Sentiment Analysis in Dravidian Code-Mixed YouTube Comments and Posts

Sanjeepan Sivapiran, Charangan Vasantharajan and Uthayasanker Thayasivam

Department of Computer Science and Engineering, University of Moratuwa

Abstract

This paper presents the methodologies implemented while doing Sentiment Analysis on Dravidian code-mixed comments and posts collected from social media. With a dataset of code-mixed Tamil, We experimented with transformer-based models such as multilingual BERT and DistilBERT and ULMFiT. This work submitted to the track **'Sentiment Analysis for Dravidian Languages in Code-Mixed Text'** organized by the Forum for Information Retrieval Evaluation. Although it received the seventh rank for the Tamil task in the benchmark, This paper improves upon the results by a margin to attain the final weighted F1 score of 0.61 for the Tamil task.

Keywords

Sentiment Analysis,, Code-Mixed, Transformers, Tamil, ULMFiT

1. Introduction

In the past few years, usage of social media platforms has drastically increased. With this trend, cyberbullying and hate speech also increased and created a need to analyze comments/posts on social media. Sentimental Analysis is a study that uses Natural Language Processing in identifying subjective opinions or emotional responses about a given topic.[1] There are already multiple steps taken to make use of sentimental Analysis in monolingual texts. But there has been an indispensable demand for sentimental Analysis in code-mixed Dravidian languages (Tamil, Malayalam, and Kannada) [2]. Code-mixing is a prevalent phenomenon in a multilingual community, and the code-mixed texts sometimes write in non-native scripts.[3] Systems trained on monolingual data fail on code-mixed data due to the complexity of code-switching at different linguistic levels in the text.

The objective of our study is to classify YouTube comments into positive, negative, neutral, mixed emotions or if the word is not in Tamil, which is in code-mixed form [4]. For this task, transformer architecture models Like multilingual BERT and DistilBERT yielded good results since they optimized for low-resourced languages like Tamil. Yet ULMFiT made the best results compared to transformer models. Since data was in code-mixed form, models had difficulty

FIRE 2021: Forum for Information Retrieval Evaluation, December 13-17,2021, India.

sanjeepan.18@cse.mrt.ac.lk (S. Sivapiran); charangan.18@cse.mrt.ac.lk (C. Vasantharajan); rtuthaya@cse.mrt.ac.lk (U. Thayasivam)
https://www.linkedin.com/in/sanjeepan/ (S. Sivapiran); https://chaarangan.github.io/ (C. Vasantharajan); https://rtuthaya.lk/ (U. Thayasivam)
0000-0001-7225-2432 (S. Sivapiran); 0000-0001-7874-3881 (C. Vasantharajan); 0000-0002-3936-8174 (U. Thayasivam)

understanding semantic relationships and their respective contexts. We used the translation and transliteration technique to convey a word from one writing system to another while preserving the context and semantics to overcome this issue.

The rest of the sections in the paper are as follows. Section 2 reviews related experiment works in Sentiment Analysis. Section 3 describes the given dataset in the Shared Task[5]. The fourth section(4) presents the system description and conducted experiments using different approaches and features as well as the results reaped from the experiments of our proposed system. Benchmark results are discussed in section 4.5 and finally, the conclusion.

2. Related Work

Cyberbullying and hateful speech are unpleasant parts of social media. To ensure the well-being of the social media users from cyberbullying, social media companies always had to invest/contribute in sentimental analysis research. Due to that, an adequate amount of studies has been already done. Historically, there have been two approaches to solve sentimental analysis problems lexicon-based and machine learning approaches [6]. Even though they produce moderately quality results, they failed against human-generated data. Due to that, new deep learning models such as Bidirectional Recurrent Neural Network(RNN)[7] and Long Short-Term Memory(LSTM) network [8] were introduced. On the other hand, [9] conducted experiments in Kannada-English using the traditional learning approaches such as Logistic Regression(LR), Support Vector Machine(SVM), Multinomial Naive Bayes, K-Nearest Neighbors(KNN), Decision Trees(DT), and Random Forest (RF).

To address the sentiment analysis problem using the above techniques, We need a corpus. Since social-media comments/posts do not follow the strict grammar rules and also they are always in non-native scripts as well as code-mixed [10]. [11] created a gold standard Tamil-English code-switched, sentiment-annotated corpus containing 15,744 comment posts from YouTube to overcome the above situation. Moreover, Chakravarthi et al. [12] created a standard corpus for Malayalam-English to increase the sentiment analysis tasks in the code-mixed contents.

[13] explored in Tamil-English, Kannda-English, and [14]Malayalam-English by using the transformer-based model mBERT. The model performed well but failed in some text where code-mixed comes[15]. As an extension work of this research work, [16] conducted experiments on different kinds of models such as Bidirectional LSTM, mBERT, DistilBERT, and ULMFiT [17] to overcome this issue. Moreover, they developed a standard Translation and Transliteration algorithm to convert the corpus into monolingual. From this approach, they could be able to improve their system's performance.

Over the past decade, different kinds of models introduced, but contrasted to conventional Recurrent Neural Network models (RNNs), the efficiency and performance of the transformer models such as BERT[18], DistilBERT[19], mBERT[20] are remarkably distinguished. BERT [21]) models designed to contextualize the text by jointly conditioning on both left and proper contexts. Due to that, transformer models can be used to produce a state-of-the-art result by just fine-tuning the output layer. After studying the above research studies, we decided to go with transformer models and ULMFiT.

3. Dataset

The Tamil-English data set is provided by the Dravidian-CodeMix-FIRE 2021 organizing committee, which extracted from Tamil YouTube comments/posts that contains three parts(Train, Validation, Test). The training, validation and testing datasets have 35,656, 3962, and 4392 comments, respectively, with annotated labels. The dataset consists of texts in five different classes as follows:

Text	**Label**
Vijay Annaa Ur Maasssss Therrrrriii	Positive
நம்ப நடே நாசாமா தான் போச்சே	Negative
Thala’s hardwork + dedication in the movie next level #Thalaaaaaaaaaaa	Mixed Feelings
மனிதனாய் வாழ்வதற்கு தேவை மனிதாபிமானம் மட்டுமே... ஜாதி இல்லை...!	Unknown State
Subtitle me traller dekhne wale like karo	Not in Tamil

Table 1
Dataset samples for each sentiment class.

The data set contains three code-mixed sentences: Inter-Sentential switch, Intra-Sentential switch, and Tag switching. They wrote in either native Tamil script or English grammar with Tamil. Some comments wrote in Tamil script with English words between them. Table 2 describes the dataset statistics and it is visualized in Figure 1. The following items show the five different classes of comments with a definition:

- Positive: Comments which are not offensive
  e.g: ennaya trailer Ku mudi Ellam nikkudhu... Vera level trailer..
- Negative: Comments which are offensive
  e.g: எந்தெந்த youtube channel காரங்க எல்லாம் இதை ஜாதி வெறி படம்கூறாங்களோளோ அவங்கெல்லாம் அந்த ஜாதி என்றறிக
- Mixed Feelings: Comments which are both negative and positive
  e.g:Kaagam karaindhu koodi unnum, Manidham ennum moodar koodam koodi serdhu pagaimai kollum... Idil yaar uyarthinai yaar agrinai
- Unknown State: Comments which are not determined
  e.g:Vandha raja vah dhaan varuven Vera level str
- Not in Tamil: Comments which are not in native Tamil
  e.g:Subtitle me traller dekhne wale like karo

4. System Description and Result Analysis

4.1. Preprocessing

Since the dataset collected from YouTube does not follow any grammar rules and is in code-mixed form. The dataset undergoes the Following steps to use the dataset efficiently.

- The first step is to stemming and lemmatization the words and lower casing the only romanized words as there is no such thing in Tamil script.

Label	Train	Dev	Test
positive	20070	2257	3190
negative	4271	480	315
unknown_state	5628	611	288
mixed-feelings	4020	438	71
Not-Tamil	1667	176	160
Total	35656	3962	4392

Table 2
Number of comment for each class in train, validation and test sets.

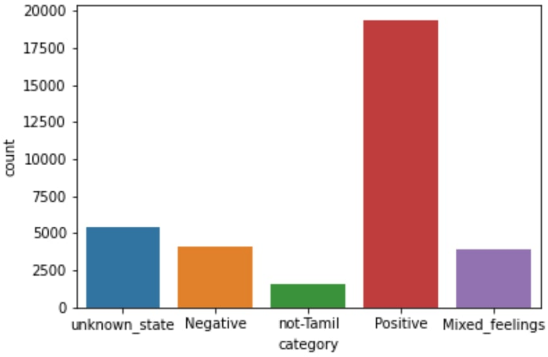

Figure 1: Class distribution on Training set. Dataset is highly imbalanced where a number of comments/posts in positive is much higher higher than other classes.

- The next step is to remove all emojis, special characters, numbers, and punctuations as they do not carry any meaning to the sentence.
- Finally, we applied the algorithm introduced by [16] to do translation and transliteration on the comments and posts to create a monolingual corpus.

4.2. Translation

After loading the dataset, we used an extensive corpus of English words from NLTK-corpus[1] to detect English words in a sentence; if the word is in the English dictionary, then we translated the word into native Tamil script; otherwise, we ignored the word. For this purpose, We used Google Translate API[2].

4.3. Transliteration

Most of the comments are in code mixed form. Comments should be in the native script to get state-of-the-art results from transformers models. Transliteration is the process of transferring a word from the alphabet of one language to another. All non-native Tamil words converted

[1] https://www.nltk.org/

[2] https://pypi.org/project/googletrans/

into the same meaning Tamil words using transliteration. To achieve this, we used AI4Bharat Transliteration[3].

4.4. Models

Recently released transformer models such as BERT achieves a state of the art results in text classification tasks. Considering the performance of transform models, we choose to start with multilingual BERT and DistilBERT. All of our transformer-based models are culled from **HuggingFace**[4] transformers library and the models' parameters are as stated in Table 3. Figure 2 depicts the architecture of our best-performed model(ULMFiT).

Parameters	Value
LSTM Units	128
Dropout	0.2
Activation Function	Softmax
Max Len	128
Learning Rate	1e-5
Optimizer	AdamW
Loss Function	Cross-Entropy
Batch Size	64
Epochs	30

Table 3
Common parameters for the models that we used during our experiments.

DistilBERT model is a small, fast, and light transformer-based model trained on the Wikipedia dataset. It has 40% fewer parameters than BERT, runs 60% faster while preserving over 95% of BERT's performances. Since our purpose is to train a model in Tamil(non-Latin script), we selected the **distilbert-base-multilingual-cased** model, which has six layers, 768 dimensions, 12 heads, and tantalizing 134M parameters.

We also experimented with **bert-base-multilingual-cased** as our pre-trained multilingual model having approximately 110M parameters with 12-layers and 768 hidden states.

4.5. Results and Analysis

Teams were ranked by the weighted average F1 score of their model, and we received 7th rank. Even though our model got above rank, the F1-score difference between the first team is relatively low.

In the beginning, we start with our BERT model and it doesn't perform well. It may have happened due to the lack of BERT multilingual based model training in the Tamil language. In the next step, We approached the problem with the ULMFiT model, a transfer learning technique [22]. ULMFiT's model architecture is different from transformer models, and it is an effective transfer learning method that can apply to any task in NLP. The table shows that ULMFiT

[3] https://pypi.org/project/ai4bharat-transliteration/
[4] https://github.com/huggingface/

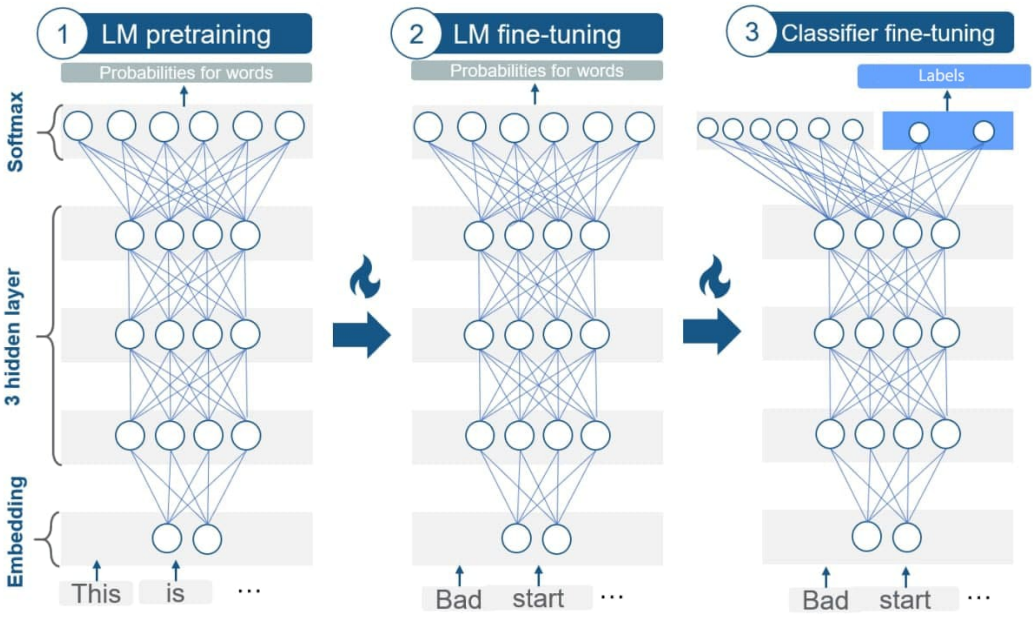

Figure 2: ULMFiT model's Architecture. To recreate this image, we used a source image from [8]. After unfreezing all the layers, we did more epochs to train the whole neural network rather than just the last few layers. This method involves fine-tuning a pre-trained language model (LM) AWD-LSTM to a new dataset in such a manner that it does not forget what it previously learned.

yielded an F1-Score of 0.6101, and DistilBert, mBERT yielded 0.60104 and 0.5963, respectively. Precision and recalls of the above models showen in Table 4.

Models	Precision	Recall	F1-Score
ULMFiT	0.6075	0.6045	0.6101
DistilBert	0.5978	0.5984	0.6014
mBERT	0.5782	0.5627	0.5963

Table 4
Weighted F1-scores according to the models on the test data-set.

5. Conclusion

In this research, we have analyzed different NLP techniques to classify offensive language in Tamil code-mixed YouTube comments[5]. We used a novel technique, transliteration, which leverages the accuracy across all three models.Also, We experimented with transformer models and transfer learning technique(ULMFiT) models. Even though transformer models are more advanced, To our task, ULMFiT yields the best results. Since Tamil is a low-resourced language [23], our research also can be applied to other low-resourced languages without much difficulty.

References

[1] B. R. Chakravarthi, R. Priyadharshini, N. Jose, A. Kumar M, T. Mandl, P. K. Kumaresan, R. Ponnusamy, H. R L, J. P. McCrae, E. Sherly, Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 133–145. URL: https://aclanthology.org/2021.dravidianlangtech-1.17.

[2] S. Banerjee, B. Raja Chakravarthi, J. P. McCrae, Comparison of pretrained embeddings to identify hate speech in indian code-mixed text, in: 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020, pp. 21–25. doi:10.1109/ICACCCN51052.2020.9362731.

[3] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in tamil, malayalam, hindi, english and german, in: Forum for Information Retrieval Evaluation, 2020, pp. 29–32.

[4] S. Suryawanshi, B. R. Chakravarthi, Findings of the shared task on troll meme classification in Tamil, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 126–132. URL: https://aclanthology.org/2021.dravidianlangtech-1.16.

[5] B. R. Chakravarthi, R. Priyadharshini, S. Thavareesan, D. Chinnappa, T. Durairaj, E. Sherly, J. P. McCrae, A. Hande, R. Ponnusamy, S. Banerjee, C. Vasantharajan, Findings of the Sentiment Analysis of Dravidian Languages in Code-Mixed Text 2021, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[6] O. Habimana, Y. Li, R. Li, X. Gu, G. X. Yu, Sentiment analysis using deep learning approaches: an overview, Science China Information Sciences 63 (2019).

[7] M. Schuster, K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45 (1997) 2673–2681. doi:10.1109/78.650093.

[8] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, Physica D: Nonlinear Phenomena 404 (2020) 132306. URL: http://dx.doi.org/10.1016/j.physd.2019.132306. doi:10.1016/j.physd.2019.132306.

[9] A. Hande, R. Priyadharshini, B. R. Chakravarthi, KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection, in: Proceedings of the Third Workshop on Computational Modeling of People’s Opinions, Personality, and Emotion’s in Social Media, Association for Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 54–63. URL: https://aclanthology.org/2020.peoples-1.6.

[10] S. Banerjee, A. Jayapal, S. Thavareesan, Nuig-shubhanker@dravidian-codemix- fire2020: Sentiment analysis of code-mixed dravidian text using xlnet, in: FIRE, 2020.

[11] 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://aclanthology.org/2020.sltu-1.28.

[12] 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://aclanthology.org/2020.sltu-1.25.

[13] C. Vasantharajan, U. Thayasivam, Hypers@DravidianLangTech-EACL2021: Offensive language identification in Dravidian code-mixed YouTube comments and posts, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 195–202. URL: https://www.aclweb.org/anthology/2021.dravidianlangtech-1.26.

[14] B. R. Chakravarthi, P. K. Kumaresan, R. Sakuntharaj, A. K. Madasamy, S. Thavareesan, P. B, S. Chinnaudayar Navaneethakrishnan, J. P. McCrae, T. Mandl, Overview of the HASOC-DravidianCodeMix Shared Task on Offensive Language Detection in Tamil and Malayalam, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[15] R. Priyadharshini, B. R. Chakravarthi, S. Thavareesan, D. Chinnappa, T. Durairaj, E. Sherly, Overview of the dravidiancodemix 2021 shared task on sentiment detection in tamil, malayalam, and kannada, in: Forum for Information Retrieval Evaluation, FIRE 2021, Association for Computing Machinery, 2021.

[16] C. Vasantharajan, U. Thayasivam, Towards offensive language identification for tamil code-mixed youtube comments and posts, 2021. arXiv:2108.10939.

[17] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, 2018. arXiv:1801.06146.

[18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.

[19] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. arXiv:1910.01108.

[20] T. Pires, E. Schlinger, D. Garrette, How multilingual is multilingual bert?, 2019. arXiv:1906.01502.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. arXiv:1706.03762.

[22] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, 2020. arXiv:1911.02685.

[23] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, E. Sherly, J. P. McCrae, Overview of the track on sentiment analysis for dravidian languages in code-mixed text, in: Forum for Information Retrieval Evaluation, 2020, pp. 21–24.