CMSAOne@Dravidian-CodeMix-FIRE2020: A Meta Embedding and Transformer model for Code-Mixed Sentiment Analysis on Social Media Text

Suman Dowlagar[a], Radhika Mamidi[a]

[a]*International Institute of Information Technology - Hyderabad (IIIT-Hyderabad), Gachibowli, Hyderabad, Telangana, India, 500032*

Abstract

Code-mixing(CM) is a frequently observed phenomenon that uses multiple languages in an utterance or sentence. CM is mostly practiced on various social media platforms and in informal conversations. Sentiment analysis (SA) is a fundamental step in NLP and is well studied in the monolingual text. Code-mixing adds a challenge to sentiment analysis due to its non-standard representations. This paper proposes a meta embedding with a transformer method for sentiment analysis on the Dravidian code-mixed dataset. In our method, we used meta embeddings to capture rich text representations. We used the proposed method for the Task: "Sentiment Analysis for Dravidian Languages in Code-Mixed Text", and it achieved an F1 score of 0.58 and 0.66 for the given Dravidian code mixed data sets. The code is provided in the Github https://github.com/suman101112/fire-2020-Dravidian-CodeMix.

Keywords
social media, code-mixed, sentiment analysis, meta embedding, Transformer, GRU

1. Introduction

Code-mixing(CM) of text is prevalent among social media users, where words of multiple languages are used in the sentence. Code-mixing occurs when conversant uses both languages together to the extent that they change from one language to another in the course of a single utterance [1]. The computational modeling of code-mixed text is challenging due to the linguistic complexity, nature of mixing, the presence of non-standard variations in spellings, grammar, and transliteration [2]. Because of such non-standard variations, CM poses several unseen difficulties in fundamental fields of natural language processing (NLP) tasks such as language identification, part-of-speech tagging, shallow parsing, Natural language understanding, sentiment analysis.

Gysels [3] defined the Code-mixing as "the embedding of linguistic units of one language into an utterance of another language". Code-mixing is broadly classified into two types, intra-sentential and inter-sentential. Intra-sentential code-mixing happens after every few words. Whereas, in inter-sentential code-mixing, one part of the sentence consists of Hindi words, and the other part is entirely English. The code-mixing helps people to express their emotions or opinions emphatically, thus leading to a phenomenal increase of use in code-mixed messages on social media platforms. With the increase in code-mixed data, the analysis of CMSM text has become an essential research challenge from the perspectives of both Natural Language Processing (NLP) and Information Retrieval (IR) communities.

There have been some research works in this direction, such as GLUECoS, an evaluation benchmark in code-mixed text [4], automatic word-level language identification for CMSM text [5, 6], parsing pipeline for Hindi-English CMSM text [7, 8], and POS tagging for CMSM text [9].

To encourage research on code-mixing, the NLP community organizes several tasks and workshops such as Task9: SentiMix, SemEval 2020[1], and 4th Workshop on Computational Approaches for Linguistic Code-Switching[2]. Similarly, the FIRE 2020's Dravidian-CodeMix task[3] was devoted to code-mixed sentiment analysis on Tamil and Malayalam languages. This task aims to classify the given CM youtube comments into one of the five predefined categories: positive, negative, mixed_feelings, not_<language>[4], unknown_state.

In this paper, we present a meta-embedding with a transformer model for Dravidian Code-Mixed Sentiment Analysis. Our work is similar to the meta embedding approach used for named entity recognition on code-mixed text [10].

The paper is organized as follows. Section 2 provides related work on code-mixed sentiment analysis. Section 3 describes the proposed work. Section 4 presents the experimental setup and the performance of the model. Section 5 concludes our work.

2. Related Work

Sentiment analysis is one of the essential tasks in the field of NLP. Sentiment analysis is the process of understanding the polarity of the sentence. Sentiment analysis helps to attain the public's attitude and mood, which can help us gather insightful information to make future decisions on large datasets [11]. Initially, sentiment analysis was used on government campaigns and news articles [12, 13]. Recently, due to social media prevalence, the research turned towards capturing the sentiment on social media texts in code-mixing scenarios [14].

The earlier approaches used syntactic rules and lexicons to extract features followed by traditional machine learning classifiers for sentiment analysis on code-mixed text. The process of rule extraction and defining lexicons is a time consuming, laborious process, and is domain-dependent. The recent work in the field of CMSA uses embeddings with deep learning and traditional classifiers [15]. The paper [16] used sub-word information for sentiment analysis on code-mixed text. The recent SentiMix 2020 task used BERT-like models and ensemble methods to capture the code-mixed texts' sentiment [14]. We used meta embeddings with state of the art transformer model for this task.

3. Proposed Model

This section presents our proposed code-mixed sentiment analysis framework. It has three main components: a sub-word level tokenizer, a text representation layer, and a transformer model.

3.1. Sub-word Level Tokenizer

To deal with the non-standard variations in spellings, we used the SentencePiece [17]. SentencePiece is an unsupervised text tokenizer and de-tokenizer mainly used for neural network models. SentencePiece treats the sentences just as sequences of Unicode characters. It implements subword units

---

[1]https://competitions.codalab.org/competitions/20654
[2]https://www.aclweb.org/portal/content/fourth-workshop-computational-approaches-linguistic-code-switching
[3]https://dravidian-codemix.github.io/2020/
[4]the language might be Tamil and Malayalam

by using byte-pair-encoding (BPE) [18] and unigram language model [19] . The byte pair encoding initializes the vocabulary to every character present in the corpus and progressively learn a given number of merge rules. The unigram language model trains the model with multiple subword segmentations probabilistically sampled during training.

3.2. Text Representation Layer

Pre-trained embedding models do not perform well on the code-mixed corpus as they consider all the code-mixed words as OOV words [20]. Thus, we have to train word representations from the code-mixed corpus. Given the complexity of the code-mixed data, it is not easy to determine which embedding model to be used for better performance. Hence, we chose the combination of fastText [21], ELMO [22], and TF-IDF [23] embeddings. fastText captures efficient text representations and local dependencies at the word and sub-word level. ELMO captures contextual representations at the sentence level. TF-IDF captures the distribution of the words in the corpus. The use of TF-IDF for sentiment analysis helps in extracting a better correlation between words and their polarity. All these diverse text representations, when combined, proved beneficial in obtaining better embeddings for the downstream tasks.

3.3. Transformer model

From [14], For the task of sentiment analysis, we saw that the attention mechanism works better in deciding which part of the sentence is essential for capturing the sentiment. Thus we chose the transformer model for our code-mixed sentiment analysis task. As the data is a classification type, we used only the encoder side from the Transformer.

The encoder encodes the entire source sentence into a sequence of context vectors. First, the tokens are passed through a standard embedding layer, and the positional embeddings are concatenated with each source sequence. The embeddings are then passed through a series of encoder layers to get an encoded sequence.

The encoder layers is an essential module where all the processing of the input sequence happens. We first pass the source sentence and its mask into the multi-head attention layer, then perform dropout, apply a residual connection, and pass it through a normalization layer. We later pass it through a position-wise feedforward layer and then, again, apply dropout, a residual connection, and layer normalization to get encoded output sequence. The output of this layer is fed into the next encoder layer.

The Transformer model uses scaled dot-product attention given in equation 1, where the query Q and key K are combined by taking the dot product between them, then applying the softmax operation and scaled by a scaling factor d_k then multiplied by the value V. Attention is a critical unit in the Transformer model as it helps in deciding which parts of the sequence are important.

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

The other main block inside the encoder layer is the position-wise feedforward layer. The input is transformed from hid_dim to pf_dim, where pf_dim is usually a lot larger than hid_dim. The ReLU activation function and dropout are applied before it is transformed back into a hid_dim representation. The intuition borrows from infinitely wide neural networks. The wide neural network grants more approximation power and helps to optimize the model faster.
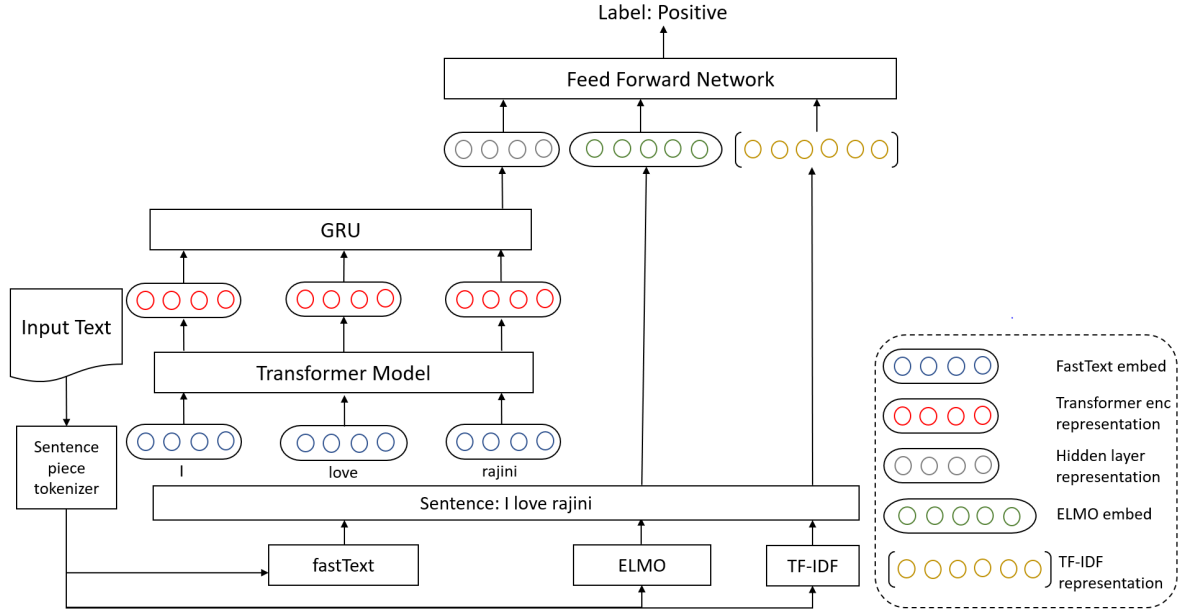
Figure 1: Meta Embedding with transformer and GRU model

3.4. Our Approach

Initially, we tokenized the sentence using the SentencePiece model. After tokenization, we extracted local dependencies between embeddings at the subword level using the fastText model. The fastText model gave embeddings at the word level. We then applied the transformer model to obtain the encoded representations. We got the encoded representations at the word level. A GRU unit is used to get the encoded representation of all the words. We considered the representation of the last hidden layer of the GRU as the final encoded representation. We then obtained the ELMO contextual and TF-IDF representations at the sentence level. We concatenated the representations of the last hidden GRU layer, ELMO, and TF-IDF giving us the meta-embeddings. The meta embeddings are then passed to the output feed-forward network to predict the polarity of the sentence.

4. Experimental Setup

4.1. Data

For Dravidian code-mixed sentiment analysis, we used the dataset provided by the organizers of Dravidian Code-mixed FIRE-2020. The training dataset consists of 15,744 Tamil CM and 6,739 Malayalam CM youtube video comments. The details of the dataset and the initial benchmarks on the corpus are given in [24, 25, 26, 27, 28]

4.2. Hyperparameters

For Embedding models Embeddings play a vital role in improving the model's performance. As mentioned above, we used fastText and ELMO embeddings. The dimensionality was set to 300 in-case

Table 1
Accuracy and weighted F1 score on Tamil Code-Mixed Text

Method	Accuracy	weighted F1
Fine Tuned BERT	0.65	0.53
fastText + Tranformer	0.66	0.57
fastText + ELMO + Transformer	0.66	0.57
fastText + ELMO + TF-IDF + Transformer	**0.67**	**0.58**

Table 2
Accuracy and weighted F1 score on Malayalam Code-Mixed Text

Method	Accuracy	weighted F1
Fine Tuned BERT	0.51	0.46
fastText + Tranformer	0.47	0.45
fastText + ELMO + Transformer	0.50	0.47
fastText + ELMO + TF-IDF + Transformer	**0.67**	**0.66**

of fastText embeddings. The embeddings are trained on training data using the parameters: learning rate = 0.05, context window = 5, epochs = 20. The ELMO model is obtained from tensorflow_hub[5], and the pre-set dimensionality of 1024 is used.

For Transformer and GRU model   After evaluating the model performance on the validation data, the optimal values of the hyper-parameters were set. We used the following list of hyper-parameters: learning rate = 0.0005, transformer encoder layer = 1, dropout rate = 0.1, optimizer = Adam, loss function = Cross-Entropy Loss, and batch size = 32, point wise feed forward dimension (pf_dim) = 2048.

4.3. Performance

We evaluated the performance of the method using weighted F1. The model performed well in classifying positive and not-language comments. The results are given in table 1 and 2 The positive comments had a lot of corpora to train. It made the classification of positive comments an easier task. The not_Malayalam and not_Tamil tweets had another language words in the data, as these language words had higher TF-IDF scores w.r.t the non-language label, their classification was straight-forward. We observed that the system could not identify the sentiment when sarcasm is used in the negative polarity comments. The words in the sarcasm are similar to those of positive comments. It made the sentiment analysis a difficult task.

Mixed feelings had both positive and negative sentences. As the classifier was trained on a lot of positive corpora, it could not deduce the negative polarity sentences with sarcasm and irony imbibed in them. Thus the classifier labeled them as positive. It affected the performance of the classifier. More training data could help resolve such issues.

---

[5]https://tfhub.dev/google/elmo/2

5. Conclusion

This paper describes the approach we proposed for the Dravidian Code-Mixed FIRE-2020 task: Sentiment Analysis for Davidian Languages in Code-Mixed Text. We proposed meta embeddings with the transformer and GRU model for the sentiment analysis of Dravidian code mixed data set given in the shared task. Our model obtained 0.58 and 0.66 average-F1 for Tamil and Malayalam code-mixed datasets, respectively. We observed that the proposed model did a good job distinguishing positive and not_Malayalam and not_Tamil youtube comments. For future work, we will explore our model's performance with larger corpora. As we observed sarcasm and irony in negative polarity sentences, we feel that it would be interesting to focus on techniques to detect irony and sarcasm in a code-mixed scenario.

References

[1] R. Wardhaugh, An introduction to sociolinguistics, volume 28, John Wiley & Sons, 2011.

[2] K. Bali, J. Sharma, M. Choudhury, Y. Vyas, "i am borrowing ya mixing?" An Analysis of English-Hindi Code Mixing in Facebook, in: Proceedings of the First Workshop on Computational Approaches to Code Switching, 2014, pp. 116–126.

[3] M. Gysels, French in urban lubumbashi swahili: Codeswitching, borrowing, or both?, Journal of Multilingual & Multicultural Development 13 (1992) 41–55.

[4] S. Khanuja, S. Dandapat, A. Srinivasan, S. Sitaram, M. Choudhury, Gluecos: An evaluation benchmark for code-switched nlp, arXiv preprint arXiv:2004.12376 (2020).

[5] U. Barman, A. Das, J. Wagner, J. Foster, Code mixing: A challenge for language identification in the language of social media, in: Proceedings of the first workshop on computational approaches to code switching, 2014, pp. 13–23.

[6] S. Gella, K. Bali, M. Choudhury, "ye word kis lang ka hai bhai?" testing the limits of word level language identification, in: Proceedings of the 11th International Conference on Natural Language Processing, 2014, pp. 368–377.

[7] A. Sharma, S. Gupta, R. Motlani, P. Bansal, M. Srivastava, R. Mamidi, D. M. Sharma, Shallow parsing pipeline for hindi-english code-mixed social media text, arXiv preprint arXiv:1604.03136 (2016).

[8] K. Nelakuditi, D. S. Jitta, R. Mamidi, Part-of-speech tagging for code mixed english-telugu social media data, in: International Conference on Intelligent Text Processing and Computational Linguistics, Springer, 2016, pp. 332–342.

[9] Y. Vyas, S. Gella, J. Sharma, K. Bali, M. Choudhury, Pos tagging of english-hindi code-mixed social media content, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 974–979.

[10] 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), IEEE, 2020, pp. 68–72.

[11] B. Liu, Sentiment analysis: Mining opinions, sentiments, and emotions, Cambridge university press, 2020.

[12] D. K. Tayal, S. K. Yadav, Sentiment analysis on social campaign "swachh bharat abhiyan" using unigram method, AI & SOCIETY 32 (2017) 633–645.

[13] N. Godbole, M. Srinivasaiah, S. Skiena, Large-scale sentiment analysis for news and blogs., Icwsm 7 (2007) 219–222.

[14] P. Patwa, G. Aguilar, S. Kar, S. Pandey, S. PYKL, B. Gambäck, T. Chakraborty, T. Solorio, A. Das, Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets, arXiv preprint arXiv:2008.04277 (2020).

[15] P. Mishra, P. Danda, P. Dhakras, Code-mixed sentiment analysis using machine learning and neural network approaches, arXiv preprint arXiv:1808.03299 (2018).

[16] A. Prabhu, A. Joshi, M. Shrivastava, V. Varma, Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text, arXiv preprint arXiv:1611.00472 (2016).

[17] T. Kudo, J. Richardson, Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, arXiv preprint arXiv:1808.06226 (2018).

[18] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, arXiv preprint arXiv:1508.07909 (2015).

[19] T. Kudo, Subword regularization: Improving neural network translation models with multiple subword candidates, arXiv preprint arXiv:1804.10959 (2018).

[20] A. Pratapa, M. Choudhury, S. Sitaram, Word embeddings for code-mixed language processing, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3067–3072.

[21] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146.

[22] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, arXiv preprint arXiv:1802.05365 (2018).

[23] A. Aizawa, An information-theoretic perspective of tf–idf measures, Information Processing & Management 39 (2003) 45–65.

[24] 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: Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India, 2020.

[25] 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: Proceedings of the 12th Forum for Information Retrieval Evaluation, FIRE '20, 2020.

[26] 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.

[27] 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.

[28] B. R. Chakravarthi, Leveraging orthographic information to improve machine translation of under-resourced languages, Ph.D. thesis, NUI Galway, 2020.