Exploring Multilingual BERT for Sentiment Analysis on Hindi-English Code Mixed Text

Christoph Schaller
christoph.schaller@posteo.de

Pavlos Musenidis
musenidis96@gmail.com

1 Introduction

Code-mixing is a phenomenon, where linguistic units of one language get embedded into sentences of another language (Myers-Scotton, 1993). Although linguistic models have gotten much better at analyzing language in the last century, this phenomenon still poses a great challenge, as it makes data much more unpredictable.

We want to do Sentiment Analysis on code-mixed data. Sentiment Analysis is the task of deciding if the sentiment of a sentence is negative, neutral or positive on a large scale. This gains more and more relevance given its importance for analysing reviews, studying trends or gaining information on a users' attitude.

We will compare both monolingual BERT (vanilla BERT hereafter) and multilingual BERT (ML-BERT). Due to the multilingual nature of the data, we think that the ML-BERT, will perform much better than the vanilla BERT model.

Using BERT with its pre-trained embeddings will possibly make up for the sparseness of our dataset. As there is not much literature on ML-BERT for sentiment analysis, we hope that our study will yield promising results and interesting insights for this task.

2 Previous work

BERT is short for Bidirectional Encoder Representations from Transformers. Devlin et al. (2019) proposed this model whose framework consists of two steps: pre-training and finetuning. During pre-training the model is trained on Masked Language Modelling and Next Sentence Prediction. The data used for this is the BooksCorpus (800M words) and English Wikipedia (2,500M words). Fine-tuning then consists of adding a classification layer and fine-tuning all parameters on the desired task. This model has been highly successful as low-resource tasks benefit a lot from this procedure.

ML-BERT uses monolingual corpora (Wikipedia pages) of 104 languages to create the embeddings. Pires et al. (2019)'s results show that ML-BERT is able to perform surprisingly well over different languages. They also investigate, why this is the case by conducting different experiments testing various hypotheses. Their results show, that the model handles transfer to code-switching fairly well and that transfer works best for languages with similar typology. They also hypothesize that shared word pieces could be the reason why ML-BERT generalizes this well. This hypothesis however, is contradicted by K et al. (2020), who want to answer the same question, but also take the model architecture into account. Their findings indicate that crucial factors for ML-BERT's performance are the structural similarity of the languages and the depth of the model.

Joshi et al. (2016) employed a Subword-LSTM for Sentiment Analysis on the same data we will use. They converted the data to subword-level representations by first using character embeddings, and then performing a 1-D convolution to form subwords of three characters. They then performed maxpooling on the subword representations to compress the information and pass these maxpooled representations to the LSTM. The LSTM propagates the most useful information of the maxpooled subwords to form a sentiment vector representation. For prediction, this is passed through a fully connected layer.

Lal et al. (2019) also proposed a CMSA model for the data we will use, which is built on three components. The first one is a convolutional neural network which serves the purpose to get subword-level representations like Joshi et al. (2016). The second one is a dual encoder, which consists of two different bidirectional LSTMs: the Collective Encoder, which is supposed to learn a representation of the overall sentiment of the sentence and the Specific Encoder which shall learn what subwords contribute most to the sentiment of the sentence. The third component is the Feature Network, where they use linguistic features to augment the neural network framework of their model. The output of the dual encoder and of the feature network get concatenated and passed through multiple fully connected layers to classify the sentences.

3 System Description

The architecture of our system is derived from Joshi et al. (2016). Its composition is visualized in Figure 1, the numerous layers of the BERT model are displayed as a single component. Additional informa-

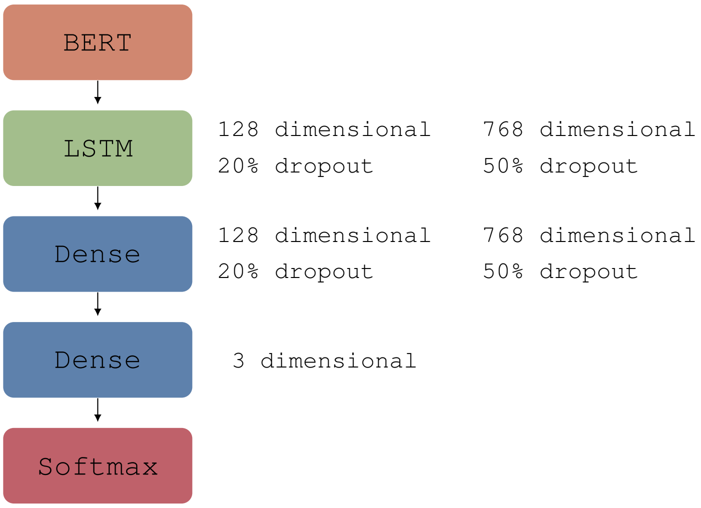

Figure 1: Schematic overview of the architecture.

tion about hyperparameters and different setups for those are displayed to the right of the respective component.

While the basic structure of our system is alike the system of Joshi et al. (2016), we add BERT instead of using convolutional and maxpooling layers before the LSTM layer. BERT also replaces the computation of word embeddings in our system. Additionally we replace the second LSTM layer with a dense layer before feeding into a final fully connected layer.

The primary objective of the system is the ability to prototype and test the outcomes of different setups in a quick fashion. For this we added the possibility of modifying core values as well as hyperparameters of the system by defining them in a single function. This allows for testing setups with different BERT models, layer configurations as well as efficient tuning of the hyperparameters. Additionally a variable number of BERT layers can be finetuned on the training data. Training metrics are logged for analysis, offering insights into the influence of the hyperparameters.

To explain the effect of different BERT models and hyperparameters on our system, we chose two different BERT models and two different sets of hyperparameters to showcase their consequences on the results. For the BERT models, we compare the results using the most recent monolingual model with the results achieved with the most recent multilingual model. The hyperparameter sets are chosen to reflect the hyperparameters of the system proposed by Joshi et al. (2016) as well as a good hyperparameter set for our system with a maximal separation from the set provided by Joshi et al. (2016). As proposed, we thus train the system with 128 dimensions for the LSTM and dense layers with a dropout of 20% for each layer and with 768 dimensions for both layers as well as 50% dropout for them. In all cases we utilize the standard Adam optimizer and regularize with early stopping. This produces four different test setups for which we discuss their results in this report. The development of accuracy and the systems stability during training is visualized in Figure 2.

4 Experimental Setup

The dataset we use, consists of 3879 Hindi-English code-mixed Facebook comments, each labeled with a number, describing the sentiment. Sentences are labeled 0 for negative, 1 for neutral or 2 for positive sentiment. This dataset was created, annotated and analyzed by Joshi et al. (2016). In the pre-processing they removed comments which were not written in roman script, consisted of more than one sentence, were longer than 50 words or were complete English sentences. However, when examining the data, we observed that there are still complete English and Hindi sentences in the data. There were two annotations for each sentence and a sentence was only taken into the dataset, if both annotations matched.

The data contains 15% negative, 50% neutral and 35% positive comments. We tried to replicate the data split Joshi et al. (2016) used, to ensure comparability, but their split couldn't be replicated for our method. We then moved on to make our own splits by first dividing the data into a randomized 80-20 train test split and then further randomly dividing the test data into development and testing splits. This leaves us with our final training, development and testing data. This way, we could ensure to have the same ratio between sets as Joshi et al. (2016).

The baseline we want to beat is a bag-of-words model which uses the decision tree classifier from the scikit-learn library. To compare our results, we will also look at the results of Joshi et al. (2016), who provided the first results and Lal et al. (2019), who provided State of the Art results on the same data.

We will evaluate our data using accuracy and F1-score to further ensure comparability with previous work.

5 Results and Analysis

As Table 2 shows, ML-BERT outperforms the vanilla-BERT in the second configuration with 768 dimensions which matches our hypothesis, that the ML-BERT also includes vocabulary of Hindi and can therefore adapt much easier to our dataset. Nevertheless, this is opposite in the first configuration.

The fact that the vanilla-BERT performs this well, can only be explained, by the diversity of domains the data is pretrained on. As wikipedia pages are written very objectively, the domains of ML-BERT may not cover enough subjective language to make out the sentiments of sentences.

Figure 2 shows that the ML-BERT reaches a higher training accuracy than vanilla-BERT in both configurations. This could be due to the higher amount of vocabulary, ML-BERT covers.

Another possible explanation could be that the English parts of the sentences yield more subjective information as users want their opinion to be globally recognized. However, Agarwal et al. (2017)'s findings, contradict this hypothesis, as they state, that bilinguals have a preference toward their dominant language when swearing and demonstrated it on Hindi-English code-switching.

Also we observe, that the dimensionality plays an important role, as the configuration with 768 dimensions yields better results than the configuration with 128 dimensions. Interestingly BERT's performance was better when not being finetuned on our data rather than when being finetuned. We thought that the amount of finetuning would play a role in this, however changing the amount of layers being finetuned did not lead to noticeable changes in the results. So our final and best configuration (bold in Table 2) is not finetuned and contains 768 dimensions and 50% dropout.

Table 3 shows that our best configuration beats the baseline accuracy by 25.2%, while the contemporary models outperform our proposed model. Joshi et al. (2016)'s accuracy is better by 5% and their F1-score is better by 20.4%. Lal et al. (2019)'s CMSA outperform our model by far. We think that the nature of the data, the sparseness aside, is too noisy for BERT as it consists of Facebook comments, which can contain spelling mistakes, neologisms and morpheme stretching.

Comment	Prediction	Truth
:'(:'(:'(:'(:'(tum gande pati ho bhot gande I hate u I really hate u	positive	negative
Amazngggggg bhaijaan.... Love uuuuh koi kuch ke le mai toh 1 st day 1 show jaunga....	neutral	positive

Table 1: Examples of noisy text in the dataset.

These phenomena seem to be captured much better on a subword level. BERT contains subword embeddings, but it also contains word embeddings which aren't as useful, when it comes to the mentioned phenomena as the examples in Table 1 show. We hoped that the size of the data the embeddings were pretrained on would compensate for this, but this was proved wrong by the results. Joshi et al. (2016) and Lal et al. (2019) both implemented methods, which only depend on subword embeddings and therefore can capture sentences with these phenomena much better. Lal et al. (2019)'s system seems to capture much more detailed features of words and sentences due to the dual encoder and the feature network they employed.

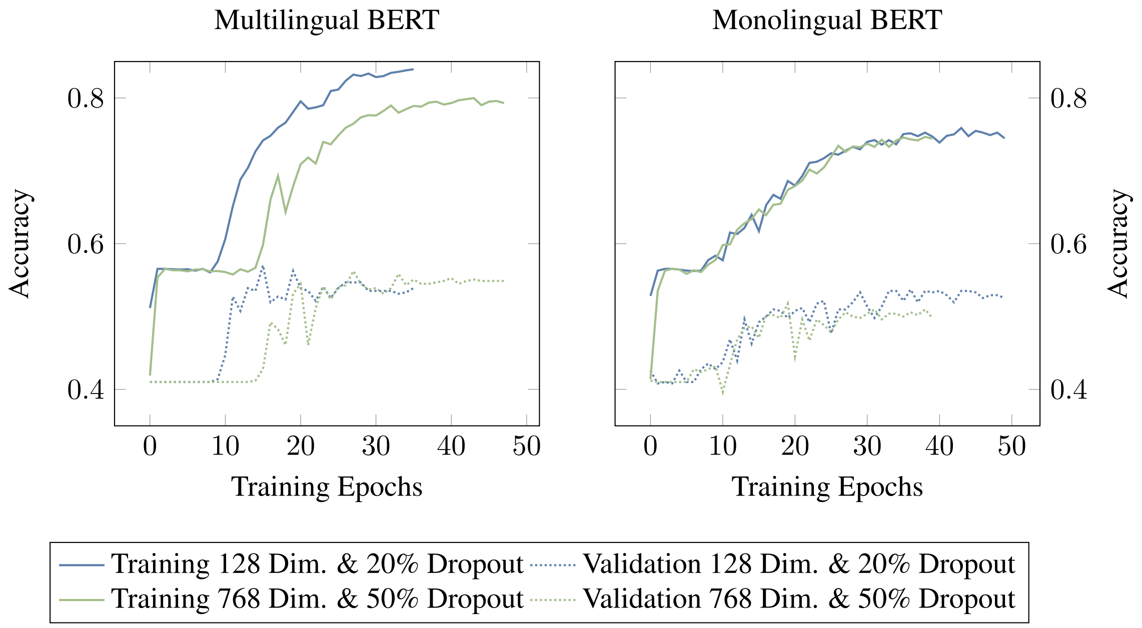

Figure 2: Development of **Accuracy** while training the system.

BERT Model	Dimensions	Dropout	Accuracy	F1-Score
monolingual	128	20%	0.621	0.439
multilingual	128	20%	0.616	0.431
monolingual	768	50%	0.626	0.44
multilingual	768	50%	**0.647**	**0.454**

Table 2: Results of the different configurations the system was trained with.

Method	Reported In	Accuracy	F1-Score
Baseline	-	0.395	-
ML-BERT-LSTM	Proposed	0.647	0.454
Subword-LSTM	Joshi et al.	0.697	0.658
CMSA	Lal et al.	**0.835**	**0.827**

Table 3: Comparison of the proposed model and the contemporary models.

6 Conclusions

We hypothesized that ML-BERT will perform much better than vanilla BERT on code-mixed data, which we found to be false. The results are really close together, which lead us to the conclusion, that the domains, the embeddings were pre-trained on in ML-BERT, are not suited enough for the task of sentiment analysis.

Further we stated, that the pre-trained embeddings could make up for the sparseness of the data, but it couldn't outperform the systems which relied on subword embeddings. We think that subword representations can deal better with the noisiness of the data, than the BERT embeddings. However this hypothesis could be further tested, by reconstructing Lal et al. (2019)'s system but replacing the convolutional neural network with BERT embeddings to make it perfectly comparable.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186, Minneapolis, USA.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pages 2482–2491, Osaka, Japan.

2020. Cross-Lingual Ability of Multilingual BERT: An Empirical Study. In Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-Lingual Ability of Multilingual BERT: An Empirical Study. In *Proceedings of the 8th International Conference on Learning Representations*.

Yash Kumar Lal, Vaibhav Kumar, Mrinal Dhar, Manish Shrivastava, and Philipp Koehn. 2019. De-mixing sentiment from code-mixed text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 371–377, Florence, Italy.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy.

Carol Myers-Scotton. 1993. Common and uncommon ground: Social and structural factors in codeswitching *Language in society*, 22(4):475–503.

Prabhat Agarwal, Ashish Sharma, Jeenu Grover, Mayank Sikka, Koustav Rudra, and Monojit Choudhury. 2017. 2017. I may talk in english but gaali toh hindi mein hi denge: A study of english-hindi code-switching and swearing pattern on social networks. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, pages 554–557.