BpHigh at WASSA 2023: Using Contrastive Learning to build Sentence Transformer models for Multi-Class Emotion Classification in Code-mixed Urdu

Bhavish Pahwa
Mindtickle / Pune, India
bhavishpahwa@gmail.com

Abstract

In this era of digital communication and social media, texting and chatting among individuals occur mainly through code-mixed or Romanized versions of the native language prevalent in the region. The presence of Romanized and code-mixed language develops the need to build NLP systems in these domains to leverage the digital content for various use cases. This paper describes our contribution to the subtask MCEC of the shared task WASSA 2023:Shared Task on Multi-Label and Multi-Class Emotion Classification on Code-Mixed Text Messages. We explore how one can build sentence transformers models for low-resource languages using unsupervised data by leveraging contrastive learning techniques described in the SIMCSE paper and using the sentence transformer developed to build classification models using the SetFit approach. Additionally, we'll publish our code and models on GitHub and HuggingFace, two open-source hosting services.

1 Introduction

The WASSA 2023 Shared Task on Multi-Label and Multi-Class Emotion Classification on Code-Mixed Text Messages (Ameer et al., 2022) aims at building multi-class and multi-label classification systems to detect if a code-mixed text message has neutral emotion or any of the eleven provided emotions which accurately describe the sentiment behind the text message and the author's emotional state. These eleven emotions are trust, joy, optimism, anticipation, disgust, sadness, fear, anger, surprise, love, and pessimism. The core purpose of the shared task is to understand how robust and accurate NLP systems can be built to perform NLU tasks like emotion detection. Many researchers have tried to approach NLU tasks like sentiment classification in code-mixed Urdu earlier and have been attempting to make robust systems to understand how accurately NLP systems can understand code-mixed Urdu (Sharf and Rahman, 2018). To begin, code-mixed Urdu may include words and phrases from many languages, including English, Urdu, and Hindi. This makes it challenging for NLP systems to reliably identify the language of each word and decide the appropriate language model to apply for text processing.

Second, code-mixed Urdu might feature complicated linguistic phenomena such as code-switching, the practice of switching between languages within a sentence or discourse. This can make it challenging for NLP systems to effectively recognize language borders and decide the appropriate language model to apply for each section of the text.

Finally, code-mixed Urdu may contain transliterated words, loanwords, and other linguistic elements not found in conventional Urdu or English. This can make it challenging for NLP algorithms to recognize and understand these phrases effectively. Researchers are building more advanced NLP models based on the transformer architecture designed to handle code-mixed text to meet these problems. These models employ transfer learning approaches, which entail pre-training a model on a vast dataset of code-mixed text before fine-tuning it for a specific purpose. Processing and interpreting code-mixed Urdu and other code-mixed languages are becoming more viable using these more complex models (Devlin et al., 2019).

Many researchers have also started building sentence transformer models by training pre-trained transformer models based on the Sentence-BERT paper (Reimers and Gurevych, 2019) using the sentence transformers framework[1]. These trained sentence transformers can generate sentence embedding vectors, which can be used for many downstream tasks like classification, clustering, and information retrieval. The significant advantage of sentence transformers is that the embedding vectors they generate can capture the respective text's syntactic and semantic meaning.

[1] https://www.sbert.net/index.html

Emotion Label	Number of Samples
neutral	3262
trust	1118
joy	1022
optimism	880
anticipation	832
disgust	687
sadness	486
fear	453
anger	226
surprise	199
love	187
pessimism	178

Table 1: Train Dataset Description

Emotion Label	Number of Samples
neutral	388
trust	125
joy	131
optimism	110
anticipation	94
disgust	113
sadness	62
fear	52
anger	35
surprise	35
love	17
pessimism	29

Table 2: Dev Dataset Description

This paper describes our approach based on training a sentence transformers model using the pretrained MURIL (Khanuja et al., 2021) transformer model based on the BERT architecture. We leverage contrastive learning techniques described in the SIMCSE paper (Gao et al., 2021) to train our sentence transformer model on unsupervised data in Romanized Urdu and Hindi. We call this trained sentence transformer model MURIL-SIMCSE. We further utilize the SetFit framework[2] (Tunstall et al., 2022) to fine-tune our MURIL-SIMCSE model on the training dataset of the Multi-class Emotion Classification (MCEC) subtask of the shared task to perform emotion detection in a multi-class prediction setting.

We will release all our code on GitHub[3] and fine-tuned models on HuggingFace[4] .

2 Dataset Description

The dataset of the MCEC track of the shared task consists of three subsets, namely the train, dev, and test set. The train and dev set consists of examples wherein we have the code-mixed sms message and the respective emotion label assigned to the message. The test set contains the code-mixed sms messages on which the approach will be tested and the gold labels against which the predicted labels will be compared. Table 1 and Table 2 describes the train and dev datasets for the number of examples in each emotion label.

3 Related Work

Reimers and Gurevych (2019) released the Sentence BERT architecture, constructed by altering BERT. The method employs Siamese and triplet network topologies on top of a BERT network to construct sentence embeddings with considerable semantic information. These sentence embeddings can be used for downstream tasks like clustering, classification, and information retrieval. Furthermore, sentence transformer models can be trained by introducing a pooling layer on top of any pretrained transformer model and by using annotated datasets that can inform the model that a pair of sentences have a degree of semantic similarity or a triplet where two sentences have a certain similarity. The third example is supposed to be dissimilar from the other two.

Khanuja et al. (2021) released a research paper and a new transformer model based on BERT architecture called MuRIL, which was trained in English and 16 other languages spoken in the Indian subcontinent region. The 16 other languages are Assamese (as), Bengali (bn), Gujarati (gu), Hindi (hi), Kannada (kn), Kashmiri (ks), Malayalam (ml), Marathi (mr), Nepali (ne), Oriya (or), Punjabi (pa), Sanskrit (sa), Sindhi (sd), Tamil (ta), Telugu (te) and Urdu (ur). It was trained using Masked language modeling and translation language modeling objectives. The authors show that MuRIL can outperform mBERT on the XTREME benchmark (Hu et al., 2020), Multilingual BERT (mBERT) achieves an average performance of 59.1, whereas MURIL achieves an average performance of 68.6. In the XNLI sentence classification task (Conneau et al., 2018), the MuRIL transformer has an accuracy of 67.7 in Urdu, whereas mBERT has an accuracy of 58.2.

Gao et al. (2021) shows how unsupervised datasets like simple text input sentences can be used to train and build sentence transformer models. In the unsupervised SIMCSE approach, the same input text sentence is passed to the pre-trained

[2] https://github.com/huggingface/setfit

[3] https://github.com/bp-high/WASSA_Code-Mixed_Shared_Task

[4] https://huggingface.co/bpHigh

encoder twice; as we use standard dropout, the two sentence embeddings of the same input sentence passed twice will be at slightly different positions. While training, the distance to other embeddings of the other sentences in the same batch (which serve as negative examples) will be maximized. The distance between these two embeddings of the same input sentence will be minimized.

Tunstall et al. (2022) released a research paper and the SetFit framework to build a robust sentence classifier for small datasets using sentence transformers. The SetFit works by fine-tuning the sentence transformer on the respective dataset using contrastive learning. The fine-tuned sentence transformer is then used to generate sentence embeddings to train the classification layer.

4 Methodology

In this section, we describe our approach to training sentence transformer model based on the MuRIL pretrained transformer and further building multi-class classifier systems by finetuning the MURIL-SIMCSE model on the training data of the multi-class emotion classification track of the shared task.

4.1 Training MURIL-SIMCSE model

For training the sentence transformer on top of MuRIL pretrianed transformer using contrastive learning approach we first build a dataset of input text sentences in Urdu by utilizing previously published code-mixed and romanized Urdu datasets. We use the following two datasets to generate the input sentences in Urdu for the train dataset:-

1. **HS-RU-20** (Khan et al., 2021) [5]

2. **Roman Urdu Hate Speech** (Rizwan et al., 2020) [6]

As both these datasets have text and labels and contain hate/toxic examples to contain bias and toxicity, we filter only the normal/ non-hateful/non-toxic examples from these datasets and curate them for the train dataset. We get 13404 input sentences in Urdu from the above-described datasets, which are relatively low for training contrastive learning-based sentence transformer models. We assume that Hindi is similar in spoken forms to Urdu to increase the number of input sentences. Although they are written in different scripts in the Romanized format, they should be similar. So we add romanized hindi sentences from the **Hing-Corpus dataset**[7] (Nayak and Joshi, 2022) to our pure romanized Urdu sentences dataset and generate a final dataset of two hundred thousand sentences(200,000).

Model	Number of Iterations	Epochs
MURIL-SIMCSE-SETFIT-V1	8	1
MURIL-SIMCSE-SETFIT-V2	15	2
MURIL-SIMCSE-SETFIT-V3	20	2
MURIL-SIMCSE-SETFIT-V4	25	2

Table 3: Hyperparameters

We train the model for one epoch, with a batch size of 32 using AdamW as the optimizer with WarmupLinear scheduler and 20000 warmup steps, the learning rate being 2e-05.

4.2 Training SetFit based classifiers

Using the trained MuRIL-SIMCSE sentence transformer model, we further develop classifiers using the **SetFit framework** (Tunstall et al., 2022) and the training dataset of the MCEC track. Figure 1 shows the training process according to the SetFit framework.

We train four versions of the SetFit-based classifier using different hyperparameters. The various hyperparameters associated with each version can be found in Table 3. In all the different versions we keep the value of batch size same and the value of batch size is 16. All versions are trained using cosine similarity loss, same learning rate of 2e-05, same seed with value 42, same warmup proportion of 0.1.

5 Results

The result for all the different MURIL-SIMCSE-SETFIT model versions on the test dataset are presented in Table 4.

We notice that the SetFit framework based model's performance improves as we increase the value of the hyperparameter 'number of iterations' while training. This hyperparameter refers to the number of iterations for which the sentence pairs would be generated for sentence transformer fine-tuning process in the SetFit training process. Even though we notice this general trend we also have to consider that although accuracy increased

[5] https://www.kaggle.com/datasets/drkhurramshahzad/hate-speech-roman-urdu

[6] https://huggingface.co/datasets/roman_urdu_hate_speech

[7] https://github.com/l3cube-pune/code-mixed-nlp

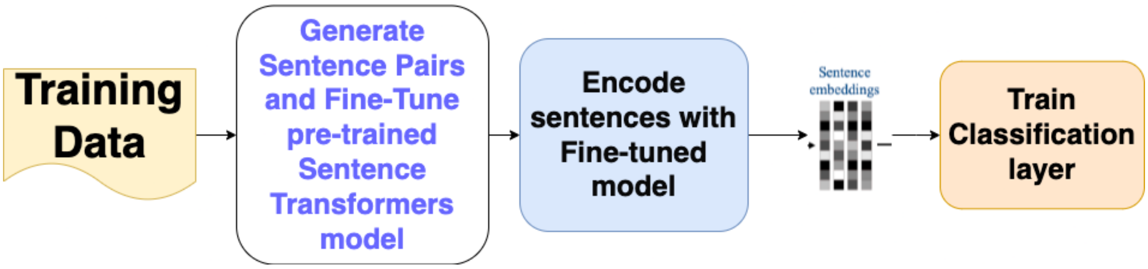
Training Data
Generate Sentence Pairs and Fine-Tune pre-trained Sentence Transformers model
Encode sentences with Fine-tuned model
Sentence embeddings
Train Classification layer

Figure 1: SetFit training process

Model	Macro F1-Score	Recall	Precision	Accuracy
MURIL-SIMCSE-SETFIT-V1	0.3764	0.5642	0.5642	0.5642
MURIL-SIMCSE-SETFIT-V2	0.5657	0.6792	0.6792	0.6792
MURIL-SIMCSE-SETFIT-V3	0.5345	0.6843	0.6843	0.6843
MURIL-SIMCSE-SETFIT-V4	**0.6400**	**0.7044**	**0.7044**	**0.7044**

Table 4: Metric Values of the different SetFit models on the test dataset

from MURIL-SIMCSE-SETFIT-V2 to MURIL-SIMCSE-SETFIT-V3 the Macro-F1 score dipped.

6 Limitations

Train Dataset for MuRIL-SIMCSE: While we try to minimize the hateful samples in this dataset by removing all the toxic/hateful samples of the respective datasets used to form this dataset, there could be samples containing certain biases like gender bias and racial bias. Also the dataset contains the respective languages written in the Roman script, so the results might not be transferable to the respective native scripts of the languages.

MURIL-SIMCSE: The model was trained on a single Tesla P100 GPU for 9 hrs. We could have trained further and on more data, but we could not due to resource and economic constraints.

7 Conclusion

We describe our approach in this paper for the MCEC track of the subtask. We leverage the unsupervised training method using contrastive learning for developing a sentence-transformer model from MuRIL pre-trained model for romanized code-mixed Urdu. We leverage this sentence-transformer model to build multi-class classifiers using the provided training data and the SetFit framework. We show how increasing the value of the hyperparameter number of iterations increases the performance of the classifiers. Further, we will examine how increasing the unsupervised text examples dataset used for training the MURIL-SIMCSE sentence transformer affects the performance of the classifiers built on top of it. We would also look into whether our assumption to mix Romanized Hindi text examples with Urdu examples produces actual benefit or more noise. At the same time, it is not necessary that text examples in Hindi and Urdu would be equivalent even in the Romanized form. For example, Urdu and Hindi speakers romanize the retroflex R differently[8] . Taking the word study as an example, it would be "parho" in Roman Urdu and "padho" in Roman Hindi.

References

Iqra Ameer, Grigori Sidorov, Helena Gómez-Adorno, and Rao Muhammad Adeel Nawab. 2022. Multi-label emotion classification on code-mixed text: Data and methods. *IEEE Access*, 10:8779–8789.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages

[8] https://www.reddit.com/r/Urdu/comments/11vnhbk/question_are_romanized_urdu_and_romanized_hindi/

4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Muhammad Moin Khan, Khurram Shahzad, and Muhammad Kamran Malik. 2021. Hate speech detection in roman urdu. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 20(1).

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha P. Talukdar. 2021. Muril: Multilingual representations for indian languages. *CoRR*, abs/2103.10730.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Hammad Rizwan, Muhammad Haroon Shakeel, and Asim Karim. 2020. Hate-speech and offensive language detection in Roman Urdu. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2512–2522, Online. Association for Computational Linguistics.

Zareen Sharf and Saif Ur Rahman. 2018. Performing natural language processing on roman urdu datasets. *International Journal of Computer Science and Network Security*, 18(1):141–148.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts.