SENTIMENT ANALYSIS OF SINHALA NEWS COMMENTS

Isuru Udara Liyanage

(168243P)

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree
Master of Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

May 2018

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

-- --

Isuru Udara Liyanage Date

The above candidate has carried out research for the Masters thesis/ Dissertation under my supervision.

-- --

Dr. Surangika Ranathunga Date

ABSTRACT

Mining sentiment values from unstructured text uncovers interesting patterns that can be effectively used for many applications. One interesting yet poorly explored area is online news comment analysis, in particular for Sinhala language. Despite the uptrend in online Sinhala news articles and related comments, no efficient method exists for analyzing and identifying the public sentiment associated with them. In this research our effort is to classify online Sinhala news comments according to its sentiment orientation.

Most of the sentiment analysis research is done for English language. As for Sinhala, only one research can be found for classification of Sinhala news comments according to its sentiment values. Since it is an initial attempt it lacks the use of advanced text analysis methods and localization, and hence can be improved in many ways.

In this research we build a complete Sinhala sentiment analysis system, from data collection to sentiment classification. First we gather a dataset by crawling through a popular online news site. Complied dataset contains news items and related comments. Sufficient amount of comments are annotated according to its sentiment values. Finally sentiment analysis is carried out to identify sentiment values associated with each comment.

This research provides many valuable outputs to the research community, sentiment analysis for Sinhala text. Dataset, the labeled data set in particular, can be used for future Sinhala text analysis research. Finally direction and a baseline will be set for future research on sentiment analysis for Sinhala text.

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to my thesis advisor Dr. Surangika Ranathunga for her continuous guidance throughout the research work. She steered me in the right direction from the beginning by providing valuable insights, resources and supervision. Without her guidance and motivation this wouldn’t have been a success.

I am grateful to Dr. Charith Chitraranjan, Dr. Shehan Perera, Dr. Malaka Walpola and Dr. Indika Perera for their motivation and guidance for making this a success.

I would also like to thank my colleges at DirectFN for their support and the understanding during the MSc duration.

Finally I express my heartfelt gratitude for my parents for their love and support throughout my life.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
POS	Part Of Speech
SVM	Support Vector Machines
SO	Sentiment Orientation
PMI	Pointwise Mutual Information
WSD	Word Sense Disambiguation
FNE	Focused Named Entities

CHAPTER 1: INTRODUCTION

1.1 News and Comments

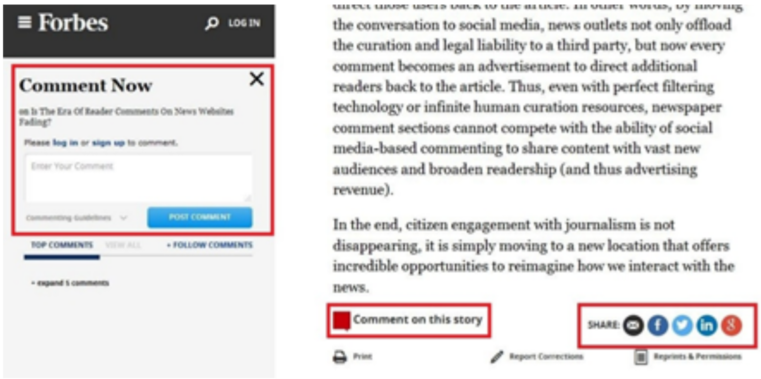
Online news sites and interactive press is one of the very active categories among the Internet traffic creators. These news sites publish news on a wide variety of categories including sports, politics, economy, society, crime, culture and much more. Ease of access and availability has made them more attractive compared with traditional news media.

With the political and social background, expressing one's opinion in public is not always very straightforward. With the fear of attracting unnecessary attention people inhibit themselves from commenting on social events, news and articles. But because of the anonymity provided, online news articles has blown this barrier apart. Users do not have to hide their opinions anymore. If a reader is uncomfortable with commenting as himself he can comment anonymously, shielding his privacy. This provides users with the freedom to speak, and freedom to express. Interactivity and facilitation for the social discourse are couple of key qualities associated with online articles.

A news item may consist of many widgets a user can interact with. Comment section is the most common widget available in most of the news articles. A comment section may include a text area where users can express their opinion in free hand, a selection of choices about the particular news item or a simply a binary selection weather the news item is good or bad.

Figure 1.1 extracted from Forbes[1] contains two interactive widgets. One widget allows user to easily share the article in one of the social networking sites. Other widget is a comment widget where user can comment on the article. To publish a comment, user needs to register first. Comment publication goes through a review process.

[1] https://www.forbes.com



Comment Now

on Is The Era Of Reader Comments On News Websites Fading?

Please log in or sign up to comment.

Enter Your Comment

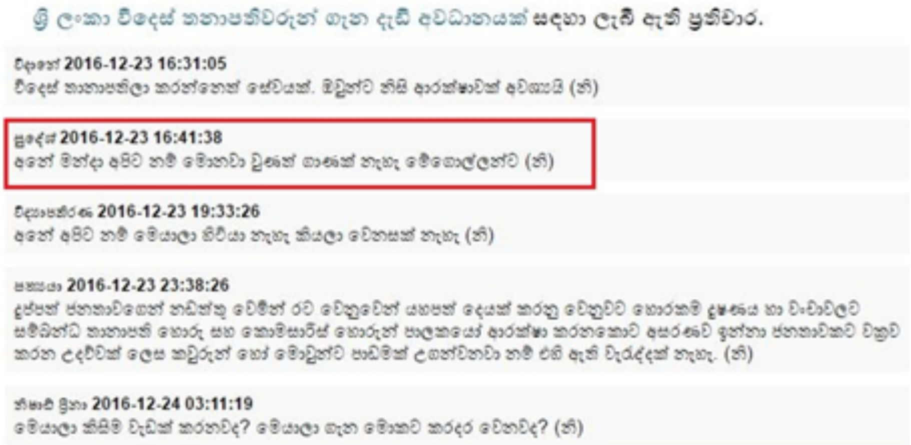
direct those users back to the article. In other words, by moving the conversation to social media, news outlets not only offload the curation and legal liability to a third party, but now every comment becomes an advertisement to direct additional readers back to the article. Thus, even with perfect filtering technology or infinite human curation resources, newspaper comment sections cannot compete with the ability of social media-based commenting to share content with vast new audiences and broaden readership (and thus advertising revenue).

In the end, citizen engagement with journalism is not disappearing, it is simply moving to a new location that offers incredible opportunities to reimagine how we interact with the news.

Comment on this story

Figure 1.1: Online news comments

Figure 1.2 contains a news article comment section extracted from Lankadeepa[2] online news site. A single comment consists of user Id, date of the comment and the comment itself. All the comments in this site are published in Sinhala media. There are no hate comments because of the manual review process. We can see that the highlighted comment in figure 1.2 bares a negative sentiment.



ශ්‍රී ලංකා විදෙස් තානාපතිවරුන් ගැන දැඩි අවධානයක් සඳහා ලැබී ඇති ප්‍රතිචාර.

විදානේ 2016-12-23 16:31:05
විදෙස් තානාපතිලා කරන්නෙත් සේවයක්. ඔවුන්ට නිසි ආරක්ෂාවක් අවශ්‍යයි (නි)

සුදේශ් 2016-12-23 16:41:38
අනේ මන්දා අපිට නම් මොනවා වුණත් ගාණක් නැහැ මේගොල්ලන්ට (නි)

විදානපතිරණ 2016-12-23 19:33:26
අනේ අපිට නම් මෙයාලා හිටියා නැහැ කියලා වෙනසක් නැහැ (නි)

ජනසය 2016-12-23 23:38:26
දූප්පත් ජනතාවගෙන් නඩත්තු වෙමින් රට වෙනුවෙන් යහපත් දෙයක් කරනු වෙනුවට හොරකම් දූෂණය හා වංචාවලට සම්බන්ධ තානාපති හොරු සහ කොමිසාරිස් හොරුන් පාලකයෝ ආරක්ෂා කරනකොට අසරණව ඉන්නා ජනතාවකට විකුට කරන උදව්වක් ලෙස කවුරුන් හෝ මොවුන්ට පාඩමක් උගන්වනවා නම් එහි ඇති වැරැද්දක් නැහැ. (නි)

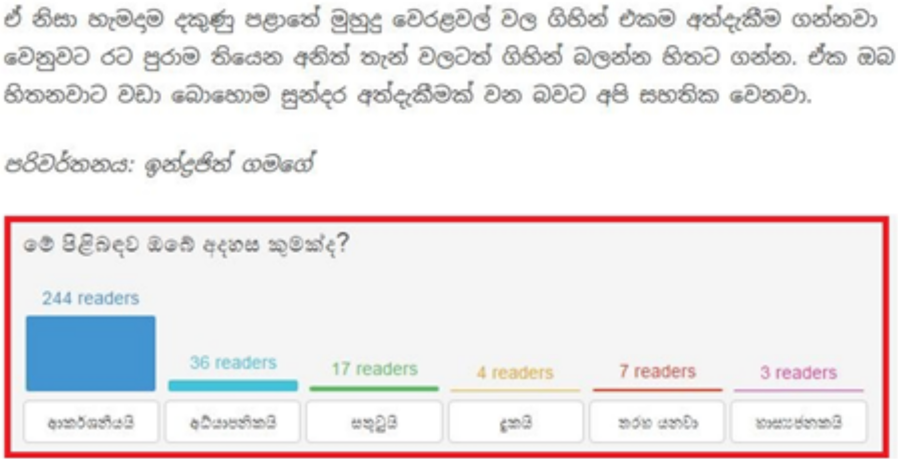
නිශාන් ප්‍රනා 2016-12-24 03:11:19
මෙයාලා කිසිම වැඩක් කරනවද? මෙයාලා ගැන මොකට කරදර වෙනවද? (නි)

Figure 1.2: News Comments

Figure 1.3 taken from roar.lk[3] contains another type of interactive widget present in news sites. Instead of commenting, user can pick an emotion from predefined set of emotions.

[2] http://www.lankadeepa.lk/

[3] https://roar.media/sinhala



ඒ නිසා හැමදම දකුණු පළාතේ මුහුදු වෙරළවල් වල ගිහින් එකම අත්දැකීම ගන්නවා වෙනුවට රට පුරාම තියෙන අනිත් තැන් වලටත් ගිහින් බලන්න හිතට ගන්න. ඒක ඔබ හිතනවාට වඩා බොහොම සුන්දර අත්දැකීමක් වන බවට අපි සහතික වෙනවා.

පරිවර්තනය: ඉන්ද්‍රජිත් ගමගේ

මේ පිළිබඳව ඔබේ අදහස කුමක්ද?

Figure 1.3: Interactive widgets

1.2 Sentiment Analysis

Sentiment analysis which is also known as opinion mining [3] is the computational study of opinions, sentiments, evaluations, attitudes, appraisal, affects, views, and emotions that are expressed in text. These includes reviews blogs discussions, news, comments, feedbacks and many more.

Basic task of sentiment analysis is to classify a given text unit as positive, negative or neutral. This is known as sentiment classification. Sentiment classification can be viewed in several granularity levels. They are sentence level, document level and feature/entity level. While sentence level classification considers each sentence separately, document level considers text unit as a whole. Feature level classification, which is also known as aspect level classification tries to classify the sentient of each of the focuses present in the document.

1.3 Problem and Motivation

The information extracted from user generated comments can be harnessed and converted into knowledge to help in the decision making process. Therefore analysis of the given data is important for many categories of people.

- General public
- Government
- Private organizations
- Police

Despite the many benefits provided by news comments analysis, it has become increasingly difficult to deal with the sheer amount of news published in the Internet. It is almost impossible to manually analyze all the news comments. Therefore an automated method is required to process and analyze the information conveyed in online news comments.

However, research on the above direction exists for English and for few other languages like Japanese [17], Chinese [30] [31], Russian [32] [33] and Hindi [34] [35]. Only one research attempt can be found for Sinhala sentiment classification [19]. This was only carried out as an introduction to Sinhala sentiment analysis and lacks use of advanced text analysis methods. Hence performance can be improved greatly by introducing state of the art algorithms and methods. Therefore its right time to develop a sufficient set of Sinhala sentiment analysis tools.

Sinhala language lacks general text analysis resources needed for sentiment analysis, such as datasets and lexicons. Even though various sentiment analysis tools exists for English language, they can't be readily used for Sinhala sentiment analysis due to major differences between two languages. Sinhala sentiment analysis will introduce resources to the text analysis community which will help the development of the language.

1.4 Objective

Overall objective of the research is to do document-level sentiment analysis on Sinhala news comments. This objective can be divided into few sub-objectives as below.

- Lexicon generation
- Collection of news articles
- Selection of suitable sentiment analysis techniques/algorithms
- Adaption of selected techniques for Sinhala language
- Preparation of test dataset
- Carrying out tests and interpreting results

CHAPTER 2: LITERATURE REVIEW

In this section we will formulate the problem and look into past work to investigate more on sentiment analysis.

2.1 Sentiment Analysis

There are three essential parts in describing an opinion [8]. They are opinion target, opinion holders and opinion words. Apart from that we have user ID and time of the opinion. For example, consider the following product review from eBay. Table 2.1 shows the breakdown of the below opinion.

ID: user123   Date: 11/12/2016
Today I bought a PlayStation 4. It's a great gaming console. It has some cool games. It's much better than the Xbox one, which is too bulky. But my friends think gaming consoles are waste of money. They think it has less features when compared with a personal computer.

Table 2.1 Sentiment Breakdown

Sentence part	Description	From Example
Opinion target	Entities and their features/ aspects	PlayStation 4, gaming console, games, Xbox one, personal computer
Opinion holder	Person who hold the opinion	I, friends
Sentiment	Positive or negative	great, cool, better, too bulky, waste, less features
Time	When opinion was expressed	Date

Now let's consider a Sinhala news article comment extracted from online news site www.lankadeepa.com. Table 2.2 shows the breakdown of this comment.

ශ්‍රියානි ජයසිංහ 2017-01-22 16:39:18

හරිම ආඩම්බරයි, ඔබ වගේ නිලධාරීන් ගැන. හිතන්ටත් අමාරුයි ලංකාවේ මෙහෙම මිනිස්සු ඉන්නවා කියල

Table 2.2 Sinhala news comment breakdown

Sentence part	From Example
Opinion target	නිලධාරීන්, ලංකාවේ මිනිස්සු
Opinion holder	ශ්‍රියානි (extracted from metadata)
Sentiment	ආඩම්බරයි, හිතන්ටත් අමාරුයි
Time	2017-01-22 16:39:18 (extracted from metadata)

Opinion target is the entity or focus of the opinion sentence. Entity can be a product, person, event, organization or just a general topic. Entity may consist of sub components and attributes. Therefore we can associate opinions of subcomponents and attributes with the main entity. In other times it may be advantageous to consider them separately. Term aspect/ feature is used to identify entity, components and its attributes throughout this discussion. In Figure 2.1, we can identify Galaxy S7 as the main entity, which has Touch Screen and Battery as subcomponents. Battery has battery life and size as the attributes.

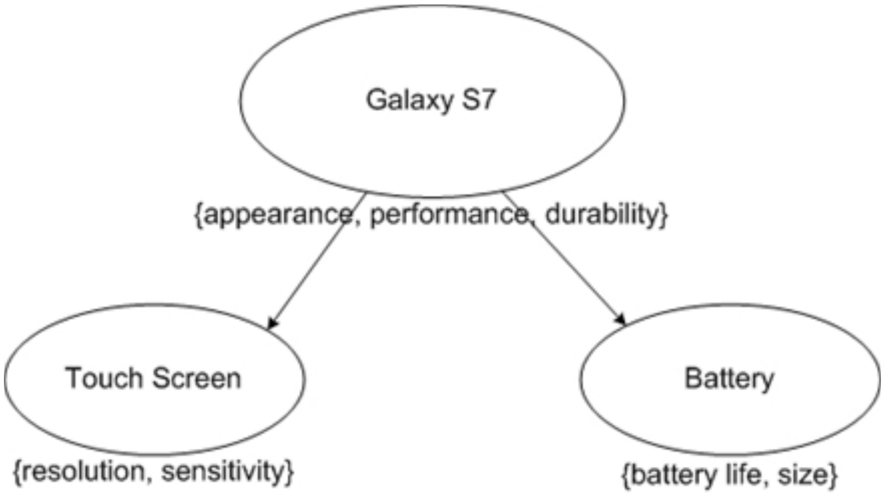

Galaxy S7
{appearance, performance, durability}
Touch Screen
Battery
{resolution, sensitivity}
{battery life, size}

Figure 2.1: Components and Features [8]

An opinion can be analyzed in few different granularity levels. Each granularity level will give insights into different features embedded in opinion. Therefore selecting a suitable granularity level before analyzing is an important task.

Document Level

Entire opinion is given a sentiment value. Individual sentences may contain different opinions on similar or closely related opinion target. Sentence level classification can be aggregated into document level.

E.g.: if we consider the above opinion, overall it conveys a positive sentiment towards the target (PlayStation)

Sentence Level

Opinion may consist of one or more sentences. Each sentence of the opinion is analyzed separately and classified whether positive or negative.

E.g.: *It's a great gaming console.* – Positive

But my friends think gaming consoles are waste of money. – Negative

Entity and feature/aspect/focus level

Instead of classifying each sentence or document as a whole, features of the entities are identified and classified. A features is an attribute of the entity or focus considered in the opinion.

E.g.: *It has some cool games.* Here feature is games and main entity is PlayStation. Therefore opinion is expressed on games of PlayStation console.

One way of categorizing opinions is by their relativeness [8]. That is weather an opinion is relative or absolute. Opinion about an absolute quality of an item is known as regular opinion. An opinion expressed comparing two or more item is a comparative opinion.

Regular opinion
Opinion consists of absolute quality of an item. Does not compare item with other items or entities.
E.g.: *It's a great gaming console.*

Comparative opinion
Relative opinions that are expressed comparing with another entity of similar nature.
E.g.: *It's much better than the Xbox one, which is too bulky.*
Here opinion is expressed comparing PlayStation with Xbox.

Here we will focus on regular opinions and continue the discussion.

2.1.1 Formal Definition

Now we have a basic understanding of the structure of a sentiment expression. We can formally define a sentiment as a quintuple as below [8].

$$(e_j, a_{jk}, h_i, so_{ijkl}, t_l) \quad \text{- (1)}$$

Where
e_j – target entity
a_{jk} – aspect/ feature of target e_j
h_i – opinion holder
so_{ijkl} – sentiment value. This could positive, negative, neutral or a more granular rating
t_l – time when the opinion is expressed
(e_j, a_{jk}) is the opinion target.

From our example we can extract few quintuples as below.

$$(PlayStation, general, positive, user123, 11/12/2016)$$

$$(PlayStation, games, positive, user123, 11/12/2016)$$

Since document comments are unstructured in nature they are difficult to analyze. Formal definition brings structure to our problem and analysis becomes simpler. With the formal definition, we can identify the main tasks in sentiment analysis as below.

- Named entity extraction (e_j)
- Information extraction (a_{jk}, h_i, t_l)
- Sentiment Identification (so_{ijkl})

2.2 Sentiment Classification

Sentiment classification of documents is to apply a label to the whole document to say whether it's positive, negative or neutral. This is a text classification problem. Mostly in topic based text classification problems we focus on the topic words. Here since we are focusing on sentiment based classification, focus is on the sentiment words. These sentiment words express desired or undesired quality of the entity being considered. If we look into quintuple representation, goal here is to find appropriate '*so'* value. Sentiment classification can be categorized into supervised and unsupervised approaches.

2.2.1 Unsupervised

Unsupervised classification does not require a training dataset. Furthermore since sentiment classification tends to be highly application dependent, models developed for one domain might not work well on another. Classical unsupervised sentiment analysis paper was published by Turney [4] which was the baseline for many research to follow.

Turney [4] introduced an unsupervised classification technique that classifies reviews as recommended (thumbs up) or not recommended (thumbs down). Proposed solution contained three steps. First step was to extract two word phrases from reviews which conforms to a given pattern. This was done using Part Of Speech (POS) tagging. Table 2.3 lists the used patterns. If we take third pattern of Table 2.3 as an example, it indicates to extract all the phrases which the first (JJ) and second (JJ) words are adjectives and third word is not a noun (NN).

- JJ - Adjective
- NN - Noun, singular or mass
- NNS - Noun, plural
- RB - Adverb
- RBR - Adverb, comparative
- RBS - Adverb, superlative
- VB – Verb

Table 2.3 Patterns of tags for extracting two-word phrases from reviews.

First Word	Second Word	Third Word (not extracted)
JJ	NN or NNS	anything
RB, RBR, RBS	JJ	not NN nor NNS
JJ	JJ	not NN nor NNS
NN, NNS	JJ	not NN nor NNS
RB, RBR, RBS	VB, VBD, VBN, VBG	anything

Step two was to estimate Sentiment Orientation (SO) of the extracted phrases using Pointwise Mutual Information (PMI). PMI between two words is given by Equation 2.1. Here *p (word$_1$ & word$_2$)* refers to the probability that word$_1$ and word$_2$ co-occur. Therefore PMI measures the degree of statistical dependence between two words. Sentiment Orientation is calculated using PMI and two reference words. 'Excellent' and 'Poor' was selected as reference words as in 5 star rating scale they refer to the extreme cases of positive and negative cases.

$$PMI(word1, word2) = \log_2 \left[\frac{p(word1 \wedge word2)}{p(word1).p(word2)} \right]$$

Equation 2.1 Pointwise Mutual Information

$$SO(phrase) = PMI(phrase, 'Excellent') - PMI(phrase, 'Poor')$$

Equation 2.2 Sentiment Orientation

Step 3 calculates the average Sentiment Orientation of all phrases in review. Algorithm classifies a review as positive if its average SO is positive and negative otherwise. Evaluation results shows classifier accuracy in between 65% to 85% in various application domains.

Lin et al [36] introduced a fully unsupervised method (joint sentiment/topic- JST) based on probabilistic modeling. Model employed both sentiments and topics to classify sentiment orientation of a document. JST was extended by adding a sentiment layer to the state of the art topic classification model, Latent Dirichlet Allocation (LDA). Accuracy was further improved by using various sources of prior information. Preprocessed movie review dataset was used for the evaluation. Evaluation results shows accuracy values up to 85% which was very close to supervised approaches. It is was identified that use of prior information such as Mutual Information increased accuracy values significantly (up to 15%).

Turney et al [37] introduced another unsupervised training method which used Pointwise Mutual Information to find out sentiment orientation. PMI was calculated using intuitively chosen seven opposing word pairs (seven positive and seven negative words). For evaluation they employed a corpus of one hundred billion words with a test word set of 3596 words (1614 positive, 1982 negative). They were able to achieve accuracy of 80%.

Even though unsupervised learning techniques are flexible [36] than their counterpart, they generate generalized models which does not fit well for a specified problem. Most of the time unsupervised techniques can be improved using prior information, making them semi-supervised or supervised. Furthermore unsupervised techniques have poor performance compared to supervised learning techniques.

2.2.2 Supervised

Supervised learning techniques use labeled input data for constructing the model. Most of the time supervised techniques gives better performance over unsupervised techniques.

Pang et al [5] proposed a supervised learning approach for sentiment classification problem. This research mostly focuses on classifying movie reviews by adapting techniques used in topic based classification. Some review systems contain rating score with reviews (comments). Therefore labeling is not required. Here they have classified movies with ratings 4, 5 as positive and 1, 2 as negative. Neutral rating of 3 is ignored. Naive Bayes, Maximum Entropy and Support Vector Machines were used as classifier algorithms. In different test setups unigrams (bag of individual words), bigrams, word frequency, POS tags, position and negation tags were used as input features. Evaluation results shows that SVM with unigrams as features has the best classifier accuracy with 83% for balance training data.

Pak et al [39] employed a dataset extracted from tweets to evaluate their supervised sentiment classification technique. Corpus for training the algorithm was collected by querying for two types of (happy and sad) emoticons using twitter API. Term presence, POS tags and n-grams (unigram, bigrams and trigrams) were used as features. After the preprocessing step multinomial Naive Bayes, SVM and Conditional Random Fields (CRF)[40] were used as classification algorithms. Naive Bayes was able to achieve higher F values compared to other two algorithms.

Hatzivassiloglou et al [38] proposed a supervised approach for predicting semantic orientation of adjectives. They employed a log linear regression model for the task. They demonstrated fact that conjunctions between adjectives provide indirect information about the orientation. Evaluation results shows accuracy values up to 90%.

Despite their superior performance over unsupervised techniques, supervised learning techniques generate problem specific solutions which are hard to transfer to an another domain [36]. Since they require prior knowledge, they cannot be applied to problems which does not have prior knowledge and need training period before classification.

2.2.1 Other Languages

Apart for the English language, few research efforts can be found for Chinese, Japanese, Russian and Hindi languages. While some attempts to directly translate English language research others have used novel approaches by utilizing native language features.

Yussupova et al [32] and Pak et al [33] experimented the effect of lemmatization on sentiment classification of Russian language. They have used SVM, Naive Bayes with n-grams, POS tags and d-grams in different test setups. Evaluation of developed algorithms were carried out for dataset containing bank loan reviews. In the results SVM outperformed Naive Bayes from a small margin.

Joshi et al. [34] employed SVM algorithm to classify comments of Hindi language. First Google translator was used to translate Hindi corpora into English language. Then existing English sentiment lexicon was used to carry out learning and evaluation. Bakliwal et al [35] generated a Hindi lexicon by projecting sentiWordNet's synsets into Hindi language. Classification was carried for different features such as n-grams and with stemming. It was mentioned in the paper that the poor performance experienced was due to the errors in translation software used.

2.2.3 Sentiment Classification of News Comments

Sentiments for news comments are expressed towards the accompanying news article or subject of the article. They share many similarities with sentiment classification problems discussed so far.

Fan and Sun [30] have developed a complete system to collect and analyze Chinese news comments. A web crawler capable of identifying dynamically generated contents through Ajax technology, was used for document collection. After word segmentation, Chinese language specific features were identified. POS tagging was carried out to identify nouns, verbs, adjectives, adverbs as well as interjection, onomatopoeia, pronouns and idioms. It was mentioned that latter four features can carry special sentiment value when it comes to Chinese language. Through experimental results they found out that out of two machine learning techniques Support Vector Machines outperformed k-nearest neighbor approach.

Zhang et al. [31] have used Word2vec [50] and SVM^{perf} [58] to classify Chinese news comments. Word2Vec is a tool based on deep learning which was released by Google. After segmenting words POS tagger was used to identify relevant words and to remove stop words. First Word2vec was used for clustering of similar features. Then both Word2vec and SVM^{perf} was used for final sentiment classification. Evaluation results shows accuracy and recall values in higher than 85%.

Moreo et al [1] introduced comment sentiment analysis system that consist of hand built lexicon, focus detection module and sentiment analysis module. They effectively used sentence focus to resolve the ambiguity of comments. These focuses are identified automatically. Hand built lexicon was enhanced by adding various domain extensions. Proposed method is as below.

- Comments with inconclusive information are filtered out
- All implicit and explicit comments focuses are automatically recognized
- Comments polarity and strength are calculated with the help of taxonomy-lexicon
- Mining techniques are used to generate interpretable summaries

Diakopoulos [43] examined the relationship between topicality, time and sentiment in online news comments. Proposed method consist of lexicon and simple classifier to assign a sentiment score (positive score and negative score) between 1 and 0 to a given sentence. Positive score was assigned by summing up the score of each positive word and then dividing by the total number of words in the sentence.

2.2.3.1 Sinhala Language

SentiWordNet [20] is a sentiment lexicon for English language built from WordNet. In sentiWordNet each word contains three scores for positivity, negativity and objectivity. Medagoda et al [19] have developed a sentiment lexicon of Sinhala language using sentiWordNet 3.0 and an online Sinhala/English dictionary. The dictionary contained the synonyms for Sinhala word and its English counterpart. Each adjective and adverb in sentiWordNet was looked up in the Sinhala dictionary and Sinhala word and related synonyms were given the sentiment scores of the original word. Carrying out this process for each word of the sentiWordNet produces a Sinhala sentiment lexicon. Following assumptions were made during the process.

- Sense of the word for both languages are same
- Sentient score for similar words in both languages are same
- POS of both languages are equivalent

Only adjectives and adverbs were considered as they are the most important language units when sentiment is considered. They were able to extract 5973 adjectives and 405 adverbs from the process. They also discovered the fact that most of the Sinhala adjectives contained more than one synonym.

Constructed lexicon was tested against 2083 news article comments extracted from www.lankadeepa.lk. Opinions supporting the article were marked as positive, criticizing the article was marked as negative and others were marked as objective. A parser was used to extract adjectives and adverbs from opinion sentences. Three classification algorithms were used in the process of sentiment classification. They are Naive Bayes, Support Vector Machine and Decision tree (J48). Precision, recall and F-measure was used for the evaluation. In the first pass evaluation results were less than 50%. Second pass was carried out by removing neutral label from the results which increased the accuracy value up to 60%. They also claims that the adjectives are more important than adverbs. Inclusion of negative words and n-grams were pointed out as the future research directions. Following is a one of rules extracted from J48 decision tree.

```
if adjective ratio of opinion is > 0.666 & sentiment score > -0.125
        then positive
else if adjective sentiment score < 0.25
        then negative.
```

2.2.4 Features

In any machine learning problem identifying features is an important task. In text classification feature vectors should be generated from unstructured text. Classification results will greatly vary according to the selected features. Therefore identifying appropriate features is utmost important. Following is a list of main features used in the literature [7] [8].

- Term presence
- Term position
- Term frequency and different IR weighting schemes
- Part-of-speech (POS) tags
- Opinion words and phrases
- Negations
- Syntactic dependency
- Word embedding

Term presence can be considered as one of the most fundamental feature that could be used for text classification. In term presence, only the term's existence in a document is considered. This generates a binary valued feature vector in which entries indicate whether a term occurs or not. Most of the time term presence will simply refer to the unigram model. That is only single word phrases are considered when creating the feature vector. This is just a specific version of the more general n-gram model. In n-gram model, feature vector may consists of variable size phrases. Most common n-grams are the unigram and bigram (two word phrases).

Instead of using just a binary value, term frequency uses the term count in feature vector. This adds few more details to the feature vector. Term frequency method can be further improved by employing different IR weighting schemes such as tf-idf weighting scheme. Positional information of a term within the sentence/document can also carry valuable information. Term presence can be further augmented by adding positional information [6] (e.g. in the middle or end of the sentence/document). Augmenting term presence feature vector does not guarantee performance increase. Instead it can have a negative effect. This was noticed by Pang et al [5] where simple term presence (unigram) model outperformed most of the other advanced feature vectors in movie reviews.

Part-of-speech tagging is widely used technique in general text analysis. It can be considered as a crude form of Word Sense Disambiguation (WSD) [9]. It has been noticed that adjectives are playing a major role in discovering semantic orientation of sentences. Research on subjectivity detection [10] has found high correlation between presence of adjective and sentence subjectivity. Turney [4] showed that instead of using isolated adjectives, POS patterns (with adjective, adverbs and nouns) with several words can be used more effectively. Apart from adjectives other POS tags can also carry important information. Nouns such as ‘love’ and ‘like’ can be considered as strong indicators of sentiment.

Negations, also known as polarity reversers play an important role in sentiment analysis. Similarity measures based on bag of words representation will consider ‘I love Jane’ and ‘I don’t love Jane’ as similar in orientation while they fall into completely opposite classes. The only difference in above two sentence is the negation word ‘don’t’. Negation can be handle in two ways [10]. That is either in initial feature extraction or in second order feature extraction. In second order feature extraction, first pass of the feature vector generation ignores the negation words. On the second pass information on the polarity reversers are added into feature vector. Das and Chen [10] proposed attaching ‘NOT’ to words occurring close to negation terms on the initial feature extraction process. According to them ‘love-NOT’ will be extracted from the sentence ‘I don’t love Jane’.

2.2.3 Domain Adaptation (transfer learning)

Sentiment classification is very much sensitive to the domain of the training data. Words used in different domains for expressing sentiment can mean something else in some other domain. As an example consider the sentence “this vacuum cleaner really sucks”. The word “sucks” will carry negative sentiment most of the time but above sentence carry a positive sentiment towards the cleaner. Therefore it is evident that sentiment words learned from one domain will have poor performance over other domain. Yet most of the time we could identify general set of opinion words that could fit into all of the domains [21] [22] [23].

2.2.4 Cross-lingual Sentiment Classification

Most of the sentiment analysis research is focused on English language. Therefore in the literature we could find many English sentiment corpora. If we take Sinhala language for an example it would be very difficult to find labeled data if not impossible. If English corpora can be transferred to the Sinhala, burden of labeling data can be relieved. Few research efforts can be found that uses machine translation techniques to transform English corpora into target language data [24] [25] [26].

2.2.5 Sentiment Shifters

Sentiment shifters also known as valence shifters are words or phrases that can shift or change the orientation of the sentiment [27][28]. Ignoring shifters can greatly decrease the classifier accuracy. Negations words like not, never, cannot are the most commonly seen shifters. Other words or phrases such as modal auxiliary verbs (ex. would, should, could) can also change the sentiment orientation. Pre-suppositional items such as hardly and barely can also change the orientation of the opinion. Some nouns can also have a similar effect (ex. fail, omit, neglect). Few examples for each case is listed below.
negators: He does not love her.
modal auxiliary verbs: She could be loved.
pre-suppositional: She barely knows him.
some nouns: She fails to treat him right.

Sarcasm is another sentiment shifter that can be commonly seen in news comments (especially on articles related to politics). Handling sarcasm could be extremely difficult as they cannot be represented in one or two words. Tsur, Davidov, Rappoport [29] introduced mechanism for identifying sarcasm in objective expressions.
E.g.: What a great car, it stopped working on the second day.

2.3 Lexicon

Most of the sentiment analysis research uses a lexicon to categorize bias of a sentence towards positive or negative direction. This pre-built dictionary/lexicon of opinion words are then mapped with sentences to capture the sentiment of the particular sentence. Success of sentiment analysis research heavily depends on the built lexicon and its structure. Therefore building a solid lexicon is the first stage in sentiment classification problem. Lexicon can consist of single words or expressions.

Sentiment words example:
Positive: *beautiful*, *handsome* and *lovely*
Negative: *ugly*, *lazy*, *cost an arm and a leg*

Here on by 'sentiment words' we will refer to both words and expressions. Sentiment words or phrases are also known as polar words or opinion bearing words. There could be an endless number of sentiment words. Many of them are context and application dependent. Therefore building an exhaustive list of sentiment words is not practical.

There are mainly three ways to build a lexicon. They are

- Manual approach
- Dictionary based approach
- Corpus based approach

Manual approach is to select the sentiment words in data set by hand. This might require considerable time and effort. Since this is a onetime effort, it's a perfectly viable option. To avoid over fitting it is important to extract the sentiment words from a data set separate from test set.

Dictionary based approach start with small seed set of sentiment words and expand the list using a thesaurus. WordNet's synsets (synsets are grouped synonyms) and hierarchies are widely used for this approach. Hu and Liu [11], Kim and Hovy [12], Kamps et al [13] used synonyms and antonyms in WordNet to iteratively build the lexicon. Andreevskaia and Bergler [14] used a more complex method with three passes. In first pass seed set was extended by using synonyms, antonyms and hyponyms. Then in the second pass lexicon was further extended by using glosses (glosses are definitions of words in synsets). On the third pass POS taggers were used to clean up and remove contradictions. Finally each word is given a fuzzy value between positive and negative ends. This process is conducted several time using non overlapping seed sets and finally an aggregate score was calculated for each of the word in lexicon.

Corpus based approach relies on syntactic patterns available on large corpora. It was shown by many researchers that corpora based approach can easily find domain dependent orientation towards positive, negative or neutral direction. Hazivassiloglou and McKeown [15] used connective operators to identify relations between sentiment words. He suggested that adjectives connected with conjunction (*and*) has same orientation. He also commented on the connection between two words if they are connected by *or, but, either-or, neither-nor* operators. Turney [4], Yu and Hazivassiloglou [16] assigned opinion orientation to words and phrases starting from seed words. They used Point-wise Mutual Information (PMI) and log likelihood ratio to measure similarity. Kanayama and Nasukawa [17] also used connecting words to identify similarities and extend sentiment words for Japanese language. Instead of only using intra sentence relatedness they further extended the idea by incorporating previous and next sentence to the modal. Ding Liu and Yu [18] suggested that domain adaptation is insufficient for some situations and context adaptation is required. They defined context as a pair that containing sentiment word and aspect or entity (*context: (adjective, aspect)*). Then opinion orientation was assigned to the pair instead of sentiment word.

Manual approach gives better results most of the time but is labor intensive. For some applications it might be impossible to manually construct the lexicon due to sheer size. Dictionary based can generate very large lexicon which covers most of the sentiment words. But it cannot adapt to domain or context specific scenarios. Even though Corpus based method can easily adapt to domain specific applications they find it hard to generate lagged set of opinion words. We can see that all of the above three methods has its ups and downs. Therefore selecting the best approach is application dependent and hybrid approach is most likely to outperform individual ones.

2.4 Focus Detection

The main focus of a document/sentence is known as focus or aspect. Document focus plays an important role in text classification. Identifying the focus is not a trivial task. Because of the implicit nature of the languages aggregation of several methods should be used to identify the focus. Based on sentence/comment target's (focus, subject) availability comments can be divided into two categories as implicit target and explicit target.

- Implicit Targets: Opinion targets do not occur in the sentence.
- Explicit Targets: Opinion target occur in the sentence.

Few research attempts on news comment classifications use focus identification to improve the classification accuracy [1]. Many more ignore it because of the complications it introduces. Few attempts are listed below.

Ma, Tengfei, and Xiaojun Wan [2] introduced an implicit and explicit target extraction mechanism in Chinese news comments. They used heuristic rules such as appearance of the subject, combination of the POS and position of the predicates to decide the sentence type. Evaluation results were 8.8% better than the baseline methods. A Chinese NLP toolkit was used along the process. Then two approaches were taken to extract target from implicit and explicit type sentences. Focused concept of the news article is used as the target for the implicit type. Then comment and the target was compared to calculate the semantic relatedness. For implicit type, nouns and pronouns of the sentences was extracted and ranked. Centering theory was used to select best candidate among them. Focused entities of an article are mostly its noun entities [42]. Since there can be large number of noun entities, priority was given to focused named entities (FNE).

Wikipedia-base explicit semantic analysis (ESA) [41] was used to calculate the semantic relatedness between sentences. ESA converted words into a series of wiki concepts which is then used to calculate the similarity. They suggested that despite having many difficulties, news comments' characteristics can be used to improve the performance of the extraction process. Using contextual information was also used eliminate noise and minimize the dependence on syntactic parser. Main characteristics that was used are as follows,

- Even though the number of potential opinion targets are large, most of them will be focused on the target/ idea of the article. Therefore article's opinion targets can be used effectively as the opinion target for comments. Article header is very important in this regard.
- It can be identified that sentences in a comment are coherent. Therefore for long comments, opinion targets of each of the sentences are highly correlated.

Moreo, Alejandro et al [1] suggest that to resolve ambiguity of comments instead extracting single focus, multi-focal methods should be used. An automated method for finding opinion targets was introduced. These focuses allow to define the context and easily isolate linguistic interferences of expressions. Opinions that do not explicitly appear in comments and spamming comments are few problems that needs to be addressed as well.

2.5 Evaluation

Most of the research efforts have used simple metrics such as accuracy, precision and F1 score for evaluation. Due to their simplicity, wide usage and interpretability they are the current norm.

Precision

Precision is the fraction of retrieved documents that are relevant to the query.

$$Precision = \frac{Relevant\ Document \cap Retrieved\ Document}{Retrieved\ Document} = \frac{tp}{tp + fp}$$

Equation 2.3 Precision

Recall

Recall is the fraction of the documents that are relevant to the query that are successfully retrieved

$$Recall = \frac{Relevant\,Document \cap Retrieved\,Document}{Relevant\,Documents} = \frac{tp}{tp+fn}$$

Equation 2.4 Recall

Accuracy

Accuracy is the proportion of total number of correctly classified instances to the total number of instances.

$$Accuracy = \frac{tp+tn}{tp+fp+tn+fn}$$

Equation 2.5 Accuracy

F1 measure

F1 is a measure of test accuracy which can be defined using precision and recall.

$$F1\,Score = \frac{2.\left(Precision\,.\,Recall\right)}{Precision + Recall}$$

Equation 2.6 F1 Score

2.6 Summary

Most of the comment sentiment analysis research is based on product review analysis. Therefore when applying these techniques to online news comments extra precautions should be taken. Even though there are few research articles on sentiment analysis in online news comments they are mostly based on English language (few can be found on Chinese and Japanese language as well [30][31][32][33]).

Supervised and unsupervised machine learning methods are being used for Sentiment analysis. While supervised learning tend to have better accuracy, unsupervised methods are easy to carry out as the test data preparation is minimum. Irrespective of the learning method used identifying features is an important task before applying any machine learning techniques.

Accuracy of any sentiment analysis task greatly depends on the quality of the lexicon. Lexicon words are mapped with words of each sentence to give a sentiment score to a particular sentence. Manual, dictionary based or corpus based approach is used to build a lexicon.

CHAPTER 3: RESEARCH METHODOLOGY

Our objective is to classify Sinhala news comments according to their sentiment values. Sub-problems related to main objective were identified in previous chapters. This chapter will discuss the strategy for battling each of the identified subproblems, which are also listed below.

- How to collect relevant news articles
- What type of preprocessing techniques to follow
- What features to extract
- What are the suitable sentiment analysis techniques/algorithms
- How to adapt selected techniques for Sinhala language
- How to prepare the test set and carry out tests

Depicted below in Figure 3.1 is the high-level architecture diagram of the proposed solution.

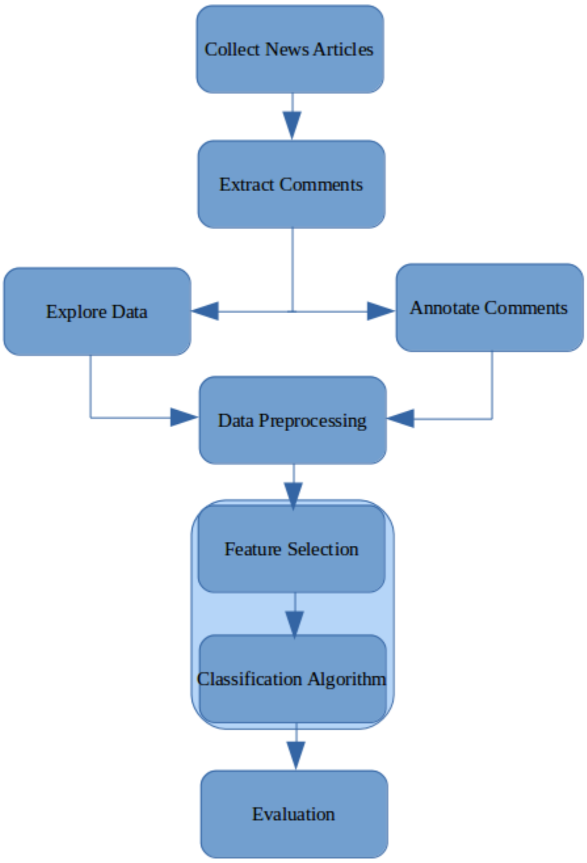


Figure 3.1: Proposed system architecture

Since we were short of a dataset of Sinhala news comments, we had to start by compiling one. Online news site Lankadeepa has large collection of articles which contain moderated comments. Since this fits well with our requirements www.lankadeepa.com was selected as our main data source. Out of two options manual collection and employing a web crawler, we sided with the latter as manual collection is tedious and time consuming.

As we identified in previous chapter supervised machine learning techniques have shown better results. To engage such technique we need an annotated dataset. Therefore few annotators were employed to tag comments with right sentiment values.

After identifying the nature of data they were prepared for further processing by cleaning and transforming to a suitable format. To simplify the feature selection process, directly available features were used. Initial experiments were carried out using bag of word model. Then tf-idf, 2-gram word vectors and finally word embeddings techniques with different aggregation methods were employed. Use of POS tags, lexicon of opinion words/phrases, negation words and syntactic dependencies were left as future enhancements.

Once feature selection/extraction is finished, identified machine learning techniques were employed for classifying comments. Due to simplicity and comparable performance, Naive Bayes was selected as a starting point of the experiments. In the literature superiority of the SVM over other algorithms were witnessed. Therefore testing was carried out using SVM to compare the results with Naive Bayes implementation. Further experiments were conducted using deep learning techniques combined with word embedding features.

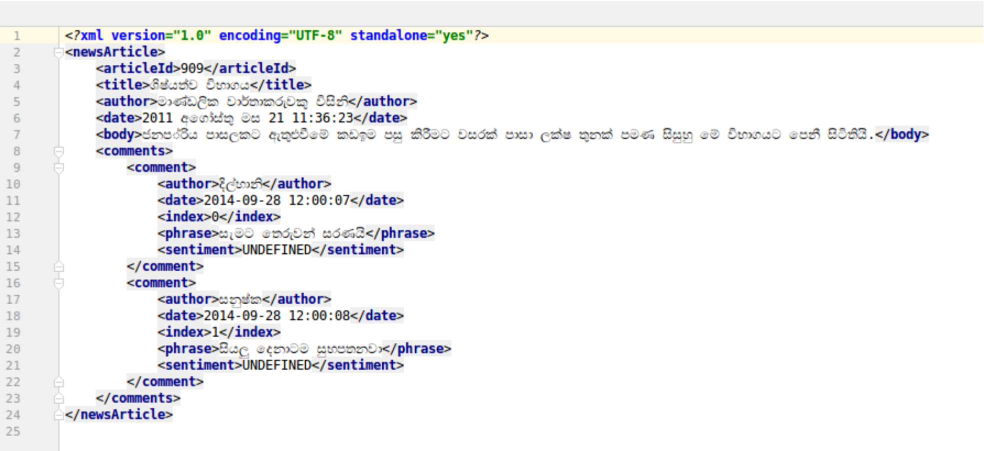
3.1 Architecture and Implementation

In this section we will discuss the implementation details in greater depth.

3.1.1 Data Collection

For the purpose of this research, online news site Lankadeepa.lk was selected as the primary data source since it has a large collection of news articles with manually moderated comments. Even-though we decided to collect data manually in initial phase since it is too cumbersome a web crawler was employed for the task. Among many crawling libraries present in Java we opt to use Crawler4j [54] because of its ease of use and popularity.

Crawler4j provides a simple interface for crawling through websites. While we kept most of the configurations in its default values number of crawlers, storage folder and seed URLs were changed to appropriate values. Crawler4j concurrently uses as many crawlers as the value passed to its number of crawlers setting. Similar number of seed URLs were provided as the starting URL for each crawler thread. Output of the Crawler4j module is a string containing HTML tags. This was parsed to extract news articles and comments. Using JSoup [55] Java library we were able to extract news articles, comments and metadata which we identified in previous section. Extracted news articles were saved in XML format for later consumption. Figure 3.2 depicts a sample article that was collected.



```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<newsArticle>
    <articleId>909</articleId>
    <title>ශිෂ්‍යත්ව විභාගය</title>
    <author>මාණ්ඩලික වාර්තාකරුවකු විසිනි</author>
    <date>2011 අගෝස්තු මස 21 11:36:23</date>
    <body>ජනප්‍රිය පාසලකට ඇතුළුවීමේ කඩඉම පසු කිරීමට වසරක් පාසා ලක්ෂ තුනක් පමණ සිසුහු මෙ විභාගයට පෙනී සිටිති.</body>
    <comments>
        <comment>
            <author>දිල්හානි</author>
            <date>2014-09-28 12:00:07</date>
            <index>0</index>
            <phrase>සැමට තෙරුවන් සරණයි</phrase>
            <sentiment>UNDEFINED</sentiment>
        </comment>
        <comment>
            <author>සනුෂ්ක</author>
            <date>2014-09-28 12:00:08</date>
            <index>1</index>
            <phrase>සියලු දෙනාටම සුභපතනවා</phrase>
            <sentiment>UNDEFINED</sentiment>
        </comment>
    </comments>
</newsArticle>
```

Figure 3.2: XML formatted news article

To carry out supervised learning task, annotated dataset is needed. Therefore our next objective is annotating the dataset that we assembled. More than 5000 comments were annotated using 3 labels POSITIVE, NEGATIVE and NEUTRAL. To carry out the experiment only POSITIVE and NEGATIVE annotated comments were used (our baseline Medagoda et al [19] used a dataset of 2083 comments). When assigning a label to a particular comment following guidelines were followed.

1. Assign a label to each comment according to its sentiment orientation towards the corresponding article.
2. If comment’s sentiment is not directly towards the news item but in a general way, assign the label according to its global sentiment orientation.
3. If a particular comment cannot be assigned a label according to above 2 rules, mark it as CONFLICT or leave it as UNDEFINED.

Following list displays few example annotated comments according to above rules.

News Item:

මෙරට ප්‍රථම විධායක ජනපති ජේ.ආර්.ජයවර්ධන මහතාගේ 110 වැනි ජන්ම දින සංවත්සරය අද (19) අගමැති රනිල් වික්‍රමසිංහ මහතාගේ ප්‍රධානත්වයෙන් කොළඹ ජේ.ආර්.ජයවර්ධන කේන්ද්‍රයේදී පැවැත්විණි. ජයවර්ධන මහතාගේ පුත් අමරිත් ජයවර්ධන මහතා ජයවර්ධන පිළිරුවට පුෂ්පෝපහාර දැක්විය.

Comments:

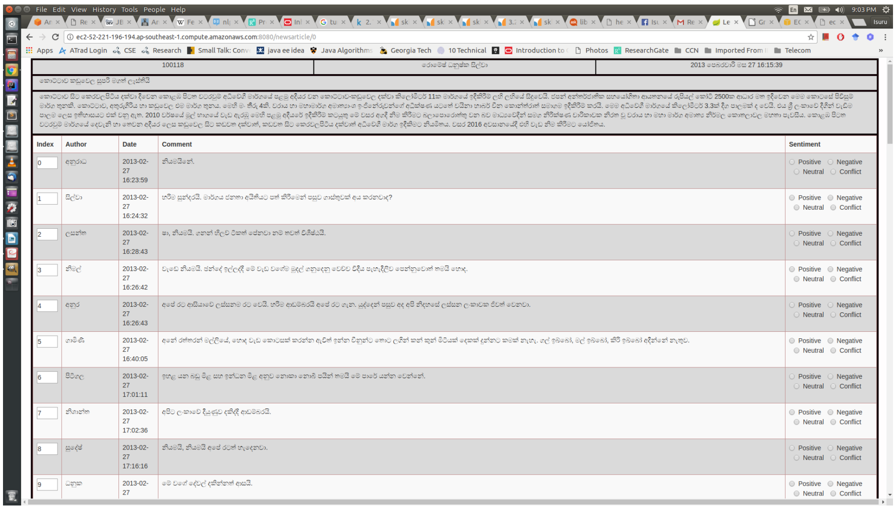
මැත ඉතිහාසයේ ලංකාවෙ පහළ වුණු ශ්‍රේෂ්ඨතම ජන නායකයා : POSITIVE

පවතින ව්‍යවස්ථා අර්බුදයේ නිර්මාතෘ. ඉතාම කෙටි දැක්මක් තිබූ ජන නායකයා. : NEGATIVE

“නිවහල් ධර්මිෂ්ට සමාජයක් ඇති කිරීම සඳහා මටත් අවස්ථාවක් දෙන්න" යන ආදර්ශ පාඨය එක්ක තමයි මෙතුමා බලය ඇල්ලුවෙ 1977 දී. ධර්මිෂ්ට සමාජයක් ඇතිකළාද තිබුණ ධර්මිෂ්ට සමාජය නැති කළාද කියලා තේරුම් ගන්න ඔබට බාරයි. රටෙ සංවර්ධනය නම් වුණා: CONFLICT

සුසන්ත.විව්‍යස්තාවෙ අර්බුද ඇතිවුනේ 18 ගෙනාවට පස්සේ එක ගෙනාවෙ කවුද .? ගෙනාවෙ රටට ද .? මොකටද .? : NEUTRAL

Annotation task was carried out manually with the help of 2 annotators. Since we could not find a suitable tool to simplify the label annotating process, simple web application was written using Spring boot framework [56]. Respectively Figure 3.3 and Figure 3.4 depicts the web tool and annotated comment sample.


File Edit View History Tools People Help
ec2-52-221-196-194.ap-southeast-1.compute.amazonaws.com:8080/newsarticle/0
Apps ATrad Login CSE Research Small Talk: Conv java ee idea Java Algorithms Georgia Tech 10 Technical Introduction to Photos ResearchGate CCN Imported From Telecom
9:03 PM Isuru

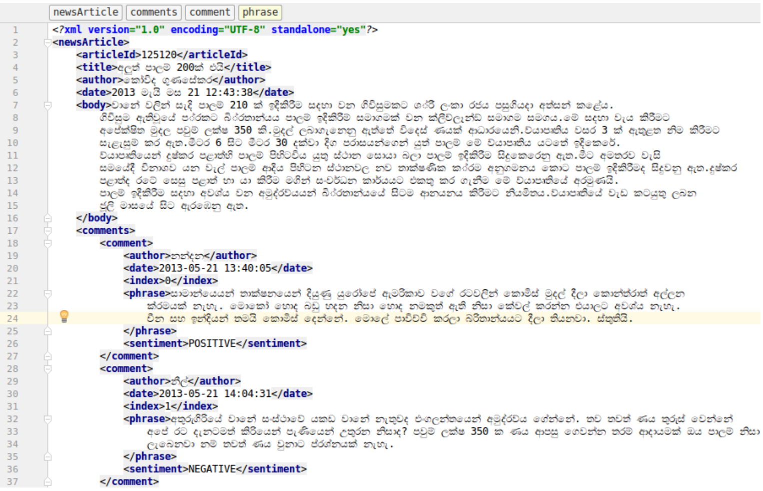
100118	රොමේෂ් ධනුෂ්ක සිල්වා	2013 පෙබරවාරි මස 27 16:15:39

කොට්ටාව කඩුවෙල සුපිරි මගක් ලෑස්තියි

කොට්ටාව සිට කෙරවලපිටිය දක්වා දිවෙන කොළඹ පිටත වටරවුම් අධිවේගී මාර්ගයේ පළමු අදියර වන කොට්ටාව-කඩුවෙල දක්වා කිලෝමීටර් 11ක මාර්ගයේ ඉදිකිරීම් ලඟි ලඟියේ සිදුවෙයි. ජපන් අන්තර්ජාතික සහයෝගිතා ආයතනයේ රුපියල් කෝටි 2500ක ආධාර මත ඉදිවෙන මෙම කොටසේ පිවිසුම් මාර්ග තුනකි. කොට්ටාව, අතුරුගිරිය හා කඩුවෙල එම මාර්ග තුනය. මෙහි මං තීරු 4කි. වරාය හා මහාමාර්ග අමාත්‍යාංශ ඉංජිනේරුවන්ගේ අධීක්ෂණ යටතේ චයිනා හාබර් චීන කොන්ත්‍රාත් සමාගම ඉදිකිරීම් කරයි. මෙම අධිවේගී මාර්ගයේ කිලෝමීටර් 3.3ක් දිග පාලමක් ද වෙයි. එය ශ්‍රී ලංකාවේ දිගින් වැඩිම පාලම ලෙස ඉතිහාසයට එක් වනු ඇත. 2010 වර්ෂයේ මුල් භාගයේ වැඩ ඇරඹූ මෙහි පළමු අදියරේ ඉදිකිරීම් කටයුතු මේ වසර අගදී නිම කිරීමට බලාපොරොත්තු වන බව මාධ්‍යවේදීන් සමග නිරීක්ෂණ චාරිකාවක නිරත වූ වරාය හා මහා මාර්ග අමාත්‍ය නිර්මල කොතලාවල මහතා පැවසීය. කොළඹ පිටත වටරවුම් මාර්ගයේ දෙවැනි හා තෙවන අදියර ලෙස කඩුවෙල සිට කඩවත දක්වාත්, කඩවත සිට කෙරවලපිටිය දක්වාත් අධිවේගී මාර්ග ඉදිකිරීමට නියමිතය. වසර 2016 අවසානයේදී එහි වැඩ නිම කිරීමට යෝජිතය.

Index	Author	Date	Comment	Sentiment
0	අනුරාධ	2013-02-27 16:23:59	නියමයිනේ.	○ Positive ○ Negative ○ Neutral ○ Conflict
1	සිල්වා	2013-02-27 16:24:32	හරිම සුන්දරයි. මාර්ගය ජනතා අයිතියට පත් කිරීමෙන් පසුව ගාස්තුවක් අය කරනවාද?	○ Positive ○ Negative ○ Neutral ○ Conflict
2	උසන්ත	2013-02-27 16:28:43	ෂා, නියමයි. ගහන් හිලව් ටිකත් පේනවා නම තවත් විශිෂ්ඨයි.	○ Positive ○ Negative ○ Neutral ○ Conflict
3	නිමල්	2013-02-27 16:26:42	වැඩෙ නියමයි. ඡන්දේ ඉල්ලද්දී මෙ වැඩ වගේම මුදල් ගනුදෙනු වෙච්ච විදිය පැහැදිලිව පෙන්නුවොත් තමයි හොඳ.	○ Positive ○ Negative ○ Neutral ○ Conflict
4	අනුර	2013-02-27 16:26:43	අපේ රට ආසියාවෙ උස්සනම රට වෙයි. හරිම ආඩම්බරයි අපේ රට ගැන. යුද්දෙන් පසුව අද අපි නිදහසේ උස්සන ලංකාවක ජීවත් වෙනවා.	○ Positive ○ Negative ○ Neutral ○ Conflict
5	ගාමිණී	2013-02-27 16:40:05	අනේ රත්තරන් මල්ලියේ, හොඳ වැඩ කොටසක් කරන්න ඇවිත් ඉන්න චීනුන්ට තොට ලගින් කන් කුන් මිටියක් දෙකක් දුන්නට කමක් නැහැ. ගල් ඉබබෝ, මල් ඉබබෝ, කිරි ඉබබෝ අඳින්නේ නැතුව.	○ Positive ○ Negative ○ Neutral ○ Conflict
6	පිටිගල	2013-02-27 17:01:11	ඉහළ යන බඩු මිළ සහ ඉන්ධන මිළ අනුව නොකා නොබී පයින් තමයි මේ පාරේ යන්න වෙන්නේ.	○ Positive ○ Negative ○ Neutral ○ Conflict
7	නිශාන්ත	2013-02-27 17:02:36	අපිට ලංකාවේ දියුණුව දකිද්දී ආඩම්බරයි.	○ Positive ○ Negative ○ Neutral ○ Conflict
8	සුදේෂ්	2013-02-27 17:16:16	නියමයි, නියමයි අපේ රටත් හැදෙනවා.	○ Positive ○ Negative ○ Neutral ○ Conflict
9	ධනුක	2013-02-27	මේ වගේ දේවල් දකින්නත් ආසයි.	○ Positive ○ Negative ○ Neutral ○ Conflict

Figure 3.3: Web application to simplify annotation task



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<newsArticle>
    <articleId>125120</articleId>
    <title>අලුත් පාලම් 200ක් එයි</title>
    <author>කෝවිද ගුණසේකර</author>
    <date>2013 මැයි මස 21 12:43:38</date>
    <body>වානේ වලින් සෑදි පාලම් 210 ක් ඉදිකිරීම සඳහා වන ගිවිසුමකට ශ්‍රී ලංකා රජය පසුගියදා අත්සන් කළේය.
        ගිවිසුම ඇතිවූයේ ප්‍රකට බ්‍රිතාන්‍යය පාලම් ඉදිකිරීම් සමාගමක් වන ක්ලිව්ලෑන්ඩ් සමාගම සමගය.මේ සඳහා වැය කිරීමට
        අපේක්ෂිත මුදල පවුම් ලක්ෂ 350 කි.මුදල් ලබාගැනෙනු ඇත්තේ විදෙස් ණයක් ආධාරයෙනි.ව්‍යාපෘතිය වසර 3 ක් ඇතුළත නිම කිරීමට
        සැළැසුම් කර ඇත.මීටර 6 සිට මීටර 30 දක්වා දිග පරාසයන්ගෙන් යුත් පාලම් මේ ව්‍යාපෘතිය යටතේ ඉදිකෙරේ.
        ව්‍යාපෘතියෙන් දුෂ්කර පළාත්හි පාලම් පිහිටවිය යුතු ස්ථාන සොයා බලා පාලම් ඉදිකිරීම සිදුකෙරෙනු ඇත.මීට අමතරව වැසි
        සමයේදී විනාශව යන වැල් පාලම් ආදිය පිහිටන ස්ථානවල නව තාක්ෂණික ක්‍රම අනුගමනය කොට පාලම් ඉදිකිරීමද සිදුවනු ඇත.දුෂ්කර
        පළාත්ද රටේ සෙසු පළාත් හා යා කිරීම මගින් සංවර්ධන කාර්යයට එකතු කර ගැනීම මේ ව්‍යාපෘතියේ අරමුණයි.
        පාලම් ඉදිකිරීම සඳහා අවශ්‍ය වන අමුද්‍රව්‍යයන් බ්‍රිතාන්‍යයේ සිටම ආනයනය කිරීමට නියමිතය.ව්‍යාපෘතියේ වැඩ කටයුතු ලබන
        ජූලි මාසයේ සිට ඇරඹෙනු ඇත.
    </body>
    <comments>
        <comment>
            <author>නන්දන</author>
            <date>2013-05-21 13:40:05</date>
            <index>0</index>
            <phrase>සාමාන්‍යයෙන් තාක්ෂනයෙන් දියුණු යුරෝපේ ඇමරිකාව වගේ රටවලින් කොමිස් මුදල් දීලා කොන්ත්‍රාත් අල්ලන
                ක්‍රමයක් නැහැ. මොකෝ හොඳ බඩු හදන නිසා හොඳ නමකුත් ඇති නිසා කේවල් කරන්න එයාලට අවශ්‍ය නැහැ.
                චීන සහ ඉන්දියන් තමයි කොමිස් දෙන්නේ. මොලේ පාවිච්චි කරලා බ්‍රිතාන්‍යයට දීලා තියනවා. ස්තුතියි.
            </phrase>
            <sentiment>POSITIVE</sentiment>
        </comment>
        <comment>
            <author>නිල්</author>
            <date>2013-05-21 14:04:31</date>
            <index>1</index>
            <phrase>අතුරුගිරියේ වානේ සංස්ථාවේ යකඩ වානේ නැතුවද එංගලන්තයෙන් අමුද්‍රව්‍ය ගේන්නේ. තව තවත් ණය තුරුස් වෙන්නේ
                අපේ රට දැනටමත් කිරියෙන් පැණියෙන් උතුරන නිසාද? පවුම් ලක්ෂ 350 ක ණය ආපසු ගෙවන්න තරම් ආදායමක් ඔය පාලම් නිසා
                ලැබෙනවා නම් තවත් ණය වුනාට ප්‍රශ්නයක් නැහැ.
            </phrase>
            <sentiment>NEGATIVE</sentiment>
        </comment>
```

Figure 3.4: Annotated article

Starting point of any machine learning problem is understanding data. We have a dataset of 16,000 news articles out of which 276 articles had annotated comments. Articles were from a wide variety of categories including politics, sports, crime, economy, society and culture. Politics was the most disputed article category attracting wide range of comments attributing both positive and negative sentiments. Each article contained variable amount of comments ranging from 1 to 200. While skimming through the articles we found that articles containing a higher number of comments held strong sentiment values. Figure 3.5 shows the article distribution against number of comments per article and date.

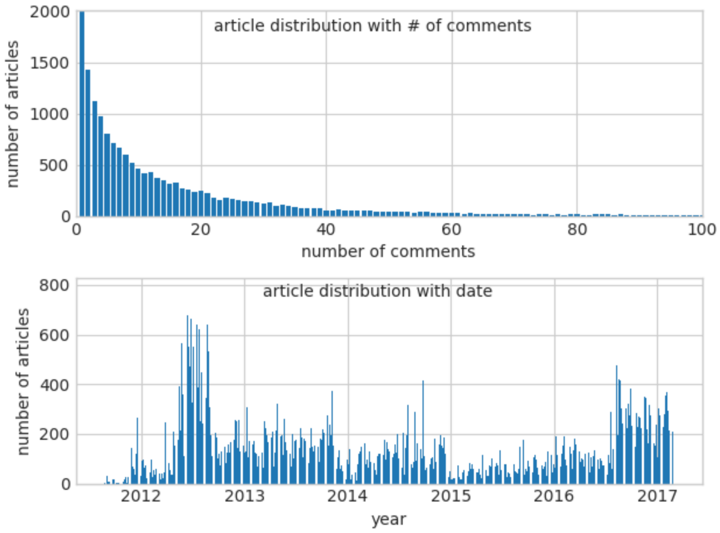


Figure 3.5: Exploring data

3.1.1.1 Inter-rater Agreement

Table 3.1 lists down the inter-rater agreement for annotated dataset. Equation 3.1 shows the calculations of Cohen's kappa measure. 366 comments were annotated by both annotators for the calculation.

Table 3.1: Inter-rater agreement

		Annotator 2		
		POSITIVE	NEGATIVE	OTHER
Annotator 1	POSITIVE	184	4	52
	NEGATIVE	3	52	41
	OTHER	2	3	25

$$Kappa(k)=\frac{P_o-P_e}{1-P_e}=\frac{0.71-0.41}{1-0.41}=0.52$$

Equation 3.1: Cohen's kappa

By analyzing annotated comments and results of table 3.1 we can see that most of the disagreements have happened between POSITIVE, OTHER and NEGATIVE, OTHER categories. Only few disagreements have occurred between POSITIVE and NEGATIVE categories. We can identify few reasons for this

- Some comments doesn't hold strong sentiment orientation. Therefore one annotator has marked them as NEUTRAL while the other marked them has POSITIVE / NEGATIVE
- When annotating, our first rule was to annotate the comments according to their sentiment orientation towards the news article. But most of the comments' sentiment orientation is towards the subjects of the article instead of news article itself. This has aroused some confusion.

Following examples show few conflicting scenarios. Conflicting labels are shown in parenthesis in front of each comment.

Example 1

Article title: තුවක්කුවෙ එල්ලුනත් කඳුළු ගෑස් කයි

News Article: රුහුණු සරසවියේ හපුගල ඉංජිනේරු පීඨ සිසුන් පොලිස් අණ නොතකා ගාල්ල දෙසට ගමන් කිරීමට සැරසීමත් සමග කඳුළු ගෑස් ගැසීය.

Comment: පෙලපාලි ගියාම මොකද වෙන්නේ? (NEUTRAL/ NEGATIVE)

Comment: විශේෂඥ වෛද්‍ය පට්ටම ලැබුණාට පස්සෙ ඕගොල්ලන්ගෙ ඔය සමාජවාදය නැහැ. ගහපු එකට ජයවේවා! (NEUTRAL/ POSITIVE)

Example 2

Article title: මත්තලින් බුද්ධගයාවට

News Article: මත්තල සිට බුද්ධගයාවට සහනදායී ගුවන්ගමන් ගාස්තු යටතේ ගුවන් ගමන් අරඹන බව මිහින් ලංකා ගුවන්සේවය කියයි. එම ගුවන්සේවය මෑතකදී.......

Comment: ගුවන් තොටුපල, නව ගුවන් ගමන් ඉතා හොදයි. හැබැයි ගුටිකන පැතිවලට, බොරු අන්තවාදීන් ඉන්න පැති වලට ගුවන් ගමන් ඕනෑ නැහැ. (POSITIVE/ CONFLICT)

Example 3

Article title: මත්තල ගුවන් තලයට එක්කළ වගයි

News Article: මහා සංඝරත්නයේ සෙත්පිරිත්, ජය සක්, මඟුල් බෙර හා මහජන ප්‍රීති ඝෝෂා මත්තල රාජපක්ෂ අන්තර් ජාතික ගුවන්තොට සිසාරා ඇසෙද්දී මහින්ද රාජපක්ෂ ජනපති කටුනායක සිට ශ්‍රී ලංකන් ගුවන් යානයකින් මත්තලට පැමිණි නව ගුවන්තොට විවෘත කළේය.......

Comment: ඇත අහසේ ඉදලා ඒ ගුවන් යානය ආපු හැටි දැක්කාද (POSITIVE/ NEUTRAL)

Comment: වැඩේ නම් හොදයි. බඩු මිළ තමයි වැඩි. (POSITIVE/ NEUTRAL)

3.1.2 Preprocessing

Preprocessing plays an important role in text classification tasks. Moreover user inputed texts such as comments and reviews lack the polish that's present in a proofread document. Therefore comments and reviews were cleaned and shaped thoroughly as the first step of the text classification process. Out of following commonly used data cleaning methods we opt to try out few depending on the context and availability of resources.

- Remove numbers
- Remove punctuations
- Strip whitespace
- Remove stop-words
- Remove sparse terms
- Stemming
- Lemmatization
- Spell checking

Removing numbers and punctuation marks is easily accomplished using regular expressions. Despite it being a trivial task we had to pay special attention when removing certain punctuation marks as they can play an important role in expressing comment sentiment. At the end of this subsection we discuss in detail the importance of punctuations.

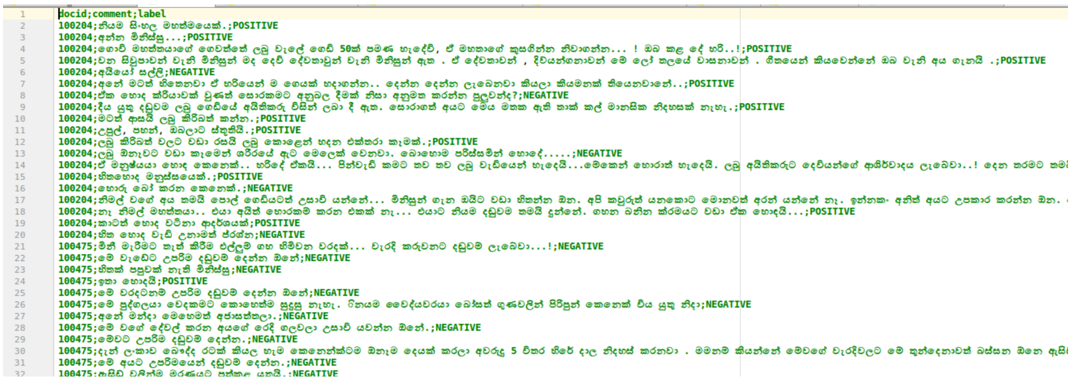
Removing whitespace was also a similar task. However our attention caught few remaining extra whitespace characters in cleaned dataset. This was due to the presence of non-breaking whitespace character (\u00A0) in several comments.

Stop word list that we employed did not has positive effect on the classification performance. This was mainly due to inclusion of words that carry strong sentiment value in the stop word list. Therefore we did not use stop word list for further experiments.

At the time of the experiment we did not have a proper Sinhala stop word list. Therefore we leave this as a future extension. Further we did not specifically remove the sparse terms from the dataset. But we employed some feature extraction methods such as Word2Vec which removes the sparse terms in the process. Word2Vec will be discussed in detail in later section.

Stemming, lemmatization and spell checking was opted out since we did not have resources to carry out these tasks for Sinhala language.

For the purpose of our experiments, we are only using news article comments. Filtering out the comments from other elements would make further processing easier. Therefore as a preprocessing step we extracted comments from XML document and exported all the comments to CSV file format as depicted in Figure 3.6. Resultant file contained rows of news article ID, comment and sentiment value each separated by a comma.



```
docid;comment;label
100204;නියම සිංහල මහත්මයෙක්.;POSITIVE
100204;අන්න මිනිස්සු...;POSITIVE
100204;ගොවි මහත්තයාගේ ගෙවත්තේ ලබු වැලේ ගෙඩි 50ක් පමණ හැදේවි, ඒ මහතාගේ කුසගින්න නිවාගන්න... ! ඔබ කළ දේ හරි..!;POSITIVE
100204;වන සිවුපාවන් වැනි මිනිසුන් මද දෙවි දේවතාවුන් වැනි මිනිසුන් ඇත . ඒ දේවතාවන් , දිව්යන්ගනාවන් මෙ ලෝ තලයේ වාසනාවන් . ගීතයෙන් කියවෙන්නේ ඔබ වැනි අය ගැනයි .;POSITIVE
100204;අයියෝ සල්ලි;NEGATIVE
100204;අනේ මටත් හිතෙනවා ඒ හරියෙන් ම ගෙයක් හදාගන්න.. දෙන්න දෙන්න ලැබෙනවා කියලා කියමනක් තියෙනවානේ..;POSITIVE
100204;ඒක හොද ක්‍රියාවක් වුණත් සොරකමට අනුබල දීමක් නිසා අනුමත කරන්න පුලුවන්ද?;NEGATIVE
100204;දිය යුතු දඩුවම ලබු ගෙඩියේ අයිතිකරු විසින් ලබා දී ඇත. සොරාගත් අයට මේය මතක ඇති තාක් කල් මානසික නිදහසක් නැහැ.;POSITIVE
100204;මටත් ආසයි ලබු කිරිබත් කන්න.;POSITIVE
100204;උපුල්, පහන්, ඔබලාට ස්තුතියි.;POSITIVE
100204;ලබු කිරිබත් වලට වඩා රසයි ලබු කොළෙන් හදන එක්තරා කෑමක්.;POSITIVE
100204;ලබු ඕනෑවට වඩා කෑමෙන් ශරීරයේ ඇට මෙලෙක් වෙනවා. බොහොම පරිස්සමින් හොදේ.....;NEGATIVE
100204;ඒ මනුෂ්‍යයා හොද කෙනෙක්.. හරිදේ ඒකයි... පින්වැඩි කමට තව තව ලබු වැඩියෙන් හැදෙයි...මෙකෙන් හොරාත් හැදෙයි. ලබු අයිතිකරුට දෙවියන්ගේ ආශිර්වාදය ලැබෙවා..! දෙන තරමට තම
100204;හිතහොද මනුස්සයෙක්.;POSITIVE
100204;හොරු බෝ කරන කෙනෙක්.;NEGATIVE
100204;නිමල් වගේ අය තමයි පොල් ගෙඩියටත් උසාවි යන්නේ... මිනිසුන් ගැන ඔයිට වඩා හිතන්න ඕන. අපි කවුරුත් යනකොට මොනවත් අරන් යන්නේ නෑ. ඉන්නකං අනිත් අයට උපකාර කරන්න ඕන.
100204;නෑ නිමල් මහත්තයා.. එයා අයිත් හොරකම කරන එකක් නෑ... එයාට නියම දඬුවම තමයි දුන්නේ. ගහන බනින ක්‍රමයට වඩා ඒක හොදයි...;POSITIVE
100204;කාටත් හොද වටිනා ආදර්ශයක්;POSITIVE
100204;හිත හොද වැඩි උනාමත් ප්‍රශ්න;NEGATIVE
100475;මිනී මැරීමට තැත් කිරිම එල්ලුම් ගහ හිමිවන වරදක්... වැරදි කරුවනට දඬුවම ලැබෙවා...!;NEGATIVE
100475;මෙ වැඩෙට උපරිම දඩුවම දෙන්න ඕනේ;NEGATIVE
100475;හිතක් පපුවක් නැති මිනිස්සු;NEGATIVE
100475;ඉතා හොදයි;POSITIVE
100475;මෙ වරදටනම් උපරිම දඩුවම දෙන්න ඕනේ;NEGATIVE
100475;මෙ පුද්ගලයා වෙදකමට කොහෙත්ම සුදුසු නැහැ. ිනයම වෛද්‍යවරයා බෝසත් ගුණවලින් පිරිපුන් කෙනෙක් විය යුතු නිදා;NEGATIVE
100475;අනේ මන්දා මෙහෙමත් අජාසත්තලා.;NEGATIVE
100475;මෙ වගේ දේවල් කරන අයගේ රෙදි ගලවලා උසාවි යවන්න ඕනේ.;NEGATIVE
100475;මෙවට උපරිම දඩුවම දෙන්න.;NEGATIVE
100475;දැන් ලංකාව බෞද්ද රටක් කියල හැම කෙනෙත්ක්ටම ඕනෑම දෙයක් කරලා අවරුදු 5 විතර හිරේ දාල නිදහස් කරනවා . මමනම් කියන්නේ මෙවගේ වැරදිවලට මෙ තුන්දෙනාවත් බස්සන ඕනෙ ඇසි
100475;මෙ අයට උපරිමයෙන් දඩුවම දෙන්න.;NEGATIVE
100475;ඇසිඩ් වතින්ම මරණයට පත්කළ යුතුයි.;NEGATIVE
```

Figure 3.6: CSV formatted extracted comments

3.1.2.1 Effect of Punctuations

While skimming the dataset, following list of commonly occurring punctuations were identified.

- Full Stop (.)
- Comma (,)
- Exclamation mark (!)
- Question mark (?)
- Colon (:)
- Semicolon (;)
- Slashs (/ \ )
- Brackets (( ) [ ])
- Quotation marks (" " ' ')

Punctuations such as question "?" and exclamation "!" play an important role in expressing comment sentiment. While skimming through the articles we noticed that question mark is mostly attributed to the negative comments and exclamation mark was seen with both positive and negative comments. Therefore to understand the behavior of these punctuations we decided to conduct few experiments. Following 4 datasets were used for testing the importance of punctuations.

- Without removing any punctuation
- All common punctuations removed
- All common punctuations removed except exclamation mark

- All common punctuations removed except question mark

To further understand the effect of punctuations most commonly occurring punctuations were analyzed. Following paragraphs contain a short description and examples of commonly occurring punctuations.

Full Stop

Full Stop (.) was the most commonly occurring punctuation mark in dataset. While it is mostly used for identifying sentence boundaries, it is also used in abbreviations, number formatting, name initials and for stressing particular words in some cases. Following list shows the usage of Full Stop in the dataset.

- Sentence end: අපේ ආණ්ඩුවක් ඉක්මණින් ඕනේ.
- Abbreviations: බී.එම්.ඩබලෙව් එකක් ළහදීම
- Name initials: නාම සෙනහ ඉන්නවා එ.ජා.ප. යට
- Number formatting: සාමාන්‍ය සරුංගලයක මිල රු. 250.00 කි.
- Stressing: මුකුත් නොදන්න තොත්ත බබාලා...........!!!!!!

While Full Stop is crucial for identifying sentence boundaries, other cases make the analysis complex most of the time. Moreover multiple Full Stops used for stress out particular word is difficult to handle. Therefore they were replaced with single Full Stop. Embedding features such as Word2Vec effectively use Full Stop at the end of the sentence to understand sentence context. But other features such as bag of word and tf-idf suffer from the existence of period. Therefore period was replaced with white space before generating bag of word and tf-idf features.

Comma

Comma (,) is one of the most commonly occurring punctuation in the dataset. It is used to separate parts of the sentence or to list an item set. While word embedding can utilize comma to identify word context, other feature models such as bag of word and tf-idf performs better once data set is cleaned of commas. Following list shows usage of comma in the dataset.

- Separate sentence part: සරත්, ඒක බොහෝදුරට ඇත්ත.
- Item list: ඉස්සර සරුංගලයක්, වෙසක් කුඩුවක් හදාගත්තේ අපිම තමයි.

Exclamation Mark

Exclamation mark is also heavily used punctuation in news comment dataset. It is used to express strong emotions or emphasis the statement. Therefore undoubtedly exclamation mark carry strong sentiment both positive and negative. Since it carry both positive and negative emotion, usefulness of the punctuation for our analysis is uncertain and should be examined. Following list shows some positive and negative comments extracted from the dataset.

- නියම පෙම්වතුන්නේ!
- ජය වේවා!
- ඇතැම්විට සතුට පවා මුදල් දී ගත හැක..!
- ලංකාවත් ප්‍රංශය වගේ නම් පාර්ලිමේන්තුවෙ අයට කාර් පර්මිට් දෙන එක නවත්වන්න ඕනෑ. අනේ මෙහෙම ලංකාවක් !!

Question Mark

Question mark (?) is used at the end of the sentence to mark the sentence as a question. In the news comment dataset question mark is heavily used to question the intent or certain aspects of the article, most of the time associating with negative comments. While skimming it was rarely noticed a question mark associated with a positive comment. Therefore it is safe to assume question mark bares a negative sentiment and will contribute greatly to the sentiment classification process. Few examples of the question mark usage is listed below

- අපේ රටට මත්පැන් හදුන්වලා දුන්නේ සුද්දද?
- යවපු නයා කොහොමද?
- ඇයි ගජබින්නය? ඒකත් හැම පෙළපාලියකම ප්‍රධානම ලක්ෂණයක්නේ.

Other Punctuations

Apart from the punctuations discussed above there are many recurring punctuations in the dataset. Most of them helps to identify the context of the sentence words, therefore useful in word embedding features. But bag of word and tf-idf models become less complex without them. Few of the punctuation occurrences in the dataset are listed below.

- මෙක ‘හිනා ඉනා‘ වැඩසටහනේ වැඩක්ද දන්නේ නෑ.
- උක්කු අම්මට ජයවේවා -නිනි.නාපොලි.
- ඔනෑ වැරදි වැඩක් කරලා කියන්නේ “ඔක තමා කියන්නේ අපි නොදන්නා ගැණු මායම්“ කියලා.
- අන්තර්ජාලයට (යූ ටියුබ් එකට) ගියාම ඔය කාගේ කාගේත් දක්ෂතා ජනතාවට දැක බලාගන්න පුළුවන්.
- ලීලාරත්න කාරියවසම් (කපු අයියා,ලීලේ අයියා) මහතාගේ වියෝව කිංග්ස්බරි පන්සලට ආවගිය ගිහි පැවිදි හැම දෙනාටම විශාල කණගාටුවක්

3.1.3 Feature Selection

Preprocessed comments were transformed such a way that they are acceptable by machine learning estimators. This process is known as feature selection/extraction. In NLP literature we can find many feature extraction methods as discussed in previous chapters.

- Term presence – Bag of word model
- Term position
- N grams
- Term frequency and different IR weighting schemes
- Part-of-speech (POS) tags
- Opinion words and phrases
- Negations
- Syntactic dependency

- Word embedding

Term presence or mostly known as bag of word model is a simple and commonly used feature in natural language processing. It can be used as a baseline to compare other complex techniques with. Bag of word model identifies all the words present in a set of documents and consider each word as a feature. In the process it disregards the word position and any grammatical structure associated with the sentence. N-gram model can be considered as a direct extension of bag of word model which adds word position into the equation. Bigram is the most commonly used n-gram technique (n=2). In bigram instead of considering each word as a feature it considers each word pair as a feature, consequently retaining word order.

Words in a particular corpus will have varying importance depending on the context. Common words such as prepositions becomes dominant features in bag of word model. But these commonly occurring words carry less value with regard to our experiments. Therefore we could introduce word weighting schemes to diminish the importance of common words. Term Frequency – Inverse Document Frequency (TF-IDF) [59] is a commonly used word weighting scheme. As the name suggests it calculates the occurrence of words across documents and introduces a diminishing factor to common words.

Techniques we have discussed so far can be applied to a given set of documents without considering the language, syntactics or grammatical structure. On the other hand POS tags, syntactic dependencies, negations and opinion words/phrases consider unique features inherent in particular language. For our experiment to reduce complexity we used only the former set of features.

3.1.3.1 Word Embedding

Features we discussed so far have very high dimensions. Most of the machine learning algorithms do not conform well in high dimensional spaces. As an solution word embedding techniques have emerged, considering it as one of key breakthroughs on taking natural language processing problems. Word embedding [50][51] techniques transform 'one dimension for word' vectors into a much denser vector space with lower dimensionality. Most importantly in doing so it considers the word context and as a consequence words with similar context will be transformed to a similar representation. Word2Vec model based on statistical methods and deep learning techniques has become the most commonly used word embedding technique [51]. Therefore we decided to evaluate the effectiveness of Word2Vec features with other traditional features.

There are two involved steps of generating a Word2Vec feature set.

1. Generate Word2Vec model
2. Employing the model to transform features into Word2Vec representation

To generate effective Word2Vec model a large corpora is needed as the performance of the model is proportional to the number of learned embeddings. Generally a generated Word2Vec model can be reused across different domains [51]. Consequently we can find many pre-built Word2Vec models freely available. Unfortunately there isn't any model for Sinhala language. Therefore we had to start with building a Word2Vec model. Word2Vec model has two different techniques for learning word embeddings. They are continuous bag of word (CBOW) model and continuous skip-gram model [50]. While CBOW model learns by predicting the current word based on its context, skip-gram model learns by predicting the surrounding words given the current word [50].

We employed all 16000 news article comments in our dataset for generating the Word2Vec model. Python Gensim[4] library was used for generating the model. Both CBOW and skip-gram models were trained with similar parameters for comparison. Table 3.2 summarizes important parameters of learned models.

[4] https://radimrehurek.com/gensim/

Table 3.2: Word2Vec model properties

Property	Sample values	Description
Number of dimensions	50, 100, 300, 400, 1000	Word vector dimensionality (number of features)
Window size	10	Number of surrounding words to consider
Minimum word count	1	Remove word if did not occur this times
Down-sampling	0.001	Down-sampling for frequent words

Now we have trained Word2Vec models, next step is to generate word embedding features from the model. Model converts each word into a word vector. But our problem require us to generate feature vector per document. As to literature [44] [45], taking a summary statistics such as minimum, maximum or average of aggregate word vectors has proven easy and reasonably well performing technique. Moreover we can add a weighting scheme such as inverse document frequency for each word vector before generating summary statistic. Finally we can move even further by combining word vectors with sparse features such as bag of word or TF-IDF to generate more complex representation. We can list down word vector aggregation methods as follows.

- min, max
- average
- IDF weighted averages
- Word embeddings + bag of word
- Word embeddings + TF-IDF

Out of five aforementioned word vector aggregation methods we commissioned mean embedding and tf-idf weighted embedding techniques. Min and max were left out as average performs better in most cases [44][45]. Others were left as future enhancements.

3.1.4 Classification Algorithms

Our next objective is to select a suitable machine learning estimator for classifying comments. Here we concentrate only on supervised machine learning techniques which we identified in chapter 2. To measure the performance of classifiers, 10 fold cross validation was used throughout all experiments unless otherwise specified. Following is the list of algorithms/estimators which we commissioned for comment classification. In the following discussion we will look into advantages and configuration options of each of the selected estimators. Here we will purposefully refrain from discussing the internals of the algorithms.

- Naive Bayes
- Logistic Regression
- Decision Tree
- Random Forest
- SVM
- Convolution Neural Networks
- Recurrent Neural Networks

3.1.4.1 Naive Bayes

Naive Bayes is the classical starting point algorithm for many machine learning problems. Prominent features of Naive Bayes are [52] [53]

- Simplicity, easy to understand, implement and require less resources.
- Highly scalable, scales well with number of features.
- Can make probabilistic predictions.
- Can be used for both binary and multi-class classifications.
- Performs reasonably well compared to other complex algorithms.
- Require features to be independent. But in practice even-though independence assumption did not hold, works well.
- Can perform online updates to model

For our experiments we used Gaussian Naive Bayes model implemented in Python sklearn library. sklearn.GaussianNB expects dense input parameters, hence sparse vectorized data had to be converted before feeding into the estimator.

3.1.4.2 Logistic Regression

Logistic regression is an commonly used regression model. Prominent features of logistic regression are [52][53]

- Used for binary classifications. Can use for multi-class classification using one vs rest strategy.
- Variables don't need to be normally distributed.
- Can handle non linear effects.
- Require more data for better results

For our experiments we used logistic regression model implemented in Python sklearn library.

Important parameters

- Inverse of regularization strength: help to reduce overfitting
- Optimization method: liblinear, newtong-cg, sag, saga, lbfgs

3.1.4.3 Decision Tree

Decision Tree is a decision support tool mainly used in operational research. This is a popular machine learning algorithm because of its easily interpretable results. Important features of decision tree are [52][53]

- Simple, easy to understand and interpret
- A multi-class classification technique
- Can easily combine with other techniques
- Can become biased for certain features
- Can become complex in the presence of larger and interdependent features

For our experiments we used decision tree classifier implemented in Python sklearn library.

Important parameters

- Quality of splits: gini impurity, entropy
- Max depth of tree
- Number of features to consider for a split
- Minimum number of samples for a split

3.1.4.4 Random Forest

Random forest, a direct extension of decision tree is an ensemble learning technique. It creates multitude of decision trees and take the mode of the prediction of each decision tree. Important features of decision tree are [52][53]

- A multi-class classification technique
- Can easily combine with other techniques
- Minimizes the biases present in decision tree technique
- Losses simplicity and interpret-ability of decision tree technique
- Can become complex in the presence of larger and interdependent features
- We used random forest classifier implemented in Python sklearn library.

Important parameters

- Number of trees

3.1.4.5 Support Vector Machines

Support Vector Machines (SVM) are set of classification techniques well known for its superiority in natural language processing. Important features of SVM are [52] [53]

- Works well for even unstructured, semi-structured data
- Can select suitable kernel depending on the problem, though this could be a difficult task sometimes
- Effectively handles high dimensional spaces
- Used for binary classifications. Can use for multi-class classification using one vs rest strategy.
- Not a probabilistic method

- Quite a lot of parameters to tune, therefore highly versatile/customizable
- Difficult to understand and interpret

We used SVM classifier implemented in Python sklearn library.
Important parameters

- Penalty parameter for the error
- Kernel: linear, polynomial, radial basis function, sigmoid, precomputed
- Kernel coefficient and independent parameter in kernel

3.1.4.6 Convolution Neural Networks

Convolution Neural Network is a feed forward artificial neural network commonly used for image classification. Kim [46] has shown that CNN can outperform most of the existing text classification models. Important features of CNN are

Effectively capture the local dependencies (context) in features. Therefore performs well for data that exhibits locality such as image, text and time series data.

- High computational cost
- Need large set of training data
- Complex and not easily interpretable

Python tensorflow library was used to implement CNN model.

CNN expects each input to be a fixed size matrix. The data set consists of sentences with variable word count where each word is a fixed length vector. To generate fixed size matrix from this we defined a maximum word count considering the average and maximum sentence lengths. If actual sentence length falls shorter 0 s are appended at the end of the matrix and if actual length is greater than maximum we will prune the remaining of the sentence. Figure 3.7 depicts a sample input matrix to the CNN model. Example in the figure assumes the length of the matrix is 12. Since the sentence “තිසල් ඔබ හරි අපේ ඉංගිනේරු ප්‍රමිතය විවාදයට ලක් කල යුතුයි” does not have 12 words, zeros are appended to the last two positions of the matrix.

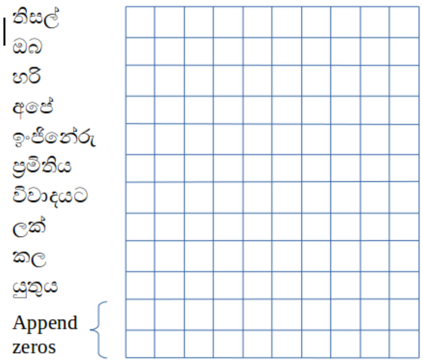


Figure 3.7: CNN input matrix

Figure 3.8 depicts the configuration of CNN model that we used for the classification.

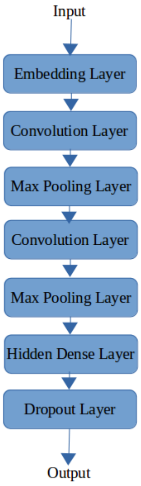


Figure 3.8: CNN model

3.2.3.7 Recurrent Neural Networks

Recurrent Neural Networks is an artificial neural network (with directed cycle) model commonly used for exploiting temporal qualities of a given dataset. In the literature [47][48][49] RNN and RNN-LSTM have been used extensively for language modeling, speech recognition and machine translation. RNN-LSTM is a special version of Recurrent Neural networks augmented using Long Short Term memory [60]. Important features of RNN are

- Performs well for data with temporal qualities, therefore has shown superior results in NLP tasks
- Always considers input with previous inputs
- Limited memory of input sequence. This can be minimized by introducing Long Short Term Memory (LSTM)

Python tensorflow[5] library was used to implement RNN model.

Though it is possible to feed variable length sentences to recurrent neural networks, to simplify the model we used input similar to CNN model which we discussed previously. Figure 3.9 depicts the configuration of RNN-LSTM model used for classification. The heart of the model is the LSTM layer which processes one word at a time to compute the probabilities of possible next words. LSTM layer was followed by a Dropout layer to prevent overfitting. Finally model contains RELU and Softmax activation functions to introduce nonlinearities and output a binary value.

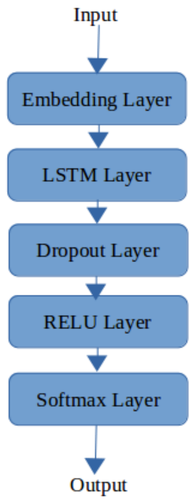


Figure 3.9: RNN model

[5] https://www.tensorflow.org/

3.2.3.8 CNN + SVM Hybrid Technique

Probability output of CNN model can be effectively used to augment the SVM classifier. This hybrid technique was successfully used by Jihan at el [58] for assigning multi-class labels for customer reviews of various domains. They have used output probabilities generated by CNN model in tandem with Word2Vec features as input to the SVM classifier. This hybrid model was able to outperform individual results of both CNN and SVM.

CHAPTER 4: SYSTEM EVALUATION

This chapter explores the experiment results comparing the performance of different features and machine learning estimators. Furthermore it compares experiment results with existing sentiment analysis research [19].

Objective of the research is to classify Sinhala news comments according to its sentiment values. This includes data cleaning/preprocessing, feature selection and machine learning algorithm selection. Final output depends on the each of the 3 variables mentioned above. Therefore experiments were carried out to select best technique in each variable class. After comparing initial experiment results with existing research, performance of different techniques are discussed in each of the subsequent subsections.

4.1 Baseline Experiment

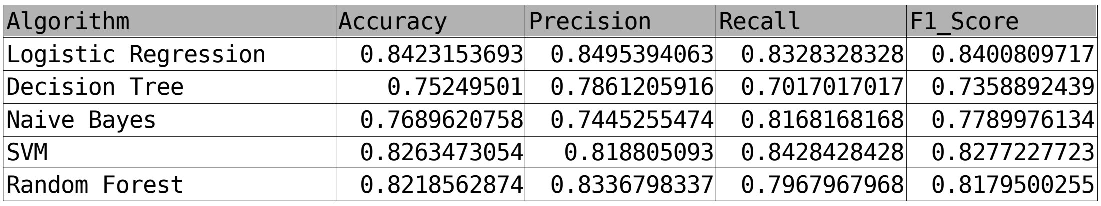
Medagoda et al [19] carried out Sinhala news comment sentiment analysis research using a pre-built lexicon. This is the only existing news comment analysis research in Sinhala language. Table 4.1 summarizes the results of Medagoda at al [19]. Since dataset of [19] is different from ours, two results are not directly comparable.

Table 4.1: Experiment results Medagoda et al [19]

Algorithm	Accuracy	Precision	F1_Score
Naïve Bayes	0.6	0.593	0.538
J48	0.58	0.581	0.578
SVM	0.56	0.541	0.412

Before diving into specifics, to get a feeling about the behavior of the dataset, a basic experiment was carried out by removing punctuations and using bag of word model. Results of the initial experiment are summarized in Table 4.2.

Table 4.2: Initial experiment results



Algorithm	Accuracy	Precision	Recall	F1_Score
Logistic Regression	0.8423153693	0.8495394063	0.8328328328	0.8400809717
Decision Tree	0.75249501	0.7861205916	0.7017017017	0.7358892439
Naive Bayes	0.7689620758	0.7445255474	0.8168168168	0.7789976134
SVM	0.8263473054	0.818805093	0.8428428428	0.8277227723
Random Forest	0.8218562874	0.8336798337	0.7967967968	0.8179500255

Results of the experiment were positive, outperforming the baseline research [19] in all of the selected algorithms. Out of selected five machine learning estimators Logistic Regression yielded best results with accuracy value of 84%. It was surprising to see Logistic Regression outperforming Support Vector Machines, the classical text classification technique. SVM and Random Forest had similar results while Decision Tree and Naive Bayes had much lower results. Even though [19] and our initial experiment both used Naive Bayes and SVM as classification algorithms, two results varies largely. Reason for this difference is either dataset or feature generation method. We cannot come to a clear conclusion until we try our dataset with the features used in [19].

4.2 Preprocessing - Effect of Punctuations

It is clear that punctuations carry special information in expressions [4]. While punctuations carry special information they can introduce noise and even confuse algorithms by meddling with generated features. Therefore it is important to experiment and identify the behavior of punctuations. However most of the past research have removed punctuation marks in preprocessing step [36][31] and some have removed selected set of punctuations to preserve context [39]. Others have replaced selected set of punctuations with special words [4]. For the purpose of identifying the effect of punctuations while keeping other variables constant, Logistic Regression and SVM were used with the TF-IDF features.

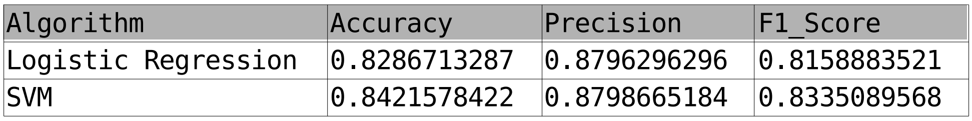
While skimming the dataset, following list of commonly occurring punctuations were identified.

- Full Stop (.)

- Comma (,)
- Exclamation mark (!)
- Question mark (?)
- Colon (:)
- Semicolon (;)
- Slash (/ \ )
- Brackets (( ) [ ])
- Quotation marks (“ ” ‘ ’)

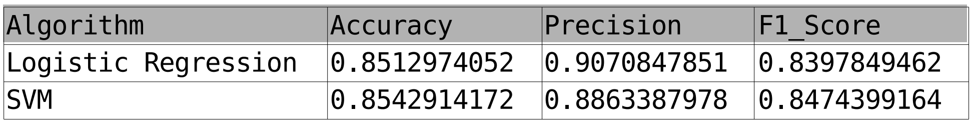
While some punctuations carry special information others just add noise to the data. By skimming the dataset period, exclamation and question marks were identified as important punctuations worth further analyzing. It was further noticed that question mark is mostly associated with comments bearing negative sentiment. Table 4.3 summarizes the results of the experiments conducted for investigating the importance of exclamation mark, question mark and punctuations as whole.

Table 4.3: Effect of punctuations



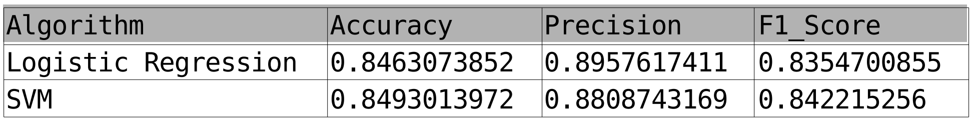
Algorithm	Accuracy	Precision	F1_Score
Logistic Regression	0.8286713287	0.8796296296	0.8158883521
SVM	0.8421578422	0.8798665184	0.8335089568

All punctuations intact



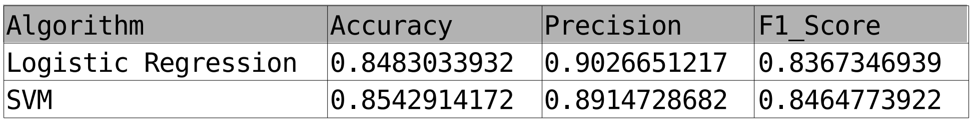
Algorithm	Accuracy	Precision	F1_Score
Logistic Regression	0.8512974052	0.9070847851	0.8397849462
SVM	0.8542914172	0.8863387978	0.8474399164

All punctuations removed



Algorithm	Accuracy	Precision	F1_Score
Logistic Regression	0.8463073852	0.8957617411	0.8354700855
SVM	0.8493013972	0.8808743169	0.842215256

All punctuations removed except exclamation mark



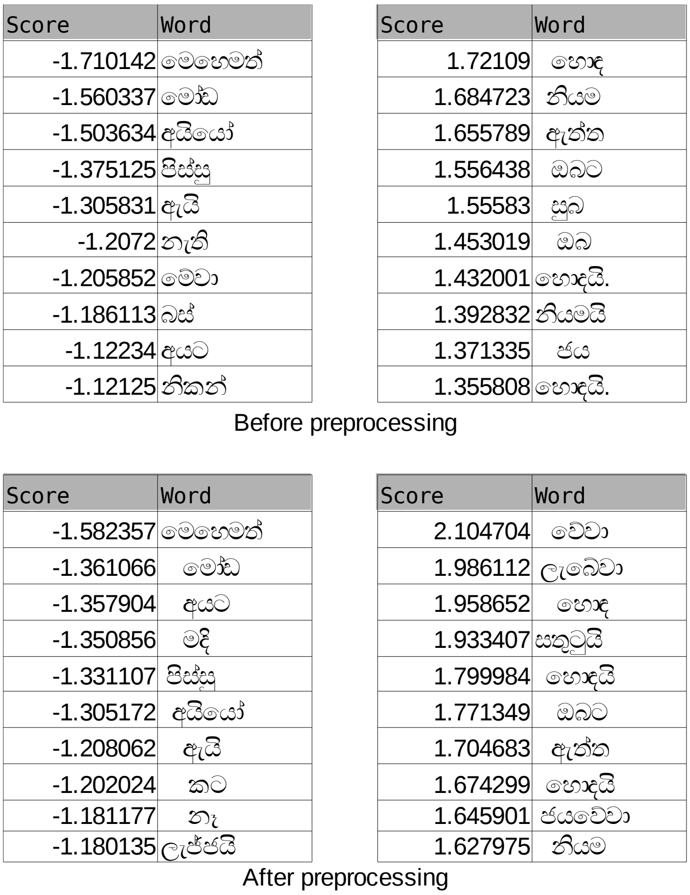
Algorithm	Accuracy	Precision	F1_Score
Logistic Regression	0.8483033932	0.9026651217	0.8367346939
SVM	0.8542914172	0.8914728682	0.8464773922

All punctuations removed except question mark

Clearly removing punctuation marks increased classification performance. This is prominent in Logistic Regression classifier resulting 2.3% increase in accuracy. Even Though exclamation mark and question marks were expected to have increased the classification performance, results do not confirm this assumption. Leaving exclamation mark in dataset lowered classification performance slightly in both cases. In case of Logistic Regression all the 3 metrics have dropped slightly when question mark was left intact. But in SVM a slight increase in precision and F1 score was observed while accuracy did not change.

To further understand how this preprocessing step affect features, most contributing 10 features in both positive and negative categories were analyzed. Table 4.4 lists down the features with high sentiment scores as identified by Logistic Regression classifier.

Table 4.4: Features with highest sentiment orientation (Logistic Regression)



Score	Word
-1.710142	මෙහෙමත්
-1.560337	මෝඩ
-1.503634	අයියෝ
-1.375125	පිස්සු
-1.305831	ඇයි
-1.2072	නැති
-1.205852	මෙවා
-1.186113	බස්
-1.12234	අයට
-1.12125	නිකන්

Score	Word
1.72109	හොඳ
1.684723	නියම
1.655789	ඇත්ත
1.556438	ඔබට
1.55583	සුබ
1.453019	ඔබ
1.432001	හොඳයි.
1.392832	නියමයි
1.371335	ජය
1.355808	හොඳයි.

Before preprocessing

Score	Word
-1.582357	මෙහෙමත්
-1.361066	මෝඩ
-1.357904	අයට
-1.350856	මදි
-1.331107	පිස්සු
-1.305172	අයියෝ
-1.208062	ඇයි
-1.202024	කට
-1.181177	නෑ
-1.180135	ලැජ්ජයි

Score	Word
2.104704	වෙවා
1.986112	ලැබෙවා
1.958652	හොඳ
1.933407	සතුටුයි
1.799984	හොඳයි
1.771349	ඔබට
1.704683	ඇත්ත
1.674299	හොඳයි
1.645901	ජයවෙවා
1.627975	නියම

After preprocessing

Both tables before and after preprocessing contains similar words. By further analyzing it can be seen that some words are in the before preprocessing list with Full Stop intact. Therefore when words such as "හොඳයි." and "හොඳයි" were encountered by the classifier they will be treated as two different words. After preprocessing list consider both as same and therefore the word "හොඳයි" has increased sentiment score in the list. Further, the word with highest sentiment score in the second list ("වෙවා") is not present in the first list. This is also due to association of exclamation mark with the word in most of the cases. Following examples shows few occurrences of word "වෙවා".

- ඔබතුමාට නිවන් සුව අත් වෙවා.
- ඔබට නිවන් සුව අත්වෙවා !!!
- ඔබ වහන්සෙට පිං. දීරිඝායු වෙවා..!
- ඔබට පින්සිදු වෙවා!

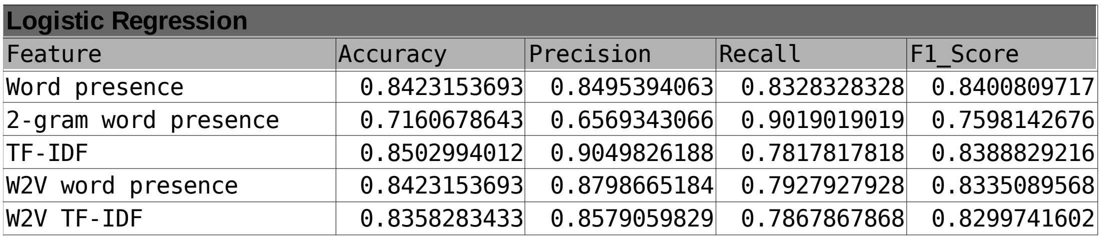
4.3 Features

In this subsection performance of selected features are measured against each machine learning algorithm. Since removing all punctuations yielded best result in previous experiment, dataset used in this section does not contain any punctuations. Performance of the classifier algorithm is highly sensitive to selected features, therefore all the classifiers are evaluated against the given features. Neural Network classifiers are considered separately in a later section. Following list of features are evaluated in this section

- Term presence – Bag of word model
- 2 gram word presence
- Term frequency - inverse document frequency
- Word embedding (Word2Vec: CBOW and skip-gram)

Logistic Regression was the best performing model so far. Therefore it was evaluated first with different available features. Table 4.5 summarizes the experiment results

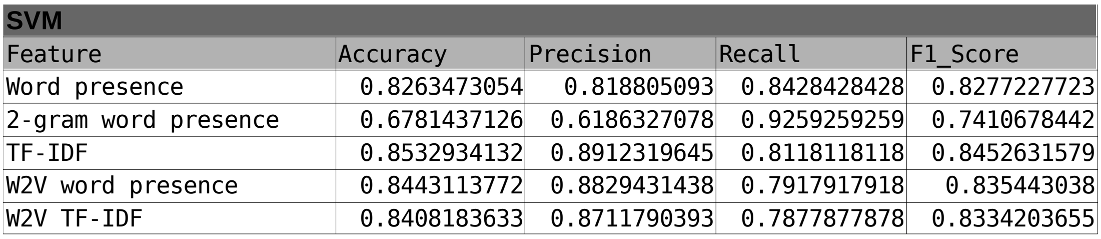
Table 4.5: Logistic Regression classifier



Logistic Regression				
Feature	Accuracy	Precision	Recall	F1_Score
Word presence	0.8423153693	0.8495394063	0.8328328328	0.8400809717
2-gram word presence	0.7160678643	0.6569343066	0.9019019019	0.7598142676
TF-IDF	0.8502994012	0.9049826188	0.7817817818	0.8388829216
W2V word presence	0.8423153693	0.8798665184	0.7927927928	0.8335089568
W2V TF-IDF	0.8358283433	0.8579059829	0.7867867868	0.8299741602

According to the results, tf-idf feature model has the highest accuracy and precision values and bag of word model has highest f1 score. Except for the 2-gram word presence model, all of the other features have good performance. Table 4.6 summarizes similar experiment results for SVM classifier.

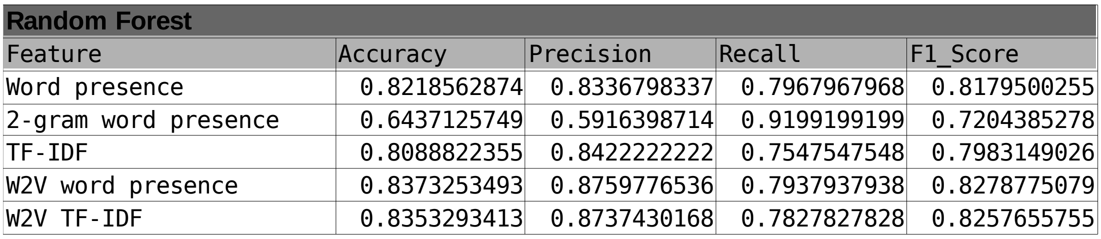
Table 4.6: SVM classifier



SVM				
Feature	Accuracy	Precision	Recall	F1_Score
Word presence	0.8263473054	0.818805093	0.8428428428	0.8277227723
2-gram word presence	0.6781437126	0.6186327078	0.9259259259	0.7410678442
TF-IDF	0.8532934132	0.8912319645	0.8118118118	0.8452631579
W2V word presence	0.8443113772	0.8829431438	0.7917917918	0.835443038
W2V TF-IDF	0.8408183633	0.8711790393	0.7877877878	0.8334203655

In the case of SVM classifier tf-idf feature model is the clear winner. Similar to previous experiment, except for the 2-gram features all others have good results. Table 4.7 summarizes the experiment results for Random Forest classifier.

Table 4.7: Random Forest classifier



Random Forest				
Feature	Accuracy	Precision	Recall	F1_Score
Word presence	0.8218562874	0.8336798337	0.7967967968	0.8179500255
2-gram word presence	0.6437125749	0.5916398714	0.9199199199	0.7204385278
TF-IDF	0.8088822355	0.8422222222	0.7547547548	0.7983149026
W2V word presence	0.8373253493	0.8759776536	0.7937937938	0.8278775079
W2V TF-IDF	0.8353293413	0.8737430168	0.7827827828	0.8257655755

Differing from the previous experiments, Word2Vec word presence model has the highest performance in Random Forest classifier. Again 2-gram features display poor performance. Table 4.8 and 4.9 summarizes the experiment results for Naive Bayes and Decision Tree classifiers.

Table 4.8: Naive Bayes classifier

Feature	Accuracy	Precision	F1_Score
Word presence	0.7689620758	0.7445255474	0.7789976134
2-gram word presence	0.7335329341	0.6744186047	0.7710120069
TF-IDF	0.7579840319	0.7471153846	0.7621383031
W2V word presence	0.7769461078	0.8407407407	0.7529021559
W2V TF-IDF	0.7729540918	0.820754717	0.753654575

Table 4.9: Decision Tree classifier

Feature	Accuracy	Precision	F1_Score
Word presence	0.75249501	0.7861205916	0.7358892439
2-gram word presence	0.6397205589	0.5896440129	0.7161949686
TF-IDF	0.751996008	0.760373444	0.7468160978
W2V word presence	0.7654690619	0.7611056269	0.7664015905
W2V TF-IDF	0.7579840319	0.7524557957	0.759543877

While Naive Bayes and Decision Tree have poor performance compared to other three models they are displayed for comparison purpose.

Even though it was expected Word2Vec features to have better results than simple word presence or tf-idf models results do not confirm this assumption hundred percent. Therefore following subsection will investigate the behavior of Word2Vec features against different parameters.

4.3.1 Word Embedding: Word2Vec

Utilizing all available articles, a Word2Vec model was generated. Table 4.10 shows the effectiveness of the Word2Vec model. It lists down the most similar 10 words for the selected words නැහැ, හොඳයි, පිස්සු, වෙවා and ඔබට. We can see that Word2Vec was able to effectively capture different forms of the given words. Moreover some words similar in meaning are also present in the list. Some words that are not directly connected are also present (Eg. පිස්සු and විහිළු). By analyzing Table 4.10 we can deduce that generated model is quite effective in reducing the feature dimensionality and identifying the context of words. By increasing the size of the trained dataset we can improve the model further.

Table 4.10: Word2Vec model - Similar words

නැහැ	ඔබට	වේවා	පිස්සු,	හොඳයි
නැහැ,	ඔබතුමාට	වේවා!	පිස්සු!	හොඳයි
නැ	ඔබලාට	වේවා!!	පිස්සු,	හොඳයි,
නැහැ!	ඔයාට	වේවා	කෙලින්නේ	හොඳා
නෑ	ඔබට	පතමි	විකාර	නරකද?
නැනේ	ඔබතුමියට	වේවායි	විහිලු	හොඳා
නැතිලු	තුමාට	වේවා!!!	මටනං	හොඳ?
නෑනේ	ඔබටත්	වේවා	කෝලම්	හොඳ?
නැහැනේ	මහතාට	යහපතක්ම	කලන්තේ	හොඳයි,
නැහැනෙ	දෙපලට	වේවා!!!!	පව	හොඳය
නැහැ?	ඔබතුමන්ට	සරණින්	කෙළින්නේ	ගුණයි

Continuous bag of word and skip gram are the two models Word2Vec can be trained. After evaluating the performance of each model, further experiments were carried out to measure how the vector size affects the classification performance. Logistic Regression, SVM and Random Forest classifiers were used for the experiment and dataset with all the punctuation marks was used to preserve the context. Table 4.11 and 4.12 shows the output of the continuous bag of word and skip gram model respectively.

Table 4.11: W2V Continuous Bag of Word model (CBOW)

Algorithm	Accuracy	Precision	Recall	F1_Score
Logistic Regression	0.8443113772	0.8820912125	0.7927927928	0.8356164384
SVM	0.8463073852	0.8886389201	0.7917917918	0.8368644068
Random Forest	0.8438123753	0.8785871965	0.7937937938	0.8356955381

Table 4.12: W2V Skip gram model

Algorithm	Accuracy	Precision	Recall	F1_Score
Logistic Regression	0.8488023952	0.9084507042	0.7727727728	0.8363047002
SVM	0.8617764471	0.9237089202	0.7877877878	0.8503511615
Random Forest	0.8473053892	0.8888888889	0.7927927928	0.8380952381

Results show the superior performance of skip gram model over continuous bag of word model. The difference is more apparent in SVM classifier resulting 1.5%

increase in accuracy and 3.5% increase in precision. It was mentioned in literature [50] that skip gram model works better for smaller datasets than CBOW model and skip gram was able to predict rare words better than CBOW. Our results further confirms this.

Table 4.13 shows how performance varies with the dimension of the word vectors. For the experiment 200, 300, 400 and 1000 sized vectors were generated and word count and tf-idf features were used.

Table 4.13: W2V vector dimension and features

Vector dimension	Accuracy word count	Accuracy tf-idf
200	0.8433133733	0.8512974052
300	0.8463073852	0.8383233533
400	0.8448103792	0.8418163673
1000	0.8483033932	0.8398203593

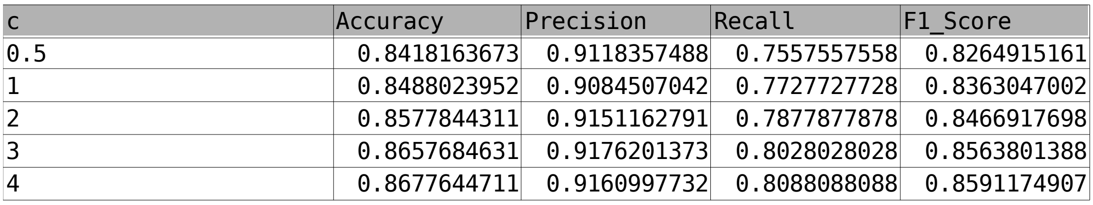
A clear conclusion cannot be drawn from the table. In the case of word count features, accuracy increased slightly with the vector dimension. As training time increases with dimension, vector dimension of 1000 cannot be justified as the best alternative. Since vector dimension of 300 is common among other research [50] [51], we decided to use the same for further experiments.

4.4 Classification Algorithms

In this subsection performance of different machine learning estimators are discussed. In the previous subsection tf-idf and Word2Vec skip gram word count features were identified as best features. Superior performance of SVM over other algorithms was also observed while Logistic Regression and Random forest closely following SVM. Therefore each of the three algorithms were further tuned to increase performance. Table 4.14, 4.15 and 4.16 respectively shows the results obtained by tuning parameters.

Table 4.14 lists the results obtained while changing the inverse of regularization strength (c) of Logistic Regression model.

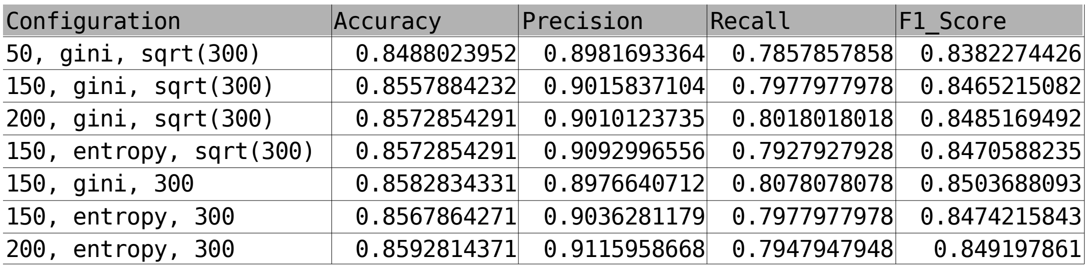
Table 4.14: Logistic Regression parameter tuning (c value)



c	Accuracy	Precision	Recall	F1_Score
0.5	0.8418163673	0.9118357488	0.7557557558	0.8264915161
1	0.8488023952	0.9084507042	0.7727727728	0.8363047002
2	0.8577844311	0.9151162791	0.7877877878	0.8466917698
3	0.8657684631	0.9176201373	0.8028028028	0.8563801388
4	0.8677644711	0.9160997732	0.8088088088	0.8591174907

Table 4.15 shows performance metrics while changing number of features, criterion and max features of Random Forest model.

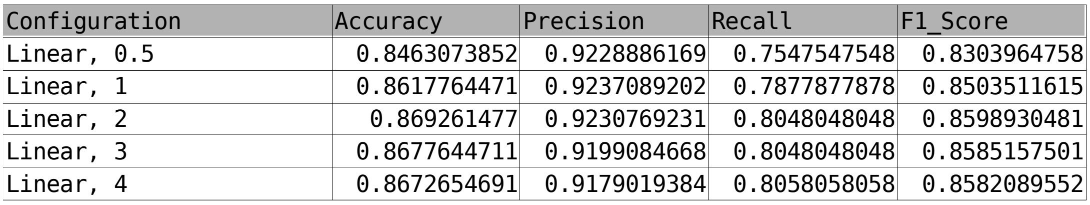
Table 4.15: Random Forest parameter tuning (# features, criterion, max features)



Configuration	Accuracy	Precision	Recall	F1_Score
50, gini, sqrt(300)	0.8488023952	0.8981693364	0.7857857858	0.8382274426
150, gini, sqrt(300)	0.8557884232	0.9015837104	0.7977977978	0.8465215082
200, gini, sqrt(300)	0.8572854291	0.9010123735	0.8018018018	0.8485169492
150, entropy, sqrt(300)	0.8572854291	0.9092996556	0.7927927928	0.8470588235
150, gini, 300	0.8582834331	0.8976640712	0.8078078078	0.8503688093
150, entropy, 300	0.8567864271	0.9036281179	0.7977977978	0.8474215843
200, entropy, 300	0.8592814371	0.9115958668	0.7947947948	0.849197861

Table 4.16 shows the metrics while changing c value of SVM model. Only 'Linear' kernel had good results. Therefore results of other kernels such as rbf, poly and sigmoid are not displayed.

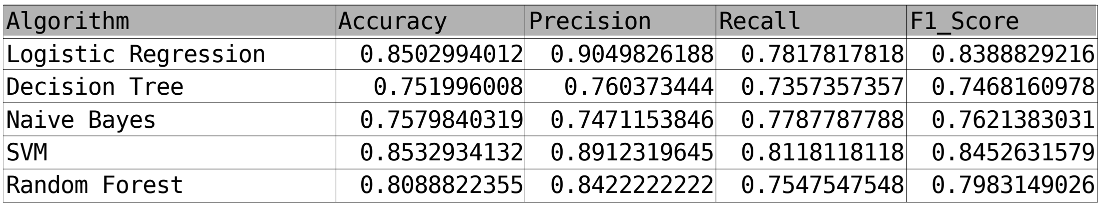
Table 4.16: SVM parameter tuning (kernel, c value)



Configuration	Accuracy	Precision	Recall	F1_Score
Linear, 0.5	0.8463073852	0.9228886169	0.7547547548	0.8303964758
Linear, 1	0.8617764471	0.9237089202	0.7877877878	0.8503511615
Linear, 2	0.869261477	0.9230769231	0.8048048048	0.8598930481
Linear, 3	0.8677644711	0.9199084668	0.8048048048	0.8585157501
Linear, 4	0.8672654691	0.9179019384	0.8058058058	0.8582089552

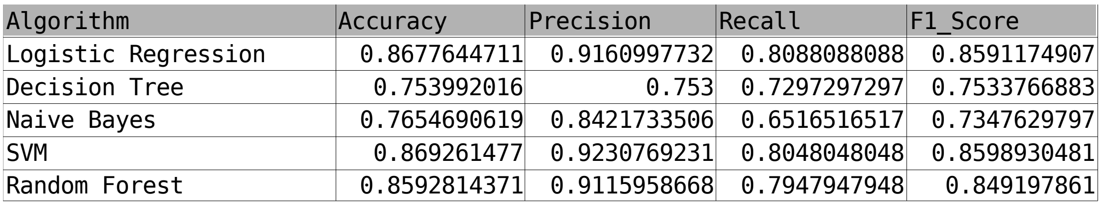
Table 4.17 and 4.18 exhibit the best results of each of the algorithms.

Table 4.17: TF-IDF features



Algorithm	Accuracy	Precision	Recall	F1_Score
Logistic Regression	0.8502994012	0.9049826188	0.7817817818	0.8388829216
Decision Tree	0.751996008	0.760373444	0.7357357357	0.7468160978
Naive Bayes	0.7579840319	0.7471153846	0.7787787788	0.7621383031
SVM	0.8532934132	0.8912319645	0.8118118118	0.8452631579
Random Forest	0.8088822355	0.8422222222	0.7547547548	0.7983149026

Table 4.18: W2V skip gram with word count features



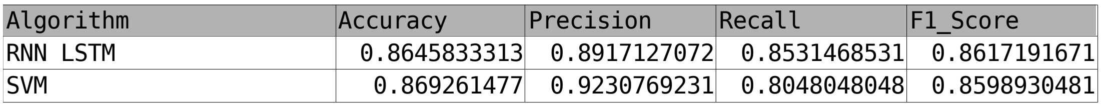
Algorithm	Accuracy	Precision	Recall	F1_Score
Logistic Regression	0.8677644711	0.9160997732	0.8088088088	0.8591174907
Decision Tree	0.753992016	0.753	0.7297297297	0.7533766883
Naive Bayes	0.7654690619	0.8421733506	0.6516516517	0.7347629797
SVM	0.869261477	0.9230769231	0.8048048048	0.8598930481
Random Forest	0.8592814371	0.9115958668	0.7947947948	0.849197861

SVM with Word2Vec (skip gram word count) model produced best results with accuracy 87%, precision 92% and f1 score 86%. Next subsection attempts to further increase performance using deep learning techniques.

4.4.1 Recurrent Neural Networks

In this subsection performance of Recurrent Neural Networks are evaluated. Word2Vec skip gram features were used for the experiment. Table 4.19 present the results of RNN with LSTM module. SVM results were also displayed alongside for the comparison. Table 4.20 and Table 4.21 shows confusion matrix of SVM and RNN-LSTM respectively. Because of the long learning time of RNN, holdout method was used instead of cross validation. Data set was divided into 3:2 ratio for training and testing. For comparison purpose SVM was also tested with similar configuration.

Table 4.19: RNN-LSTM with W2V skip gram word count features



Algorithm	Accuracy	Precision	Recall	F1_Score
RNN LSTM	0.8645833313	0.8917127072	0.8531468531	0.8617191671
SVM	0.869261477	0.9230769231	0.8048048048	0.8598930481

Table 4.20: Confusion matrix (SVM with W2V skip gram word count)

		Prediction	
		POSITIVE	NEGATIVE
Actual	POSITIVE	787	212
	NEGATIVE	65	940

Table 4.21: Confusion matrix (RNN-LSTM with W2V skip gram word count)

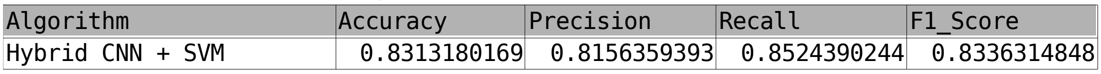
		Prediction	
		POSITIVE	NEGATIVE
Actual	POSITIVE	885	108
	NEGATIVE	152	775

RNN-LSTM displays very similar results to SVM, slightly beating SVM in recall and in f1 score while SVM has higher accuracy than RNN.

4.4.2 Hybrid CNN + SVM

Hybrid model used by Jihan et al [57] was evaluated using Word2Vec skip gram features. Model had promising results, but was not able to outperform RNN-LSTM. Results are displayed in Table 4.22.

Table 4.22: Hybrid CNN + SVM with Word2Vec



Algorithm	Accuracy	Precision	Recall	F1_Score
Hybrid CNN + SVM	0.8313180169	0.8156359393	0.8524390244	0.8336314848

4.5 Error Analysis

In this subsection we briefly examine few misclassified instances. Table 4.23 list downs few misclassified instances. Since accompanying news items are large they are not displayed here.

Table 4.23: Misclassified instances

Index	Comment	Label	Prediction
1	නිමල් වගේ අය තමයි පොල් ගෙඩියටත් උසාවි යන්නේ මිනිසුන් ගැන ඔයිට වඩා හිතන්න ඕන අපි කවුරුත් යනකොට මොනවත් අරන් යන්නේ නෑ ඉන්නකං අනිත් අයට උපකාර කරන්න ඕන මෙක ආදර්ශයට ගත යුතු ක්‍රියාවක්	POSITIVE	NEGATIVE
2	ඔන්න රටකට ගිය කල	NEGATIVE	POSITIVE
3	මෙන්න ලංකාව උතුම් යැයි මොරදෙන්නන්ට හොද කණේ පහරක්	NEGATIVE	POSITIVE
4	හිත හොද වැඩි උනාමත් ප්‍රශ්න	NEGATIVE	POSITIVE
5	ලංකාවෙ කාන්තා පොලිස් කොස්තාපල් වරියන් දැක්කම හිනා යනවා කෙලින් හිට ගන්න පන නැහැ වගේ	NEGATIVE	POSITIVE
6	සාධු සාධු සාධු	POSITIVE	NEGATIVE
7	සාදු සාදු	POSITIVE	POSITIVE

Comment 1 does not contain any specific word with strong sentiment value. To correctly classify such a sample aspect level analysis might be needed.

Comments 3 and 4 contain words that have positive sentiment as well as negative sentiment. But word “උතුම්” in comment 3 and “හොද” in comment 4 have strong positive sentiment. They have contributed much to the final prediction.

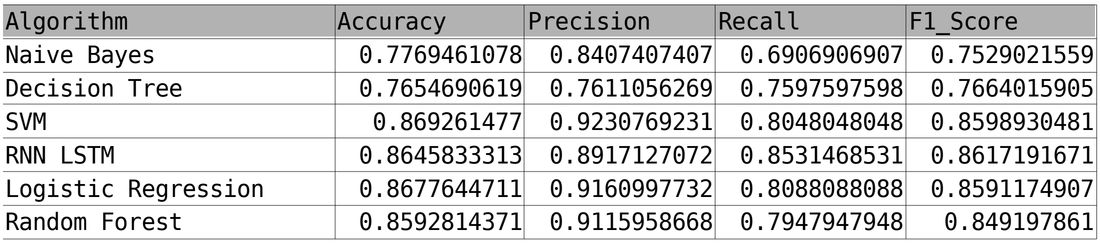
Though last item is not a misclassified comment, it is listed here to compare with item number 6. Even though two items have similar words they differ from spellings. Word “සාදු” is common among many comments and it is identified as an positive feature. But word “සාධු” was not available at the training time therefore model had trouble assigning a sentiment score to it.

By examining the misclassification instances we can conclude that to further increase performance of the models we need to consider word context instead of only considering individual word features. Further by incorporating spell correcting step we could improve results.

4.6 Summary

Before concluding the evaluation section, best results obtained so far are displayed in Table 2.24. All of the algorithm categories have better results than the baseline research, SVM and RNN-LSTM having the best results.

Table 4.24: Best results of each of the algorithm



Algorithm	Accuracy	Precision	Recall	F1_Score
Naive Bayes	0.7769461078	0.8407407407	0.6906906907	0.7529021559
Decision Tree	0.7654690619	0.7611056269	0.7597597598	0.7664015905
SVM	0.869261477	0.9230769231	0.8048048048	0.8598930481
RNN LSTM	0.8645833313	0.8917127072	0.8531468531	0.8617191671
Logistic Regression	0.8677644711	0.9160997732	0.8088088088	0.8591174907
Random Forest	0.8592814371	0.9115958668	0.7947947948	0.849197861

CHAPTER 5: CONCLUSION

This research focused on developing a better sentiment classification process for online Sinhala news comments. Since the dataset for such a process does not exists for Sinhala language, a dataset was compiled as the first step. By employing a web crawler, close to 16,000 news articles were collected from www.lankadeepa.com. More than 5000 comments were annotated with one of four labels, POSITIVE, NEGATIVE, NEUTRAL or CONFLICT. Only POSITIVE and NEGATIVE annotated comments were used in the classification process.

Preprocessing step mostly consisted of removing unnecessary whitespaces, repeated characters and punctuation marks. Even though exclamation mark and question marks carry a special sentiment in comments, experiment results did not confirm this. Therefore final conclusion is to remove all punctuations.

Common features available for text classification such as term frequency, 2-gram term frequency and tf-idf features were tested against five different machine learning estimators. In most cases tf-idf features outperformed others. Out of five different machine learning estimators SVM, Logistic Regression and Random Forest had most promising results.

Since word embedding features have shown much promise recently we decided to measure the classification performance using Word2Vec. Similar to previous research we noticed skip-gram model of Word2Vec performed better than Bag of word model. Out of six algorithms used with Word2Vec mean embedding, RNN had the best results while SVM closely followed.

As seen in the previous research for English, Word2Vec performed really well for Sinhala as well. From this we can conclude that Word2Vec can be effectively used without using any features specific to the language been tested. This greatly simplifies the text analysis process for low resourced languages such as Sinhala. Moreover superior performance of RNN confirms the temporal qualities of Sinhala language and advantages of RNN in natural language processing.

In this research we were able to develop a complete system from data gathering to sentiment classification of Sinhala news comments. We were able to prove the effectiveness of the word embedding features and performance of SVM and RNN in general text classification tasks. We have also contributed a dataset for the community which can be used for further Sinhala text analysis.

5.1 Future Work

Following list exhibits a list of possible future enhancements and improvements to the discussed model.

- News article, article related metadata, comment related metadata were not used for the classification task. They can be used appropriately to improve the performance.
- Further preprocessing the data such as applying lemmatization, stemming and stopword list can improve the performance.
- Different word embedding aggregation methods and combination of word embedding and traditional features can be experimented.
- Features can be further enriched by using a lexicon of word sentiments, similar to Medagoda et al [19]
- Current dataset contains large number of comments which were not annotated. A larger dataset will enable many new avenues.

REFERENCES

[1] Moreo, Alejandro, M. Romero, J. L. Castro, and Jose Manuel Zurita. "Lexicon-based comments-oriented news sentiment analyzer system." *Expert Systems with Applications* 39, no. 10 (2012): 9166-9180.

[2] Ma, Tengfei, and Xiaojun Wan. "Opinion target extraction in Chinese news comments." In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 782-790. Association for Computational Linguistics, 2010.

[3] Liu, Bing. "Sentiment analysis." Invited talk at the 5th Annual Text Analytics Summit (2009).

[4] Turney, Peter D. "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews." In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 417-424. Association for Computational Linguistics, 2002.

[5] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79-86. Association for Computational Linguistics, 2002.

[6] Kim, Soo-Min, and Eduard Hovy. "Automatic identification of pro and con reasons in online reviews." In *Proceedings of the COLING/ACL on Main conference poster sessions*, pp. 483-490. Association for Computational Linguistics, 2006.

[7] Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis."*Foundations and Trends® in Information Retrieval*2, no. 1–2 (2008): 1-135.

[8] Liu, Bing. "Sentiment analysis." Invited talk at the 5th Annual Text Analytics Summit (2009).

[9] Wilks, Yorick, and Mark Stevenson. "The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation."*Natural Language Engineering* 4, no. 2 (1998): 135-143.

[10] Das, Sanjiv, and Mike Chen. "Yahoo! for Amazon: Extracting market sentiment from stock message boards." In *Proceedings of the Asia Pacific finance association annual conference (APFA)*, vol. 35, p. 43. 2001.

[11] Hu, Minqing, and Bing Liu. "Mining and summarizing customer reviews." In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168-177. ACM, 2004.

[12] Kim, Soo-Min, and Eduard Hovy. "Determining the sentiment of opinions." In *Proceedings of the 20th international conference on Computational Linguistics*, p. 1367. Association for Computational Linguistics, 2004.

[13] Kamps, Jaap, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. "Using WordNet to Measure Semantic Orientations of Adjectives." In *LREC*, vol. 4, pp. 1115-1118. 2004.

[14] Adreevskaia, Alina, and Sabine Bergler. "Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses." In *11th*

conference of the European chapter of the Association for Computational Linguistics. 2006.

[15] Hatzivassiloglou, Vasileios, and Kathleen R. McKeown. "Predicting the semantic orientation of adjectives." In *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics*, pp. 174-181. Association for Computational Linguistics, 1997.

[16] Yu, Hong, and Vasileios Hatzivassiloglou. "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences." In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 129-136. Association for Computational Linguistics, 2003.

[17] Kanayama, Hiroshi, and Tetsuya Nasukawa. "Fully automatic lexicon expansion for domain-oriented sentiment analysis." In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pp. 355-363. Association for Computational Linguistics, 2006.

[18] Ding, Xiaowen, Bing Liu, and Philip S. Yu. "A holistic lexicon-based approach to opinion mining." In *Proceedings of the 2008 international conference on web search and data mining*, pp. 231-240. ACM, 2008.

[19] Medagoda, Nishantha, Subana Shanmuganathan, and Jacqueline Whalley. "Sentiment lexicon construction using sentiwordnet 3.0." In *Natural Computation (ICNC), 2015 11th International Conference on*, pp. 802-807. IEEE, 2015.

[20] Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." In *LREC*, vol. 10, no. 2010, pp. 2200-2204. 2010.

[21] Gamon, Michael, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. "Pulse: Mining customer opinions from free text." In *international symposium on intelligent data analysis*, pp. 121-132. Springer, Berlin, Heidelberg, 2005.

[22] Blitzer, John, Mark Dredze, and Fernando Pereira. "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification." In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440-447. 2007.

[23] He, Yulan, Chenghua Lin, and Harith Alani. "Automatically extracting polarity-bearing topics for cross-domain sentiment classification." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 123-131. Association for Computational Linguistics, 2011.

[24] Banea, Carmen, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. "Multilingual subjectivity analysis using machine translation." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 127-135. Association for Computational Linguistics, 2008.

[25] Wei, Bin, and Christopher Pal. "Cross lingual adaptation: an experiment on sentiment classifications." In *Proceedings of the ACL 2010 conference short papers*, pp. 258-262. Association for Computational Linguistics, 2010.

[26] Duh, Kevin, Akinori Fujino, and Masaaki Nagata. "Is machine translation ripe for cross-lingual sentiment classification?." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 429-433. Association for Computational Linguistics, 2011.

[27] Polanyi, Livia, and Annie Zaenen. "Contextual valence shifters." In *Computing attitude and affect in text: Theory and applications*, pp. 1-10. Springer, Dordrecht, 2006.

[28] Jia, Lifeng, Clement Yu, and Weiyi Meng. "The effect of negation on sentiment analysis and retrieval effectiveness." In *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 1827-1830. ACM, 2009.

[29] Davidov, Dmitry, Oren Tsur, and Ari Rappoport. "Semi-supervised recognition of sarcastic sentences in twitter and amazon." In *Proceedings of the fourteenth conference on computational natural language learning*, pp. 107-116. Association for Computational Linguistics, 2010.

[30] Fan, Wen, and Shutao Sun. "Sentiment classification for online comments on Chinese news." In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, vol. 4, pp. V4-740. IEEE, 2010.

[31] Zhang, Dongwen, Hua Xu, Zengcai Su, and Yunfeng Xu. "Chinese comments sentiment classification based on word2vec and SVMperf."*Expert Systems with Applications*42, no. 4 (2015): 1857-1863.

[32] Yussupova, N., Diana Bogdanova, and M. Boyko. "Applying of sentiment analysis for texts in russian based on machine learning approach." In *Proceedings of Second International Conference on Advances in Information Mining and Management*, pp. 8-14. 2012.

[33] Pak, A., and P. Paroubek. "Language independent approach to sentiment analysis."*Komp'uternaya Lingvistika i Intellektualnie Tehnologii: po materialam ezhegodnoy mezhdunarodnoy konferencii "Dialog* 11, no. 18 (2011): 37-50.

[34] Joshi, Aditya, A. R. Balamurali, and Pushpak Bhattacharyya. "A fall-back strategy for sentiment analysis in hindi: a case study."*Proceedings of the 8th ICON* (2010).

[35] Bakliwal, Akshat, Piyush Arora, and Vasudeva Varma. "Hindi subjective lexicon: A lexical resource for hindi polarity classification." In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, pp. 1189-1196. 2012.

[36] Lin, Chenghua, and Yulan He. "Joint sentiment/topic model for sentiment analysis." In *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 375-384. ACM, 2009.

[37] Turney, Peter D., and Michael L. Littman. "Unsupervised learning of semantic orientation from a hundred-billion-word corpus."*arXiv preprint cs/0212012* (2002).

[38] Hatzivassiloglou, Vasileios, and Kathleen R. McKeown. "Predicting the semantic orientation of adjectives." In *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth*

conference of the european chapter of the association for computational linguistics, pp. 174-181. Association for Computational Linguistics, 1997.

[39] Pak, Alexander, and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining." In *LREc*, vol. 10, no. 2010. 2010.

[40] Lafferty, John, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." (2001).

[41] Gabrilovich, Evgeniy, and Shaul Markovitch. "Computing semantic relatedness using wikipedia-based explicit semantic analysis." In *IJcAI*, vol. 7, pp. 1606-1611. 2007.

[42] Zhang, Li, Yue Pan, and Tong Zhang. "Focused named entity recognition using machine learning." In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 281-288. ACM, 2004.

[43] Diakopoulos, Nicholas, and Mor Naaman. "Topicality, time, and sentiment in online news comments." In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pp. 1405-1410. ACM, 2011.

[44] De Boom, Cedric, Steven Van Canneyt, Thomas Demeester, and Bart Dhoedt. "Representation learning for very short texts using weighted word embedding aggregation."*Pattern Recognition Letters* 80 (2016): 150-156.

[45] Clinchant, Stéphane, and Florent Perronnin. "Aggregating continuous word embeddings for information retrieval." In *Proceedings of the workshop on continuous vector space models and their compositionality*, pp. 100-109. 2013.

[46] Kim, Yoon. "Convolutional neural networks for sentence classification."*arXiv preprint arXiv:1408.5882* (2014).

[47] Lai, Siwei, Liheng Xu, Kang Liu, and Jun Zhao. "Recurrent Convolutional Neural Networks for Text Classification." In *AAAI*, vol. 333, pp. 2267-2273. 2015.

[48] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory."*Neural computation* 9, no. 8 (1997): 1735-1780.

[49] Liu, Pengfei, Xipeng Qiu, and Xuanjing Huang. "Recurrent neural network for text classification with multi-task learning."*arXiv preprint arXiv:1605.05101* (2016).

[50] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space."*arXiv preprint arXiv:1301.3781* (2013).

[51] Mnih, Andriy, and Geoffrey E. Hinton. "A scalable hierarchical distributed language model." In *Advances in neural information processing systems*, pp. 1081-1088. 2009.

[52] Qiu, Junfei, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng. "A survey of machine learning for big data processing."*EURASIP Journal on Advances in Signal Processing* 2016, no. 1 (2016): 67.

[53] Sharma, Seema, Jitendra Agrawal, Shikha Agarwal, and Sanjeev Sharma. "Machine learning techniques for data mining: A survey." In *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*, pp. 1-6. IEEE, 2013.

[54] Yasser Ganjisaffar (2010-2017). Open Source Web Crawler for Java. Retrieved from https://github.com/yasserg/crawler4j

[55] Jonathan Hedley (2009-2017). jsoup HTML parser. Retrieved from https://jsoup.org/

[56] Pivotal Software (2018). Spring Boot. Retrieved from https://projects.spring.io/spring-boot/

[57] Jihan, Nadheesh, Yasas Senarath, Dulanjaya Tennekoon, Mithila Wickramarathne, and Surangika Ranathunga. "Multi-Domain Aspect Extraction using Support Vector Machines." In *Proceedings of the 29th Conference on Computational Linguistics and Speech Processing (ROCLING 2017)*, pp. 308-322. 2017.

[58] Joachims, Thorsten. "Training linear SVMs in linear time." In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 217-226. ACM, 2006.

[59] Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge university press, 2014.

[60] Gers, Felix A., Nicol N. Schraudolph, and Jürgen Schmidhuber. "Learning precise timing with LSTM recurrent networks."*Journal of machine learning research* 3, no. Aug (2002): 115-143.