measure of 0.95. This has suggested that an ensemble classification approach is the most suitable approach to classify hate speech with their feature set. The purpose of the classifier is to assist policy in monitoring the public reactions to large scale emotive events.

2.5. Discussion

There are quit lots of articles related to the topic of hate speech detection. According to the review on literature following key points are noted down.

In all the researches done in this research area they have first consider a lot about narrowing down the definition of hate speech with regard to their research. It is really important to define hate speech in a particular way since, the data set is annotated according to this definition and all the results will rely on the annotated data set and assumptions, definitions made in first stage of the research. Z. Waseem et al [17] have done their research to show the influence of the annotator on online hate detection. According to all the researches considered a large data set is used in every research making it clear that usage of large amount of data gives the better results.

When considering on how different researches have done text preprocessing. In all most all the researches they have used python with NLTK library for preprocessing. Mainly three different types of features are extracted from text data for this task as N-gram features, linguistic features and syntactic features. To extract the context level features some have tried out with paragraph level feature extracting [14] while in most of the researches sentence level or word level features are extracted. Except to them character level features are extracted in the study Mehdad et al. [19] and few others.

Basically, many researchers have focused on machine learning and lexicon based models while only few have concerned on deep learning approaches [19,20] for online hate detection. Supervised learning with the combination of a lexicon based approach is used in most of the researches. Still there is no proper exploration on unsupervised learning for hate speech detection.

Since the task of hate speech detection has been framed as a supervised learning task, this study will focus on exploring an unsupervised learning technique with four other supervised learning techniques and compare the accuracies of the models built with our own local English comment dataset. At the same time different features will be used to train and test the models.

Chapter 3 - Design

In this chapter a detailed description will be given on dataset used in our experiment explaining origin of data and annotation process of data. Then data preprocessing and feature extraction steps will be presented in detail. The entire details about the design of the experimental setup, algorithms used will be presented.

3.1. Research Design

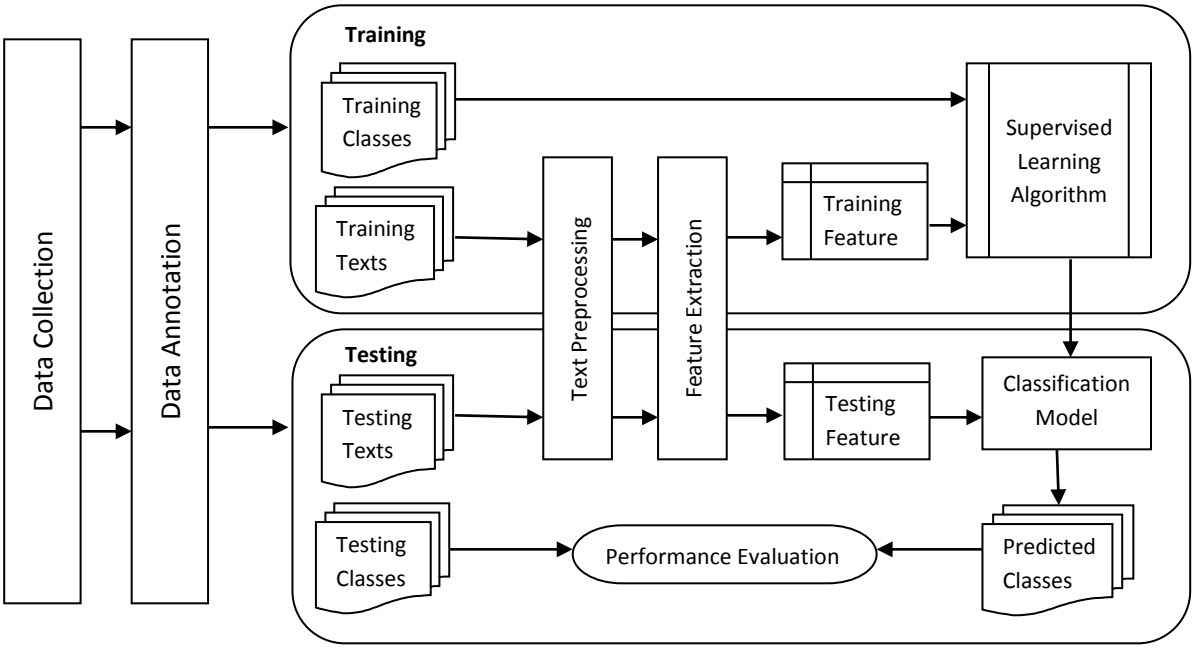

Figure 3.1 Research Design for Supervised Models

Figure 3.1 presents the high-level overview of the system design for the model built using a supervised machine learning algorithm. Each and every step of the given process will be explained in detail in the latter part of the chapter.

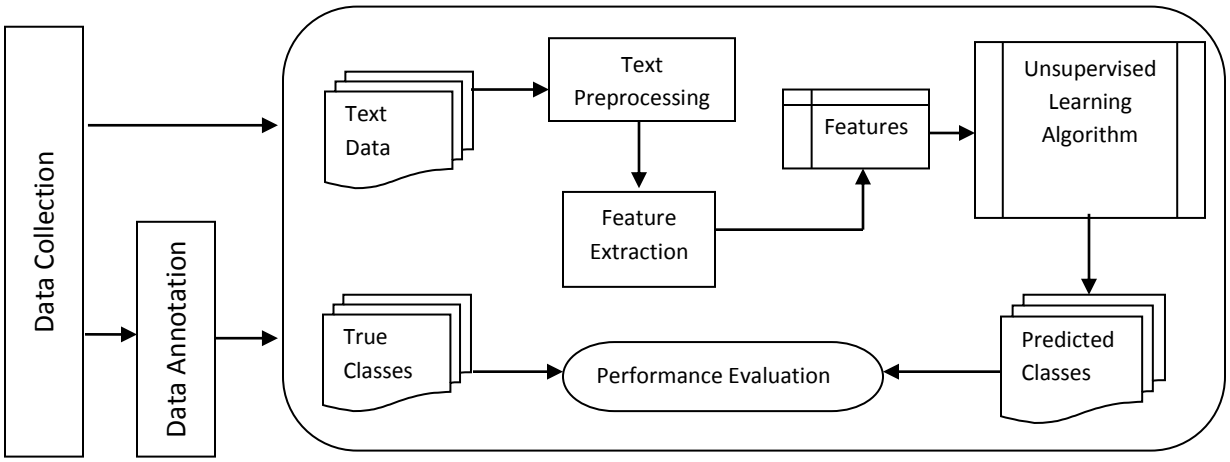

Figure 3.2 Research Design for Unsupervised Models

Figure 3.2 presents the high-level overview of the system design for the model built using an unsupervised machine learning algorithm. Here the text preprocessing and feature extraction steps will be same as the steps in supervised learning model. Only the way of learning is changed.

In both supervised and unsupervised designs data collection, data annotation, text preprocessing, feature extraction and performance evaluation are common steps. Only the difference is to train the supervised model features of text data is used with the classes of training data and in unsupervised model classes are not used and there is no specific training and testing phase there. All these mentioned steps in the diagrams will be discussed in detail in this chapter and the 4th Implementation chapter.

3.2. Data set

To perform a successful experiment on hate speech detection availability of a labeled corpus is really important.

Our data set is comprised of user written comments from different articles in Colombo Telegraph website [5]. Colombo Telegraph is a public interest website which is full of articles related to Sri Lankan matters. It is run by a group of exiled journalists and they are working on volunteer basis. Colombo Telegraph provides an opportunity for readers to discuss the content published or debate issues more generally. Ensuring the safe of platform is one of their aims. So, that they have come up with few simple guidelines for users to abide. What they have mentioned in their guidelines in short is as follows.

– If you **act with maturity and consideration** for other users, you should have no problems.
– **Don't be unpleasant.** Demonstrate and share the intelligence, wisdom and humor we know you possess.
– **Take some responsibility for the quality of the conversations in which you're participating**. Help make this an intelligent place for discussion and it will be.

According to their guidelines it is clear that they are highly concerned on the stuff published. In another way CT doesn't like any sort of hate speech to be published. Although guidelines are provided CT editors have plenty of work to do such as reading comments and removing comments which are not according to their comment policy. Through a small inspection in the comment section we can clearly see the number comments removed manually since those comments are not according to the comment policy. But still there are quite lots of comments which haven't abide the guidelines. So, that we selected CT website as source of our data.

All articles which have more than 25 comments published in April and May of 2017 were selected to collect comments. The prepared dataset consists of 1500 comments. Among them 1000 comments were manually annotated as hate or no hate. Table 3.1 presents an overview of the number of annotated comments with hate and without hate.

Table 3.1 Classes of Data set

Number of Comments with Hate	**421**
Number of Comments without Hate	579

According to the classes of data we can clearly see that dataset is not 100% balanced but it is not highly unbalanced too.

3.3. Definition for Hate Speech

With the understanding gained through literature review definition for hate speech was built as follows. Whole dataset was manually annotated according to the definition of hate speech given below.

"Hate speech is the usage of language to insult or spread hatred towards a particular group or individual based on religion, race, gender or social status."

3.4. Data Preprocessing

All text data should be cleaned before they are fed to the classifiers in order to reduce noise. It is one of the essential tasks in this type of a research. NLTK proposed by S.Bird [23] is one of the best preprocessing libraries which can be used for this task. Tokenization, expansion of contractions, special character removal, stops word removal, lemmatization are basic preprocessing steps which were used.

3.5. Feature Extraction

A feature can be named as a property of the instance that is being classified. In our case the instance is a comment. Features like words in a sentence are highly specific while features like POS tags are less specific. When training the models for hate speech detection the input to the models is the set of features which represent comments in our corpus. We have explored different feature types in our experiment. All feature extractions were done using scikit learn python library which is used widely for machine learning purposes with python.

3.6. Classifiers

Five different classification frameworks were tested in our experiment as Support Vector Machine, Naïve Bayes Classifier, Logistic Regression Classifier, Decision tree Classifier and KMeans Clustering algorithm. It is clear that we have used four supervised learning techniques and one unsupervised learning technique. The task was considered as a binary classification task, using the implementations and libraries present in scikit-learn toolkit Pedregosa et al. [13].

Support Vector Machine is a supervised learning algorithm used for classification problems. SVM is very suitable for cases where the number of dimensions is greater than the number of samples. Making a single linear plane in the x-dimensional space where each x features of a given feature set corresponds to one dimension in an x-dimensional space is the main task accomplished by SVM. The plane should be positioned in way such that very few numbers of samples are on wrong side of the plane. The scikit-learn library which used in this study for machine learning task is available of few different support vector machine classes as SVC, LinearSVC and NuSVC. The SVM used in our study is Linear SVM and Stochastic Gradient Descent Classifier (SGDClassifier) with its default loss parameter (hinge) to build the SVM was used.

Naïve Bayes Classifiers are based on Baye's theorem with "naïve" assumption and it is a supervised learning algorithm. Naïve assumption is all features are independent of each other. Membership probabilities for each class of data points are predicted using models which use Naïve Bayes algorithm. Both Gaussian Naïve Bayes and Multinomial Naïve Bayes were used for the experiment.

Decision tree is a supervised learning algorithm which uses a decision tree to make predictions. It can be used for both classification and regression tasks. In decision tree algorithm a model is created to predict the value of a target variable by learning simple decision rules inferred from the data features. Decision Tree Classifier is capable of performing multi-class classification on a dataset. Leaves of the tree represent class labels and branches represent conjunctions of features that lead to those class labels. Decision Tree Classifier of scikit-learn toolkit with 'entropy' criterion was used in this study.

Logistic Regression model is a statistical method for analyzing a dataset where there are one or more independent variables to determine the outcome. Or else it can be called as a regression model where dependent variable is categorical. Since our task is a binary classification task, binary logistic model was used to estimate the probability of a binary response based on one or more or independent variables (features). Finding the best fitting model to describe the relationship between the independent variables and dependent variable is the main goal of logistic regression. Scikit-learn toolkit's Logistic Regression Classifier with its default parameters is used in this research.

KMeans clustering algorithm was used to build the model using unsupervised learning. KMeans tries to cluster data through separating data with equal variance by minimizing a criterion

known as the sum-of-squares with-in-cluster. The number of clusters to be specified should be given for this algorithm as a parameter which is generally known as K. Since the need is to cluster our data into to two clusters as hate and no hate, the value of K was set as two. KMeans class of scikit-learn library assigning n_clusters parameters with two and using default values for other parameters was used to build the unsupervised model.

3.7. Evaluation

Since the research follows a quantitative approach, what is done is a systematic investigation, where statistical, mathematical techniques to accomplish our task can be used. Collected dataset was annotated manually and then fed to the built model in order to get predictions. So, then the data and results with an evaluation metric in order to check the performance of the model, biasness of the results and to what extent study's results can be generalized had to be analyzed.

The evaluation metric built was used throughout the experiment. Since this research is related to data analytics and natural language processing accuracy, precision, recall and F-score was chosen for evaluation metric. In our dataset a comment contains either hate or they do not. So, that our classifier was a binary classifier. All values which were checked, accuracy, precision, recall and F-score all relied on the notion of positives and negatives. So, that a positive was defined as a comment with hate and a negative was defined as a comment that does not contain hate.

Table 3.2 Structure of Confusion Matrix

		Predicted Class	
		With Hate	No Hate
True Class	With Hate	True positive	False Positive
	No Hate	False Negative	True Negative

True negatives, true positives, false negatives and false positives are defined as given in the Table 3.7. At the same time the confusion matrix which was built according to the above table were observed.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

According to the given formula accuracy can be defined as the fraction of predictions that are correct. Although accuracy is used in many natural language processing researches for

evaluation, there are few problems with accuracy which are very common. The main problem is accuracy is not a good measure of the classes of data is unbalanced. 57.9% of our annotated data is in no hate class while 42.1% of annotated data belong to hate class. The data set is not 100% balanced, but according to the percentages of data it is fairly balanced. So, we can use accuracy as a measure. Accuracy measure gives more weight to the correctly classified positives and negatives, so that when the data set is unbalanced it can give a higher accuracy considering one class although other class is misclassified.

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

Fraction of predicted hate comments which were actually hate comments is defined as precision. From this measure the correctness of the positive predictions can be observed. To look over the correctly predicted positives precision is the best measure to use, since it does not consider about negatives.

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

The fraction of hate comments that were detected is known as recall. From this measure the idea of number of hate comments identified and number of hate comments the classifier missed can be examined. Since recall also does not consider about true negatives this measure is also better for our task.

$$F - score = \frac{2 \text{ x Precision x Recall}}{\text{Precision} + \text{Recall}}$$

When harmonic mean of precision and recall is calculated it is called as F-score. F-score ensures that there will be no overly rely on either precision or recall. So, that F-score was considered as our main evaluation measure.

Chapter 4 - Implementation

In this chapter the various components that go into constructing the model and making classifications are described. A detailed description of all the implementations, codes, processes used and technologies used are included into this chapter. The entire process of the experiment constructed will be described step by step from start to end. The main steps of the experiment are as follows.

- Data collection and annotation
- Data preprocessing
- Feature Extraction
- Build the classification model and evaluation

Details of the first step data collection and annotation were discussed in the Design chapter. So, this chapter will continue from second step onwards.

4.1. Implementation Environment

As mentioned earlier the language chosen is Python (python 2.7.9.) mainly because of its available libraries. NLTK offers most of the preprocessing activities which is very important in text analytics and Scikit Learn offers implementations of Support vector machine, Naïve Bayes algorithm, Decision tree algorithm, Logistic regression algorithm, Kmeans algorithm and feature extraction techniques like BoW and Tfidf.

4.2. Data preprocessing

Steps used for preprocessing the corpus are presented in this section. Tools and packages used for the experiment will be presented. After considering different toolkits for Natural Language Processing, Natural Language Processing Toolkit, NLTK library with python was selected for the preprocessing task. NLTK can perform sentence splitting, tokenization, pos-tagging, lemmatization and many other preprocessing activities. Since, the language used for all preprocessing and other experimental work was python; the data after preprocessing was stored in-memory since it can be directly accessed for the next steps of the experiment.

All the comments were stored in a csv file. So, before text preprocessing all the comments in the csv file were read into a data frame in python. Then all the next activities were done using this data frame. Both training and testing data were stored in a single csv file. After reading the csv file into a data frame, data was split into training and testing datasets using a function.

Expanding Contractions

Preprocessing was stared through examining each comment individually and expanding verb contractions. Since English texts are used, expanding verb contractions is applicable. A part of the contraction map used for this task is given below.

```
CONTRACTION_MAP = {
"ain't": "is not",
"aren't": "are not",
"can't": "cannot",
"can't've": "cannot have",
"'cause": "because",
"could've": "could have",
"couldn't": "could not",
"couldn't've": "could not have",
"didn't": "did not",
"doesn't": "does not",
"don't": "do not",
```

Tokenization

Breaking up strings into words and punctuations is known as tokenization. Words in every comment were tokenized. For this task, the tokenizer of NLTK toolkit word_tokenize() was used.

Lemmatization

The process of identifying the root or stem of a word is known as lemmatization. For lemmatization WordNet Lemmatizer was used. Before lemmatization POS tagging was used and words were tagged using WordNet. POS tagging was done since lemmatization was done base on the POS tags. After lowercasing all words, nltk.pos_tag() was directly used since this function loads the pre-trained tagger from a file where it was trained with Treebank corpus. Then using the POS tags the Treebank tags to WordNet POS names were mapped. For this tagging process only adjectives, verbs, nouns and adverbs of Treebank POS tagging were considered. This was done since verbs, nouns, adjectives and adverbs play a major role in hate speech detection when

compared with other types of words. Finally, the POS tag obtained from WordNet was passed with the particular word for lemmatization. Following example will explain the process clearly.

```
word = going
wnl = WordNetLemmatizer()
postag = nltk.pos_tag(word)
```

For the given word “going” this will return VBP as the POS tag. Since this is a verb then we map the POS tag from WordNet corresponding to the tag, verb.

```
if pos_tag.startswith(‘V’):
        return wn.VERB
```

Then this returned POS tag will be passed for lemmatization with the corresponding word.

```
wnl.lemmatize(‘going’, wn.VERB)
```

As the final result we will get the word “go” as the lemmatized word.

Removing Special Characters and Stop words

Punctuations and stop words in English were removed in this step. Special characters were removed by checking with a regular expression. Stop words were removed by checking with the stop word dictionary of NLTK.

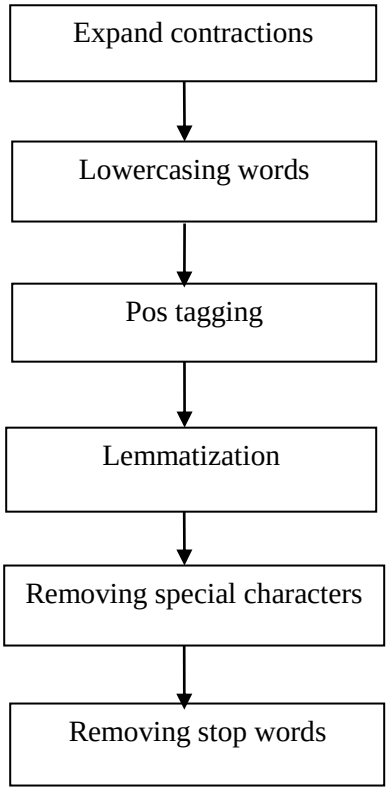

Figure 4.1. Steps of preprocessing

4.3. Feature Extraction

All the feature extraction activities were done using scikit learn toolkit of python. The feature extraction codes basically rely on the functions of scikit learn toolkit. Countvectorizer and Tfidf vectorizers are two main widely used inbuilt vectorizers used for natural language processing feature extraction purposes. Other than these two vectorizers, another function was used to extract features.

CountVectorizer – Bag of Word Features (BoW)

CountVectorizer implements both tokenization and occurrence counting in a single class. A collection of text documents can be converted to a matrix of token counts using CountVectorizer. Using this vector space model, the number of the unique words in all comments and the frequency of each term in vector can be observed. So, bag-of-words features were extracted using countvecorizer.

Tf-idf Vectorizer – Term Frequency Features (Tf-idf)

Term frequency-inverse document frequency vector is a way to measure the importance of a word or term. How much rare a word is present in a document can be checked using tfidf. So, using this vectorizer, the words with highest importance as a feature can be obtained. The specialty of Tfidf is frequency of the term is off-set by the frequency of the word in the corpus which clearly says that some words appear more frequently in general.

Bag of Features (BoF)

Then few features from features presented in T. Davidson et al. [9] were used for our model since in their study they have gained good results with these features. Following are the features that extracted considering the T. Davidson et al. [9] feature set.

- Flesch Reading Ease Score (FRE)

  Modified FRE score presented in J. P. Kincaid et al. [22] was used. This score indicates how difficult a passage in English to understand. A rate is given to the comment in the scale of 1-100 by this score. Higher the number, it's easier to read and understand the comment. To compute this score average number of words per sentence and syllables per sentence are analyzed.

$$FRE\ Score = 206.835 - (1.015 * WPS) - (84.6 * SPW)$$

  WPS – Average words per sentence

  SPW – Average syllables per word

- Flesh-Kincaid Grade Level (FKRA)

  FKRA score is used to capture quality of each comment.

$$FKRA\ Score = (0.39 * WPS) + (11.8 * SPW) - 15.59$$

  WPS – Average words per sentence

  SPW – Average syllables per word

- Sentiment score

  Sentiment score was used to capture the polarity score of each comment. If the comment has a negative polarity score, this indicates that the comment can contain hate speech. VaderSentiment presented by C.J.Hutto et al. [21] was used for the purpose of sentiment analysis. VaderSentiment is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains.
- Number of characters, words and syllables in each comment
- TF-IDF vector

Hate word count was added as an extra feature to the above mentioned features. All the mentioned features were only used in both supervised learning and unsupervised learning algorithms. Since the unsupervised learning algorithm used, Kmeans clustering is sensitive for noise another feature set was prepared from above features as sentiment score and hate word count (BoF2). Then all models were trained and tested using BoF2 features also to compare the results obtained. For the task of extracting the feature hate word count, "Google Bad Words List" was used as a lexicon for hate words. Since Google is one of the biggest internet companies who takes care of internet search to offer useful and clean results, the word list banned by Google considering those words as bad words, swear words, offensive words or profanities was selected in this study. There are 550 words available in this list.

BoF1 = <FRE, FKRA, Sentiment score, syllables, num_chars, num_words, num_unique_terms, Tf-idf, hate word count >

BoF2 = <Sentiment score, hate word count >

According to the features considered four types of features were extracted as BoW, Tfidf, and BoF features. All the models were trained using all of these feature types.

4.4. Classification Models and Evaluation

As mentioned in the Design chapter five machine learning algorithms were selected to build classifier models. Pipelines were used to extract BoW and Tf-idf features and train the models while BoF features were extracted through functions. Then all four supervised learning models were trained using the features of training dataset and evaluated using testing dataset. Whole dataset was used for the unsupervised learning model and the clustering behavior was analyzed.

At the same time an exploration on the effect of size of the dataset for the accuracies of the models was carried out. In the first phase out of 1000 comments in our dataset only 500 comments were used for both training and testing purpose of the models. All the evaluation measures were recorded and then in the second phase all 1000 comments were used for training and testing the models and evaluation measures were recorded to compare the accuracies and F-score values.

Chapter 5 - Results and Evaluation

In this chapter the experiments and the results of the experiments, following the experimental setups described in Chapter 4 will be discussed. Our main objective is to examine the behavior of different algorithms with different feature types in order to detect hate speech. So, in this section different feature sets and different classifiers will be compared with regard to accuracy, precision, recall and F-score measures.

5.1. Results

As mentioned in the Design Chapter all the comments were stored in one csv file and read into one data frame and then split into two for testing and training. For the interpretation of results the whole dataset including all 1000 comments will be represented as "DS1000"

Table 5.1 Partitioning of Whole Dataset (DS1000)

	Number of Comments
Training dataset	670
Testing dataset	330

Details about the Testing dataset

Table 5.2 Classes of Testing Dataset

Class	Support
Hate (1)	124
No Hate (0)	206

To see the difference of F-score and accuracies with the size of the dataset, half (500 comments) of the dataset was used and the experiment was done again. This dataset will be named as "DS500".

Table 5.3 Partitioning of DS500

	Number of Comments
Training dataset	335
Testing dataset	165

Details about the Testing dataset

Table 5.4 Classes of Testing Dataset

Class	Support
Hate (1)	86
No Hate (0)	79

BoW features

All the bag of words features is extracted using countvectorizer in Scikit-learn package. Then same feature vector was passed for five different models and using testing data performance of the models was evaluated.

Table 5.5 Results of BoW Features

	Accuracy		Precision		Recall		F-Score	
	DS500	DS1000	DS500	DS1000	DS500	DS1000	DS500	DS1000
SVM	0.52	0.67	0.52	0.67	0.52	0.67	0.51	0.67
Logistic Reg.	0.55	0.67	0.56	0.66	0.55	0.67	0.55	0.66
Naïve Bayes	0.579	**0.709**	0.579	**0.709**	0.579	**0.709**	0.579	**0.709**
Decision tree	0.5	0.66	0.51	0.66	0.5	0.66	0.5	0.66
KMeans	0.5	0.66	0.51	0.66	0.5	0.66	0.5	0.66

When examining the results of DS500 and DS1000 it's clear that DS1000 has performed better than DS500. All the F-score values of DS1000 are higher than the F-score values of DS500. So, it's clear that when the size of the dataset get increased BoW features performs better. Decision tree model and KMeans model has shown poor performance from all five models. According to the results, Naïve Bayes Model has the best performance with an F-score of 0.709 for the dataset with all 1000 comments and the confusion matrix of predicted results of Naïve Bayes Model is as follows.

Table 5.6 Confusion Matrix 01

		Predicted Class	
True Class		1	0
	1	79	45
	0	51	155

According to the confusion matrix the number of correctly classified comments is always higher than the misclassified comments. At the same time, it is clear that unsupervised learning

model's performance is also in an acceptable state as it has shown similar results like in Decision tree supervised learning model with BoW features.

Tf-idf Features

All the tfidf features is extracted using Tfidfvectorizer in Scikit-learn package. Then same feature vector was passed for five different models and using testing data performance of the models was evaluated.

Table 5.7 Results of Tf-idf Features

	Accuracy		Precision		Recall		F-Score	
	DS500	DS1000	DS500	DS1000	DS500	DS1000	DS500	DS1000
SVM	0.56	0.67	0.56	0.68	0.56	0.67	0.56	0.67
Logistic Reg.	0.52	0.69	0.62	0.689	0.52	0.69	0.429	0.68
Naïve Bayes	0.54	**0.739**	0.64	**0.75**	0.54	**0.739**	0.46	**0.719**
Decision tree	0.479	0.63	0.479	0.63	0.479	0.63	0.479	0.63
KMeans	0.56	0.45	0.56	0.46	0.56	0.45	0.56	0.46

All models except KMeans model has performed better with more data since F-score values of DS1000 is greater than F-score values of DS500 except in KMeans model. So, it's clear that when the size of the dataset gets increased supervised learning models perform better than unsupervised learning model with tfidf features. Among supervised learning models Decision tree classifier has the lowest results when compared to other three models. According to the results, Naïve Bayes Model has the best performance with an F-score of 0.719 and the confusion matrix of predicted results of Naïve Bayes model is as follows.

Table 5.8 Confusion Matrix 02

		Predicted Class	
		1	0
True Class	1	54	70
	0	16	190

According to the confusion matrix of the best performing model, the classifier has classified almost most of the comments with no hate correctly while comments with hate are also misclassified as No hate. So, the F-score value of the Naïve Bayes classifier is higher may be due to the correct classification of True Negatives but not True Positives. This may be a result of the nature of data in the testing dataset. The number of No hate comments is 206 while number of hate comments is 124. There's a considerable difference between the numbers of data in two classes. So that the classifier

may have tend to classify Hate comments as No hate comments increasing the number of False Positives in the prediction set.

BoF Features

BoF1 Features Set

Table 5.9 Results of BoF1 Features

	Accuracy		Precision		Recall		F-Score	
	DS500	DS1000	DS500	DS1000	DS500	DS1000	DS500	DS1000
SVM	0.5	0.53	0.51	0.55	0.5	0.53	0.469	0.54
Logistic Reg.	0.579	**0.69**	0.579	**0.689**	0.579	**0.69**	0.579	**0.689**
Naïve Bayes	0.569	0.62	0.569	0.62	0.569	0.62	0.56	0.62
Decision tree	0.51	0.599	0.51	0.589	0.51	0.599	0.51	0.599
KMeans	0.5	0.5	0.5	0.52	0.5	0.5	0.5	0.51

As in BoW features it is seen that always DS1000's F-score values are greater than DS500's F-score values making it clear that the classifier models perform better with BoF1 features when the amount of data fed to the models get increased. KMeans model has the worst performance with the F-score value 0.5 and Logistic regression model has achieved the best performance among the five models with a F-score values of 0.689. The confusion matrix of predicted results of Logistic Regression Model is as follows.

Table 5.10 Confusion Matrix 03

		Predicted Class	
True Class		1	0
	1	56	68
	0	31	175

Here also the problem which was in the best performing model with Tfidf features is seen. The number of misclassifications of hate comments is higher than the number of correctly classified hate comments. The model has given an overall good F-score value due to its ability to correctly classify True Negatives.

BoF2 Feature Set

Table 5.11 Results of BoF2 Features

	Accuracy		Precision		Recall		F-Score	
	DS500	DS1000	DS500	DS1000	DS500	DS1000	DS500	DS1000
SVM	0.579	0.66	0.579	0.65	0.579	0.66	0.569	0.66
Logistic Reg.	0.599	**0.69**	0.599	**0.69**	0.599	**0.69**	0.599	**0.689**
Naïve Bayes	0.569	0.609	0.569	0.609	0.569	0.609	0.560	0.609
Decision tree	0.5	0.579	0.5	0.589	0.5	0.579	0.5	0.589
KMeans	0.409	0.39	0.409	0.429	0.409	0.39	0.409	0.40

Except the KMeans model all other four models have performed better when the size of the dataset is increased. At the same time KMeans model has the worst performance compared to other models. According to the results, Logistic Regression Model has the best performance with an F-score of 0.689 and the confusion matrix of predicted results of Logistic Regression Model is as follows.

Table 5.12 Confusion Matrix 04

		Predicted Class	
True Class		1	0
	1	54	70
	0	28	178

According the confusion matrix, it is clear that the number of misclassifications of True positives as false positives is greater than that of its correct classification. F-score value of the model has increased due to the correct classification of true negatives. This may has happened due to the imbalance nature of the testing dataset.

Comparison of Different Features

Out of all the best performing models with different feature types the Best Models were selected. As mentioned before under each feature type always almost all the models have performed better with DS1000 dataset. So, the best models for particular feature type were selected with respect to DS1000 dataset.

Table 5.13 Comparison of Models

Model	Accuracy	Precision	Recall	F-Score
Naïve Bayes-BoW	0.709	0.709	0.709	0.709
Naïve Bayes-Tf-idf	0.739	0.75	0.739	**0.719**
Logistice Reg. BoL1	0.69	0.689	0.69	0.689
Logistic Reg. BoL2	0.69	0.69	0.69	0.689

Naïve Bayes Model with Tfidf features performs best out of all models. At the mean time it was clear that Naïve Bayes Models performs well with almost all the feature types more than other models when compared to other four models.

Chapter 6 - Conclusion

This chapter includes a review of the research aims and objectives, research problem, limitations of the current work and implications for further research.

6.1. Conclusions about Research Questions(aims/objectives)

The main aim of the research was to explore and see whether online hate speech can be identified automatically or not. Five models were built using both supervised and unsupervised machine learning algorithms in order to accomplish this task and according to the results we can conclude that online hate speech can be identified automatically.

One of the main objectives of the study was to create a text dataset using comments available in Sri Lankan news sites. A local English dataset was created by collecting reader responses of articles published in Colombo Telegraph website. Totally there were 1500 comments collected and out of them 1000 comments were manually annotated mentioning whether comments contain hate or not.

Then our objective was to identify an appropriate lexicon based method for hate speech identification. Google bad word list was used as the hate lexicon to build the lexicon based method. Each and every comment of the dataset was read one by one and a count of hate words in the comment was extracted. This count was used as a feature named 'Hate word count' to be used as a feature in machine learning models for each and every data in the dataset. Through this mechanism the combination of the lexical based method and machine learning method was done.

As mentioned before five models were built using four supervised learning algorithms and one unsupervised learning algorithm. Support Vector Machine, Logistic Regression, Naïve Bayes algorithm, Decision Tree algorithm were used for supervised learning models and KMeans clustering algorithm was used for the unsupervised learning model. Supervised learning models performed better than the unsupervised learning model with all the feature types considered.

Finally, accuracies and F-scores of all the models were compared in order to explore the most suitable classifiers for the task of hate speech identification. According to the analysis done on

results it was identified that Naïve Bayes Model with Tfidf features performs best out of all models with an F-score value of 0.719. Mean time the effect of the dataset for the accuracy of the classifiers was explored and was able to come with a general conclusion that classifier models perform better when there is more data. But this was not always true for the model which used KMeans clustering algorithm. Out of all five models KMeans clustering model was the model which had the worst performance in almost all the scenarios. This may be the reason behind the framing of the problem online hate speech identification as a supervised learning task.

6.2. Limitations

The main limitation of this research was the availability of annotated data. Collection of data is not a big issue. But manual annotation is time consuming and difficult to do. Since it was noticed that the classifier models perform better when the size of the dataset is increased, if there are more annotated data better results can be achieved. But amount of annotated data is very limited.

Google bad word list was used to build the lexical based method for the task. It is just a word list which contains words banned by Google Company. Currently there is no hate lexicons build for at least English language with a rate which represents to what extent this word is hate. If there is such a hate lexicon the accuracy of lexical based approaches can be increased.

6.3. Implications for further research

This research mainly focused on comparing different models for hate speech identification on a local English dataset. Although it was noticed that supervised learning models perform better than unsupervised learning models it is better to try out other unsupervised learning techniques for the task since KMeans clustering model performed little bit better with few feature types. At the same time combining different feature types together and then training and testing the models can be done as a future work.

Since it was clear that the amount of data is not enough to gain better results, the dataset should be expanded further. A semi-supervised classification approach can be used accomplish the task of annotating the dataset and training the models. As mentioned in Limitations section data annotation is the most difficult task rather than collection. So, if a large amount of comments can be collected at least 6000 (for example), then using the current 1000 annotated comments train a

classifier and run it on un-annotated comments to label them. For each and every labeled comment that's most likely correct, add it to the annotated text. By repeating this process until no more, most likely correct comments are achieved an annotated corpus can be created easier than manual annotation.

References

[1] YouTube Community Guidelines [Online]. Available: https://www.youtube.com/yt/policyandsafety/communityguidelines.html

[2] The Twitter Rules [Online]. Available: https://support.twitter.com/articles/18311#

[3] Facebook Comment Policy [Online]. Available: https://www.facebook.com/help/

[4] No Hate Survey Results [Online]. Available: http://www.nohatespeechmovement.org/surveyresult

[5] Colombo Telegraph Comment Policy [Online]. Available: https://www.colombotelegraph.com/

[6] N. D. Gitari, Z. Zuping, H. Damien, and J. Long, "A Lexicon-based Approach for Hate Speech Detection," *Int. J. Multimed. Ubiquitous Eng.*, vol. 10, no. 4, pp. 215–230, 2015.

[7] E. Riloff and J. Wiebe, "Learning extraction patterns for subjective expressions", Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 105-112, 2003

[8] E. Cambria, F. K. Uk, R. Speer, and F. K. Uk, "SenticNet: A Publicly Available Semantic Resource for Opinion Mining," AAAI fall symposium: commonsense knowledge, pp. 14–18, 2010.

[9] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," *arXiv Prepr. arXiv1703.04009*, 2017.

[10] Hatebase.org [Online]. Available: https://www.hatebase.org/ 15

[11] A. Wester, L. Øvrelid, E. Velldal, and H. L. Hammer, "Threat detection in online discussions," *WASSA@ NAACL-HLT*, pp. 66–71, 2016.

[12] Hugo Lewi Hammer.,"Detecting threats of violence in online discussion using bigrams of importantwords". Proc. Intelligence and Security Informat- ics Conference (JISIC), pp 319–319, 2014.

[13] Pedregosa, F., et al., "Scikit-learn: Machine learning in Python". Journal of Machine Learning Research" 12:2825–2830, 2011.

[14] W. Warner and J. Hirschberg, “Detecting hate speech on the world wide web,” Proceeding LSM ’12 Proc. Second Work. Lang. Soc. Media, no. Lsm, pp. 19–26, 2012.

[15] David Yarowsky, Decision Lists for Lexical Ambiguity Resolution: Application to Ac- cent Restoration in Spanish and French. In ACL-94, Stroudsburg, PA, pp. 88-95, 1994

[16] Z. Waseem and D. Hovy, “Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter,” Proc. NAACL Student Res. Work., pp. 88–93, 2016.

[17] Z. Waseem, “Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter,” Proc. 2016 EMNLP Work. Nat. Lang. Process. Comput. Soc. Sci., pp. 138–142, 2016.

[18] P. Burnap and M. L. Williams, “Hate Speech, Machine Classification and Statistical Modelling of Information Flows on Twitter: Interpretation and Communication for Policy Decision Making,” pp. 1–18, 2014.

[19] Y. Mehdad and J. Tetreault, “Do Characters Abuse More Than Words ?,” *SIGDIAL Conf.*, no. September, pp. 299–303, 2016.

[20] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, “Deep Learning for Hate Speech Detection in Tweets,” *Proc. 26th Int. Conf. World Wide Web Companion*, no. 2, pp. 759--760, 2017.

[21] C. J. Hutto and E. Gilbert, “VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text,” Eighth international AAAI conference on weblogs and social media, pp. 216–225, 2014.

[22] J. P. Kincaid, R. P. F. Jr, R. L. Rogers, and B. S. Chissom, “Derivation Of New Readability Formulas (Automated Readability Index , Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel RESEARCH BRANCH REPORT 8-75 READABILITY INDEX , FOG COUNT AND FLESCH READING EASE FORMULA) FOR NAVY ENLISTED PERSONNEL,” 1975.

[23] S. Bird. "NLTK: the natural language toolkit." In Proceedings of the COLING/ACL on Interactive presentation sessions, pp. 69-72. Association for Computational Linguistics, 2006.