# NITP-AI-NLP@Dravidian-CodeMix-FIRE2020: A Hybrid CNN and Bi-LSTM Network for Sentiment Analysis of Dravidian Code-Mixed Social Media Posts

Abhinav Kumar[a], Sunil Saumya[b] and Jyoti Prakash Singh[a]

[a]National Institute of Technology Patna, Patna, India
[b]Indian Institute of Information Technology Dharwad, Karnataka, India
[a]National Institute of Technology Patna, Patna, India

**Abstract**

The sentiment analysis is one of the important tasks in the field of natural language processing. Many works have been proposed recently by the research community to find the sentiment from English social media posts. Nevertheless, very little work has been proposed to find sentiments from the Dravidian code-mixed Malayalam and Tamil social media comments. In this work, we have proposed two-hybrid neural network models based on Convolutional Neural Network (CNN) and Bidirectional Long-Short-Term-memory (Bi-LSTM) network. We utilized both character and word embedding of the YouTube comments to learn robust features from the text. The proposed hybrid CNN-CNN network achieved a promising weighted $F_1$-score of 0.69 for Malayalam code-mixed text, whereas the CNN-Bi-LSTM network achieved a promising weighted $F_1$-score of 0.61 for Tamil code-mixed text.

**Keywords**

Sentiment analysis, Code-mixed, Tamil, Malayalam, YouTube, Machine learning, Deep learning

## 1. Introduction

Sentiment analysis helps to recognize opinions or answers on a specific subject. It is one of the most researched topics in natural language processing due to its significant impact on businesses like e-commerce, spam detection [1, 2], recommendation system, social media monitoring [3], and name a few. English is the most preferable and acceptable language worldwide and very prevalent in the digital world. However, in a country like India, having over 400 million internet users speaks more than one language to communicate their thoughts or emotions, producing a new code-mixed language [4, 5]. The issue with the code-mix language is that it contains more than one script and language constructs. Most of the existing models trained to extract a single language's sentiment fail to capture a code-mixed language semantics. Extracting sentiments from code mixed user-generated texts becomes more difficult due to its multilingual nature.

Recently, the sentiment analysis of code-mixed language [6, 7] has drawn attention from the research community. Joshi et al. [8] presented a model with subword representation of

*FIRE 2020: Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India*
abhinavanand05@gmail.com (A. Kumar); sunil.saumya@iiitdwd.ac.in (S. Saumya); jps@nitp.ac.in (J.P. Singh)

code-mix data and long short term memory (subword-LSTM) for sentiment analysis of Hinglish (Hindi-English) dataset. Priyadharshini et al. [9] used subword representation for named entity recognition in code-mixed Hindi-English text. A model with a support vector machine that uses character n-grams features for Bengali-English code mixed data was reported by [4]. Advani et al. [10] used logistic regression with handcrafted lexical and semantic features to extract sentiments from Hinglish and Spanglish (Spanish + English) data. Goswami et al. [11] proposed a morphological attention model for sentiment analysis on Hinglish data.

The Malayalam language is one of the Dravidian languages spoken in the Indian state of Kerala. There are almost 38 million Malayalam speakers over the globe. Another famous Dravidian language in India's southern region is Tamil, which is being spoken by Tamil people in India, Singapore, and Sri Lanka [12]. The scripts of both Dravidian languages are alpha-syllabic, which is partially alphabetic and partially syllable-based [13]. However, people on social media frequently utilize Roman script for writing because it is easy to write through keyboards available on the devices [14]. For these under-resourced languages, thus, the majority of the data available in social media are code-mixed.

The objective of the current study is to extract sentiment from code-mixed Dravidian languages Tanglish and Manglish. The data of the *Dravidian-CodeMix-FIRE2020 challenge* [15, 16] was collected from the social media platform YouTube. Each instance or post in the data typically has one sentence, and in a few cases, it is more than one. Every instance is labeled with one of the sentiment polarities "positive, negative, mixed emotion, unknown state, and if the post is not in the said Dravidian languages". The current paper develops two different hybrid neural networks based on Convolutional Neural Network (CNN) and Bi-directional Long-Short-Term-Memory (Bi-LSTM) networks. In the proposed hybrid models, both character and word embedding vectors of the text are used to get the text's robust textual features.

In the rest of the paper, the dataset description, the proposed methodology is explained in Section 2. The various experiments and their finding is presented in Section 3. Finally, Section 4 concludes the discussion by highlighting the main findings of this study.

## 2. Methodology

The detail description of the proposed hybrid Convolutional Neural Network (CNN) and Bidirectional Long-Short-Term-Memory (Bi-LSTM) networks are discussed in this section. We have proposed two different hybrid deep neural network models: (i) CNN (c) + CNN (w) model: in this model, two parallel CNN networks are used to extract the character level (c) and word level features (w) from the text. For the first CNN network character embedding of the text is given as the input to the network, whereas for the second CNN network, word embedding of the text is given as the input to the network. The model diagram for the hybrid CNN (c) + CNN (w) can be seen from Figure 1. (ii) CNN (c) + Bi-LSTM (w): in this model, similar to the previous CNN (c) + CNN (w) model, charcter embedding is given as input in CNN whereas word embedding is given as input in Bi-LSTM network. The model diagram for the hybrid CNN (c) + Bi-LSTM (w) can be seen from Figure **??**. The detailed description regarding the number of layers, parameters, and type of embedding can be seen in 2.2 and 2.3.

We have removed multiple spaces between the words into one for the data pre-processing,

**Table 1**
Data statistics used in this study

| | Class | Training | Development | Testing |
|---|---|---|---|---|
| Malayalam (code-mixed) | Mixed feelings | 289 | 44 | 70 |
| | Negative | 549 | 51 | 138 |
| | Positive | 2022 | 224 | 565 |
| | Not-Malayalam | 647 | 60 | 177 |
| | Unknown state | 1344 | 161 | 398 |
| | Total | 4851 | 540 | 1348 |
| Tamil (code-mixed) | Mixed feelings | 1283 | 141 | 377 |
| | Negative | 1448 | 165 | 424 |
| | Positive | 7627 | 857 | 2075 |
| | Not-Tamil | 368 | 29 | 100 |
| | Unknown state | 609 | 68 | 173 |
| | Total | 11335 | 1260 | 3149 |

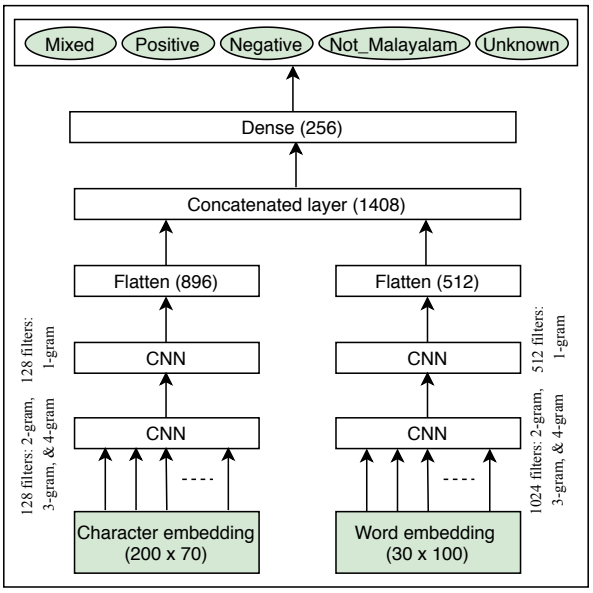


**Figure 1:** Model diagram for hybrid CNN(c)+CNN(w)

and we replaced &, @ symbols to their English words *'and'* and *'at'*, respectively. We also replaced numeric values into their corresponding English words (e.g., *'1'* is replaced by *'one'*, *'2'* is replaced by *'two'* and so on). The data statistic for the given task is presented in Table 1.

## 2.1. Character and Word embedding vectors

Each character of the YouTube comments is encoded into one-hot vector to get the character embedding. We fixed 200-characters for each posts. In our character vocabulary we found seventy different characters such as alphabets, numbers, and special symbols. Therefore, each

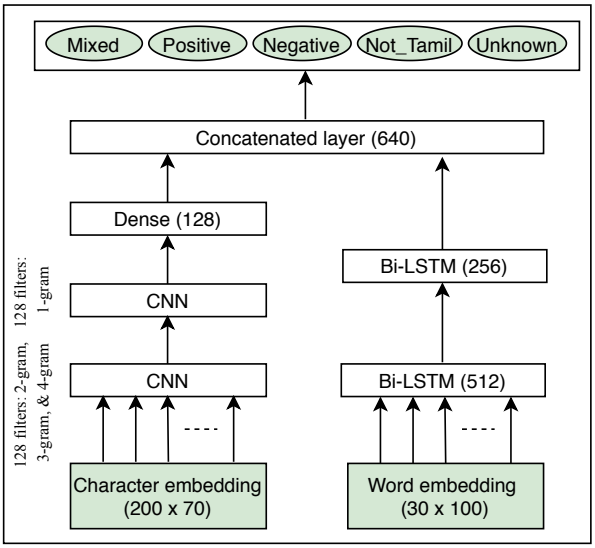


**Figure 2:** Model diagram for hybrid CNN(c)+ Bi-LSTM(w)

**Table 2**
Best suited hyper-parameters for the proposed models

| Hyper-parameters | CNN(c) + CNN (w) | CNN(c) + Bi-LSTM(w) |
|---|---|---|
| Number of CNN layers | 2, 2 | 2 |
| Number of Bi-LSTM layers | - | 2 |
| Batch size | 32 | 32 |
| Epochs | 200 | 200 |
| Loss | Categorical crossentropy | Categorical crossentropy |
| Optimizer | Adam | Adam |
| Activation function | ReLU, Softmax | ReLU, Softmax |
| Dropout rate | 0.2 | 0.2 |
| Pooling window | 5 | 5 |

social media post is converted into a 200 × 70 dimensional character embedding matrix. Then this matrix is used by the CNN network for their convolution process. For word embedding, we trained a FastText[1] model by using Tamil code-mixed and Malayalam code-mixed corpus separately. Each word of the corpus is converted into a 100-dimensional vector. In our case, we fixed 30-words for each of the YouTube comments. Therefore, each post is converted into (30 × 100) dimensional word embedding matrix. These character embedding and word embedding matrix is then used in our proposed hybrid models.

[1]https://fasttext.cc/

**Table 3**
Results of Malayalam and Tamil code-mixed sentiment analysis

| | | CNN (c) + CNN (w) | | | CNN (c) + Bi-LSTM (w) | | |
|---|---|---|---|---|---|---|---|
| | Class | Precision | Recall | $F_1$-score | Precision | Recall | $F_1$-score |
| Malayalam | Mixed feelings | 0.42 | 0.41 | 0.42 | 0.34 | 0.39 | 0.36 |
| | Negative | 0.62 | 0.54 | 0.57 | 0.55 | 0.54 | 0.54 |
| | Positive | 0.72 | 0.82 | 0.77 | 0.73 | 0.79 | 0.76 |
| | Not-Malayalam | 0.79 | 0.65 | 0.71 | 0.79 | 0.71 | 0.74 |
| | Unknown state | 0.66 | 0.62 | 0.64 | 0.68 | 0.63 | 0.66 |
| | Weighted avg. | **0.69** | **0.69** | **0.69** | 0.69 | 0.68 | 0.68 |
| Tamil | Mixed feelings | 0.21 | 0.04 | 0.06 | 0.19 | 0.11 | 0.14 |
| | Negative | 0.36 | 0.22 | 0.27 | 0.33 | 0.26 | 0.29 |
| | Positive | 0.71 | 0.91 | 0.80 | 0.73 | 0.85 | 0.79 |
| | Not-Tamil | 0.58 | 0.49 | 0.53 | 0.63 | 0.59 | 0.61 |
| | Unknown state | 0.26 | 0.11 | 0.15 | 0.29 | 0.16 | 0.21 |
| | Weighted avg | 0.57 | 0.65 | 0.59 | **0.59** | **0.64** | **0.61** |

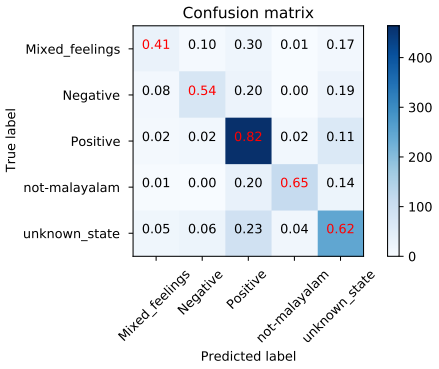


**Figure 3:** Confusion matrix for Malayalam code-mixed post in case of CNN (c) + CNN (w)

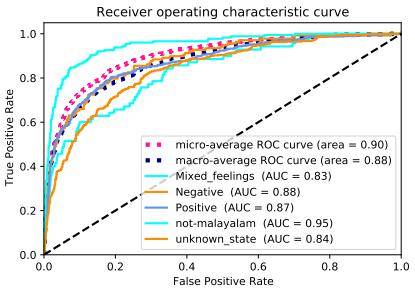


**Figure 4:** ROC curve for Malayalam code-mixed post in case of CNN (c) + CNN (w)

## 2.2. CNN (c) + CNN (w) model

The overall diagram for the hybrid CNN (c) + CNN (w) model can be seen from Figure 1. Two parallel CNN network is used one to process character embedding matrix and other one is to process word embedding matrix. To process character embedding matrix, two layers of CNN is used. In the first CNN layer, 128 filters of 2-gram, 3-gram, and 4-gram are used, whereas in the second CNN layer 128 filters of 1-gram are used. Similarly, to process word embedding matrix, two layers of CNN in used. In the first CNN layer, 1024 filters of 2-gram, 3-gram, and 4-gram filters are used, where in the next CNN layer 512 filters of 1-gram are used. Finally, the flattened vectors from both the parallel CNN networks are concatenated and passed to a dense layer having 256-neurons. Finally, the output of dense layer is passed to a softmax layer to get its class probability. As the performance of the deep neural networks are very sensitive to the selected hyper-parameters, we experimented by varying the batch sizes, dropout rates,

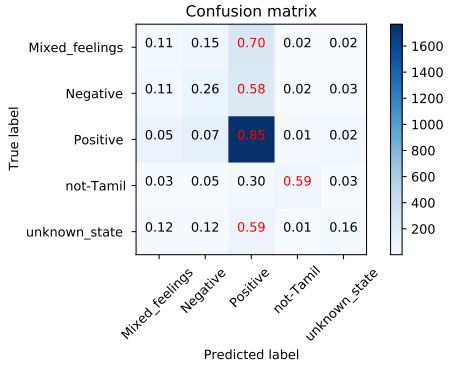


**Figure 5:** Confusion matrix for Tamil code-mixed post in case of CNN (c) + Bi-LSTM (w)

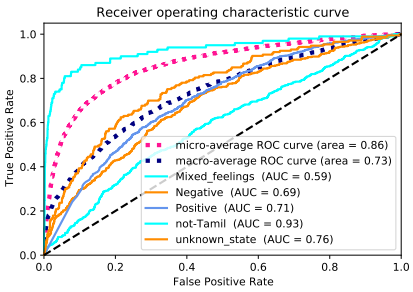


**Figure 6:** ROC curve for Tamil code-mixed post in case of CNN (c) + Bi-LSTM (w)

pooling window, and epochs. The best suited hyper-parameter of the proposed CNN (c) + CNN (w) model is listed in Table 2.

### 2.3. CNN (c) + Bi-LSTM (w) model

The overall diagram for the hybrid CNN (c) + Bi-LSTM (w) model can be seen from Figure **??**. The CNN network of the proposed hybrid CNN (c) + Bi-LSTM (w) model takes the character embedding matrix as an input. In the first CNN layer, 128 filters of 2-gram, 3-gram, and 4-gram are used, whereas as in the second CNN layer, 128 filters of 1-gram are used in our model. Then the extracted features from two consecutive CNN layer is passed through a dense layer having 128 neurons. Similarly, word embedding is given input to two Bi-LSTM layers with a 512-dimensional output vector at the first layer and a 256-dimensional output vector at the second layer. The second Bi-LSTM layer's output vector is then concatenated with the 128-dimensional output vector of CNN (followed by dense) model, as can be seen in Figure **??**. Finally, the concatenated vector is passed through a softmax layer to get the class probability. The best suited hyper-parameters for the proposed CNN (c) + Bi-LSTM (w) can be seen in Table 2. The detailed description of the CNN and Bi-LSTM network can be seen in [17, 18, 19, 20, 21].

## 3. Results

In the given task of *Dravidian-CodeMix-FIRE2020 workshop*, participants had to classify code-mixed (written in roman script) Tamil and Malayalam social media posts into five different sentiment classes: (i) Mixed feelings, (ii) Positive, (iii) Negative, (iv) Not related to that language (Not-Tamil/Not-Malayalam), and (v) Unknown state. The results of code-mixed Malayalam posts for both the CNN (c) + CNN (w) and CNN (c) + Bi-LSTM (w) models are listed in Table 3. After comparing the results of both proposed models, it was found that CNN (c) + CNN (w) performed better for code-mixed Malayalam posts with precision, recall, and $F_1$-score of 0.69. The confusion matrix and ROC curve for the code-mixed Malayalam posts can be seen in Figures 3 and 4, respectively.

The results of code-mixed Tamil post for both the CNN (c) + CNN (w) and and CNN (c) + Bi-LSTM (w) models are listed in Table 3. The proposed CNN (c) + Bi-LSTM (w) model performed better as compared to another model and achieved a precision of 0.59, recall of 0.64, and an $F_1$-score of 0.61. The confusion matrix and ROC curve for the code-mixed Tamil posts can be seen in Figures 5 and 6, respectively.

## 4. Conclusion

Sentiment analysis of the textual contents has significant uses in various natural language processing tasks. In this work, we proposed two-hybrid deep neural networks based on CNN and Bi-LSTM networks. We used both character and word embedding vectors in the proposed hybrid CNN (c) + CNN (w) and CNN (c) + Bi-LSTM (w) models that achieved promising performance in the classification of code-mixed Malayalam and Tamil YouTube comments. The proposed CNN (c) + CNN (w) network achieved a weighted $F_1$-score of 0.69 for Malayalam code-mixed text, whereas the CNN (c) + Bi-LSTM (w) network achieved a weighted $F_1$-score of 0.61 for Tamil code-mixed text.

## References


[1] S. Saumya, J. P. Singh, Spam review detection using lstm autoencoder: an unsupervised approach, Electronic Commerce Research (2020) 1–21, doi.org/10.1007/s10660–020–09413–4.

[2] S. Saumya, J. P. Singh, Detection of spam reviews: A sentiment analysis approach, CSI Transactions on ICT 6 (2018) 137–148.

[3] S. Saumya, J. P. Singh, P. Kumar, Predicting stock movements using social network, in: Conference on e-Business, e-Services and e-Society, Springer, 2016, pp. 567–572.

[4] B. G. Patra, D. Das, A. Das, Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task@ icon-2017, arXiv preprint arXiv:1803.06745 (2018).

[5] N. Jose, B. R. Chakravarthi, S. Suryawanshi, E. Sherly, J. P. McCrae, A survey of current datasets for code-switching research, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020.

[6] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, J. P. Sherly, Elizabeth McCrae, Overview of the track on Sentiment Analysis for Davidian Languages in Code-Mixed Text, in: Working Notes of the FIRE 2020. CEUR Workshop Proceedings., 2020.

[7] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, J. P. Sherly, Elizabeth McCrae, Overview of the track on Sentiment Analysis for Davidian Languages in Code-Mixed Text, in: Proceedings of the 12th FIRE, FIRE '20, 2020.

[8] A. Joshi, A. Prabhu, M. Shrivastava, V. Varma, Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 2482–2491.

[9] R. Priyadharshini, B. R. Chakravarthi, M. Vegupatti, J. P. McCrae, Named entity recognition for code-mixed Indian corpus using meta embedding, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020.
[10] L. Advani, C. Lu, S. Maharjan, C1 at semeval-2020 task 9: Sentimix: Sentiment analysis for code-mixed social media text using feature engineering, arXiv preprint arXiv:2008.13549 (2020).
[11] K. Goswami, P. Rani, B. R. Chakravarthi, T. Fransen, J. P. McCrae, Uld@ nuig at semeval-2020 task 9: Generative morphemes with an attention model for sentiment analysis in code-mixed text, arXiv preprint arXiv:2008.01545 (2020).
[12] B. R. Chakravarthi, Leveraging orthographic information to improve machine translation of under-resourced languages, Ph.D. thesis, NUI Galway, 2020.
[13] B. R. Chakravarthi, M. Arcan, J. P. McCrae, Comparison of Different Orthographies for Machine Translation of Under-Resourced Dravidian Languages, in: 2nd Conference on Language, Data and Knowledge (LDK 2019), volume 70 of *OpenAccess Series in Informatics (OASIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019, pp. 6:1–6:14. URL: http://drops.dagstuhl.de/opus/volltexte/2019/10370. doi:10.4230/OASIcs.LDK.2019.6.
[14] A. Kumar, J. P. Singh, S. Saumya, A comparative analysis of machine learning techniques for disaster-related tweet classification, in: 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129), IEEE, 2019, pp. 222–227.
[15] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, Corpus creation for sentiment analysis in code-mixed Tamil-English text, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France, 2020, pp. 202–210. URL: https://www.aclweb.org/anthology/2020.sltu-1.28.
[16] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, A sentiment analysis dataset for code-mixed Malayalam-English, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France, 2020, pp. 177–184. URL: https://www.aclweb.org/anthology/2020.sltu-1.25.
[17] A. Kumar, J. P. Singh, Location reference identification from tweets during emergencies: A deep learning approach, International journal of disaster risk reduction 33 (2019) 365–375.
[18] S. Saumya, J. P. Singh, Y. K. Dwivedi, Predicting the helpfulness score of online reviews using convolutional neural network, Soft Computing 24 (2020) 10989–11005, https://doi.org/10.1007/s00500–019–03851–5.
[19] A. Kumar, J. P. Singh, Y. K. Dwivedi, N. P. Rana, A deep multi-modal neural network for informative twitter content classification during emergencies, Annals of Operations Research (2020) 1–32.
[20] J. P. Singh, A. Kumar, N. P. Rana, Y. K. Dwivedi, Attention-based lstm network for rumor veracity estimation of tweets, Information Systems Frontiers (2020) 1–16.
[21] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint

arXiv:1508.01991 (2015).