Print- ISSN: 1391-586X, E-ISSN: 2602-9030, Copyright © 2018 by Faculty of Science, Eastern University, Sri Lanka.

REVIEW ON SENTIMENT ANALYSIS IN TAMIL TEXTS

Sajeetha Thavareesan[1], Sinnathamby Mahesan[2]

[1]Department of Mathematics, Eastern University, Sri Lanka.

[2]Department of Computer Science, University of Jaffna, Sri Lanka.

ABSTRACT

Sentiment Analysis (SA) is an application of Natural Language Processing (NLP) to analyse the sentiments expressed in the text. It classifies into categories of qualities and opinions such as good, bad, positive, negative, neutral, etc. It employs machine learning techniques and lexicons for the classification. Nowadays, people share their opinions or feelings about movies, products, services, etc. through social media and online review sites. Analysing their opinions is beneficial to the public, business organisations, film producers and others to make decisions and improvements. SA is mostly employed in English language but rare for Indian languages including Tamil. This review paper aims to critically analyse the recent literature in the field of SA with Tamil text. Objectives, Methodologies and success rates are taken in consideration for the review. We shall conclude from the review that SVM and RNN classifiers taking TF-IDF and Word2vec features of Tamil text give better performance than grammar rules based classifications and other classifiers with presence of words, TF and BoW as features.

Keywords: Sentiment Analysis (SA), Natural Language Processing (NLP), Tamil, machine learning

[1]*Corresponding author:* sajeethas@esn.ac.lk

https://orcid.org/0000-0002-6252-5393

1.0 INTRODUCTION

Social media and e-commerce have a great impact on human life, especially with the high usage of smartphones all over the world. These facilitate the users to share and access the opinions of individuals around the world. This makes an explosive growth of data in the internet. Opinions are the key influences to almost all human activities and behaviours. We seek opinions before we make decisions. In the past, people rely only on their friends and relatives for opinions and comments about anything they wanted. When a business organisation needed public opinions about its products and services, it conducted surveys and opinion polls, which is a tedious, time consuming and costly matter [1].

Expeditious development of social media like face book, twitter, fora, *etc*. enables individuals to share their opinions publicly on the Web, which can be accessed by others for their decision making. Individuals are viewing these sites for opinions of others about a product or service before making a purchase or others' opinions about political candidates. Business organizations need them for improving their marketing. However, the major problem here is that even though volumes of data are available, the knowledge that can be acquired from the data is still need to be extracted and the extraction is not a straight forward matter, rather it poses challenges [1].

Each site contains the bulk of opinionated text about products, places, movies, *etc*. but the reader is unable to extract the opinions when the data become large. Thus, Sentiment Analysis (SA) systems are employed to extract and summarise needed information for the reader. SA has drawn attention in recent decades as an active research area in the field of Natural Language Processing (NLP) to analyse people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organisations, individuals, issues, events, topics, and their attributes [1]. Main reason for SA being a very active research area is that it incorporates a wide range of applications in every domain. Research in SA has an important impact on NLP, management sciences, political science, economics, and social sciences as they are all affected by people's opinion.

Sentiments may be analysed in document level, sentence level or aspect level. In the document level the sentiment is analysed by considering the whole document as a basic information unit. In the sentence level analysis each and every sentence considered separately whereas aspect level is to classify the sentiments with respect to aspects of entities. Researchers choose them depending on the nature of analysis. Document level analysis is performed by many authors. The papers that.

Analysing and extracting the sentiments from the text document is a challenging task. Subjectivity and objectivity are the two concepts related to SA. An objective sentence presents factual information whereas a subjective sentence expresses personal thoughts of opinion. "*'Nota' is a Tamil movie*" is an example for objective sentence whereas "*I like the movie 'Nota'* " is a subjective sentence.

Review papers on Sentiment analysis are rare. We have found only two papers. Hussein [2] and Medhat *et al.* [3] presented survey papers on SA applications, challenges, techniques and the details of the corpus. In [3], 54 research papers were analysed and categorised based on the classification techniques, feature selection, language, *etc*. In [2], 47 papers were analysed and the challenges were presented in each paper along with the techniques, corpus, *etc*. used. Most of the research papers discussed in [2] and [3] are SA on corpora in English language.

With the popularity of sharing opinions in native languages across web sites, there is a need for Sentiment classification such as positive, negative, harm, no harm, *etc*. in local languages too. There are quite a few approaches for the classification of sentiments in English language [2], [3], but rare for Indian languages including Tamil. This review paper mainly aims to discuss the algorithms and corpora used in SA researches in Tamil language as no review papers are found reported in our search for survey.

This survey categorises the recent articles according to the techniques, keys of sentiment challenges and the resources used. This could help the researchers to choose the appropriate techniques for their applications and to give a panoramic view to the new comers in this field. Moreover, the available benchmark data sets are also discussed. It includes the most used SA techniques and applications along with the reviews.

2.0 METHODOLOGY

We are interested in doing research on SA using Tamil text. We consider paper on SA using mainly Tamil text. We have so far found nine papers in SA on Tamil text. We discussed about the objectives, corpus, features, techniques and challenges along with the accuracy or F-measure of these papers.

It is found in the papers that different approaches were used for feature representation including presence of words, Bag of Words (BoW), Term Frequency (TF), Term Frequency- Inverse Document Frequency (TF-IDF) and Word2vec [4] vectors and different classifiers were used for classification. In order for getting clear understanding, we begin with the brief descriptions of the feature representation approaches and the classifiers, in Section 2.1 and Section 2.2.

This is followed by the review of the papers in Section 3 in which we look into each paper separately for what approaches were used, under what conditions they tested and what the success rates were.

Findings of the review are discussed in Section 4, where we discuss what challenges they faced and about the corpora used, and the reported results.

A summary of the findings from Section 3 and results of discussion in Section 4 are tabulated in Table 1, for easy perusal. It is followed by the conclusion reached by reviewing the papers.

2.1 Feature representation methods

In order to perform classification on text documents, we first need to convert them into numerical feature vectors. There are many techniques to create feature vectors. Bag of Words (BoW), Term Frequency (TF), Term Frequency- Inverse Document Frequency (TFIDF) and Word2vec are such techniques used for creating feature vectors from categorical data. These feature vectors are given as inputs for classification.

Presence of words: The most intuitive way to do so is presence of words which represents the presence or absence of words in the documents. Usually one is used to represents presence of word and zero to represents absence of word.

BoW: BoW is used to represent the number of times a word appears in a document.
BoW = No: of times word w occurred

TF: TF algorithm is the ratio of number of times the word appeared in a document compared to the total number of words in that document.
TF = No. of times word w occurred in a document/ Total no. of words in that Document

TF-IDF: It is the multiplication of TF and IDF scores whereas TF is a scoring of the frequency of the word in the current document and IDF is a scoring of how rare the word is across documents.
IDF(w) = log(No. of documents/ No. of documents containing word w)

Word2vec: Word2vec is a group of related models that are used to produce word embedding. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space [3].

2.2 Classifiers

Classification is the process of predicting the class (also called, targets or labels or categories) of given data points. The task of classifiers is, approximating a mapping function (f) from input variables (X) to discrete output variables (y). We have listed some popular classifiers used in the reviewed papers.

SVM: The objective of SVM is to identify the hyper plane that has the maximum margin in an N-dimensional space that distinctly classifies the data points. The Generalized SVM equation is given by, min $½||w||^2$ subject to, $yi((w_i^T \cdot xi)-b)-1 >= 0$. For given training data (xi, yi) from 2 classes such that $yi = \pm 1$, SVM finds out hyper plane $w^t(x) + b = 0$

Naive Bayes (NB): NB classifier is derived from Bayes theorem. It assumes that the features are independent. Which is mathematically represented as, $p(c\backslash d) = p(c).\ p(d\backslash c)$ where $p(c)$ represents the individual probability of c and $p(d\backslash c)$ is probability of d given c.

Multinomial Naive Bayes (MNB): Suppose the feature never occurred in a given class, probability will be estimated to zero in NB. To normalise this Laplace smoothing is introduced which is then called as MNB. $p(c\backslash d) = p(c).\ p(t_k\backslash c)$ where $1 <= k <= nd <= n$

Bernoulli Naive Bayes (BNB): Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence features are used rather than term frequencies. If x_{i} is a Boolean expressing the occurrence or absence of the i'th term from the vocabulary, then the likelihood of a document given a class C_{k} is given by $p(x\backslash C_{\mathrm{k}}) = \Pi^{n}_{i=1}\ p^{ki}{}_{xi}(1-\ p_{ki})^{(1-xi)}$ where p_{ki} is the probability of class C_k generating the term x_i. This event model is especially popular for classifying short texts. It has the benefit of explicitly modelling the absence of terms. Note that a naive Bayes classifier with a Bernoulli event model is not the same as a multinomial NB classifier with frequency counts truncated to one.

Decision Tree: A decision tree is a tree where each node represents a feature (attribute), each link (branch) represents a decision (rule) and each leaf represents an outcome (categorical or continues value). It repetitively divides the working area into sub part by identifying lines. Entropy is calculated using, $H = -p(x)\ log\ p(x)$. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers. This causes very poor performance on the test data. It can be avoided by pre-pruning: halts tree construction early or post-pruning: removes branches from the fully grown tree.

Random Forest: Random forest can be identified as a collection of decision trees as its name says. Each tree tries to estimate a classification and this is called as a "vote". Ideally, we consider each vote from every tree and chose the most voted classification.

Logistic Regression: Logistic regression uses some function to squeeze values to a particular range. “Sigmoid” is one of such function which has “S” shape curve used for binary classification. It converts values to the range of 0, 1 which interpreted as a probability of occurring some event. Below is a simple logistic regression equation where b0, b1 are constants. $y = e^{(b0+b1\_x)}/(1 + e^{(b0+b1\_x)})$

Recurrent Neural network: RNN is a type of advanced artificial neural network (ANN) that involves directed cycles in memory. One aspect of recurrent neural networks is the ability to build on earlier types of networks with fixed-size input vectors and output vectors. RNNs can use their internal state (memory) to process sequences of inputs and in RNN, all the inputs are related to each other.

3.0 Review of papers

In this section we present about the approaches, techniques and corpus used in the papers we found. From a technical point of view, the approaches to performing SA can be grouped into two categories, namely, those of machine learning based approach and those of rule based approach. The machine learning method uses learning algorithms such as SVM, NB to determine the sentiment by training on a known labelled dataset, whereas rule based approach involves calculating sentiment polarity for a review using rules or lexicons. The papers [5], [6] and [7] are based on rule based approach whereas [8], [9], [10], [11], [12] and [13] are based on machine learning based approach. All the papers listed here except [8] were used accuracy to evaluate their model. Accuracy was calculated using the equation of (*Correctly predicted test data/ Total number of test data*) × 100.

Uma and Kausikaa [8] proposed a model for SA of English and Tamil tweets. They proposed SVM based classification model to classify tweets into positive, negative and neutral. They collected the Tamil and English tweets that contains mixture of English and Tamil words/ texts using Twitter API and the Tamil tweets were separated using a Tamil corpus. Google Translator was used to translate Tamil tweets to English, and the lowercase version of translated text was used for tokenising, stemming and Part of Speech (POS) tagging. Word Sense Disambiguation (WSD) was employed to overcome the misunderstanding of word sense. The classification model was evaluated using Tweets of sizes of 200, 400, 600, 800, 1000 and different Precision, Recall and F-measure were obtained. It was found that the F-measure value for system using SVM was 0.741.

Arunselvan *et al.* [9] proposed, SA on movie reviews using the machine learning algorithms such as Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), Logistic Regression (LR), Random Kitchen Sink (RKS) and SVM. Term Frequency (TF) of unigram as well as bigram for unique words of the movie review corpus were used as features by varying values of N (N - number of counts). They tested SVM with Linear, Polynomial, Radial Basis Functions, Sigmoid and Pre-computed Kernels and obtained better accuracy of 64.69% for bigram features using Radial Basis Kernel with N=10 than other kernels. RKS and LR obtained 61.88% and 57.14% accuracy respectively using bigrams. MNB and BNB obtained 61.01% and 60.19% accuracy respectively for unigram features.

Seshadiri *et al.* [10] intended to build a model to predict sentiments from Tamil movie reviews as positive or negative. Context words of training data, punctuations and apostrophe were used as features. Different approaches using SVM, Maximum Entropy classifier (Maxent), Decision tree and NB were used along with Tamil SentiWordNet dictionary to categorise the reviews into positive and negative. SVM obtained an accuracy of 75.96% whereas Decision tree gave an accuracy of 66.29%. Authors tested the model without using SentiWordNet and obtained 71.91% for SVM and 64.04% for Decision tree.

Shanta Phani *et al.* [13] analysed sentiments of tweets in three Indian languages: Bengali, Hindi and Tamil. They used training set of SAIL dataset for training and development dataset for testing. They experimented two types of classifications: two-class classification and three-class classification using word n-grams, character n-grams, surface features and SentiWordNet features. They used binary, TF and TF-IDF representations, and six different classifiers: Multinomial Naive Bayes (NB), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), SVM SVC (SV), and SVM Linear SVC (LS). Binary representation of two class classification of Tamil obtained 62.16% using Naive bayes with Word unigrams of all words as features whereas three class classification obtained 44.24% using Random forest with Character bigrams of all characters except space as features. They reported the following as best-performing feature-classifier combinations for Tamil: Two-class: Word unigrams, all words including stop words, binary, NB classifier (62.16%). Three-class: Character unigrams, all characters, TF, RF classifier (45.24%).

Nivedhitha *et al.* [11] proposed unsupervised word embedding to detect polarity of Tamil tweets. They used SAIL-2015 training data set, proposed data and SAIL-2015 test data along with SentiWordNet to test their model. They fitted the input data to the model after pre-processing in parallel with SentiWordNet. Word vectors were created using Word2vec [4] embedding. Centroid vectors were calculated for all the three categories: positive, negative and neutral. Distance vector was computed using sentence vector and centroid vectors. The values were there normalized for classification. Finally the decision making algorithm was used to predict the results. SAIL-2015 test data obtained an accuracy of 60.35%. Training data of SAIL-2015 obtained 64.27% and finally their own corpus obtained 70.62%. They achieved better accuracy for their data as they updated and customized the dictionary data based on the tweets.

Seshadri *et al.* [12] used SAIL dataset for SA on three languages: Tamil, Hindi and Bengali. They classified data into positive, negative and neutral. Recurrent Neural Network (RNN) was iterated 1000 times to obtain the results. They obtained F-measure 0.802 and 88.23% accuracy for Tamil, 0.714 and 72.01% for Hindi and 0.644 and 65.16% for Bengali.

Ravishankar and Shriram [5] proposed a corpus based sentiment classification of Tamil movie tweets into five categories: ‘Sandai’ (Stunt), ‘Kadhal’ (Romance), ‘Masala’ (Formula), ‘Kudumbam’ (Family) and ‘Comedy’. They collected about 7,000 tweets about around 100 Tamil movies for their study. They used TF-IDF, Domain Specific Tags (DST) and Tweet Weight Models (TWM) for classification along with Tamil Agarathi. Overall accuracy obtained for TF-IDF was 29.87%, TF-IDF with DST was 35.64% and TWM was 40.07%. Maximum length of a tweet is 140 characters. So that the possibilities for getting more frequency value for words is rare. This yielded poor performance to their model.

Ravishankar and Shriram [6] proposed grammar rule based sentiment categorisation of tweets into the genres of action, love, sentiment, comedy and commercial. TF-IDF, negation rules and adjective rules were applied to categorise the sentiments. They reported that adjective rules produce better average accuracy of 64.72% than TF-IDF and negation rules.

Ravishankar *et al.* [7] proposed n-gram based sentiment categorisation. They collected reviews about 100 movies. Corpus was cleaned by removing reply, retweets and external links and user defined symbols. TF-IDF and unigrams, bigrams and trigrams were used in feature extraction phase. Grams were used for negation handling. Tweets of length less than four words were classified using grammar rules. Trigrams were applied for length less than or equal to 8 words and longer ones were ignored. Sentiment categorizer algorithms were applied to identify the categories and polarities. Accuracy of 29.87% and 61.29% were obtained for TF-IDF and N-grams respectively. Authors failed to mention about the type of grammar used and did not provide any detail about sentiment categoriser algorithms.

4.0 Discussion

In this section we discuss about the challenges the authors faced, success rates and details that are not stated clearly.

In [8], Google translator was used to translate Tamil text into English. It should be noted that Google Translator still has much lapses in translating Tamil to English or vice versa. Tweets are usually written using only Tamil words or mixed with English words or English words written in Tamil letters. They did not mention how they handled each of these types of tweets rather than merely mentioning they used ‘a Tamil corpus’ and Google Translator to process the tweets.

In [9], MNB and BNB yielded their best performance when using unigram as feature while RKS and LR yielded their best performance while using bigram. RKS and MNB can be employed for SA in Tamil as they yielded above 61% which is comparably closer to the accuracy of SVM. When analysing the elapsed time, RKS yielded lowest when compared to MNB and SVM. MNB took time of all classifiers. The results reported by the authors in the graphs and the table’s mismatches in several places.

In [10], the authors failed to describe what features were considered as context words and how features were represented. The importance of using punctuations and apostrophe as features were not mentioned. No instance was mentioned with apostrophe used in Tamil text (enclosing phrases in single or double quotes is different). They used 114 reviews for training and 420 reviews for testing. Even though the size of the training dataset is less compared to the size of testing dataset SVM achieved 75.96% accuracy. They mentioned about POS of words in SentiWordNet but did not mention about the role of it in their model. The adverbs and adjectives listed in Tamil SentiWordNet could have been used as they form a complete sentiment indicators, and also stop words, which carry no sentiments, could have been removed to make analysis more rigorous. DT, NB and Maxent obtained 66.29%, 66.17% and 64.04% of accuracy respectively.

The SAIL data set was used in [12]. This SAIL consists of corpus of tweets for three languages, namely, Tamil, Hindi and Bengali. SAIL were categorised into three categories: Positive, Negative and Neutral. Authors did not mention about the features used as input to the RNN model. RNN performed well (88.23%) compared to NB (39.28%).

Tweets contain informal, slang and short forms of words. Authors of [8], authors considered all the slang words and non-English words to build a dataset for DST which yielded higher accuracy (35.64%) than TF-IDF. They did not use the TWM method with DST. The sample dataset of DST mentioned in their paper contains a column to represent the second category of the tags, how why these tags were used was not mentioned at all. Nor did they mention about the strategy used for polarity calculation, whether lexicon based or any other method used and how they used the TF-IDF scores to determine the polarity and category of the test data.

‘*Agarathi*’ Tamil dictionary was used by authors of [6] to categorise the tweets but tweets are made up of slang words and transliterated words, which do not normally exist in a formal dictionary. Thus, the accuracy of the categorisation is questionable. Seven rules were created to calculate polarity. Syntax parser, Tamil SentiWordNet, POS tags and Tamil Agarathi were employed in the sentiment evaluation phase. Authors used a different form of equation for calculating TF-IDF values. They did not report the accuracy of negation and TF-IDF based models. But they mentioned the accuracy of TF-IDF, negation rules and adjective rules of a selected movie ‘*Veeram*’ as 27.13%, 47.32% and 60.82% respectively. How the POS tagger was used to build this model was not mentioned in the paper.

In [7], an n-gram model was proposed to handle negations. The authors considered tweets with length less than nine words and created trigrams. Negation rules and adjectives rules were employed to calculate the polarity and category. But the authors failed to mention what negation rules and adjectives rules were used and did not give details of sentiment categorizer algorithms. Each movie was listed with the percentage of being in all the five categories: action, love, sentiment, comedy and commercial. Likewise for polarity also they mentioned the percentage of being in all three categories rather than displaying overall polarity and category with a single label. The algorithm used was not well illustrated or mentioned. They did not try the algorithm without applying length cut off. Sometimes length cut off might cause a decrease in the accuracy as it loses some information.

In [13], authors tested their model with a very few developed dataset and did not test for Tamil corpus. For Tamil they only reported accuracy values from 10-fold cross-validation as they did not have the test data. They found that English words played an important role in discriminating between sentiment classes and also the test data accuracy depends on the size of the training Data. Authors did not check the performance of Tamil tweets classification using stop words as features and by using test data. We can notice that they obtained constant values for classifiers when using SentiWordNet as features. It is a puzzle that how different classifiers that use different techniques for classification resulted in the same value. They obtained notable performance while using stop words as vocabulary and all characters except space and punctuation. We can conclude that stop words and punctuation influenced the polarity calculation. Evaluation on words without punctuation could be made to check whether punctuation was beneficial in word level analysis also. LR performed well for Hindi and Bengali whereas NB performed well for Tamil.

In [11] the authors did not explicitly mention about calculating centroid vectors and the distance vectors. Word2vec approach was employed to represent the semantic meaning of words which was employed to enhance the feature representation.

The techniques, corpus, challenges and the results of reviewed papers are tabulated in Table 1. Among the reviewed papers five were used BoW and TF-

IDF as feature representation for their model. In [11], authors implemented unsupervised model using Word2vec [4] as feature.

According to papers [9], [10] and [8] SVM performed well compared to other classifiers for Tamil language. n-gram, POS, sentence length, Tamil SentiWordNet and word sense disambiguation were also employed to enhance the accuracy of SA. Negation rules and adjective rules were also employed which gained good results in SA according to [6], [7] and [5]. The results of the models were affected by improper pre-processing steps, stop words, mixed scripts, translation and transliteration. Authors used SAIL dataset and their own corpus for implementing their model. Unavailability of the resources for Tamil such as corpus, stop words is huge challenge to the researchers.

Word2vec feature representation seemed more suited compared to binary representation of SA on SAIL dataset. RNN [12] and NB [13] obtained 88.23% and 62.16% respectively whereas unsupervised Word2vec based model obtained 64.27% [11]. In [11] their own corpus yielded 70.62% of accuracy which is greater than the accuracy obtained while using SAIL dataset. The unstructured nature of SAIL dataset may be the reason for this.

SVM with Tamil SentiWordNet of [10] produced better results than SVM with TF of bigram of feature representation [9].

From the papers [5], [6] and [7] we can conclude that adjective rules performed better compared to n-gram, TWT, DST and TF-IDF. DST and thus adjective rules may be taken as an appropriate approach to enhance the performance.

Table 1: Details of the papers reviewed

Title	Objective	Corpus	Features and Methods Used	Observations	Results
Sentiment analysis of Tamil movie reviews via feature frequency count. [9]	Classify the reviews as positive or negative.	Own corpus developed using websites. 1160 positive and 1160 negative reviews.	Features: TF of unigram and bigram. Classifiers: MNB, BNB, Logistic regression, RKS and SVM.	Neutral reviews in both classes, unstructured Corpus, implicit meaning, negations, stop words, short abbreviations and unlemmatised words. RKS yielded lowest when compared to MNB and SVM. MNB yielded highest time of all classifiers.	SVM accuracy-64.69% (Using bi-grams)
Predicting the Sentimental Reviews in Tamil movie using Machine Learning Algorithms. [10]	Classify movie reviews into positive or negative.	Own corpus developed using websites. 267 positive and 267 negative	Features: TamilSentiWordNet and context words of corpus. Classifiers: SVM, Maxent, Decision tree and NB.	Size of the training data used is smaller than the size of the testing data, stop words, Features considered as context words and	SVM accuracy-75.96% (Using Tamil SentiWordNet) SVM accuracy-71.91% (Without

		reviews.		apostrophe used as feature.	Tamil SentiWordNet)
Sentiment Analysis of English and Tamil Tweets using Path Length Similarity based Word Sense Disambiguation. [8]	Predict the sentiments of Tamil and English tweets as positive or negative or neutral.	Own corpus developed using tweets. 1000 tweets for training.	Features: POS and word sense disambiguation Classifier: SVM	Reliability of google translator and efficiency of Tamil corpus.	SVM F-measure- 0.741 (> NB with 0.68)
Analysing sentiment in Indian languages micro text using Recurrent Neural Network. [12]	Classify the SAIL dataset into positive, negative and neutral.	SAIL dataset.	Features: Not mentioned Classifier: Recurrent Neural Network (1000 times iterated on SAIL dataset.)	Emoticons, stop words, mixed scripts and negations.	Tamil:F-score- 0.802 accuracy- 88.2% Hindi: 0.714 and 72.01% Bengali: 0.644 and 65.16%
Grammar Rule-based Sentiment	Sentiment categorization	Own corpus developed	Features: TF-IDF, POS, Tamil SentiWordNet,	Building POS tagger, negation rules and	Accuracy: TF-IDF- 29.87%

Categorisation model for classification of Tamil Tweets. [6]	and polarity detection using negation and adjective rules.	using tweets. 7000 Tamil tweets.	Tamil Agarathi, negation and adjective terms. Classifier: Negation rules and adjective rules	adjective rules.	N-gram- 61.29%
Grammar Rule based Sentiment Categorization model for Tamil Tweets. [7]	Sentiment categorization and polarity detection using TF-IDF and modified n-grams.	Own corpus developed. 7418 tweets about 100 movies.	Features: TF-IDF, n-gram, Sentence length. Classifier: Negation rules and adjective rules	Building negation rules and adjective rules.	Accuracy: Adjective rules- 64.72%
Corpus based sentiment classification of Tamil movie tweets using syntactic patterns. [5]	Corpus based sentiment and polarity classification.	Own corpus developed using tweets about 100 movies. 7,000 tweets.	Features: TF-IDF values, newly created lexicon (DST) and sentence length.	Corpus categorisation, Creating Domain Specific Tags, Calculation of accuracy using TF-IDF scores. Length cut-off.	Accuracy: TF-IDF- 29.87% TF-IDF+DST- 35.64% Tweet weight-age model- 40.07%
Unsupervised Word Embedding based polarity	Unsupervised word embedding	SAIL-2015 training data set, proposed	Features: Word2vec Classifier: Distance	SentiWordNet, Centroid calculation, distance vector	Accuracy: SAIL-2015 training data set - 64.27%

detection for Tamil tweets. [11]	based polarity detection into three categories: Positive, Negative and Neutral.	data and SAIL-2015 test data.	vector.	calculation and decision making algorithm.	Own corpus - 70.62% and SAIL-2015 test data - 60.35%.
Sentiment analysis of tweets in three Indian languages. [13]	Analyse sentiments of tweets in three Indian languages: Bengali, Hindi and Tamil.	SAIL training dataset and development dataset.	Features: word n-grams, character n-grams, surface features and SentiWordNet features. Classifier: Multinomial Naive Bayes (NB), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), SVM SVC (SV), and SVM Linear SVC (LS).	Very few amount of test data used, stop words, negations and 3-class classification for Tamil.	Accuracy: Tamil, 2-class: Word unigrams, all words including stop words, binary, NB classifier (62.16%). Tamil, 3-class: Character unigrams, all characters, if, RF classifier (45.24%).

5 Conclusion

This review paper aims to critically analyse the recent literature in the field of SA with Tamil text. Pre-processing, Corpus, Methodologies and success rates are taken in consideration for the review. Performance of SA model depends on the pre-processing steps such as negation handling techniques and stop words removal. Word level feature representation using TF-IDF and Word2vec gives notable improvement compared to feature representation techniques using character level and word level feature representation using presence of words, BoW or TF. SVM and RNN give best results when compared to other classifiers. The performance of SVM disproportionate to the size of the corpus. In [9] authors reported an accuracy of 64.69% with 2320 reviews whereas in [10] it was 75.96% using SVM with just 534 tweets. SAIL gave better accuracy of 88.23% for RNN in [12] compared to distance vector approach in [11] (60.35%). Grammar rules for handling negations and adjectives is a good attempt to improve the performance as mentioned in [5], [6] and [7].

Moreover, gram model, domain specific tags and Tamil SentiWordNet can be used to enhance the performance.

We shall conclude that SVM and RNN classifiers taking TF-IDF and Word2vec features of Tamil text give better performance than grammar rules based classifications and other classifiers with features presence of words, TF and BoW. Tamil SentiWordNet, adjective rules and n-grams also can be used in SA on Tamil text as it proves a notable performance in the papers [5], [6], [7] and [10].

As there is no review paper on SA in Tamil text found, this paper will be a good resource to the researchers to get better understanding of the classifiers, corpora and challenges of SA in Tamil.

References

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

[2] Hussein, D. M. E. D. M. (2016). A survey on sentiment analysis challenges. *Journal of King Saud University-Engineering Sciences*. http://dx. doi. org/10.1016/j. jksues, 2:330-338.

[3] Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093-1113.

[4] Word2vec (2018). [Online; accessed 30-May-2018].

[5] Ravishankar, N., and Shriram, R. (2017). Corpus based sentiment classification of Tamil movie tweets using syntactic patterns. *A Journal of Multidisciplinary Science and Technology*, 8(2):172-178.

[6] Ravishankar, N., and Shriram, R. (2018). Grammar rule-based sentiment categorisation model for classification of Tamil tweets. *International Journal of Intelligent Systems Technologies and Applications*, 17(1):89-97.

[7] Ravishankar, N., Shriram, R., Vengatesan, K. B., Mahajan, S. B., Sanjeevikumar, P., and Umashankar, S. (2018). Grammar rule-based sentiment categorization model for Tamil tweets. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, 687-695. Springer.

[8] Kausikaa, N., and Uma, V. (2016). Sentiment analysis of English and Tamil tweets using path length similarity based word sense disambiguation. *International Organization of Scientific Research Journal*, 1:82-89.

[9] Arunselvan, S. J., Anand Kumar, M., and Soman, K.P. (2015). Sentiment analysis of Tamil movie reviews via feature frequency count. *Innovations in Information, Embedded and Communication Systems*.

[10] Shriya, S., Vinayakumar, R., Anand Kumar, M., and Soman, K. P. (2016). Predicting the sentimental reviews in Tamil movie using machine learning algorithms. *Indian Journal of Science and Technology*, 9(45).

[11] Nivedhitha, E., Sanjay, S. P., Anand Kumar, M., and Soman, K. P. (2016). Unsupervised word embedding based polarity detection for Tamil tweets. *International Journal of Computer Technology and Applications (IJCTA)*, 9(10):4631-4638.

[12] Seshadri, S., Anand Kumar, M., and Soman, K. P. (2016). Analyzing sentiment in Indian languages micro text using recurrent neural network. *IIOAB Journal: A Journal of Multidisciplinary Science and Technology*, 7:313-318.

[13] Phani, S., Lahiri, S., and Biswas, A. (2016). Sentiment analysis of tweets in three Indian languages. In *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, 93-102.