2018 International Conference on Advances in ICT for Emerging Regions (ICTer) : 273 - 278

Identification of Hate Speech in Social Media

N.D.T. Ruwandika[1], A.R. Weerasinghe[2]

[1-2]*University of Colombo School of Computing, 35, Reid Avenue, Colombo 00700, Sri Lanka*

[1]tharushiruwandi26@gmail.com, [2]arw@ucsc.cmb.ac.lk

Abstract— An exploration of different approaches to detect hate speech in social media is present in this paper. Due to the rapid growing of online content hate speech has become a common issue which can influence variety of hate crimes. So, there is a need to find an accurate and efficient technique to detect online hate content and flag them automatically. The experiment was carried out using a local English text dataset. Hate speech is defined as the usage of language to insult or spread hatred towards a group or individual based on religion, race, gender or social status for the experiment. Then a comparison of both supervised and unsupervised learning techniques with different feature types for the task of hate speech detection was done. From all the supervised and unsupervised models Naïve Bayes classifier with Tf-idf features performed best with an F-score of 0.719.

***Keywords* — Hate speech, Naïve Bayes Classifier, Supervised Learning, Tf-idf, Unsupervised Learning,**

I. INTRODUCTION

Social media is a platform which provides a kind of virtual life for people to openly express feelings, opinions and beliefs. Websites dedicated as forums, social networking, wikis and micro blogging are examples for different types of social media. A website dedicated as a news forum was selected in this research. Finding as internationally accepted definition for hate speech is difficult. According most national and international legislations hate speech is referred as expressions that prompt to harm, discrimination, hostility and violence based on an identified social group or demographic group. Hate speech can include any form of expression such as images, videos, songs as well as written comments. Written comments are considered in this research since; the task of hate speech detection is considered as a text analytics task using natural language processing techniques.

Different social media platforms like Facebook, Twitter, and YouTube have used different policies to handle hate speech. According to YouTube community guidelines [1], free speech is encouraged while hate speech is not permitted. The content that promotes violence, hatred against individuals or groups based on race, religion, disability, gender, age, veteran status, sexual orientation is defined as hate speech. Manual flagging and filing abuse reports are the options currently given by YouTube to report about hate content in YouTube. Twitter [2] strictly prohibits the promotion of hate content, sensitive topics and violence globally through twitter. They also have come up with few different policies for this. Facebook [3] too has mentioned their policies for users about the content the user posts. In case of any policy abuse sending messages to the responsible person of posting, un-friend persons and block persons are few options given by Facebook to control promoting hate content through Facebook. Not only these three-social media, but almost all the social media have come up with their own definitions, policies for hate speech. It represents the importance of detection and removal of online hate content.

Currently there are quite lot of researches done for automatic detection of online hate speech. But finding a generalized mechanism for the task is difficult since hate speech is context dependent and language dependent. So, this research was conducted using a dataset prepared by collecting reader responses of a Sri Lankan news forum. Meantime according to literature it was identified that this task is usually framed as a supervised learning problem and there are no considerable number of researches conducted with unsupervised leaning techniques. So, applying an unsupervised learning algorithm and exploring the results were one of our intentions.

Newly created our own dataset was used for the research. Data collection and annotation was done to prepare the dataset. Then the research will be focused on a lexicon-based approach with the combination of a machine learning technique. Both supervised learning and unsupervised learning algorithms were used to build the classifier models. Four different feature types were used with the five different models to evaluate the models. Lexicon based approach was used to extract features from text data which were to be used in the machine learning models. Then finally results of five models were compared and analyzed.

The next sections of this paper will be structured as follows. The reviewed literature will be described in Section 2 of the paper. Section 3 of the paper presents experimental setup used for the research explaining dataset, definition of hate speech, preprocessing, feature extraction, classifier models and evaluation models. Results of the research will be discussed in Section 4 and finally in Section 5 conclusion of the research will be presented.

978-1-5386-7352-2/18/$31.00 ©2018 IEEE

II. Related work

Automatic hate speech detection does not have a long history, but there has been a huge interest in the recent ten years. Since in most scenarios comments, posts were used for hate speech detection, the problem is classified as a natural language processing problem. Three main types of approaches were identified for hate speech detection.

A. Lexical 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. N.D.Gitari et al [4] presents a classifier model for hate speech detection using a lexicon. The methodology proposed by them is comprised of three steps. A rule based and learning approach is used for subjectivity detection as first step. Then a lexicon for hate speech had been built in the 2nd step. Negative polarity words, hate verbs and theme based grammatical patterns were used as features to build the lexicon. Using those three types of features rules were generated to classify a sentence as hate or not as the 3rd step. An F-score of 70.83 was achieved for the combination of all three feature types.

B. Machine Learning Approaches

Machine learning approaches are the most commonly seen approach used then. A multi-class classifier to distinguish between hate speech, offensive language and none of them is presented by Davidson et al [5]. Logistic regression with L2 regularization has been used to build the final model. Their best performing model has an overall precision of 0.91, recall of 0.90 and F1-score of 0.90.

Z. Waseem et al [6] have evaluated the influence of different features for prediction of hate. A logistic regression classifier with 10-fold cross validation had been 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.

C. Hybrid Approaches

Hybrid approaches are used by many researchers. 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.

Results of the research conducted by A. Wester et al. [7] shows that combination of lexical features outperforms the use of more complex syntactic and semantic features for the task of detecting online hate. Maximum Entropy, SVM and Random Forest are the three different classification frameworks used. Basic lexical features, word forms, lemmas and n-grams were used as initial set of features. Then in the second-round different combinations of mentioned features were used. According to their analysis Bag-of-Word model and lexical n-gram model with both Maximum Entropy and SVM classifiers were selected to build the final model. From them n-gram model has outperformed BoW model with both SVM and MaxEnt with F-scores of 0.6885 and 0.6860 respectively.

Usage of paragraph level features for the first stage of classification was proposed by Warner et al [8]. They have used the hypothesis that hate speech resembles a word sense disambiguation task. A template-based strategy presented in Yarowsky et al. [9] has been used to generate features from the corpus. SVM classifier has given the best performance with an F-score of 0.63.

D. Discussion

In all the researches done in this research area they have first consider a lot about narrowing down the definition of hate speech regarding their research. It is important to define hate speech in a 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 [10] 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 [8] 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. [11] and few others.

Basically, many researchers have focused on machine learning and lexicon-based models while only few have concerned on deep learning approaches [11,12] 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.

III. Experimental Design

Entire process of experiment can be divided into four main steps as Data collection and annotation, Data preprocessing, Feature extraction, Classification and Evaluation. NLTK library and Sci-kit learn library was used with Python 2.7.9 for implementation activities. Fig. 1. And Fig. 2. Shows the models used in the research for supervised learning and unsupervised learning respectively.

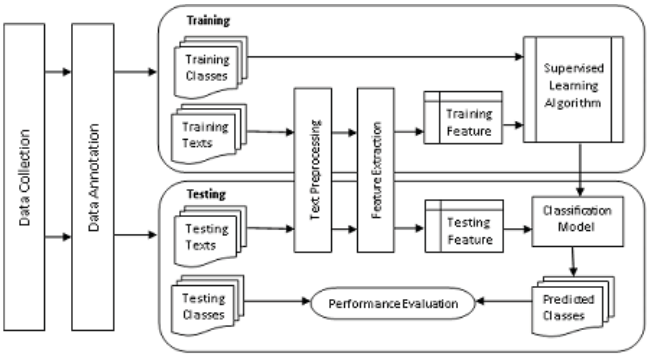

Fig. 1. Research Design for Supervised Models

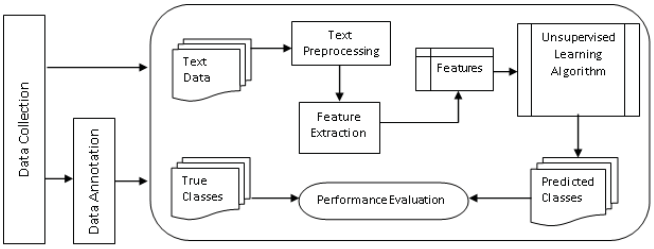

Fig. 2. Research Design for Unsupervised Models

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.

A. Dataset

Our data set is comprised of user written comments from different articles in Colombo Telegraph website. 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.

All articles in which more than 25 comments have 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 I. presents an overview of the number of annotated comments with hate and without hate.

TABLE I
CLASSES OF DATASET

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

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."

B. Data Preprocessing

Natural Language Processing Toolkit, NLTK library was used all preprocessing activities. Preprocessing was started through examining each comment and expanding verb contractions. Since English texts were used, this was applicable and needed. Then tokenization of words was carried out and words were lemmatized. Punctuations and stop words of English were removed then.

C. Feature Extraction

All the feature extraction activities were done using sci-kit learn toolkit. Bag-of-Words features (BoW) were extracted using countvectorizer. Tf-idf vector which is used to measure the importance of a word or term was extracted using Tf-idf vectorizer. Then the 3rd feature type Bag-of-Features (BoF) was created. Here Flesch reading Ease score (FRE), Flesh-Kincaid Grade Level (FKRA), Sentiment score, Number of characters, Number of words, Number of syllables and Tf-idf vector of each comment was extracted to create the 3rd feature type. The mentioned features used to create the 3rd feature set are a subset of features presented in T. Davidson et al. [5]. VaderSentiment presented by C.J.Hutto et al. [13] was used to obtain the sentiment score of a comment.

Hate word count was added as an extra feature to the 3rd feature type. To extract hate word count, a lexicon was used. Google Bad Word List was used as the lexicon. Using the features of 3rd feature type two feature sets were created as follows.

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, Tf-idf, and BoF features. All the models were trained using these feature types.

D. Classification Models

Five different classification frameworks were tested in our experiment and they were Support Vector Machine, Naïve Bayes Classifier, Logistic Regression Classifier, Decision tree Classifier and K-Means Clustering algorithm. So, one

unsupervised learning model was created with four supervised learning models. The task was considered as a binary classification problem.

All five models were trained using the all four types of features of training dataset and evaluated using features of testing dataset. 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.

E. Evaluation Procedure

Since this research is related to data analytics and natural language processing accuracy, precision, recall and F-score was chosen for evaluation metric. In the dataset a comment contains either hate or they do not. So, that our classifier is 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. Since one unsupervised model was built a special procedure was conducted to evaluate the model. Experiment was done several times using small number of data (100 comments at a time) and labels given by clustering process was analyzed first. Then according to the labels of clusters evaluation metrics were created for the unsupervised model.

IV. RESULTS

Main objective is to examine the behavior of different algorithms with different feature types to detect hate speech. So, different feature sets and different classifiers were compared regarding accuracy, precision, recall and F-score measures.

For the interpretation of results the of whole dataset including all 1000 comments will be represented as “DS1000”. Table II represents the partitioning of the whole dataset as training dataset and testing dataset. Table III. represents the number of comments in 2 different classes in testing dataset.

TABLE II
PARTITIONING OF WHOLE DATASET (DS1000)

	Number of Comments
Training dataset	670
Testing dataset	330

TABLE III
CLASSES OF TESTING DATASET OF DS1000

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 IV. shows the partitioning of the dataset with 500 comments in to training and testing while Table V. shows the number of comments in each class of testing dataset of DS500.

TABLE IV
PARTITIONING OF WHOLE DATASET (DS500)

	Number of Comments
Training dataset	335
Testing dataset	165

TABLE V
CLASSES OF TESTING DATASET OF DS500

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

A. BoW Features

Results of both datasets with BoW features are represented in Table VI.

TABLE VI
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
K-Means	0.5	0.66	0.51	0.66	0.5	0.66	0.5	0.66

DS1000 has performed better than DS500, since 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 K-Means 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.

B. Tf-idf Features

Results of both datasets with Tf-idf features are represented in Table VII.

TABLE VII
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
K-Means	0.56	0.45	0.56	0.46	0.56	0.45	0.56	0.46

All models except K-Means model has performed better with more data since F-score values of DS1000 is greater than F-score values of DS500 except in K-Means model. So, it's clear that when the size of the dataset gets increased supervised learning models perform better than unsupervised learning model with Tf-idf 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.

C. BoF1 Features

Results of both datasets with BoF1 features are represented in Table VIII.

TABLE VIII
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
K-Means	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. K-Means 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.

D. BoF2 Features

Results of both datasets with BoF2 features are represented in Table IX.

TABLE IX
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
K-Means	0.409	0.39	0.409	0.429	0.409	0.39	0.409	0.40

Except the K-Means model all other four models have performed better when the size of the dataset is increased. At the same time K-Means 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.

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 feature type were selected with respect to DS1000 dataset. Table X. shows a comparison of best performed models.

TABLE X
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**
Logistic 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 Tf-idf 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.

V. CONCLUSION

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 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 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 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 K-Means 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 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 Tf-idf 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 could 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 K-Means clustering algorithm. Out of all five models K-Means 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.

VI. FUTURE WORK

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 K-Means 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 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://help.twitter.com/en/rules-and-policies/hateful-conduct-policy

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

[4] 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.

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

[6] 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.

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

[8] 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.

[9] 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

[10] 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.

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

[12] 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.

[13] 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.