Paper No: SC-02 **Smart Computing**

2022 International Research Conference on Smart Computing and Systems Engineering (SCSE) | 978-1-6654-7375-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/SCSE56529.2022.9905170

Extraction of Sentiments in Tamil Sentences Using Deep Learning

Hirushayini Loganathan
Department of Mathematics
Faculty of Science, Eastern University, Sri Lanka
hirushahiru5@gmail.com

Ratnasingam Sakuntharaj
Centre for Information and Communication Technology
Eastern University, Sri Lanka
sakuntharaj@esn.ac.lk

***Abstract* - Sentiment analysis is the process of extracting information from the given text in which the text consists of various sensations such as happiness, perturbation, pride, worry, and so on about various functions, human beings, systems, and facts. Sentimental analysis or opinion mining uses data mining and natural language processing techniques to discover, retrieve and filter the information and opinions from the World Wide Web's vast textual information. The sentiment analysers for European languages and some Indic languages are fully developed. However, Tamil, which is an under-resourced language with rich morphology, has not experienced these advancements. A few experiments have been conducted to determine the sentiments for Tamil text. An approach to doing the sentiment analysis for the Tamil language is proposed in this paper. The proposed approach uses Long Short-Term Memory, Convolutional Neural networks, and simple Deep Neural Network techniques. Test results show that the Long Short-Term Memory-based deep learning model performs well than the Convolutional Neural Network and simple Deep Neural Network for sentiment analysis of Tamil language with 94.10% accuracy.**

Keywords - BLSTM, deep learning, sentiment analysis, Tamil

I. INTRODUCTION

Tamil is a constituent of the Dravidian language family, primarily spoken by Tamils mostly in Sri Lanka, India, Malaysia, and Singapore. It has a significant number of speakers in emigrant communities around the world. It is one of the official languages in Sri Lanka [1] [2] [3] [4] [5] [6].

Sentiment analysis is the operation of extracting the information from the given text in which the text consists of various sensations such as happiness, perturbation, pride, worry, and so on about various functions, human beings, systems, and facts [7] [8]. If the sentence is subjective, sentence-level sentiment analysis defines whether the sentence determines a positive or negative opinion [9] [10] [11] [12].

Research in sentiment analysis has an essential impact on natural language processing, management sciences, political science, economics, and social sciences, as they all are affected by people's opinions. Social media comments, movie reviews, and emotions exposed by the text significantly impact human life, especially with the high usage of smart devices all over the world. These facilitate the users to share and access the opinions of individuals worldwide. This makes an explosive growth of data on the internet. Opinions are the key influences to almost all human activities and behaviours. We seek opinions before we make decisions. People rely only on their friends and relatives for opinions and comments about anything they wanted in the past. When a business organization needs public opinions about its products and services, it conducts surveys and opinion polls, which are tedious, time-consuming, and costly.

The fastest development of social media like Facebook, Twitter, LinkedIn, Instagram, and WhatsApp enables individuals to share their opinions publicly on the Web, which others can access for their decision-making. Individuals view these sites for the opinions of others before spending time, money, and interest on them. However, the major problem here is that even though volumes of data are available, the knowledge that can be acquired from the data still needs to be extracted, and the extraction is not a straightforward matter [13].

Every site contains huge opinionated text about reviews, emotions, places, products, and movies, but the reader cannot extract the opinions when the data becomes large. Thus, sentiment analysis systems are employed to extract and summarize information needed by the reader. Sentiment analysis has drawn attention in recent decades as an active research area in the field of Natural Language Processing to analyse people's opinions, reviews, evaluations, appraisals, attitudes, and emotions towards entities such as movies, songs, products, services, organizations, individuals, current issues, events, topics, and their attributes. Easily emotions could be understood when they were read, and they could be able to identify and separate. Moreover, most of the people shared their emotions in their native language and Tamil language the most. The important thing was that there are quiet approaches to finding the emotions in English but rare in Tamil. So, our model was developed for the Tamil language. The important thing in this work is the features of Tamil text are not enumerated and the text was only considered to extract the sentiments. The bidirectional Long Short-Term Memory model was proposed after comparing with another two models

After this introductory section, the rest of this paper is organized as follows: Section II summarizes the previous work in sentiment analysis of the Tamil Language. Section III describes our methodology. Test results are discussed in Section IV, with the conclusion in Section V.

II. LITERATURE REVIEW

Six reported works [14] [15] [16] [17] [18] [19] related to Tamil sentiment analyser were found in the literature review.

Anbukkarasi and Varadaganapathy [14] have proposed a method to analyse sentiments for Tamil tweets using the deep learning technique. Bi-directional long short-term memory with combined character-based neural networks is used to analyse the Tamil tweets. Moreover, unknown words were handled by using the n-gram technique. How far the system works successfully was not reported.

Arunselvan *et al.* [15] have used Multinomial Naive Bayes, Bernoulli Naive Bayes, Logistic Regression, Random Kitchen Sink, and Support Vector Machines to analyse sentiments on movie reviews. Term Frequency of unigram and bigram for unique words of the movie review corpus were used as features by varying values of N (N number of counts). The authors claimed that the combination of Polynomial, Radial Basis Functions, Support Vector Machines with Linear, Sigmoid and Pre-computed Kernels approach was 64.69% more successful for bigram features using Radial Basis Kernel with N=10 than other kernels. Moreover, the authors claimed that this approach was 61.88% and 57.14% successful for Random Kitchen Sink and Logistic Regression approaches, respectively for bigrams, and 61.01% and 60.19% successful for Multinomial Naive Bayes and Bernoulli Naive Bayes, respectively for unigram features.

Shriya *et al.* [16] have proposed a model using Support Vector Machines, Maximum Entropy classifier, Decision Tree, and Naive Bayes to predict sentiments from Tamil movie reviews as positive or negative. Context words of training data, apostrophes, and punctuation were used as features. The authors claimed that this approach was 66.29% and 75.96% successful for Decision Tree techniques and Support Vector Machines, respectively.

Anand Kumar and Soman [17] have used the SAIL dataset for sentiment analysis in three languages such as Tamil, Hindi, and Bengali. Recurrent Neural Network was used to determine the reviews into positive, negative, and neutral. The authors claimed that this approach was 88.23%, 72.01%, and 65.16% successful for Tamil, Hindi, and Bengali languages respectively.

Ravishankar and Raghunathan [18] have proposed a corpus-based sentiment classification of Tamil movie tweets into five categories such as Sandai (Stunt), Kadhal (Romance), Masala (Formula), Kudumbam (Family), and Comedy. Seven thousand of tweets were collected from 100 Tamil movies for their study. TF-IDF, Domain Specific Tags, and Tweet Weight Models techniques were used for classification along with Tamil Agarathi. The authors claimed that this approach was 29.87%, 35.64%, and 40.07% successful for TF-IDF, TF-IDF with Domain Specific Tags, and Tweet Weight Models respectively.

Ravishankar *et al.* [19] have proposed n-gram-based sentiment categorization. Corpus was created from 100 movie reviews. TF-IDF and unigrams, bigrams, and trigrams were used in the feature extraction phase, and Grams were used for negation handling. Grammar rules were used to classify the Tweets of length less than four words. Trigrams were applied for length less than or equal to eight words, and longer ones were ignored. The authors claimed that this approach was 29.87% and 61.29% successful in TF-IDF and n-grams, respectively.

III. METHODOLOGY

The objective of this work is to predict sentiments from Tamil movie reviews using a deep learning approach. The proposed approach uses *Bi-directional Long Short-Term Memory* (BLSTM) for developing a sentiment analyser for the Tamil Language. For this, the annotated corpus provided by the shared task Dravidian CodeMix-FIRE2020 was used [20]. It consists of 14720 sentences. These sentences are based on the movie reviews, and they contain five different kinds of sentiments: *Positive*, *Negative*, *Neutral*, *Conflict*, and *Unknown*. The overall workflow of the proposed sentiment analyser is projected in Fig.1.

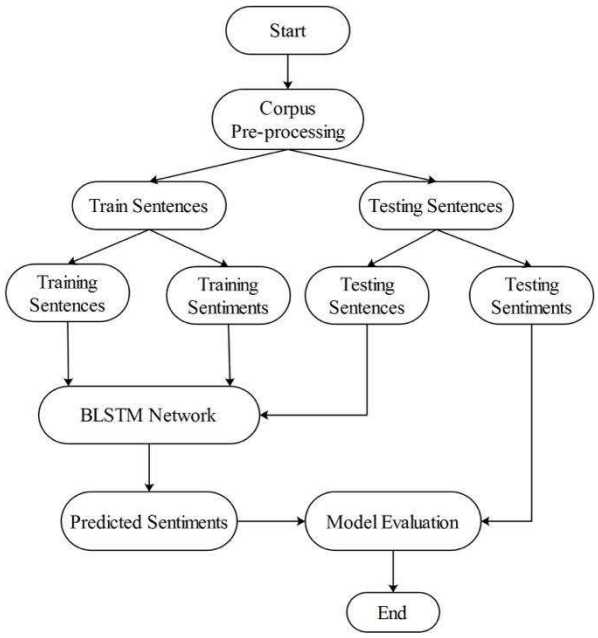

Fig. 1. Overall workflow of the proposed sentiment analyser.

A. Pre-Processing

The dataset which was used in the experiment consists of lots of unnecessary tokens. It will reduce the algorithm's capability. In the pre-processing step, all the unnecessary tokens such as punctuation and HTML tags were removed from the dataset. After the pre-processing, the trained sentences were tokenized into individual sentences and sentiments. In testing, this unit removes the sentiments and then tokenizes the sentences.

B. BLSTM Network

The deep neural network Bi-directional Long Short Memory (BLSTM) was used for Tamil sentiment analysis in this research work. BLSTMs are an extension of LSTMs. LSTM processes the sequence of data in a *forward* direction only, whereas BLSTM can do this in both *forward* and *backward* directions. The BLSTM can track both past and

future information by processing the sequence of data in both directions [21] [22] [23]. The BLSTM network consists of an input layer, hidden layers, and an output layer shown in Fig.2.

The input layer of the BLSTM model takes sentences and their corresponding sentiments as input during the training phase. The embedding layer converts the textual data into numeric data, and then a word-to-index dictionary was created. In the word-to-index dictionary, each word in the corpus was used as a key, while the corresponding unique index was used as the value for the key. The training set contained the number of lists, where each list contains integers that corresponded to each sentence in the training set. The size of each list varied because of the different lengths of each sentence. Then the longest sequence is found from the variable length of sequences, and the rest of the sequences are padded to that size because the deep learning model only works with the vectorized representation of the data with a fixed size of sequences. Then *Glove Embedding* was used to create the feature matrix [24]. Further, to compile our model, the Adam optimizer was used. Adam is the best among the adaptive optimizers in most cases. Optimizers are classes or methods of deep learning models such as weight and learning rate to reduce the losses and help to get results faster. Adam optimization is a stochastic gradient descent method that is based on the adaptive estimation of first-order and second-order moments [25].

C. DNN and CNN Network

Moreover, to evaluate the proposed model (BLSTM), simple Deep Neural Network (DNN) and Convolutional Neural Network (CNN) was applied to the dataset. The feature matrix was created for the variables using GloVe embedding. GloVe embedding was loaded, and the dictionary was created, and it contained words as keys and their corresponded embedding lists as a value. After the embedding, the deep learning model was introduced to the data. Then sequential model and embedding layer were created. The densely connected layer was connected directly to the embedding layer, and the embedding layer was flattened. After that sigmoid function was activated and finally, the model was compiled. For the next part, the convolutional neural network was created with one convolutional layer and one pooling layer. The embedding layer was created, and then the pooling layer was added to the rest. At this point, the processes created a simple neural network again up to the embedding layer to the CNN model. Here also, the sequential model was created and followed by the embedding layer. Next one-dimensional convolutional layer with 128 features or kernels was created. Kernel size is 5 and the 'sigmoid' function was activated and then global max pooling layer and dense layer were added with 'sigmoid activation'. Finally, the compilation was done. Here the important thing was that flattening the embedding layer is not needed, and the feature size is also reduced with the use of the pooling layer.

After training the model, the test sentences were given to the model to evaluate its performance of the model. Finally, the model evaluation unit measured the performance of the system by comparing predicted sentiments with actual sentiments in the test sentences.

IV. TEST RESULTS AND DISCUSSION

TABLE I. NUMBER OF SENTENCES IN EACH SENTIMENT

Sentiments	**No. of Sentences**
Positive	10064
Negative	1914
Neural	1710
Conflict	688
Unknown	344

The proposed deep learning-based sentiment analyzer for the Tamil language was developed using python 3.9 with Keras deep learning library. The system was tested using *k-fold cross validation*, and the *k* value is *five*. Hence, the entire data in the corpus was split into five-folds, and then out of five folds, four folds were used for training and the remaining one-fold was used for testing. This experiment process was repeated until every fold served as a test set. Then the average of the recorded score was calculated to determine the performance of the proposed model. The experiments were run for *fifty epochs,* and the epoch size was determined using a callback called *Early Stopping* [26]. Table II shows the cross-validation score for k=5 for Dravidian CodeMix-FIRE2020 Corpus.

According to the obtained results for the cross-validation score for k=5, the proposed system gives approximately 94% accuracy for each iteration. Moreover, the proposed system was tested with the following amount of training and testing sentences.

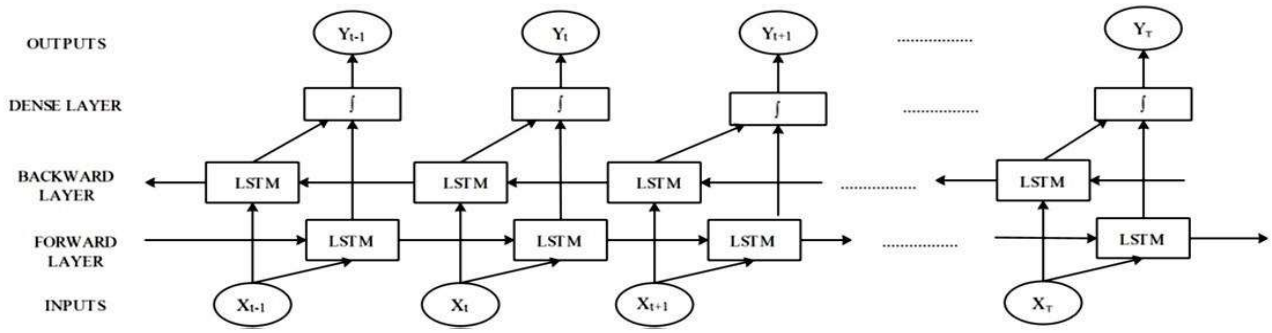

Fig. 2. The BLSTM Network

TABLE II. CROSS VALIDATION SCORE FOR k=5 FOR DRAVIDIAN CODEMIX-FIRE2020 CORPUS

Iteration	Accuracy
Iteration 1	94.26%
Iteration 2	94.14%
Iteration 3	94.37%
Iteration 4	94.22%
Iteration 5	94.19%

- **Test 1:** corpus is split into 10% for training and 90% for testing
- **Test 2:** corpus is split into 20% for training and 80% for testing
- **Test 3:** corpus is split into 30% for training and 70% for testing
- **Test 4:** corpus is split into 40% for training and 60% for testing
- **Test 5:** corpus is split into 50% for training and 50% for testing
- **Test 6:** corpus is split into 60% for training and 40% for testing
- **Test 7:** corpus is split into 70% for training and 30% for testing
- **Test 8:** corpus is split into 80% for training and 20% for testing
- **Test 9:** corpus is split into 90% for training and 10% for testing

Test results for each test case (from Test 1 to Test 9) are shown in Table III.

TABLE III. TEST RESULTS FOR EACH TEST CASE

Test Cases	Accuracy
Test 1	40.29%
Test 2	44.30%
Test 3	45.97%
Test 4	52.87%
Test 5	63.15%
Test 6	72.64%
Test 7	81.59%
Test 8	94.13%
Test 9	90.10%

According to the obtained results, the accuracy of the sentiment analyser increases when the amount of training data increases. Therefore, the performance of the proposed system also depends on the amount of training data.

When considering the performance of the models for sentiment analysis for Tamil language, the simple DNN model gives 81.21% maximum accuracy in epoch size 13, the CNN model gives 85.26% maximum accuracy in epoch size 14, and the BLSTM model gives 94.10% maximum accuracy in epoch size 10. When comparing the performance, the BLSTM model performs better than the CNN and Simple DNN for sentiment analysis of the Tamil language. In this research, an approach has been proposed for sentiment analysis of the Tamil language using BLSTM model based on the obtained test results.

V. CONCLUSION

In this paper, we have proposed a deep learning-based sentiment analyser for the Tamil language. The BLSTM network was used as a deep learning model in this approach. We used Dravidian CodeMix-FIRE2020 annotated corpus for this experiment. Test results show that the sentiments for

Tamil sentences determined by this approach are found to be with 94% accuracy. Moreover, the test shows that the amount of training data affects the performance of the sentiment analyser.

REFERENCES

[1] M. A. Nuhman, "Basic Tamil Grammar". Department of Tamil, University of Peradeniya: Readers Association, Kalmunai, 2013.

[2] V. Sangar, "Tamil Grammar".Puduchcheri, India: Nanmozi Printers, 2006.

[3] A. Navalar, "Tamil Grammar Questions, and Answers". No. 366, Kankesanthurai Road, Jaffna: Vannai Santhayarmadam, 1998. [4] M. Balasubramanium, Studies in Tholkappiyam, Annamalai University, 2001.

[5] B. Krishnamurti, "Tamil language." [Online]. Available: https: //www.britannica.com/topic/Tamil-language

[6] Wikipedia, "List of countries where Tamil is an official language." [Online]. Available: https://simple.wikipedia.org/wiki/List_of_countries_where_Tamil_is_an_official_languagee

[7] F. Miedema and S. Bhulai, "Sentiment analysis with long short-term memory networks," 2018.

[8] L. Senevirathne, P. Demotte, B. Karunanayake, U. Munasinghe, and S. Ranathunga, "Sentiment analysis for sinhala language using deep learning techniques," ArXiv, vol. abs/2011.07280, 2020.

[9] B. Lutz, N. Pröllochs, and D. Neumann, "Sentence-level sentiment analysis of financial news using distributed text representations and multi-instance learning," CoRR, vol. abs/1901.00400, 2019. [Online]. Available: http://arxiv.org/ abs/1901.00400

[10] J. Shen, X. Liao, and Z. Tao, "Sentence-level sentiment analysis via BERT and BiGRU," in 2019 International Conference on Image and Video Processing, and Artificial Intelligence, R. Su, Ed., vol. 11321, International Society for Optics and Photonics. SPIE, 2019, pp. 658 – 663. [Online]. Available: https://doi.org/10.1117/12.2550215

[11] V. S. Shirsat, R. S. Jagdale, and S. N. Deshmukh, "Document level sentiment analysis from news articles," in 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), 2017, pp. 1–4.

[12] G. Choi, S. Oh, and H. Kim, "Improving document-level sentiment classification using importance of sentences," CoRR, vol. abs/2103.05167, 2021. [Online]. Available: https://arxiv. org/abs/2103.05167

[13] N. Sadat and J. B. Ibrahim, "The impact of social media on society," Imperial journal of interdisciplinary research, vol. 3, 2017.

[14] S. Anbukkarasi and S. Varadhaganapathy, "Analyzing sentiment in tamil tweets using deep neural network," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), pp. 449–453, 2020.

[15] S. J. Arunselvan, M. A. Kumar, and K. P. Soman, "Sentiment analysis of tamil movie reviews via feature frequency count," International journal of applied engineering research, vol. 10, 2015.

[16] S. Se, R. Vinayakumar, M. A. Kumar, and K. P. Soman, "Predicting the sentimental reviews in tamil movie using machine learning algorithms," Indian journal of science and technology, vol. 9, 2016.

[17] S. Seshadri, A. b Madasamy, S. K. Padannayil, and M. A. Kumar, "Analyzing sentiment in indian languages micro text using recurrent neural network," 2016.

[18] N. Ravishankar and S. Raghunathan, "Corpus based sentiment classification of tamil movie tweets using syntactic patterns," 2017.

[19] N. Ravishankar, R. Shriram, K. Vengatesan, S. Mahajan, P. Sanjeevikumar, and S. Umashankar, "Grammar rule-based sentiment categorization model for tamil tweets," 2018.

[20] "Dravidian-codemix - fire 2020," 2020. [Online]. Available: https://dravidian-codemix.github.io/2020/datasets.html

[21] H. Visuwalingam, R. Sakuntharaj, and R. G. Ragel, "Part of speech tagging for tamil language using deep learning," in 2021 IEEE 16th International Conference on Industrial and Information Systems (ICIIS), 2021, pp. 157–161.

[22] K. K. Akhil, R. P. Rajimol, and V. S. Anoop, "Parts-of-speech tagging for malayalam using deep learning techniques," International Journal of Information Technology, pp. 1–8, 2020.

[23] R. Rajan, A. J. Joseph, E. K. Robin, and N. T. K. Fathima, "Part-of-speech tagger in malayalam using bi-directional lstm," in 2020 23rd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA), 2020, pp. 22–27.

[24] T. Shi and Z. Liu, "Linking glove with word2vec," ArXiv, vol. abs/1411.5595, 2014.

[25] I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, "Adam optimization algorithm for wide and deep neural network," Knowl. Eng. Data Sci., vol. 2, pp. 41–46, 2019.

[26] J. Brownlee, "Use early stopping to halt the training of neural networks at the right time," 2020. [Online]. Available: https://machinelearningmastery.com/how-to-stop-trainingdeep-neural-networks-at-the-right-time-using-early-stopping/