Sentiment Analysis on Multilingual Code Mixing Text Using BERT-BASE: participation of IRLab@IIT(BHU) in Dravidian-CodeMix and HASOC tasks of FIRE2020

Anita Saroj, Sukomal Pal

Indian Institute of Technology (BHU), Varanasi, India

Abstract

This paper discusses our participation in the "Sentiment Analysis in Dravidian-CodeMix", Dravidian-CodeMix and "Hate Speech and Offensive Content Identification in Indo-European Languages"- FIRE 2020 tasks of identifying subjective opinions or reactions on a given topic. Several techniques are applied for sentiment analysis including the recent word embeddings-based methods. BERT, Word2Vec, and ELMo are currently among the most promising and ready-to-use word embedding methods that can convert words into meaningful vectors. We used the BERT_BASE model for sentiment classification of Dravidian-CodeMix data and for HASOC task, our team submitted systems for all the two sub-tasks in three languages - Hindi, English, and German with BERT-based system. We report our approach and results which are promising.

Keywords

BERT, Classification, Dravidian, CodeMix, Sentiment Analysis

1. Introduction

Over the last two decades, there has been unprecedented growth in the number of online media and people's use of them to express and share their opinions, thoughts, or attitudes on various subjects. Online social media platforms, such as YouTube, Twitter, Facebook, Instagram, and LinkedIn, as well as on instant messaging tools, such as WhatsApp, Skype, and WeChat have been very popular where users produce and consume contents on movies, politics, celebrities, products, market surveys, social studies, etc. Most of these content reflect users' sentiment, mostly positive or negative with some being neutral. People write in their native languages. For example, Indians write in Hindi, English, Tamil, Gujarati, Malayalam, etc. Some also use two or more languages. Writing in a mixed language like Hindi-English, English-Tamil, English-Spanish, English-Malayalam, English-Chinese etc is also quite common. Several content-based text classification techniques have been successfully applied in sentiment detection and classification. Nevertheless, identification of sentiments within a collection or a stream of short texts or tweets is a challenging task, especially from within a code-mixed data. Dravidian-CodeMix task at FIRE-2020 provides an experimental setting for the same where we participated in the task involving English-Tamil and English-Malayalam data. In this paper, we explore an word embeddings-based technique BERT_BASE on two code-mixed data: English-Tamil

FIRE 2020: Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India

EMAIL: anitas.rs.cse16@iitbhu.ac.in (A. Saroj); spal.cse@iitbhu.ac.in (S. Pal)

© 2020 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-2826/T4-18.pdf

and English-Malayalam and build models for classifying "posts" into positive, negative, mixed feelings and unknown_state, and other_language [1].

In a related note, while the social media helps quickly spread the information in society, it has also become hotbeds for hate speech and offensive content. Hate speech is defined as "language that is used to express anger towards a targeted group or is meant to be derogatory, to humiliate, or to insult the members of the group [2]." People convicted of using hate speech are often enforced to face large fines and even imprisonment. These laws should spread to the internet and social media, leading many sites to create their own provisions against hate speech. Both Twitter and Facebook have responded to criticisms for not doing enough to prevent hate speech on their sites by instituting policies to prevent their platforms for attacks on people based on characteristics like race, ethnicity, gender, and sexual orientation threats of violence towards others. The HASOC task at FIRE 2020 provides another platform to build and test robust automatic hate speech and offensive content identification systems where content could range from political and religious to caste and gender or any other issues that potentially divide and polarise a society. It is, therefore, imperative to use such systems that can help automatically filter out hate and offensive contents so that they do not spread in the community. HASOC - FIRE 2020 sets up the task divided into two subtasks: hate speech language identification and automatic categorization of hate speech types.

The rest of the paper is organized as follows. Section 2 describes the existing work on Code-mix and HASOC content identification respectively. Section 4 focuses on the methodology. In Section 5 we describe the experiments. Section 6 discusses the results of the task. Section 7 concludes, discussing the lessons learned.

2. Related Work

There exist a few relevant past literature of research work done in sentiment analysis for the CodeMix language.
Garain et al. (2020) [3] used feature extraction algorithms in conjunction with traditional machine learning algorithms such as the Support Vector Regression and Grid Search to solve the Hindi-English codemix sentiment analysis and garnered an F_1-score of 66.2%. Nguyen and Dogruoz (2013) [4] analyzed Turkish-Dutch posts from an online chat forum and compared dictionary-based methods with language models, adding logistic regression and linear-chain CRF. They achieved the best result with word-level accuracy of 97.6% and post-level classification 0.89%. Yadav and Chakraborty (2020) [5] used multilingual and cross-lingual embeddings to efficiently transfer knowledge from monolingual text to code-mixed text for sentiment analysis of the code-mixed text. They achieve an F_1-score of 0.58 (without a parallel corpus) and 0.62 (with the parallel corpus) on the same benchmark in a zero-shot way compared to 0.68 F_1-score in supervised settings.

The development of social media has witnessed the proliferation of two separate but connected phenomena: first, they helped to create a more open and connected world, and second, they contributed to the spread of hate speech and rude behavior [6]. Previous work on hate-speech language detection and related phenomena has seen different system architectures'

deployment with varying performance levels. We can observe three significant waves of the system: discrete linear models, neural networks, pre-trained language models [7, 8, 9, 10, 11]. Linear models (like SVM, logistic regression, or ensemble models) are very competitive and robust methods to identify off-line successfully. An offensive language that, in many cases, improves more complex approaches based on neural networks [12].

3. Task definitions

Although Dravidian CodeMix and HASOC were simultaneously run in FIRE 2020, they attempted to focus on two related but different aspects with clearly distinct task definitions. Below we describe them.

3.1. Dravidian Code-mix Task

The task is primarily of sentiment detection from code-mixed social media text. We attempt the message-level polarity classification task [13]. Given a YouTube comment, the system must classify it into positive, negative, neutral, mixed feelings, or not in intended languages. The track consists of multi-class classification tasks:-

- `Positive state`: Positive state shows the positive opinion, thoughts or attitude of the person, such as relaxing, happiness, thanking to someone, forgiveness.
- `Negative state`: Negative state shows the negative opinion, thoughts, or attitude of the person, such as sad, angry, worried, and violent.
- `Mixed feelings`: Mixed feelings show the speaker is experiencing both positive and negative emotions, such as comparing two products.
- `Unknown state`: There is no clear or implicit indicator of the emotional state of the speaker. Examples are liking or membership or asking questions about the release date or the film dialogue.
- `Other language`: For Malayalam and Tamil, if the sentence does not contain Malayalam and Tamil, it is not Malayalam and not Tamil.

3.2. HASOC Task

The task of identifying hate and offensive posts were done in two different languages: Hindi and German. Each language has two subtasks. A brief description of each work is given below [14].

- `Sub-task A`: Identifying Hate, offensive and profane content. Sub-task A is a coarse-grained binary classification. Teams are required to classify tweets into two classes: Hate and Offensive (HOF) and Non- Hate and offensive (NOT).

  `(NOT) Non Hate-Offensive`: The post does not contain any hate speech, profane, offensive content.

  `(HOF) Hate and Offensive`: Post contains Hate, offensive, and profane content.

- `Sub-task B:` There is no clear or implicit indicator of the emotional state of the speaker. Examples are liking or membership or asking questions about the release date or the film dialogue.
  - `HATE:` Posts under this class contain Hate speech content.
  - `OFFN:` Posts under this class contain offensive content.
  - `PRFN:` These posts contain profane words.

4. Methodology

This section describes the general model and architecture commonly followed for identification of sentiments in both CodeMix and HASOC tasks. and then segregates them according to the requirement of the tasks.

- **Preprocessing:** We first convert all the texts into lowercase. We also replace all the links with the word "URL" and all the numbers with the word "number", remove the leading and trailing white-spaces, and replace multiple white-spaces between words with a single whitespace. We remove all the punctuation symbols using a pre-initialized string, string punctuation available in the string library. After removing non-letters from the data, all tokens are lemmatized.

- **Model Architecture:** BERT stands for Bi-directional Encoder Representations from Transformers [15]. Figure 1 shows the architecture. It is based on a transformer and reads the input from both directions at once. It uses two training approaches, namely Masked Language Model and Next Sentence Prediction. In our approach, it is a pre-trained model, so fine-tuning is done to use it for a specific task.

  We used Rectified Linear Units (ReLU), Sigmoid activation function, and Adam algorithm as an optimizer. Input data is converted in the form of input representation of BERT. In the training phase, the model receives a pair of sentences as input and learns to predict whether the second sentence in the pair is the subsequent sentence in the original document. The model differentiates between two sentences during training by adding a CLS token at the beginning of the first sentence and a SEP token at the end of each sentence. Approximately 15 percent of the words in the input are masked. The input data is converted into a combination of token embedding plus sentence embedding, the transformer positional embedding. The transformer encoder reads the input embedding. The final embedding is fed into our model which makes the predictions.

5. Experiments

In this section, we present a BERT_BASE experiments on Dravidian-CodeMix and HASOC data.

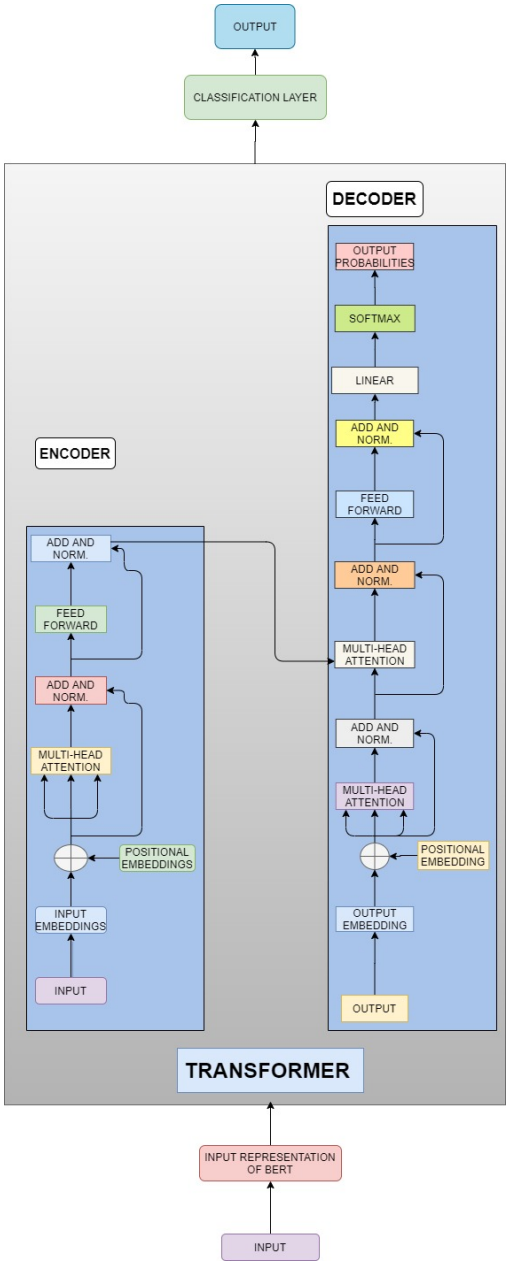

Figure 1: Working module of BERT

5.1. Dataset

5.1.1. Dravidian-CodeMix

The dataset is created from YouTube comments and shared by the task organizers in a tab-separated format for a code-mixed dataset for Tamil [16] and Malayalam [17]. The dataset

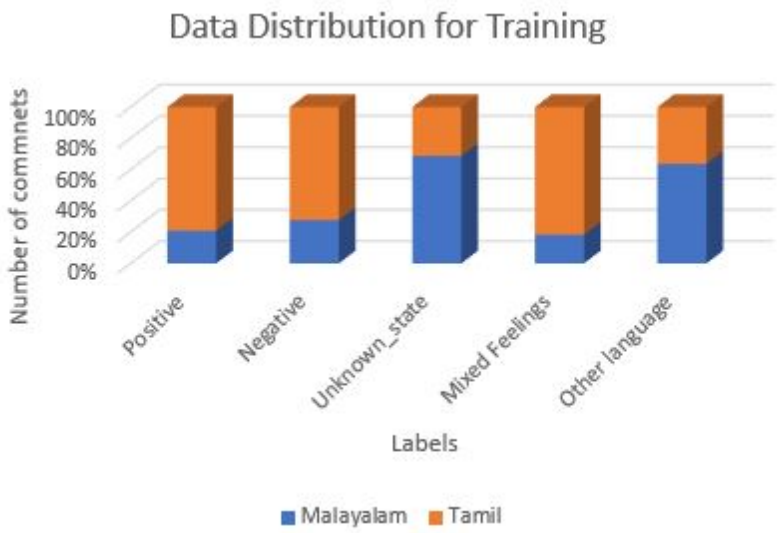

Figure 2: Data distribution of Training set for both the language. Here Malayalam, Tamil indicate Manglish and Tanglish, respectively.

was collected from the comments from the Tamil and Malayalam trailers of movies released in 2019. The sentences written in code-mixed Tamil-English (Tanglish) and Malayalam-English (Manglish). Figure 2 shows the distribution of training instances of the task.

5.1.2. HASOC

HASOC provides a platform for multilingual research on the identification of problematic content and a data challenge. There are 2 sub-tasks for each language such as English, German and Hindi, altogether over 10.000 annotated tweets from Twitter. There are five columns in the CSV file for each language, i.e., tweet_id, text, task1, task2, and ID. Table 1 shows the data distribution of the task.

5.2. Experimental Settings

For both the tasks, we used the released BERT_BASE pre-trained model (Un-cased: 12-layer, 768-hidden, 12-heads, 110M parameters). Based on the maximum size of the sentence in the dataset, performed padding of the sentence to forge the sentence of equal measure. We use recurrent dropout of 0.5 and Sigmoid as an activation function. The dropout layer, with a rate of 0.2, is used to avoid overfitting of the model and set the number of epochs to 10. The initial learning rate is $2e^{-5}$, and the batch size is 8. We used a Softmax activation function at the output layer. We use the Adam optimizer and categorical cross-entropy loss function for training.

6. Results

6.1. Dravidian-CodeMix

The metric for evaluating the systems was as follows. The organizers used F-scores across the positive, negative, unknown_state, mixed_feelings, and the other_language. The final ranking was based on the average F-score. Our submitted system garnered an F-score of 0.59% for English-Tamil and 0.60% for English-Malayalam. The detailed results are shown in Table 3. We compare our result with the team SRJ result, which is rank 1. We found that our submitted result is 0.06, 0.14 (F-score) low compared to rank 1 for English-Tamil, English-Malayalam, respectively.

6.2. HASOC

The metric for evaluating the systems was as follows. The organizers used F-scores across the sub-task A and sub-task B. The final ranking was based on the average F-score. Our submitted system garnered an F-score of 0.5028% for Hindi sub-task A and 0.3840% for German sub-task A. The detailed results are shown in Table 3.

7. Conclusion

In this task, we attempted to explore the sentiment analysis of code-mixed English-Tamil and English-Malayalam data while participating in Dravidian-CodeMix - FIRE 2020. Our system was based on the BERT_BASE model. The system, when evaluated by the organizers, garnered an F_1-score of 0.59 for English-Tamil and 0.6 for English-Malayalam. There was a choice of developing an unconstrained approach, but we only used the provided data to train the system. As a future work, we would like to use other embeddings: ELMo, BERT_LARGE, and other deep learning models. We attempted to explore hate speech and offensive content identification in the Indo-European language while participating in HASOC - FIRE 2020. Our system was based on the BERT_Base model for multi-class classification. The system, when evaluated by the organizers, achieved

$$F_1$$

-score of 0.50 for Hindi. There was a choice of developing an unconstrained approach, but here also we restricted ourselves only to the provided data for training. As future work, we want to work on multitask learning with other embeddings.

References

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

[2] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, arXiv preprint arXiv:1703.04009 (2017).

[3] A. Garain, S. K. Mahata, D. Das, Junlp@ semeval-2020 task 9: Sentiment analysis of hindi-english code mixed data, arXiv preprint arXiv:2007.12561 (2020).
[4] D. Nguyen, A. S. Doğruöz, Word level language identification in online multilingual communication, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 857–862. URL: https://www.aclweb.org/anthology/D13-1084.
[5] S. Yadav, T. Chakraborty, Unsupervised sentiment analysis for code-mixed data, arXiv preprint arXiv:2001.11384 (2020).
[6] D. Colla, T. Caselli, V. Basile, J. Mitrovic, M. Granitzer, Grupato at semeval-2020 task 12: Retraining mbert on social media and fine-tuned offensive language models, in: Proceedings of the International Workshop on Semantic Evaluation (SemEval), 2020.
[7] A. Cimino, L. De Mattei, F. Dell'Orletta, Multi-task learning in deep neural networks at evalita 2018, Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18) (2018) 86–95.
[8] M. Karan, J. Šnajder, Cross-domain detection of abusive language online, in: Proceedings of the 2nd workshop on abusive language online (ALW2), 2018, pp. 132–137.
[9] P. Liu, W. Li, L. Zou, Nuli at semeval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers, in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 87–91.
[10] J. Mitrović, B. Birkeneder, M. Granitzer, nlpup at semeval-2019 task 6: a deep neural language model for offensive language detection, in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 722–726.
[11] Z. Waseem, D. Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on twitter, in: Proceedings of the NAACL student research workshop, 2016, pp. 88–93.
[12] J. P. Montani, P. Schüller, Tuwienkbs at germeval 2018: German abusive tweet detection, in: 14th Conference on Natural Language Processing KONVENS, volume 2018, 2018, p. 45.
[13] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, J. P. Sherly, Elizabeth McCrae, Overview of the track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text, in: Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India, 2020.
[14] T. Mandl, S. Modha, G. K. Shahi, A. K. Jaiswal, D. Nandini, D. Patel, P. Majumder, J. Schäfer, Overview of the HASOC track at FIRE 2020: Hate Speech and Offensive Content Identification in Indo-European Languages), in: Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, CEUR, 2020.
[15] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[16] 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.
[17] 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.

Table 1
Data distribution of HASOC task

Language	Training	Testing
Hindi	2963	663
English	3794	526
German	2452	526

Team name	**Model**	**Hindi**	**German**
-	-	Sub-task A	Sub-task A
IRLab@IITV	BERT_BASE	0.5028	0.3840

Table 2
Classifier result of HASOC dataset at F1 Macro average

Team name	**Model**	English-Tamil				English-Malayalam			
		Precision	Recall	F-score	Rank	Precision	Recall	F-score	Rank
IRLab@IITV	$BERT_{BASE}$	0.59	0.61	0.59	7	0.68	0.6	0.6	12
SRJ	-	0.64	0.67	0.65	1	0.74	0.75	0.74	1

Table 3
Classifier result of Dravidian-CodeMix dataset at Precision, Recall, F-score and Accuracy in %.