Identification of Abusive Sinhala Comments in Social Media using Text Mining and Machine Learning Techniques

H.M.S.T. Sandaruwan[#1], S.A.S. Lorensuhewa[#2], M.A.L. Kalyani[#3]

***Abstract*— With the technology revolution, most of the natural languages that are used all over the world have won the digital world. Therefore, people use modern technologies such as Social Media and the Internet with their native languages. As a result, people who are with self-ego on their tradition, race, caste, religion and other social factors, tend to make abusiveness on others who do not belong to the same social group by their native languages. Since the Social Media platforms do not have centralized control, it has become a good platform to advertise their backward ideas without being governed and monitored. The Sinhala language has also been added to most famous Social Media platforms. Though the Sinhala language has more than 2500 years of history, it does not have rich resources for computer-based natural language processing. Therefore, it has been a very difficult task to automatically detect Sinhala abusive comments which are being published and shared among Social Media platforms. Therefore, here, we have used evenly distributed 2000 Sinhala comment corpus among offensive and neutral classes to train three different models: Multinomial Naïve Bayes (MNB), Support Vector Machine (SVM) and, Random Forest Decision Tree (RFDT) and the features were extracted from Bag of Word model, word n-gram model, character n-gram model, and word skip-gram model to automatically detect Sinhala abusive comments. After the training process, each model was tested with 200 evenly distributed comment corpus and MNB showed the highest accuracy of 96.5% with 96% average recall for both character tri-gram and character four-gram models. Further, two lexicon-based approaches called cross-lingual lexicon approach and corpus-based lexicon approach were considered to detect Sinhala abusive comments. From these two lexicon based approaches, the corpus-based lexicon gave the highest accuracy of 90.5% with an average recall of 90.5%.**

***Keywords*— Sinhala Abusive Comment detection, Machine Learning, Text Mining, Natural Language Processing.**

I. Introduction

With the rapid growth of technology, traditional communication media such as newspapers, radio channels, and television have been invaded by the Internet-based communication media. Since these new Internet-based communication media make person based telecommunication platforms, people who do not have a space to address the mass community, welcome the Social Media platforms warmly. Therefore, almost all the people around the world now are connected by some kind of Social Media.

Despite traditional ideologies, the young generation of each country joins the Social Media networks massively day by day. Since non-international languages are facilitated in Social Media, people who only know their mother tongue, have been able to work on these platforms without having any difficulties. It is a great opportunity for the people who do not have a place to share their ideologies, experiences or any other thing like to be shared.

Though the basic and primary concept of Social Media is connecting people, some social media users misuse this primary goal by making offense towards others. Since the number of users who have involved with Social Media is higher than traditional mass media, the impact that can be put on society by these Social Media are very serious.

When we narrow down our focus into Sri Lanka, according to the online statistics [15], more than 16 million people are using Sinhala as their communication language and more than 7 million people use the Internet. The Sinhala language has a remarkable history and it belongs to the Indo-Aryan language family which is a subfamily of Indo-European [1]. It consists of 18 vowels and 42 consonants, but now only uses 16 vowels and 41 consonants [17]. Since Sinhala is a morphologically rich language [1], it is a very complex task to make algorithmic natural language processing resources.

Like other languages, Sinhala has also been used in Social Media networks to post abusive comments. From 2014 to 2018, Sri Lanka faced two racism based riots due to these kinds of abusive comments and the government as well as Social Media authorities could not control the situation, as there was not a possible method available to identify Sinhala abusive comments which were being shared. In the 2018 incident, the government has to ban Social Media for a few days, but users were able to access their accounts by using third party *Virtual Private Networks* (*VPN*). Hence, it is clear that banning Social Media will not be an optimal solution to control such a situation in the future. It was proven by a few subsequent incidents that occurred after the 21st April Easter bomb attack in Sri Lanka 2019. Even the government banned Social Media at that time, people shared a lot of abusive comments by enabling VPN.

One of the major problems in Social Media is the lack of governance in the content which is being published and shared. Until someone reports that a particular content violates the policies, that content will not be monitored and governed. Also, Social Media providers do not have enough linguistic expertise to handle such kind of situations. Most of the time, when a complaint is received, they say the reported content does not violate the policies even the content violates the policies.

Manuscript received on 14 Nov. 2019. Recommended by Dr. D.D. Karunaratna on 02 April 2020.

This paper is an extended version of the paper "Sinhala Hate Speech Detection in Social Media using Text Mining and Machine learning" presented at the ICTer 2019.

H.M.S.T. Sandaruwan, S.A.S. Lorensuhewa, and M.A.L. Kalyani are from the Department of Computer Science, University of Ruhuna, Sri Lanka.. (stsandaruwan001@gmail.com, aruna@dcs.ruh.ac.lk, kalyani@dcs.ruh.ac.lk).

Although everyone has a right to share their opinions and ideologies, that ideas should not invade others' feelings, because we all are human beings and have a right to be prevented from any kind of offensiveness. Since there are no perfect men and women, we should have a way to control these situations, such as social riots which occurred because of a few abusive comments.

A social study [1] done in 2014 shows that Sri Lankan people also use the Sinhala language in Social Media to spread abusive comments and it discusses the importance of identifying online Sinhala abusive comments in an automated manner. In this research, we focused on the identification of Sinhala abusive comments using text analytical models and lexicon models, hence it is a binary classification.

II. Literature Review

Abusive speech detection on web content is an ongoing research area nowadays. There is a considerable amount of researches has been done for the English language ([3],[7],[13]), but very few for other languages such as Arabic [16], German [7], and Chinese. Since 2015 to 2018, a lot of researches has been carried to detect abusive speeches, therefore significant attention has been attracted by the researchers who involve with the text mining area.

Though it is difficult to compare different methodologies that have been used for hate speech detection due to the different data sets they used, we can identify the main approaches that were used to detect online abusive comments. According to the literature that we studied, abusive speech identification has been carried through two different approaches namely, lexicon-based abusive speech detection and machine learning based abusive hate speech detection. Further, some researchers have used a combination of these two approaches to detect abusive comments.

According to the research [3] which was published in 2015, the lexicon-based approach has been considered to detect online hate speeches. They started the research with comments collection phase and concerned online forums, blogs and comment sections in news reviews to gather the comments. As the research data sources, they have created two data sets. In the first source, it includes 100 blog postings from different 10 web sites and each site, 10 blog postings were collected. These 10 web sites were taken from a list provided in the Hate Directory [4]. The next source was the web page content which was related to the Israel-Palestinian conflict and it was a 150-page web document. Annotation of content had been done by two graduate students in their university and 30% of each source was annotated.

This research has been conducted in three phases, as the first phase subjectivity detection has been done. In that case, they needed a sentiment lexicon and therefore they have used [5] and *SentiWordNet* [6] as resources. To determine whether a given sentence is subjective or not, they extracted the positive or negative scores from each sentiment word in the sentence. Then they calculated both positive and negative scores and subtract the negative score from a positive score. If the *synset* score is greater than 0.5 or less than -0.5, the given sentence is considered as positive or negative respectively. Therefore, if a sentence has either a positive or a negative score according to the given rules, that sentence was considered as subjective and not objective. About 56% of the first corpus and 75% of the second corpus were subjective in this research.

The lexicon of hate speech was built as the second phase. Therefore, they have used three different sets of features. They identified negative opinionated words from the subjectivity analysis and that words were selected as the first feature set of the hate speech lexicon. For the second set, they selected all verbs that are related to their hate corpus but not in the first feature set, then they gathered hypernyms. If those words exist in their corpus, then that words are added into the lexicon. The third set was created with hate nouns, which were related to three types such as religion, race, and nationality. By using Named Entity Recognition (*NER*) software, they have identified the source and the recipients of opinions. The experiment was done as follows.

At the first stage of the experiment, they only considered negative semantic features, but the accuracy was less than 70%. Then hate verbs were included in the semantic features, so the accuracy was increased above 70%. The best results obtained were an F-Score of 70.83 for the first corpus with three features.

Since we do not have a sentiment lexicon as *SentiWordNet* for the Sinhala language to evaluate a sentence as positive or negative before the phase of abusive speech detection, we have to directly apply the lexicon with given comments. As Sinhala does not have a good *NER*, we have not been able to obtain the source and the recipient as they did in their research.

However, the lexicon-based approaches are not enough in the process of the online abusive speech detection process, because the process of identifying abusive comments from the lexicon approach is limit to the words that exist in the pre-built lexicon. Therefore, some researches have focused on machine learning techniques to identify online abusive comments more precisely.

Machine learning allows computers to learn from the data without applying all the steps or the instructions by a programmer that is needed to perform some specific task. The foundation of machine learning is training data. Therefore, training data plays a huge role in machine learning and the learning algorithm generates a new set of rules, based on the inference of the data. These algorithms are formally known as the model and the same algorithms can be used by different data sets to generate different models.

Further, machine learning approaches can be categorized into two strategies as supervised learning and unsupervised learning. Supervised and unsupervised learning is divided based on a feature of the training data. If the training data is labeled with its belonging class, the learning model will be considered as supervised learning. Otherwise, the model is known as an unsupervised model. Among various supervised learning algorithms, Naïve Bayes, Decision Tree algorithms, Support Vector Machine algorithm, and Logistic regression algorithm have been considered in abusive speech classification. Thus unsupervised learning algorithms such as K-means clustering and hierarchical clustering are very few in the abusive speech detection process.

In 2018, research [7] has been done on the topic of hate speech detection and they have concerned the German language as the targeting language and have tried to apply the approaches that were used in English hate speech detection researches to build their models. In that study, they have shown the importance of hate speech detection on other languages as English due to some crisis. Therefore, the main

goal of their research was to "*investigate the potential value of automatic analytics of German texts to detect hate speech*".

As the dataset, the user-generated comments were taken from the news platforms on the Internet and they had to use web scraping technology since most German news platforms do not offer APIs. Therefore they have used a Python framework called *Scrapy*; with that web scraping technology, 376,143 comments and 21,740 articles have been collected and that comments were annotated by 247 individual participants. In that process, they have only concerned two class labels as "Hate" and "Non-hate". Then they received a total of 11,973 rated comments and they were distributed among three classes as 3875 hate comments, 6073 non-hate, and 2025 unclassified comments. For the feature extraction process, they have used Bag of Words (*BOW*), N-grams, Linguistic features such as number of punctuation marks, number of words in a comment and Word2Vec/Doc2Vec.

From annotated comments, they have used 811 hate and 1561 non-hate comments to create the model. Because of the unbalance of the two classes, comments were used under the sampling method. Then 1622 comments that include both hate and non-hate comments were applied to the logistic regression model and the highest accuracy was taken for the *BOW* feature with 0.7608.

Since they did not have a hate speech corpus, they used web crawling techniques to extract the comments from online forums. We also had to use the same web scraping approach as we did not have a comment corpus that contains Sinhala abusive speeches. This research [7] was based on the German language and it shows that the *BOW* features are good at detecting hate speeches as it works with English despite the language difference. Hence it guided us to use the BOW feature with the Sinhala abusive speech detection.

As the very first attempt in hate speech detection in Sinhala comments that are published with the Sinhala UNICODE, research [8] was published in 2018. In that study, they have focused on Sinhala racism based speech detection. However, as the first attempt on Sinhala abusive detection, it comes with few limitations. The first limitation is, though they could get 70.8% accuracy, that accuracy cannot be considered as a good measurement, because they have used an unbalanced data set which contains 73 racism based comments with 111 non-racism based comments. The second limitation is the feature extraction method that they used. Though a lot of similar researches ([19]) which was done on other languages has tried with several feature extraction methods, this study [8] has only considered the word bi-gram models. From that, we cannot get a clear idea about the best fitting features that can be used in Sinhala abusive comment identification process. Therefore, there is a significant need to identify other best fitting features, which can be used in Sinhala abusive comment detection process.

Since abusive comments are not limited to racism based comments, we need to consider other characteristics of abusive comments such as religion, gender, sexual orientation or any other disability since these types of comments should also be removed from Social Media. As we identified, the third limitation of this research is the model that they have used to train the model. Since *SVM* is considered as the only one model, we cannot get a clear idea of other existing models that can be applied with Sinhala abusive comments identification process.

Once we went through the literature, we found that all most all the researches have used PYTHON with NLTK libraries [18] for pre-processing and model-building. Further, supervised learning methods are widely used in abusive speech detection and rather than deep learning techniques, machine learning approaches have been used in this task. Therefore, we also considered the statistical model-based machine learning approaches due to the lack of a large number of comments to consider the deep learning approaches.

In [13], it has used natural process feature selection algorithms like character n-gram, word n-gram, word skip-gram and brown cluster with the Support Vector Machine classifier. Here, they have obtained 78% accuracy for the character 4-gram feature.

As the research [18] shows, unigram, bigram, and trigram features can also be used with the *TF-IDF* method and as per the results, they have found that Logistic Regression works better and the best model has resulted with 0.91 precision, 0.90 recall, and 0.90 F1-Score.

Recent research [19] done for the Indonesian language discusses the importance of identifying hate speeches in the Social Media that are published with the Indonesian language. Since they have not much-related researches regarding the Indonesian language, they were guided by other language researches. Therefore, they have used word n-gram and character n-gram feature extraction methods with the Naïve Bayes, *SVM*, Bayesian Logistic Regression (*BLR*), and *RFDT* classifiers. The best performance of 93.5% of F-measure was scored for the word n-gram features with the *RFDT* algorithm.

By considering all most all the factors, we proceeded the research with a few different feature groups; Character n-gram, word n-gram, word-skip gram, and *BOW* features as experimented in ([7],[8],[19]). These extracted features were trained and tested with a few statistical models; *MNB*, *SVM*, and *RFDT* as these classifiers have shown good performances ([8],[18],[19]) despite the language difference to fill the blanks in Sinhala abusive comment identification process. Other researches such as ([7],[18],[19]) on various languages shows the application of pre-processing techniques such as stemming and removing *stop words* are best for model training. Therefore, we also have considered these pre-processing techniques with our study.

III. Methodology

The goal of our research was to compare multiple models and investigate the effect of features that can be used to train models in the Sinhala abusive speech detection process. Therefore a methodology based on training, validation, and testing have been considered in our research. At the training phase, we conducted experiments with three different variations. As the method 1, we built a corpus-based lexicon with hate and offensive words, hence it is an abusive lexicon for Sinhala, and then that lexicon was used for abusive speech classification. As the method 2, machine learning techniques were used. By applying extracted features, three different models: Multinomial Naïve Bayes (MNB), Support Vector Machine (SVM) and Random Forest Decision Tree (RFDT) were trained. As the method 3, using trained models and created lexicon, we tested the testing data set that is containing 200 evenly distributed comments. Since we could not find any Sinhala abusive or offensive related corpus online, we initiated the research with the corpus construction.

A. Data Set Construction

In supervised learning methods, there is a need for a labeled data set that is required for the training. The resulting system's ability depends on the content of the data set and its annotation. When the data set is highly correlated with the considering topic, the predicting results can be guaranteed. Though there is a lot of hate speech annotated data set available on the Internet for many languages such as English, German and Arabic, there is no such resource available for the Sinhala language. Therefore, we extracted and annotated the comments according to the details given in the following section.

1) Collection of Sinhala comments:

Here we considered two online Social Media platforms: Facebook and YouTube with a Sri Lankan gossip site. Since Facebook and YouTube have APIs, we could access them by creating API keys. But for Sinhala gossip site, we had to build an in-house web crawler to extract the comments.

2) Comments filtering process:

Since some of the collected comments were not in Sinhala Unicode encoding, we filtered them out from our corpus. Further, some special characters such as emoji, URLs and other non-Sinhala characters were removed from our data corpus by using regular expressions.

B. Comments Annotation.

After the comment extraction process, extracted comments were annotated with the help of three annotators, since we are focusing on the supervised learning approach. Based on the majority vote, the class of comment was elected. If a comment was elected as a neutral one it was annotated with the label "0" and else if it is an abusive comment, it was labeled as "1". From the annotated comments, 2000 comments were selected to train the model while selecting 200 comments for testing the model randomly. TABLE I shows the individual instances of each class label.

TABLE I
ANNOTATED COMMENTS

Comment Type	Count
Offensive Comments	1100
Neutral Comments	1100

From the annotated comments, randomly selected 100 comments from each class and that comments set was used as the testing data set.

After the annotation process, we did a corpus analysis of collected data.

C. Corpus Analysis

To identify the characteristics of our corpus, we applied a few corpora analyzing techniques as follows.

1) Length Analysis:

At the first step of the corpus analysis, we did a length analysis of the corpus. The annotated data set was used for the analysis and we divided the comments into two corpora such as offensive and neutral. This comment separation was done based on the comments' class and it proceeded with a Python script. If the comment's label is "1" that comment is put to the offensive corpus and if the label is "0", it is put to the neutral corpus by this script.

Each corpus contained 1000 comments. At the length analysis process, we considered the following criteria and the results obtained are listed in TABLE II.

- Average word length

  In this analysis, we considered the average word length of words in each corpus. It is important to identify the number of characters that are used in each corpus, because every language has a fixed set of the abusive word list, though they may vary according to the context.
- Average sentence length

  We analyzed sentence length to check whether offensive and neutral speeches use a small number of words or not, as we wanted to discover the average sentence length that is used in the offensive and neutral comments.

TABLE II
LENGTH ANALYSIS OF EACH CORPUS

	Offensive	Neutral
Average sentence length	11.03	12.377
Average word length	4.973	5.033

According to the analysis, we found that neutral comments use more words than offensive comments and Sinhala abusive words have fewer characters than normal words.

2) Vocabulary Analysis:

Here, we have considered the number of words used in each corpus. Our objective was to identify the number of words used in all two types and we wanted to analyze the word count behavior in both corpora. Therefore, we did the following two analyses for each offensive and neutral corpus. In TABLE III, analyzed results are listed.

- Total number of words

  Here we considered the total number of words used in each corpus. Our objective was to compare and contrast the total number of words used in an offensive speech and a neutral speech.

- Total number of unique words

  We found the number of unique words in each corpus. We did this analysis to check whether people use the same set of words to make comments or not.

TABLE III
VOCABULARY ANALYSIS

	Offensive Corpus	Neutral Corpus
Total num. of words	11041	12389
Total num. of unique words	4642	5346

Though each corpus contained 1000 comments, from these results, we can conclude that offensive speeches use only a few words since it has less unique words than the neutral comments corpus and these words are used again and again.

3) Zipf's Law Analysis:

This law was invented in 1935 and it was the first academic study of word frequencies. It states that the frequency (f) of any word in a given natural language corpus is inversely proportional to its rank (r) in the frequency table. The formula of Zipf's law is as follows.

$$f \approx \frac{1}{r^a}$$

Where 'f' is the frequency of a word ranked 'rth' and the exponent 'a' is almost 1.

Zipf's law behavior on the Sinhala language has been experimented in 2004 [9] and it shows that the Sinhala language almost follows the law. Since their corpus is based on grammatically correct Sinhala sentences, we decided to study the behavior of Zipf's law on an ungrammatical, neutral and offensive corpus. Therefore, we divided our sentences into two corpora as we used in the above analysis. Each corpus consisted of 1000 comments and that comments were tokenized. The tokenization process was done in two approaches. In the first approach, stop words removing and stemming processes were not considered and in the second approach, words were tokenized after applying stop word removal and stemming techniques.

- Zipf's law behavior of abusive comments without stemming and stop word removing

  We took 1000 comments to study the behavior of Zipf's law in Sinhala offensive comments. In this analysis, we did not consider any stemming techniques and stop word removing techniques. Top 10 words with the highest frequency are shown in TABLE IV and some letters of some words are replaced by the "#" symbol due to the abusiveness of the words.

TABLE IV
TOP 10 OFFENSIVE WORDS WITHOUT STEMMING AND STOP WORD REMOVAL.

Rank	Word	Frequency
1	මේ	151
2	කෑ#	92
3	පොහ් න	90
4	හු#නෝ	67
5	පොහ්නයා	60
6	එපා	53
7	හු#නො	52
8	පු#	51
9	ගොන්	50
10	අනේ	50

Then we plotted the log value of offensive words frequency against the log value of word ranks. Following Fig. 1 shows the retrieved graph.

According to the graph that we obtained, we can identify that even the corpus is not grammatically correct; it follows Zipf's law.

When we further investigated, we found that there are a lot of words that frequently occurred are just used to construct the order of a sentence. In other words, some of the frequently occurred words are not useful in the separation of comment class since they are used in both abusive and neutral comments. Here, we considered them as stop words.

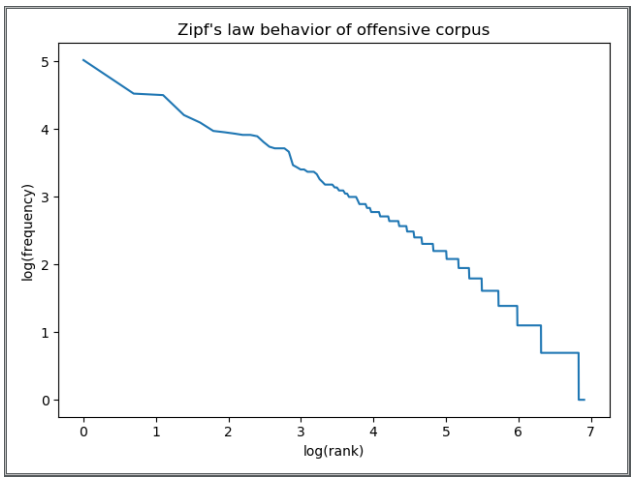

Fig. 1 Zipf's law behaviour of offensive corpus without stemming.

After removing stop words from the corpus, we did another experiment to analyze Zipf's law behavior.

- Zipf's law behavior of abusive comments with stemming and stop word removing

  Here, we applied the same approach in the offensive corpus. After applying the stemming and stop word removing techniques on the offensive corpus, we got word tokens with their frequencies. Obtained results are shown in TABLE V. Some characters of words in the table are replaced by the "#" symbol since they are disgraceful words.

TABLE V
TOP 10 OFFENSIVE WORDS WITH STEMMING AND STOP WORD REMOVING.

Rank	Word	Frequency
1	පොහ්න	227
2	හු#න	215
3	ප#	127
4	වේ#	106
5	කෑ#	103
6	පු#	102
7	එක	99
8	අම්ම	63
9	උබ	58
10	එප	55

According to the results, we observed that applying stemming as pre-processing is also a better practice since it reduces the feature vector by replacing words by their root word.

As done previously, we plotted the log value of offensive words frequency against the log value of word ranks. Following Fig. 2 shows the graph obtained.

According to the graph, we can conclude that, even after applying pre-processing techniques, the corpus follows Zipf's law.

D. Sinhala Abusive lexicons construction

Since there is no lexicon resource for Sinhala abusive speech detection, we made two lexicons by following two approaches. In the following sub-sections, these approaches are discussed.

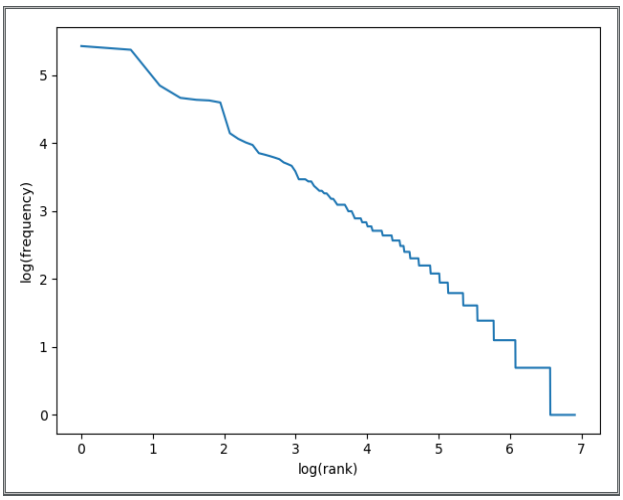

Fig. 2 Zipf's law behaviour of offensive corpus with stemming.

1) Dictionary-based abusive lexicon creating:

As there is no online dictionary for Sinhala hate speeches, we used "google bad word list" [10] based on the user policies of them and also other resources that contain Social media banned English words. Therefore, we used an online Sinhala English dictionary to map each banned word in English list to Sinhala.

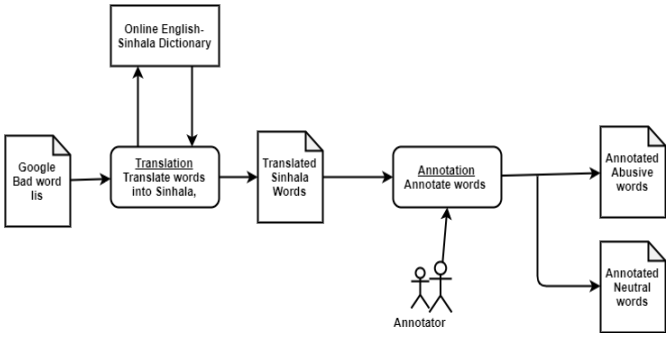

Fig. 3 Dictionary-based lexicon building.

Though we sent 1703 English bad words to the online Sinhala-English dictionary, it gave 1128 translated words. Then that translated words were given to the annotators to get classified. Among 1128 words, only 157 words were annotated as abusive. TABLE VI shows the annotator's agreement on each Sinhala word.

TABLE VI
ANNOTATOR'S AGREEMENT ON TRANSLATED SINHALA WORDS

Word Type	Count
Abusive words	157
Neutral words	971
Total translated words	1128

Since dictionary-based lexicon does not cover all words that can be used in the context of Sinhala abusive comments, we decided to consider corpus-based lexicon construction.

2) Corpus-based abusive lexicon creating:

In this approach, we used our 2000 comment corpus as the resource to identify words that are specific to Sinhala abusive speeches. We separated the comment corpus into two corpora called abusive and neutral by considering their class. Each corpus contains 1000 comments and using a seed word set taken from an online source, we searched their variations in the offensive corpus. Fig. 4 shows the steps that were used to build the corpus-based lexicon. This seed word set contains base forms of Sinhala abusive words. By adding suffixes to these seeds words, their variations were identified.

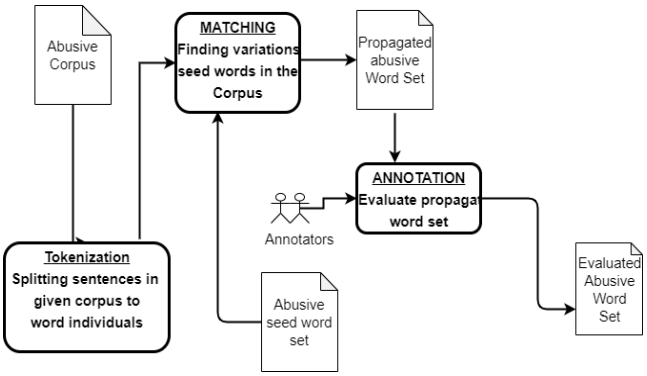

Fig. 4 Corpus-based lexicon building.

Annotators' agreement on each word is listed in TABLE VII. Then both dictionary-based and corpus-based lexicons were used to identify Sinhala abusive speeches.

TABLE VII
ANNOTATORS' AGREEMENT ON PROPAGATED WORDS

Seed Words	Propagated words	Annotators' agreement	Accuracy
64	279	277	99.28%

E. Text Pre-processing.

To reduce the dimensionality of the feature vector, we applied several famous pre-processing techniques, before the models are trained. All non-Sinhala characters, emojis, and URLs were removed as the basic step of pre-processing by using regular expressions. Further, the following standard techniques were used.

1) Stop words and Stop word removing:

Stop words are words that have a little meaning but they are essential to maintain the structure and grammatical relationship among other words in a sentence. In general, stop words are known as the most common words in a given language. In Natural Language Processing, these words are dropped to reduce the dimensionality of the feature vector. Since the sense of these words affects the sentiment of a given sentence, before the classification, it is essential to decide whether these *stop words* should be removed or not.

Even though Sinhala is a less resource language, in this study we have used a *stop word* list which is compiled and published by [11]. That list contains 425 words. But some words in that list are important as they are the cause for the negativity of a sentence as well as their presence in Sinhala abusive comments. Therefore those words were removed from the standard *stop word* list before the classification. The

removed words from that stop word list are listed in the following TABLE VIII.

TABLE VIII
REMOVED STOP WORDS FROM THE STOP WORD LIST

Removed words from Stop Word List	
නැත	අඩෝ
නොවේ	විහ්
බැහැ	ඕනෑ
බෑ	එපා
අපොයි	ඌයි
අයියෝ	ෂික්
අම්මෝ	ෂා
ආනේ	නැහැ
අප්පච්චියේ	නෑ

2) Stemming:

Stemming is the process of reducing a word into its word root by stripping recognized prefixes and suffixes. It is very important to identify stems of words because it reduces the dimensionality of the feature vector by converting words into its relevant word stem. Since Sinhala is a less resource language in natural language processing, it does not have a stemmer such as Porter's Stemmer for English. Therefore, in this study, we used a shallow stemming method proposed [12] for Sinhala. The main problem it has is, if the document does not contain the stem word itself, the algorithm is unable to find the stem of a particular word. Since its programmatic implementation is not available publically, we implemented the concept and applied it to our corpus. TABLE IX shows few stem roots that were found through our corpus.

TABLE IX
STEMS AND THEIR VARIATIONS

Stem	Words
අන්තවාදීන්	අන්තවාදීන්ගේ, අන්තවාදීන්ට
අමන	අමනයන්ගේ, අමනයනි, අමනයා, අමනා
කෙල්ල	කෙල්ලක්, කෙල්ලට, කෙල්ලන්ට, කෙල්ලයි

Identified stems were listed alphabetically and saved in a text file and it was used as a stemming dictionary in the process of feature extractions. The steps that we followed to apply stemming are shown in the following figure, fig. 5.

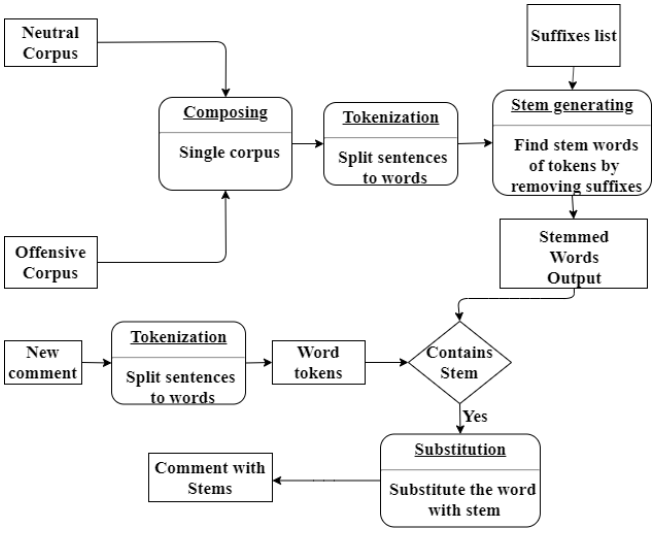

Fig. 5 Stemming.

Since this algorithm is based on removing suffixes, we needed a standard suffixes list for Sinhala. Therefore, we used a standard list that contains 413 suffixes, published by [11]. Some sample suffixes are listed in TABLE X.

TABLE X
SAMPLE SET OF SINHALA SUFFIXES

ක	ය
ක්	යක
කක	යකි
කක්	යක්
කකට	යකගෙන්

F. Feature Extraction.

In our study, we use machine learning algorithms to detect abusive Sinhala comments and therefore identified features through the literature could be used to train these models. Since Sinhala abusive comment detection is a novel area, it is a challenging task to identify best fitting feature types. However, in this study, we considered four different feature types: Bag of Words (BOW), word n-gram, character n-gram, and word skip-gram to train the models.

1) Bag of Words (BOW) features:

BOW is the most famous [7] and the simplest feature that can be used in natural language processing. Therefore, it has been used in many text mining related researches. In this approach, a text is represented as a bag of its words disregarding grammar and the word order. Though it neglects the grammar and the word order, the frequency of each word's occurrence is recorded with the word.

2) Word n-gram features:

Word n-gram is a word model that captures the structure of sentences or corpus. Though BOW is good at feature extraction, it is not sufficient since natural languages do not contain just words but words with some structure. These n-gram features can be a unigram, bigram, trigram or combination of these features. Similar researches in other languages such as English [13] show that considering word n-gram features to identify abusive speeches makes good results. Therefore we considered several groups of word n-gram features and they are listed in TABLE XI.

TABLE XI
WORD N-GRAM FEATURE GROUPS

Group	Pre-processing	Feature set
W01	RN+SR+ST	UG
W02	RN+SR+ST	BG
W03	RN+SR+ST	TG
W04	RN+SR+ST	UG+BG
W05	RN+SR+ST	UG+BG+TG

UG- Unigram **TG**- Trigram **BG**- Bigram
ST- Stemming **SR**- Stop word Removing **RN** – Removing non-Sinhala symbols

The main difference between word unigram and BOW is the word count that is considered in each approach. In BOW, all the words are considered when the feature vector is constructed. But

in word unigram, it does not consider all the words in the corpus to make the feature vector.

3) Character n-gram features:

Most of the languages such as English and Sinhala are composed of characters or letters, digits, punctuations, and spaces. In Social Media comments, where many words very often to be misspelled, character n-grams are especially powerful at detecting patterns in such things and substantially less sparse than previously introduced word n-gram features.

Though many kinds of research in abusive speech detection domain have used word n-grams for feature extraction, character n-grams were considered by less number of researches. A hate speech identification study [7] in the German language has used character 2-gram and 3-gram features to identify hate speeches with 0.62 and 0.65 accuracies respectively.

In our study, we used up to 4-gram character levels separately and combined them to identify hate, neutral and offensive speeches. TABLE XII shows the feature groups that were considered in our study.

TABLE XII
CHARACTER N-GRAM FEATURE GROUPS

Group	Pre-processing	Feature set
C01	RN+SR+ST	2G
C02	RN+SR+ST	3G
C03	RN+SR+ST	4G
C04	RN+SR+ST	2G+3G
C05	RN+SR+ST	2G+3G+4G

2G- Bigram **3G**- Trigram **4G**- four-gram
ST- Stemming **SR**- Stop word Removing **RN** –Removing non-Sinhala symbols

4) Word skip-gram features:

These features are similar to word n-gram features and the difference is that skip-gram models extract features from a text by parsing some words from its current position. As an example, consider the sentence "I went to school in the morning", if the features are taken with 1-skip gram then the features will be as this: "I to", "went school", "to in", and "school the", "in morning". Here we considered 1-skip gram and 2-skip grams for bigrams and these were combined with normal bigram features and with unigram features. TABLE XIII shows the word skip-gram feature groups that were considered in our study.

TABLE XIII
WORD SKIP-GRAM FEATURE GROUPS

Group	Pre-processing	Feature set
S01	RN+SR+ST	1SG
S02	RN+SR+ST	2SG
S03	RN+SR+ST	1SG+UG+BG
S04	RN+SR+ST	2SG+UG+BG
S05	RN+SR+ST	1SG+UG+BG+TG

1SG- Skip-1-gram **2SG**- Skip-2-gram

These features were extracted by using SciKit learns count vectorizer and skip-gram features were obtained by customizing the count vectorizer.

G. Feature Vectorization.

The process of converting natural language text into numbers is called as vectorization in machine learning. Machines are not able to identify natural languages as a human does and, also statistical methods that are used for classification of given texts require input data in the form of numeric. Therefore feature vectorization plays a major role in natural language processing. In this study, we used two feature vectorization methods, Scikit learns *CountVectorizer* [14] and *TfidfTransformer* to vectorize extracted features.

1) CountVectorizer:

It is an implementation of SciKit learn's machine learning package and it is being used in a lot of ongoing researches all over the world. It provides a simple way to tokenize a collection of text documents and build a vocabulary of known words. When new documents are needed to be classified using that vocabulary, these documents are encoded.

2) TfidfVectorizer:

TfidfVectorizer is another implementation of feature vectorization. It consists of the term frequency-inverse document frequency (*TF-IDF*) concept as the basis. Typically *TF-IDF* is composed of two terms and calculates a weight. These two terms can be explained as follows.

- **TF: Term Frequency,** measures how frequently a term occurs in a document. Since every document is in different lengths, it is possible that a term would appear much more times in a lengthy document than shorter ones. Therefore, the term frequency is always divided by the document length (total number of words) for normalization

Tf(t,d) = frequency of term (t) in document (d) / total number of terms in (d)

- **IDF: Inverse Document Frequency,** measures the importance of a term. TF considers all terms are in an equal manner. But it is known that some terms have less importance while classifying sentences. Therefore, we need to weigh down the frequent terms while scaling up the rare ones, by computing the following:

IDF (t, D) = log_e (Total number of documents (D) / Number of documents with term (t) in it).

The *TF-IDF* score of a term (t) is calculated according to the following equation.

Tf-idf (t, d, D) = TF (t, d). IDF (t, D)

Since we use Scikit learn *CountVectorizer* to vectorize features, we did not use *TfidfVectorizer* directly, but *TfidfTransformer*, another implementation in Scikit learn package is used to convert CountVectorizer into TfidfVectorizer.

H. Classifiers and Machine Learning.

Machine learning is a necessary component of advanced text classification. The primary aim of machine learning is to allow computers to learn automatically without involving

humans.. In this study, we focused on supervised machine learning algorithms for the process of identifying offensive and neutral comments. Therefore, we have used three different machine learning algorithms: Support Vector Machine (SVM), Multinomial Naïve Bayes (MNB), and Random Forest tree as per the analysis we did in the literature review, most of the researches have shown best performance for these classifiers. In the following subsections, these algorithms are described separately. Here we used the algorithm implementations of Scikit learn.

1) Naïve Bayes (NB):

Naïve Bayes classifiers are simple probabilistic classifiers based on the application of Bayes theorem with strong independence assumptions between the features. In this study, Multinomial Naïve Bayes (MNB) classification technique has been used. It considers word frequency information in the document for analysis, where a document is considered to be an ordered sequence of words obtained from the vocabulary. The main difference between Multinomial NB and Bernoulli NB is that Bernoulli NB cares only the presence or absence of a particular feature (word) while Multinomial NB considers the occurrence (frequency count) of the features (words). Here we used the Scikit learn's implementation of MNB to classify a given sentence.

2) Support Vector Machine (SVM):

Support Vector Machine is a very famous classification method that is being used in the area of natural language processing. Unlike the Naïve Bayes algorithm, SVM is a non-probabilistic classifier algorithm. It is an efficient classification method when the feature vector is high dimensional. SVM separates data points using a hyperplane with the largest amount of margin. It constructs a hyperplane in multidimensional space to separate different classes. One of the main advantages of SVM is the robustness in general and effectiveness when the number of dimensions is greater than the number of samples. Here we used the Scikit learn's SVM implementation to achieve the classification goal.

3) Random Forest algorithm (RFDT):

Random forests also known as random decision forests are famous as an ensemble method that can be used to build classification models. It consists of several decision trees and based on the majority vote, a particular sentence or a document is classified. This algorithm differs from previous MNB and SVM since this decides by considering the majority vote from several trees. Since more trees are there, the random forest will not overfit the model and it is a reason for using the random forest in text mining researches as well as in hate speech detection.

I. Experiments

We started our experiments with generated lexicons and later we used feature extraction methods with classifiers to identify abusive speeches in Sinhala. Therefore, we selected a new 200 comments set randomly, that are annotated by previous annotators. This comment set also balanced among abusive and neutral classes.

1) Experiment 01:- Dictionary-based lexicon approach for abusive speech detection:

Despite the inability to find context-based opinion words in the dictionary, we made the first experiment for 200 annotated comments. Therefore, we selected 100 neural comments and 100 abusive comments which were not used to construct the corpus-based lexicon. Finally, we got 200 comments, 100 as neutral and 100 as abusive.

We used the dictionary that we built to classify the given comments. That dictionary contains 157 abusive words, together all of them we considered as an abusive word dictionary. After that, we built an algorithm to check whether any comment consists of an offensive word or not. If a sentence contains an offensive word that is listed in the dictionary, the algorithm identifies it as an offensive that is abusive, otherwise as a neutral comment.

In the first phase of the algorithm, we tokenized the sentences and sent them through the algorithm.

2) Experiment 02: Corpus-based lexicon approach for abusive speech detection:

We did the same experiment here with the same algorithm by changing the lexicon to a corpus-based lexicon.

3) Experiment 03:- word n-gram features for abusive speech identification:

Here we used feature groups that are listed in TABLE XI with three classifiers: MNB, SVM, and RFDT. To extract features, we use the comment corpus (TABLE I) which contains evenly distributed 2000 comments. Before the word n-gram extraction, we applied previously described pre-processing techniques. Then the extracted features were vectorized and weighted. Thereafter, each classifier was trained separately using those weighted features.

4) Experiment 04:- character n-gram features for abusive speech identification:

We trained each classifier by using character n-gram feature groups that are introduced in TABLE XII. In this experiment, characters with word boundaries were considered. Therefore *CountVectorizer*'s "*char_wb*" controlling value was used.

5) Experiment 05:- word skip-gram features for abusive speech identification

As the final experiment of our study, we used word skip-gram features as listed in TABLE XIII to train each classifier. Since we do not have prior knowledge on best features that can be used to identify Sinhala abusive speeches, we used five-word skip-gram feature groups here.

In experiments 03, 04 and 05, features were extracted by using the comment corpus that is listed in TABLE I. Then extracted features were applied to three classifiers and they were trained. Evenly distributed 200 test comment corpus was used to test every model that is trained through the experiments. Results obtained through these experiments are discussed in the next section.

IV. Experimental Results and Evaluation

As our research follows a quantitative approach, we can use statistical and mathematical techniques to evaluate what is done in the study. Therefore, we have used precision, recall, and accuracy as performance evaluation measurements. Since binary classification is considered, two confusion matrixes that are with two class labels are constructed in this study. The TABLE XIV gives the structure of the confusion matrix and based on that matrix, precision, recall, and accuracy are obtained.

Here we considered the binary classification's confusion matrix for describing the precision, recall, and accuracy since it is the standard way to demonstrate confusion.

TABLE XIV
CONFUSION MATRIX STRUCTURE

		Predicted Class	
		Offensive	**Neutral**
True Class	**Offensive**	True Positive (TP)	False Negative (FN)
	Neutral	False Positive (FP)	True Negative (TN)

At each experiment, a unique confusion matrix is built and based on its true positive, false positive, false negative and true negative values, accuracy, precision, and recall are calculated.

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

Accuracy of a model is calculated according to the above equation and it is known as the measurement of the fraction of correct predictions. But it has some very common problems. Major problem is that accuracy is not a good measure when the data set is unbalanced through the classes. But our data set is balanced through all two classes, accuracy can be used as a performance evaluation method.

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

Precision is the fraction of relevant instances among the retrieved instances and here precision for offensive classification is known as the fraction of actual offensives among predicted offensives.

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

The recall is the fraction of the relevant instances that are successfully retrieved. Since it does not contain true negatives, it is a good measurement in our study. It measures the predicted actual instances among all actual instances for a particular class label. Therefore it can be taken as the main measurement here.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

F1-Score is the harmonic mean of precision and recall. It ensures that there will be no overly rely on either precision or recall. Therefore F1-Score is considered as another performance measurement in this study.

Results that are obtained through every experiment is discussed below.

A. Experiment 01 Results

As experiment 01 described, we tested the dictionary-based lexicon with 200 evenly distributed comment corpus. Results obtained through the experiment are listed in TABLE XV and its confusion matrix is listed in TABLE XVI.

TABLE XV
PERFORMANCE MEASUREMENTS OF THE DICTIONARY-BASED APPROACH

	Precision	*Recall*	*F1-score*	*Accuracy*
Abusive	0.83	0.10	0.18	0.54
Neutral	0.52	0.98	0.68	

Though the model has 0.54 accuracy, the recall of abusive is 0.10. As described previously, recall plays a major role among other performance measurements, and having high recall value makes trust in the model. Therefore, with this less recall value of the offensive class, we cannot conclude that a cross-lingual dictionary is a good approach for Sinhala abusive speech detection.

TABLE XVI
CONFUSION MATRIX OF EXPERIMENT01

		Predicted Class	
		Offensive	**Neutral**
True Class	**Offensive**	10	90
	Neutral	2	98

Translated words contain the definitions of words than their practical forms. That is why these measurements gave low values. Therefore, we can conclude that abusive speech identification cannot be done efficiently by translating one language's hate or offensive words into another language.

B. Experiment 02 Results

Since experiment 01 is not efficient and sufficient in Sinhala abusive speech identification, the constructed corpus-based lexicon was tested with the same 200 comment corpus. Results obtained through the process are listed in following TABLE XVII and the confusion matrix is listed in TABLE XVIII.

TABLE XVII
PERFORMANCE MEASUREMENTS OF THE CORPUS-BASED APPROACH

	Precision	*Recall*	*F1-score*	*Accuracy*
Abusive	0.98	0.83	0.90	0.905
Neutral	0.85	0.98	0.91	

The corpus-based lexicon gives 0.905 accuracies in the process of abusive speech identification and this accuracy is greater than the dictionary-based lexicon approach. Therefore, it is clear that corpus-based lexicons are effective than the

dictionary-based translated lexicons in the process of Sinhala hate speech detection.

TABLE XVIII
CONFUSION MATRIX OF EXPERIMENT 02

		Predicted Class	
		Offensive	**Neutral**
True Class	**Offensive**	83	17
	Neutral	2	98

Though the corpus-based approach gave 90.5% accuracy, it is not sufficient in the process of abusive language detection, since the lexicon approaches depend on already classified words. Once these approaches meet strange abusive words, they will not be able to identify them as abusive. Therefore we need to consider machine learning approaches.

C. Experiment 03 Results

In this experiment, we trained two different classifiers with five different feature groups. Therefore, we obtained ten models to test. Since we have limited space here, we considered only one confusion matrix. Performance measurements for each feature group with each classifier are listed in TABLE XIX.

As TABLE XIX shows, word n-gram feature with MNB gives the highest accuracy for the W01 feature group which contains unigram features. Also feature group W04 and W05 have shown better accuracies and recall values. Though W02 and W03 contain more information than W01, they show low accuracies in Sinhala hate speech detection.

Although W01 shows the highest accuracy for RFDT as well as MNB, W04 gave the highest accuracy for the SVM classifier. Therefore, it is clear that the same feature groups may not be suitable when classifiers are changed. Among three classifiers, MNB has given the highest accuracy with the highest f1-score for the W01 feature group.

TABLE XIX
PERFORMANCE MEASUREMENTS OF WORD N-GRAM FEATURES

	Class	Feature Groups				
		W01	**W02**	**W03**	**W04**	**W05**
MNB Classifier						
Precision	**A**	0.94	0.63	0.53	0.93	0.92
	N	0.95	0.84	1.00	0.93	0.94
Recall	**A**	0.95	0.91	1.00	0.93	0.94
	N	0.94	0.47	0.11	0.93	0.92
F1-Score	**A**	0.95	0.75	0.69	0.93	0.93
	N	0.94	0.60	0.20	0.93	0.93
Accuracy		**0.945**	**0.69**	**0.555**	**0.93**	**0.93**
SVM Classifier						
Precision	**A**	1.00	0.95	1.00	0.96	0.95
	N	0.79	0.55	0.51	0.86	0.86
Recall	**A**	0.74	0.18	0.05	0.85	0.84
	N	1.00	0.99	1.00	0.96	0.96
F1-Score	**A**	0.85	0.30	0.10	0.90	0.89
	N	0.88	0.70	0.68	0.91	0.91
Accuracy		**0.87**	**0.585**	**0.525**	**0.905**	**0.90**
RFDT Classifier						
Precision	**A**	1.00	0.59	0.52	0.88	0.83
	N	0.85	0.89	1.00	0.88	0.85
Recall	**A**	0.83	0.96	1.00	0.88	0.86
	N	1.00	0.34	0.06	0.88	0.82
F1-Score	**A**	0.91	0.73	0.68	0.88	0.84
	N	0.92	0.49	0.11	0.88	0.84
Accuracy		**0.915**	**0.65**	**0.53**	**0.88**	**0.84**

A-Abusive **N-** Neutral

Since all the confusion matrices cannot be listed, in TABLE XX we presented MNB's WO1 feature group confusion matrix just because it gave the highest accuracies among other feature groups and models.

TABLE XX
CONFUSION MATRIX OF EXPERIMENT 03'S MNB WITH W01 FEATURE GROUP

		Predicted Class	
		Offensive	**Neutral**
True Class	**Offensive**	95	5
	Neutral	6	94

D. Experiment 04 Results

According to the performance measurements, we can conclude that all the feature groups that we considered in character n-gram are good at Sinhala abusive comments identification process as it gives higher Precision, Recall, and F1-Score values regardless of the model that we used to train.

When we compare the precision, recall, F1-score, and accuracy values of the MNB, SVM, RFDT classifiers from the TABLE XXI, it is clear that the MNB has outperformed the other two models for the character n-gram model.

TABLE XXI
PERFORMANCE MEASUREMENTS OF CHARACTER N-GRAM FEATURES

	Class	Feature Groups				
		C01	**C02**	**C03**	**C04**	**C05**
MNB Classifier						
Precision	**A**	0.97	0.99	0.98	0.99	0.98
	N	0.95	0.93	0.95	0.94	0.94
Recall	**A**	0.95	0.93	0.95	0.94	0.94
	N	0.97	0.99	0.98	0.99	0.98
F1-Score	**A**	0.96	0.96	0.96	0.96	0.96
	N	0.96	0.96	0.97	0.97	0.96
Accuracy		**0.96**	**0.96**	**0.965**	**0.965**	**0.96**
SVM Classifier						
Precision	**A**	0.99	0.98	0.99	0.96	0.95
	N	0.76	0.92	0.88	0.95	0.97
Recall	**A**	0.69	0.92	0.86	0.95	0.97
	N	0.99	0.98	0.99	0.96	0.95
F1-Score	**A**	0.81	0.95	0.92	0.95	0.96
	N	0.86	0.95	0.93	0.96	0.96
Accuracy		**0.84**	**0.95**	**0.925**	**0.955**	**0.96**
RFDT Classifier						
Precision	**A**	0.94	0.99	0.98	0.98	0.99
	N	0.89	0.90	0.88	0.91	0.89
Recall	**A**	0.88	0.89	0.87	0.90	0.88
	N	0.94	0.99	0.98	0.98	0.99
F1-Score	**A**	0.91	0.94	0.92	0.94	0.93
	N	0.91	0.94	0.93	0.94	0.94
Accuracy		**0.91**	**0.94**	**0.925**	**0.94**	**0.935**

A-Abusive **N-** Neutral

Further, all three models have given good performances for C03, C04 and C05 feature groups. The confusion matrix for MNB with Character four-gram is listed in TABLE XXII.

TABLE XXII
CONFUSION MATRIX OF EXPERIMENT 04'S MNB WITH CO3 FEATURE GROUP

		Predicted Class	
		Offensive	**Neutral**
True Class	**Offensive**	95	5
	Neutral	2	98

E. Experiment 05 Results

In this section, we discuss the results obtained through the models that were trained by word skip-gram features. As previously described in two subsections, here we also tested ten models with a test comment corpus which has 200 evenly distributed comments. TABLE XXIII shows the performance measurements that were obtained.

TABLE XXIII
PERFORMANCE MEASUREMENTS OF WORD SKIP-GRAM FEATURES

	Class	**Feature Groups**				
		S01	**S02**	**S03**	**S04**	**S05**
MNB Classifier						
Precision	**A**	0.59	0.54	0.95	0.95	0.95
	N	0.85	0.82	0.97	0.96	0.96
Recall	**A**	0.94	0.96	0.97	0.96	0.96
	N	0.35	0.18	0.95	0.95	0.95
F1-Score	**A**	0.73	0.69	0.96	0.96	0.96
	N	0.60	0.30	0.96	0.95	0.95
Accuracy		**0.645**	**0.57**	**0.96**	**0.955**	**0.955**
SVM Classifier						
Precision	**A**	0.52	0.54	0.84	0.99	0.99
	N	0.80	0.79	0.95	0.88	0.88
Recall	**A**	0.98	0.95	0.96	0.86	0.86
	N	0.08	0.19	0.82	0.99	0.99
F1-Score	**A**	0.68	0.69	0.90	0.92	0.92
	N	0.15	0.31	0.88	0.93	0.93
Accuracy		**0.53**	**0.57**	**0.89**	**0.925**	**0.925**
RFDT Classifier						
Precision	**A**	0.51	0.52	0.89	0.89	0.86
	N	0.78	1.00	0.90	0.86	0.89
Recall	**A**	0.98	1.00	0.90	0.86	0.89
	N	0.07	0.07	0.89	0.89	0.86
F1-Score	**A**	0.68	0.90	0.87	0.88	0.67
	N	0.13	0.89	0.88	0.87	0.13
Accuracy		**0.525**	**0.535**	**0.895**	**0.875**	**0.875**

A-Abusive **N-** Neutral

Though all skip-gram feature groups show more than 50% accuracy, recall for Neutral comments are less than 40% in both S01 and S02 feature groups. According to the results, feature groups S01 and S02 with MNB tend to classify neutral comments as abusive comments. From all measurements, it shows that feature group S03 is the best to identify offensive speeches in Sinhala with Multinomial Naïve Bayes classifier.

By focusing on the S01 and S02 feature groups, we can conclude that S01 and S02 are not good to detect Sinhala offensive speeches with SVM since their recall measurement of Neutral speech is less than 20%. These two models tend to classify neutral comments wrongly as offensive comments. Both S04 and S05 feature groups show the best accuracy as 92.5% with the SVM classifier.

Random forest classifier performs worst with S01 and S02 feature groups since they have low recall values for neutral speeches. The highest average f1-score and accuracy has obtained for the S03 feature group. Therefore S03 ssssis the best word skip-gram feature that can be used to identify Sinhala offensive speeches.

From these experiments, it is clear that character four-gram (C03), feature group C04 and feature group C05 are the best features for Sinhala abusive speech detection since it gives high accuracies and f1-scores for all classification models. Therefore, we can conclude that these features can be taken in the process of Sinhala abusive speech identification effectively though the classification models are changed.

As the test results show, we can conclude that MNB performs well with each feature group than the other two classifiers.

V. CONCLUSION AND FUTURE WORKS

In this study, we have identified the best features and classification models that can be used in Sinhala abusive speech identification with machine learning. Further, we built two new lexicons which contain Sinhala abusive words. Abusive speech detection was done by using these two lexicons and we observed that corpus-based lexicons are the best approaches in Sinhala abusive speech detection process.

As per the results we obtained through the experiments, character four-gram (C03), C04 and C05 features have outperformed all other feature types that were considered in this study. Since many abusive speeches are published with spelling mistakes and substituted with similar characters, we can conclude that character n-gram features perform well in abusive Sinhala speech detection.

Regardless of the feature extraction method, when we focus on the performance measurement values of TABLE XIX, TABLE XXI, and TABLE XXIII, it is clear that the RFDT classifier has the least accuracy values for the Sinhala abusive comments detection.

Since we considered only supervised methods, it is an open area to apply unsupervised learning techniques to identify Sinhala hate speeches.

As the features of this research, we only considered macro features such as word n-grams and character n-grams. Therefore, micro features such as patterns of the speeches using POS tags, presence and frequencies of punctuation marks and word counts in a sentence can be used to identify Sinhala hate speeches.

Here we only considered abusive speeches which are published using Sinhala Unicode, because we did not have enough comments to consider Singlish (Sinhala words are written in English) speeches. Therefore, Singlish is also an open area to be considered in hate speech detection domain. Working with these transliterated forms (Singlish words) may be a challenging task as Singlish comments contain both pure English and Sinhala comments. It will make the stemming and other pre-processing techniques as well as feature vectorization techniques too complex.

Further, comparison between stop word removing and without removing stop words should also be investigated, since Sinhala is a rich morphological language.

s

ACKNOWLEDGMENT

We would like to thank the people who support us to make research data set available.

REFERENCES

[1] Medagoda, N. (2017). Framework for Sentiment Classification for Morphologically Rich Languages: A Case Study for Sinhala. http://aut.researchgateway.ac.nz/handle/10292/10544

[2] "Liking violence: A study of hate speech on Facebook in Sri Lanka" [Online] Available: https://www.cpalanka.org/wp-content/uploads/2014/09/Hate-Speech-Final.pdf

[3] Gitari, N. D. *et al.* (2015) 'A lexicon-based approach for hate speech detection', *International Journal of Multimedia and Ubiquitous Engineering*, 10(4), pp. 215–230. doi: 10.14257/ijmue.2015.10.4.21.

[4] "The Hate Directory" [Online] Available: http://www.hatedirectory.com/

[5] Riloff, E. and Wiebe, J. (2003) 'Learning extraction patterns for subjective expressions', *Proceedings of the 2003 conference on Empirical methods in natural language processing* -, 10, pp. 105–112. doi: 10.3115/1119355.1119369.

[6] Cambria, E. *et al.* (2010) 'SenticNet : A Publicly Available Semantic Resource for Opinion Mining', *Artificial Intelligence*, pp. 14–18. doi: 10.1038/leu.2012.122.

[7] Köffer, S. *et al.* (2018) 'Discussing the Value of Automatic Hate Speech Detection in Online Debates', *Multikonferenz Wirtschaftsinformatik*, (October), pp. 83–94. doi: 10.1111/j.1365-2923.2008.03277.x.

[8] Dias, D. S., Welikala, M. D. and Dias, N. G. J. (2019) 'Identifying Racist Social Media Comments in Sinhala Language Using Text Analytics Models with Machine Learning', (September), pp. 1–6. doi: 10.1109/icter.2018.8615492.

[9] Gallege, S. (2004) 'Analysis of Sinhala Using Natural Language Processing Techniques'.

[10] "Google Bad Word List" [Online] Available: https://www.freewebheaders.com/full-list-of-bad-words-banned-by-google/

[11] Language Technology Research Laboratory [Online] Available: http://ltrl.ucsc.lk/

[12] Welgama, V. (2011) 'Evaluation of a shallow stemming algorithm for sinhala', *Language*, (35), p. 2009.

[13] Malmasi, S. and Zampieri, M. (2017) 'Detecting Hate Speech in Social Media', pp. 467–472. doi: 10.26615/978-954-452-049-6_062.

[14] CountVectorizer Documentation [Online] Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[15] Central Intelligence Agency - World Fact Book (https://www.cia.gov/library/publications/the-worldfactbook/geos/ce.html)

[16] Mubarak, H., Darwish, K. and Magdy, W. (2017) 'Abusive Language Detection on Arabic Social Media', Proceedings of the First Workshop on Abusive Language Online, pp. 52–56. Available at: https://drive.google.com/open?id=0B4xDAGbwZJjQZkRzLTRZcTQ0ZkE.

[17] Nandasara, S. T. (2015) 'Bridging the Digital Divide in Sri Lanka : Some Challenges and Opportunities in using Sinhala in ICT Bridging the Digital Divide in Sri Lanka : Some Challenges and Opportunities in using Sinhala in ICT', (May). doi: 10.4038/icter.v8i1.7162.

[18] Davidson, T. *et al.* (2017) 'Automated Hate Speech Detection and the Problem of Offensive Language', (Icwsm), pp. 512–515. doi: 10.1561/1500000001.

[19] Alfina, I. *et al.* (2018) 'Hate speech detection in the Indonesian language: A dataset and preliminary study', *2017 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2017*, 2018–Janua(October), pp. 233–237. doi: 10.1109/ICACSIS.2017.8355039.