Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

TWITTER SENTIMENT ANALYSIS USING ENSEMBLE CLASSIFIERS ON TAMIL AND MALAYALAM LANGUAGES

V. Gokula Krishnan[1], Pinagadi Venkateswara Rao[2], J. Deepa[3] and V. Divya[4]

[1]Associate Professor in the Department of Computer Science and Information Technology, CVR College of Engineering, Mangalpally, Hyderabad, Telangana, India.
Email – gokul_kris143@yahoo.com
[2]Associate Professor in the Department of CSE, ACE Engineering College, Ghatkesar, Hyderabad, Telangana, India. Email – venkat.pinagadi@gmail.com
[3]Assistant Professor in the Department of CSE, Eswari Engineering College, Chennai, Tamil Nadu, India. Email – deeparavindhran@gmail.com
[4]PhD Research Scholar in the School of Electrical and Electronics Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India. Email – vdivya6891@gmail.com

Abstract- The proliferation of social network is generating a huge amount of texts and drawing attentions. Sentiment Analysis (SA) extracts useful information from such data. Maximum researches on SA have been done on the English language, but others main languages such as Tamil and Malaya requests obligation too. It is pivotal to work on Tamil and Malaya social posts because it is the most spoken language by native speakers and heavily used in social media. Although such a crowd, modest work has been done on different languages SA. This paper proposes to automatically classify the overall polarity of sentiments expressed in Tamil and Malaya tweets posts by Twitter users in three classes: Positive, Negative and Neutral, and determine a fruitful approach to solve this problem. Two samples of Tamil and Malaya languages are collected and later divided into two different types of corpuses. Each sample in both corpuses is annotated with a polarity labels. We take supervised deep learning approach to this classification problem for each corpus. In deep learning approach, a popular Recurrent Neural Network, Recursive Neural Network and Long-Short Term Memory (LSTM) are tested. LSTM performs best for this classification task in the pre-processed corpus; producing a maximum accuracy of 97%.

Keywords- English Language, Malaya, Polarity, Tamil, Twitter, Sentiment Analysis, Long-Short Term Memory.

1 INTRODUCTION

The popularity of social media has rapidly increased in today's digital world. People actively participate in online activities and surveys by voluntarily posting messages in their mother languages and emoticons for example product reviews, movie reviews, political issues, etc [1]. Twitter is a famous social media service that supports users to send messages up to 140 characters in length, called tweets. As of January 2016, Twitter had 313 million monthly active users and they are posting an average of 550 million of Tweets per day [2]. Most of the Sentiment Analysis (SA) aims to find the polarity (positive, neutral and negative) of a keyword or text, or to make sentence or text level classification based on their polarity score [3]. The proliferation of Twitter has generated a source of extremely large data and most of the information is publicly accessible to all users. Twitter users express their opinions about a wide range of issues like movie, politics, technology, books, religion, food, sports etc. Mining this volume of sentiments provides information for understanding collective human behavior and it is of valuable commercial interest to a particular movie, product or service [4]. Most of the aspects shared by all these opinions are the subjectivity, since the sentiments expressed by the users about a product or service is not biased. Sentiment analysis of data collected from Twitter users can be able to extract the general emotion of users in relation to a range of topics [5]. These sentiments can be very useful for companies to decide about their competitors and user feelings about their product and also useful for individuals to decide whether to purchase the product or not.

Natural Language Processing is a domain of computer science and scientific study of human language i.e. linguistics which is related with the interaction or interface between the human (natural) language and computer [6]. Sentiment Analysis is one of the major areas of focus in Natural Language Processing (NLP). It deals with opinion mining to determine the sentiment regarding the various topics/subjects in discussion [7]. The basic task is classification of piece of text stating an opinion on an issue with one of the two opposing sentiments (Thumbs up/positive or Thumbs down/negative). With increase in the number of users in social media, user generated web contents such as chat, blogs, microblogs, etc. are increasing rapidly [8-9]. This opens a new room for Sentiment Analysis in the social media contents. In addition to this, users use many variations of spellings, various emoticons and improper uses of punctuation marks. These things together made the Sentiment Analysis task more challenging and open new scope for research [10]. Much of the research work on sentiment analysis has been applied to the English language, but construction of resources and tools for sentiment analysis in languages other than English is a growing need [11] since the microblog posts are not just posted in English, but in other languages as well. However, during last few years user generated content in other languages is growing at a very rapid rate. Tamil is one of the most important Indian languages with a substantially large number of speakers. This paper describes the system we used for sentiment classification of Tamil text messages into one of the sentiment classes namely positive, negative or neutral. In order to accomplish the task, we used ensemble deep learning techniques namely Recurrent Neural Network, recursive neural network and Long-Short Term Memory (LSTM) for classification of Tamil and Malaya tweets.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

The remaining paper is consists of: section 2 presents the study of existing techniques of tweets classification, Section 3 provides the detailed explanation of proposed methodology. The validation of proposed method along with its existing techniques is given in Section 4. Finally, the conclusion of the research study is described in Section 5.

2. LITERATURE REVIEW

In this section, several approaches used in twitter sentiment classification are reviewed. We review recent research on twitter sentimental analysis approaches that are described as follows:

M.bouazizi and T. ohtsuki have proposed the tweets which contain more than one sentiment called as multi class sentiment analysis. Where they have identify the exact sentiment conveyed by the user rather than the whole sentiment of the tweet. To identify this thing they have also used SENTA tool. They proposed an approach, with the help of this approach they have calculated the sentiment score whoever sentiment is having highest score that will be considered this process is called as “Quantification” [12]. Akshay Amolik, et al., a highly suitable model have discussed in this paper which will take the twitter data of upcoming Hollywood and Bollywood movies. They are able to this task with the help of classifier and features like SVM and naive Bayes. Both of them are used for high accuracy but in terms of precision naive Bayes is better than SVM and if we talk about recall then SVM is better than naive Bayes. By increasing the dataset we can increase the classification accuracy [13].

Pulkit et al. built and proposed a model which extract tweet from twitter based on the post terror activities. They made their study on terrorist attack which was occurred in URI on 18 September 2016. They considered 59,988 tweet which had taken after the attack. They consider only those tweets which has #UriAttack, #uriattack. #uriattacks. They have used the naïve bayes and SVM to extract the last Re-tweet time and number of Re-tweet [14]. Sudarshan et al. proposed a technique in sentiment analysis on twitter data where they have collected reviews of the product. They have used naïve bayes algorithm which perform better in term of accuracy and efficiency. They have extracted 200 tweets where the average length of tweet was 70.105. The aim of this research is to identify the characteristic of tweet like how many times the tweet was liked and how many times they have Re-tweet the tweet [15].

Hetu et al., built and proposed a model in sentiment analysis on twitter data based on anaconda python. They extract the dataset from kaggle in which they classify the people emotions based on positive and negative reviews. This model gives high accuracy on large dataset [16]. Ali hasan et al. proposed a model using the hybrid approach that comprise sentiment analyzer machine learning. They took only those tweet that is followed by the hashtag(#) and contain the current political trends. Basically this model converts the urdu tweet into English tweet. They took 1690 tweet for training data and 400 for testing the data. They have used the naive Bayes and SVM classifier for training the dataset in weka and building a model. They have used three different libraries to calculate the subjectivity and polarity [17].

The research work [18] uses a hybrid approach by combining CNN and RNN for sentence classification. The authors stated that the pooling of layers in CNN which is done to extract high level features, can be the drawback as it concentrates only on the main features of the sentence and ignore the other features. This leads to the lost data in CNN. Hence, the authors proposed RNN as an alternative for the pooling layers to avoid loss of data. It is also mentioned that the vanilla RNN version suffers from vanishing and exploding gradient. Hence Long short-term memory LSTM is used to overcome the drawback of vanilla RNN. Two input datasets Stanford movie review IMDB dataset and Stanford Sentiment Treebank are classified using this approach. The performance of the model is compared with traditional methods like SVM, Naive Bayes, Bag of words and other CNN forms like CNN random and CNN static. This model outperforms the aforementioned models in terms of prediction accuracy. Hence it was proved that the use of pooling layers only degrades the performance due to the loss of long-term dependencies. Thus using a much smaller architecture the same level of classification accuracy can be obtained by replacing the pooling layers with RNN-LSTM.

In the paper [19], the authors Wint et al. propose another flavour of hybrid deep learning architecture called Hybrid two Convolutional Neural Networks and Bidirectional LSTM (H2CBi) for sentiment classification to detect hateful languages. This uses two CNN layers and a bidirectional LSTM layer. The authors suggest that using this model, can help CNN overcome two main drawbacks 1) its lack of semantic understanding 2) inability to understand misspelled words. The authors are experimented with three pre-trained word vectors Word2Vec, GloVe and fast-Text against seven datasets, three product review datasets from Amazon, movie review and Yelp and four social network site data from Twitter I and II, Facebook and FormSpring.me. Of all the three Word2Vec is observed to more effective than other two word vectors. The architecture uses, two pre-trained word matrices embedding layers which are parallely fed to two CNN layers of filter region size 3. This generates two sets of feature maps which are then concatenated and using maximum pooling function the largest feature map is formed. The output from here is fed as input to the bidirectional LSTM- BLSTM it is so called as it uses two LSTM, one from left to right and the other from right to left to get the maximum benefit out of the LSTM. The authors conclude that best results are achieved for product review

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

datasets using Word2Vec and H2CBi and for social network sites data the best result was achieved using both Word2Vec and fastText with H2CBi. As a future work, the authors suggest to use large and imbalanced data sets and to alter the filter region size to get better results out of CNN layers.

In this paper [20], the authors Zhao et al. use Global Vectors for Word Representation (GloVe) word vector and deep convolution neural network to perform sentiment analysis on five twitter data sets. Five twitter datasets take The Stanford Twitter Sentiment Test (STSTd) data set, SemEval 2014 dataset, the Stanford Twitter Sentiment Gold dataset, The Sentiment Evaluation Twitter dataset and The Sentiment Strength Twitter dataset are used. These five datasets are tested against BOW with SVM and Logistic Regression LR combination and GloVe with SVM and LR combinations and the GloVe with deep convolution neural network DCNN. The results obtained show that the GLoVe DCNN outperforms the conventional models in terms of accuracy, F1 measure in classifying the tweets as positive and negative tweets. The GloVe word vector is chosen for twitter sentiment analysis as it has the advantages of local context window and global matrix factorization methods. It uses co-occurrence probability rather than the probability to learn word vector. Tweets need an exhaustive pre-processing as tweets are highly informal in nature. After pre-processing, the words are converted to word embeddings and fed to deep convolution network. Three convolution layers and max pooling layers are used. The convolution layers use different filter size and feature maps. The fully connected layer is used along with dropout and softmax output.

This paper [21] used deep CNN and bidirectional LSTM. Deep CNN was applied on character-level embeddings to increase the information for word embeddings. Followed by bidirectional LSTM to classify the sentences based on sentiment. This paper stresses on standardizing data to achieve high performance and hence the authors have designed a module called tweet processor to remove the non-essential words from the tweet however retaining the emoticons and necessary information. Further, the tweet processor applies a set of semantic rules SR on the sentences. The output of the tweet processor is fed to DeepCNN. In the proposed framework, the DeepCNN is trained on the character embeddings unlike word embeddings in the other models. Two wide convolution layers were used in the deep CNN. This way the morphological and shape information of the words were obtained. This is helpful in understanding the word formation and the relationship with other words. The output of the DeepCNN is a fixed size feature vector for all the words. These fixed size vectors along with the word embeddings obtained from pre-trained word vectors like word2vec or GloVE are concatenated and fed to bidirectional LSTM. The model was tested against three datasets from Twitter. The results obtained were compared with machine learning algorithm approaches. It was observed that Stanford STS gives a

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

promising result whereas Sanders and HCR datasets outperform the results obtained from that of the machine learning approaches. The datasets are also tested against separate models of DeepCNN and bidirectional LSTM just to show that the combination of both yields best results. It was observed that for twitter datasets, the word vector Glove works better than word2vec. And the use of character embedding has made a remarkable difference in the field of natural language processing.

In this paper [22], the authors Jagadeesh et al. have used word2vec and CNN on product sentiment analysis. The input datasets were taken from Amazon product review. The input domain is mobile phone review. Two input dataset one consisting of 998 mobile phone reviews and the other consisting of 5761 reviews are taken. In this model only static dataset is used. After the pre-processing of the input datasets, they are fed to word2vec to generate word embeddings. Short reviews are padded with zeroes. The output of the word2vec is a 300 dimensional vector which is stored in array. The matrix formed out of the word vectors and polarity is sent to CNN as input. The authors have used a CNN architecture containing convolution layer, ReLU activation, Max-pooling layer, fully connected layer and Softmax activation function. Accuracy, precision and recall metrics are measured. Two datasets with 998 reviews and 5761 reviews are tested against the model and their accuracy is compared. It is observed that with large dataset the accuracy is better. To show that the deep learning models work better than machine learning algorithms, the input datasets is tested with Naïve Bayes. As a future enhancement big and dynamic dataset was to be used. They also like to extend the model to work on aspect level of the products.

3. PROPOSED METHODOLOGY

In this section, the detailed description of proposed ensemble classifiers for tweets classification is provided. Figure 1 shows the working procedure of proposed methodology, where the process has four major steps such as collection of tweets, pre-processing, feature extraction and classification. Each process is briefly described in the following sections.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

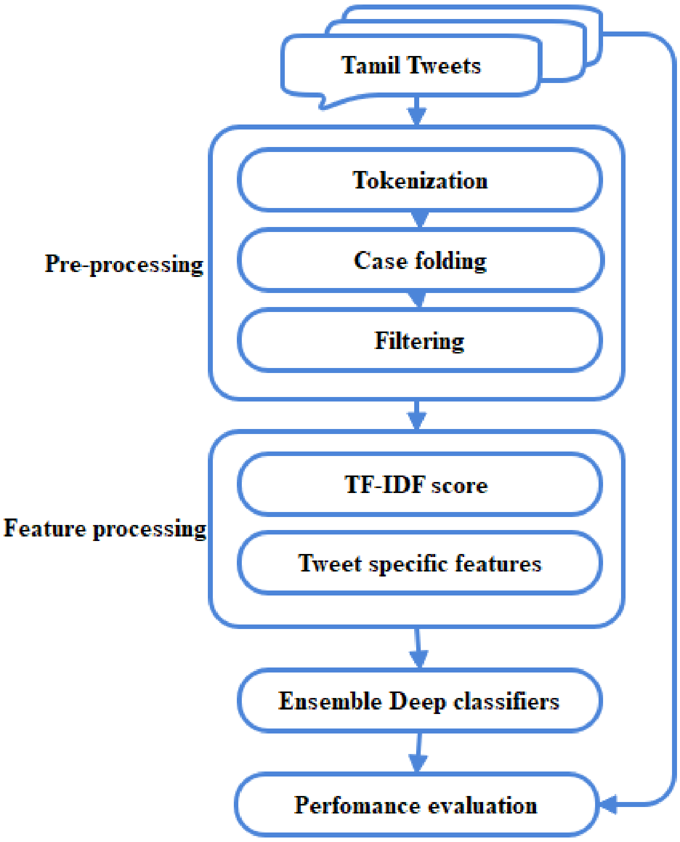

Fig.1. Proposed Flow Diagram.

3.1. Dataset Description

The proposed system has evaluated with two different kinds of datasets such as kracekumar and Malaya-Dataset. The descriptions of the data sets are described below.

3.1.1. Tamil Dataset (Kracekumar)

To evaluate the proposed system kaggle kracekumar dataset has used. The dataset contains 1K tweets from twitter. Each tweet has a relevant label as Happy or Sad. The dataset's purpose is to use the tweets for binary classification, sentiment analysis, and other NLP tasks. The dataset is downloaded from the link: https://www.kaggle.com/kracekumar/tamil-binary-classification-1k-tweets-labels-v1. Some of the sample tweets are shown in the figure.2.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

```
tweet,sentiment
உன்னைத்தொட்டால் உன்னுள்ளத்தை நொருக்கமாட்டியோ!! என்னைப் போல பெண்ணைப்ப்
பார்த்து மயங்க மாட்டியோ!!  #RaOne #chammakChallo #tamilLyrics,Happy
"நதியா நதியா நயில் நதியா
…
இடை தான் கொடியா
கொடி மேல் கனியா
#RDBurnam #HindMusic #TamilLyrics",Happy
"உறக்கம் விற்று கனவுகள் வாங்கலையா?! #TamilLyrics RT @JanuShath: கனவுகள் விற்றுக் க
விதைகள் வாங்குவதும், கவிதைகள் விற்றுக் காதலை வாங்குவதுமாய்.",Sad
மீண்டும் உன்னை காணும் மனமே ... வேண்டும் எனக்கே மனமே மனமே !!! #TamilLyrics,Sad
உயிரை தொலைத்தேன் அது உன்னில் தானோ ... இது நான் காணும் கனவோ நிஜமோ...அ
ன்பே உயிரை தொடுவேன் உன்னை தாலாட்டுதே பார்வைகள் ! #TamilLyrics,Sad
```

Fig. 2. kaggle kracekumar dataset Sample tweets.

3.1.2. Malaya-Dataset

We extracted the dataset from huseinzol05's GitHub repository named Malaya-Dataset for the training set and testing set for this study were extracted. The public can access this repository at https://github.com/huseinzol05/Malaya-Dataset. From the readme file, the repository claimed to gather and store Bahasa Malaysia corpus. We also discussed the method used to gather these data in the same file. The data are mostly collected using crawlers, and these data are semi-supervised by paid linguists. We extract two repository folders data, which are Sentiment Twitter and Sentiment Multi-domain. These data are all in. json format, and the number of data in total is 1,231,396, and all the data are pre-labeled. The number of negative tagged sentences is 693,249, and the number of positive tagged sentences is 538,147.

We extract the data from Twitter profiles for real-world implementation without using Twitter's API through Twint. Twint is an advanced Twitter scraping tool written in Python, and it utilizes Twitter's search operators to allow scraping from specific users and tweets relating to certain topics, hashtags, and trends. In this research, we scraped tweets that contain the keyword'Celcom', 'Digi', and 'Maxis' dated from March 18, 2020, until August 18, 2020.Throughsearchingforthosekeywords,tweets, directly and indirectly, mentioned to the 3 CSP can be extracted. The scraped data stored in .csv files. The total number of data scraped for Celcom, Digi, and Maxis is 101,768, 45,783, and 36,582, respectively. There are 34 columns, and some examples of the columns are 'timezone', 'user_id', 'username', 'name', and 'tweet'.

3.2. Pre-processing

The following steps are performed for the pre-processing module of the tweets both from training and test data.

1) **Translation:** The input Tamil and Malaya tweets are translated into English tweets by using Google Translator, which is given as input for further processing.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

2) **Tokenization:** As tweets are user generated content, sometimes user types two or more terms without any white space between them.
3) **Case folding:** Terms having upper-case letters are converted into its lower-case version.
4) **Filtering:** Tweets contains URL links, user names and punctuation marks that do not contribute to sentiment and hence need to be removed.
5) **Format Conversion:** .arff conversion is carried out.

3.3. Feature Extraction

This module focuses on identifying features that contributed to the sentiment of the tweets.

1) **TF-IDF score of Unigram and Bigram:** TF-IDF score gives a statistical measure of an n-gram. This score can be used to identify the inclination of a particular n-gram to any one of the sentiment classes. TF-IDF can be calculated using eq. 1.

$$tf - idf = tf \times \log(\frac{N}{df}) \tag{1}$$

2) **Tweet specific features:** Here we considered two features that are specific to tweets namely Hashtags and Emoticons.

Hashtags: Hashtags are the terms starting with a # symbol (for eg. #new_year). It is one of the important features of twitter. We have considered the terms present in a hashtag to find its inclination towards a particular class after removing the # symbol.

Emoticons: Since twitter restricts users to only 140 characters, users frequently uses various emoticons to express their sentiments. We identified this as an important feature and classified the frequently appearing emoticons into the three sentiment classes.

3.4. Classification of Final Tweets

In this section, an ensemble classifiers namely RNN, LSTM and Recurrent Neural Network are used for the input tweets, which is described as follows:

A. Recurrent Neural Networks

It is obvious that the training of the Recurrent Neural Network model consists of two parts - Forward Propagation and Back Propagation. Forward Propagation is responsible for calculating the output values, and Back Propagation is responsible for passing the residuals that were accumulated to update the weights, which is not fundamentally different from the normal neural network training. Figure 3 shows the basic architecture of this network.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

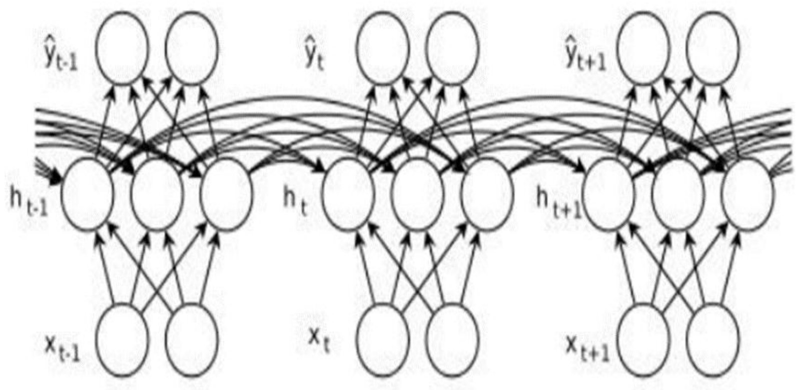

Fig. 3. The Unfolded Recurrent Neural Network.

The standard RNN is formalized as follows: Given training samples $x_i(i = 1, 2, \ldots, m)$, a sequence of hidden states$h_i(i = 1, 2, \ldots, m)$, and a sequence of predictions$\widehat{y}_i(i = 1, 2, \ldots, m)$. W_{hx} is the input-to-hidden weight matrix, W_{hh} is the hidden-to-hidden weight matrix, Wyh is the hidden-to-output weight matrix, and the vectors b_h and b_y are the biases. The activation function e is a sigmoid, and the classification function g engages the SoftMax function.

The objective function associated with RNNs for a single training pair $(x_{i,}y_i)$ is defined as $f(\theta) = L(y_i : \hat{y}_i)$, where L is a distance function which measures the deviation of the predictions $\hat{y}_i$from the actual labels y_i. Let η be the learning rate and k be the number of current iterations. Given a sequence of labels $y_i(i = 1, 2, \ldots, m)$.

B. Recursive Neural Network (RNN)

As a classical neural network framework, the standard RNN is applied to solve inductive inference tasks on complex symbolic structures of arbitrary size (such as logical terms, trees, or graphs). Fig. 4. Illustrates this approach. When a phrase is given, the RNN parses it into a binary semantic tree and computes the vector representation of each word. During the forward-propagation training period, the RNN computes parent vectors in a bottom-up fashion. The composition equation is as follows:

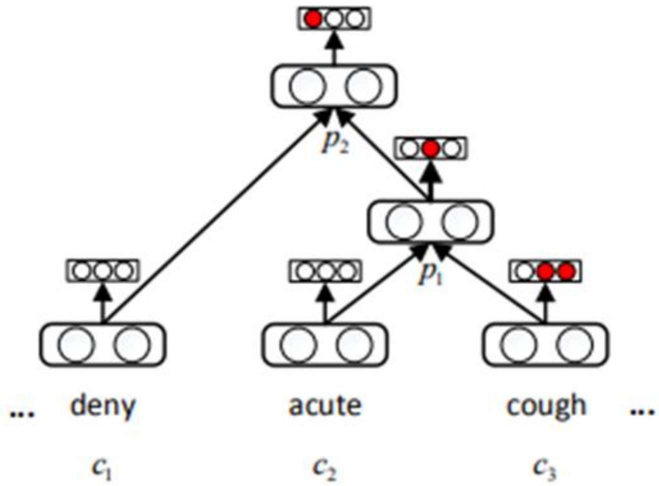

Fig. 4. Semantic Tree of RNN Models for Classification Task.

$$p_1 = f\left(W\begin{bmatrix}c_2\\c_3\end{bmatrix} + b\right), p_2 = f(W(\begin{bmatrix}c_1\\p_1\end{bmatrix} + b) \quad (2)$$

Where f is the activation function; $W \in R^{d\times 2d}$ is the weight matrix, where d is the dimensionality of the vector; and b is the bias. Then, each parent vector $p_i$is given as a feature to a softmax classifier such as that defined in Eq. 16 to compute its label probabilities:

$$y^p = softmax(W_s.p) \quad (3)$$

Where $Ws \in R^{3\times d}$ is the classification matrix. In this recursive process, the vector and classifying result of the node will gradually converge. After the vector of the leaf node is given, the RNN can ultimately map the semantic representation of the entire tree into the root vector.

C. Long Short-term Memory (LSTM)

RNN is the neural feedback network's extension. The gradient disappears or explodes, however, in the ordinary RNN. Long Shortened Memory Network (LSTM) has been designed to resolve the problems and has been performed superiorly. Three gates and one cell memory state exist in the LSTM architecture. Figure 5 appearances the LSTM standard architecture.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

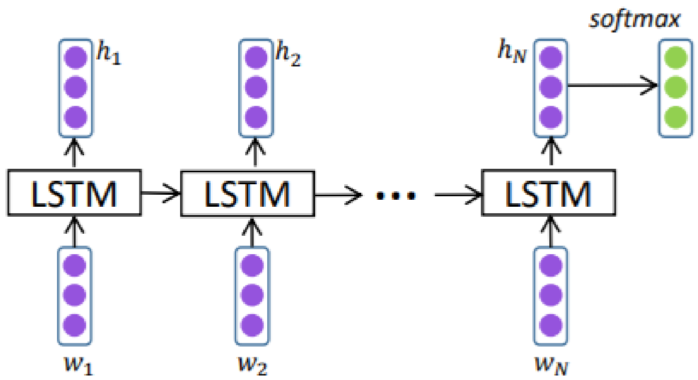

Fig. 5. The architecture of a standard LSTM.

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \quad (4)$$

$$f_t = \sigma\left(W_f . X\right) + b_f \quad (5)$$

$$i_t = \sigma(W_i . X) + b_i \quad (6)$$

$$o_t = (W_o . X) + b_o \quad (7)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c . X + b_c) \quad (8)$$

$$h_t = o_t \odot \tanh(c_t) \quad (9)$$

where W_i , W_f , $W_o \in \mathbb{R}^{\times 2d}$ dare the weighted matrices and b_i , b_f , $b_o \in \mathbb{R}^{d}$ are biases of LSTM, which is the short-term memory solution. They have inbuilt systems, which can control the flow of information, called gates. These gates can find out which data to keep or throw away in one sequence. This enables it over the extended chain of sequences to transmit relevant information to make forecasts of tasks. The main notion of LSTM is the cell condition and its different gates. The cell state serves as a transportation route throughout the flow of information. During processing of the sequence, the cell state can contain useful data. Even early knowledge can thus lead to later temporal stages, which reduce the impact of short-term memory. The information will be added to or withdrawn by gates as the cell state moves on its trip. The gates can find out which knowledge is important in training to remember or forget. The next section will describes the validation of proposed methodology.

4. RESULTS AND DISCUSSION

The proposed system is implemented with the help of python with 3.0 GHz Intel i3 processor, 1TB hard disc and 8 GB RAM. Python is one of programming language with multiple paradigms, which support object, procedural, and functional oriented

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

programming. From the research that has been done, it is apparent that researcher's IDE is Jupyter Notebook and researcher is using Python 3.6.8 as the programming language. Aiming to make the sentiment analysis application easy to use, researcher has made a program based on command prompt (CMD).

4.1. Performance measure

According to the confusion matrix in Table 1, several performance measures can be defined. Among them, accuracy, recall, precision, and F-score are well known to researchers [23] and are the most used in research in the field of twitter sentiment analysis like [24].

Table.1. Confusion matrix.

	Actual positive	**Actual negative**
Real positive	True positive (TP)	False positive (FP)
Real negative	False negative (FN)	False negative (FN)

Therefore, in this article, the performance of all methods on all datasets is evaluated based on these four performance measures. The accuracy measure that is utilized as a fitness function for the proposed method has been defined in (10), and the other performance measures are computed as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

$$Recall = \frac{TP}{TP+FN} \quad (11)$$

$$Precision = \frac{TP}{TP+FN} \quad (12)$$

$$F-Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (13)$$

4.2. Experiment analysis of Proposed Ensemble Classifier

In this section, two datasets are used for validation process and we are experimenting the datasets into the ratio of 60-40 and 80-20 of training and testing dataset in terms of accuracy, precision, recall and F-score. Initially, the ratio of 60-40 is taken and validated the proposed ensemble classifier for kracekumar dataset, which is shown in Table 2.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

Table 2 Performance Analysis of Proposed Ensemble for the ratio of 60-40 on kracekumar

Method	**Accuracy (%)**	**Recall**	**Precision**	**F-score**
Recurrent Neural Networks	83.18	0.754	0.780	0.765
Recursive Neural Network	81.65	0.764	0.740	0.748
LSTM	**87.67**	**0.815**	**0.814**	**0.789**

From the above table, it is clearly shows that the proposed LSTM achieved better performance than other two algorithms namely RNN and Recurrent Network. For instance, RNN achieved 81.65% of accuracy, 75% of recall, 74.0% of precision and 74.8% of F-score, where Recurrent Network achieved better than RNN in terms of accuracy, precision and F-score i.e. 83.18% of accuracy, 78.0% of precision and 76.5% of F-score. However, LSTM achieved nearly 81% of precision and recall, where accuracy is 87.67% and F-score is 78.9% that shows its better performance. Table 3 shows the performance analysis for Malaya dataset on the ratio of 60-40.

Table 3 Performance Analysis of Ensemble Classifier in the ration of 60-40 on the Malaya-Dataset

Method	**Accuracy (%)**	**Recall**	**Precision**	**F-score**
Recurrent Neural Networks	77.18	0.832	0.759	0.791
Recursive Neural Network (RNN)	79.56	0.7719	0.732	0.782
LSTM	**83.49**	**0.82**	**0.791**	**0.8712**

The RNN technique achieved nearly 73% to 79% of accuracy, precision, recall and F-score on Malaya Dataset. The Recurrent Network achieved nearly 75% to 77% of accuracy and precision, 83.2% of recall and 79.1% of F-score. However, LSTM achieved better performance than RNN and Recurrent Network, for instance, LSTM achieved 83.49% of accuracy, 82% of recall, 79.1% of precision and 87.12% of F-score. When comparing with Malaya dataset, the LSTM achieved high accuracy on Tamil dataset. The next table 4 shows the classification of ensemble technique for the ratio of 80-20 on kracekumar dataset.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

Table 4 Performance Analysis of Proposed Ensemble for the ratio of 80-20 on kracekumar

Method	**Accuracy (%)**	**Recall**	**Precision**	**F-score**
Recurrent Neural Networks	91.65	0.880	0.935	0.905
RNN	96.58	0.983	0.944	0.963
LSTM	97.71	0.991	0.961	0.975

When comparing with three techniques, Recurrent Network and RNN achieved nearly 95% of accuracy, nearly 90% to 97% of recall, nearly 93% of precision and nearly 95% of F-score. But, the LSTM achieved 97.71% of accuracy, 99.1% of recall, 96.1% of precision and 97.5% of F-score for kracekumar dataset. This proves that the LSTM achieved better performance even for Tamil dataset. But, the ensemble classifiers achieved high performance only when the 80% of training dataset and 20% of testing data. This proves that when the training number increases, the performance of proposed methodology is increased. The next table 5 provides the validated results of proposed technique for Malaya Dataset on the ratio of 80-20% of data.

Table 5 Performance Analysis of Ensemble Classifier in the ration of 80-20 on the Malaya-Dataset

Method	**Accuracy (%)**	**Recall**	**Precision**	**F-score**
Recurrent Neural Networks	90.70	0.935	0.957	0.892
RNN	95.46	0.972	0.934	0.952
LSTM	97.23	0.995	0.962	0.982

The F-score of Recurrent Network and RNN is nearly 90-95%, where the LSTM is 98% of F-score. In addition, Recurrent Network and RNN achieved nearly 96% of both precision and recall, but the LSTM achieved 99% of recall and 96.4% of precision. The accuracy of all three techniques achieved nearly 92% to 97.23% of accuracy. However, the LSTM achieved better performance than Recurrent Network and RNN, because it has more number of hidden layers and easily trained the dataset than other two techniques. In both datasets, the ensemble techniques achieved better performance than the ratio of 60-40% of training and testing data. The next Table 6 will explain the comparative analysis of our proposed ensemble techniques in terms of accuracy.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH

Table 6 Accuracy comparison of Proposed Ensemble Techniques for both datasets.

Method	Dataset	Data	Accuracy
Recurrent Neural Networks	Tamil	60-40	83.18
RNN			81.65
LSTM			**87.67**
Recurrent Neural Networks		80-20	91.65
RNN			96.58
LSTM			**97.71**
Recurrent Neural Networks	Malaya	60-40	77.18
RNN			79.56
LSTM			**83.49**
Recurrent Neural Networks		80-20	90.70
RNN			95.46
LSTM			**97.23**

In Tamil dataset, among the three ensemble techniques, LSTM achieved better performance for the both ratio of 60-40% and 80-20%. However, LSTM improves nearly 10% of accuracy on the ratio of 80-20% of training and testing data. Not only the performance of LSTM is improved, but both RNN and Recurrent Network's performance are also improved while using the ratio of 80-20%. When comparing with Malaya Dataset, ensemble techniques achieved better performance on Tamil dataset for the ratio of 60-40%. In Malaya Dataset, the LSTM achieved 83.49% of accuracy on 60-40% ratio and same technique achieved 97.23% of accuracy on 80-20% ratio of training and testing data. This comparison proves that the LSTM achieved better performance than other techniques for both datasets on the both ratios.

5. CONCLUSION

Twitter sentiment analysis offers organizations ability to monitor public feeling towards the products and events related to them in real time. The approach in this paper presents the sentiment analysis of the different data consisting of monolingual tweets in two Indian languages (Malayalam and Tamil). The collected data have been used as input to the sequential model. The approach investigates ensemble classifiers of neural network layers (RNN, Recurrent Network and LSTM). The highest accuracy has been obtained using the LSTM achieved 97.71% on the Tamil dataset, and the LSTM achieved 97.23% on the Malaya dataset. The results have been closely scrutinized to conclude that as the complexity of the text in the data set increases, the accuracy decreases for each model. Models with more number of hidden layers provide better accuracy for more complex text as an increase in the number of hidden layers adds to the non-linearity of the model to predict complex situations. It has also been observed that LSTM network perform better than other techniques in predicting the polarity of a sentence. Presently, pre-processing of the textual data is done to remove the emoticons, exclamations. However, they may be playing a role in determining the sentiment of the sentence. For example: 'What!' and 'What?' represent two different meanings, which cannot be distinguished using present-day sentiment analysis approaches as they tend to pre-process the sentences to remove the punctuation. Some people use only emoticons to express their sentiments, which can be embedded and worked upon for making the sentiment analysis approaches even more reliable. This work can also be extended to classify sentences into multiple classes such as extremely negative, negative, neutral, positive, and extremely positive, which gives a more precise sentiment to a sentence.

REFERENCES

[1] M.Ramadhani and H.S.Goo, H.S, "Twitter sentiment analysis using deep learning methods," *In 2017 7th International annual engineering seminar (InAES),* pp. 1-4, 2017.

[2] A.Deshwal and S.K.Sharma, "Twitter sentiment analysis using various classification algorithms. *In 5th international conference on reliability, infocom technologies and optimization (Trends and Future Directions)(ICRITO),* pp. 251-257, 2016.

[3] B. Liu, "Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press, pp. 381-398, 2015.

[4] Giachanou, A. and Crestani, F., 2016. Like it or not: A survey of twitter sentiment analysis methods," *ACM Computing Surveys (CSUR*), vol.49, no.2, pp.1-41.

[5] Krouska, A., Troussas, C. and Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," *In 2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA),* pp. 1-5, 2016.

[6] A.Trivedi, A.Srivastava, I.Singh, K.Singh, and S.K.Gupta, "Literature Survey on Design and Implementation of Processing Model for Polarity Identification on Textual Data of English." *IJCSI*, 2011.

[7] M.Kanakaraj and R.M.R.Guddeti, 2015, "Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP techniques," *In Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pp. 169-170, 2015.

**ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:08:2021
DOI 10.17605/OSF.IO/W4BEH**

[8] P.D.Turney. "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, 2002.
[9] K.K.Pawar, P.P.Shrishrimal and R.R.Deshmukh, “Twitter sentiment analysis: A review,” *International Journal of Scientific & Engineering Research*, vol. 6, no. 4, pp.957-964, 2015.
[10] S.M.Mohammad, “Challenges in sentiment analysis. In A practical guide to sentiment analysis” *Springer*, Cham, pp. 61-83, 2017.
[11] P.Tripathi, P.Vishwakarma, S.K.and A.Lala, “Sentiment analysis of english tweets using rapid miner,” *In 2015 international conference on computational intelligence and communication networks (CICN*), pp. 668-672, 2015.
[12] M.Bouazizi, T.Ohtsuki, “Multi-Class Sentiment Analysis in Twitter: What if Classification is Not the Answer,” *IEEE Access.* Vol.6, pp. 64486-64502, 2018.
[13] A.Amolik, A.Akshay, Niketanjivane, Mahavir Bhandari and M.venkatesan “Twitter sentiment analysis of movie reviews using machine learning techniques.” *International Journal of Engineering and Technology, vol.* 7, no.6, pp. 1-7, 2016.
[14] G.PulkitGarg, HimanshuGarg, VirenderRanga “Sentiment Analysis of the Uri Terror Attack UsingTwitter” *International Conference on Computing, Communication and Automation (ICCCA2017),* 2017.
[15] S.Sirsat, G.SujataRao, BhartiWukkadada, “Sentiment Analysis on Twitter Data forproduct evaluation,” *IOSR Journal of Engineering*, pp. 22-25, 2019.
[16] A. Hasan, Sana Moin, Ahmad Karim and ShahaboddinShamshirband” Machine Learning-Based Sentiment Analysis forTwitter Accounts” 2018 by the authors. Licensee MDPI, Basel, Switzerland.
[17] B.HetuBhavsar, RichaManglani” Sentiment Analysis of Twitter Data using Python”International Research Journal of Engineering and Technology (IRJET) Mar 2019e-ISSN: 2395-0056 p-ISSN: 2395-0072.
[18] A. Hassan and Ausif Mamhmood, “Convolutional recurrent deep learning model for sentence classification,” *in IEEE Access,* vol. 6, pp. 13949-13957, 2018.
[19] Z.Jianqiang, G.Xiaolin and Z.Xuejun, "Deep convolution neural networks for twitter sentiment analysis," *in IEEE Access*, vol. 6, pp. 23253-23260, 2018.
[20] Z. Z.Wint, Y.Manabe and M.Aritsugi, "Deep learning based sentiment classification in social network services datasets," *IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD)*, Yonago, 2018, pp. 91-96.
[21] H.Nguyen and M.Le Nguyen, “A deep neural architecture for sentence-level sentiment classificaion in twitter social networking,” *Communications in computer and information science (CCIS*), vol. 781, pp. 15–27, 2018.
[22] J.Panthati, J.Bhaskar, T.K.Ranga and M.R.Challa, "Sentiment analysis of product reviews using deep learning," I*n 2018 International Conference on Advances*

in Computing, Communications and Informatics (ICACCI), Bangalore 2018, pp. 2408-2414, 2018.
[23] J.Han, M.Kamber, “Data mining: concepts and techniques,” *2nd edn. University of Illinois at Urbana-Champaign, printed on Elsevier Inc*, 2006.
[24] Keshavarz H, Abadeh, “M-SALGA: adaptive lexicon learning using genetic algorithm for sentiment analysis of microblogs,” *Knowl-Based System,* pp. 1–16, 2017.