Aligning Multilingual Embeddings for Improved Code-switched Natural Language Understanding

Barah Fazili
IIT Bombay
barah@cse.iitb.ac.in

Preethi Jyothi
IIT Bombay
pjyothi@cse.iitb.ac.in

Abstract

Multilingual pretrained models, while effective on monolingual data, need additional training to work well with code-switched text. In this work, we present a novel idea of training multilingual models with alignment objectives using parallel text so as to explicitly align word representations with the same underlying semantics across languages. Such an explicit alignment step has a positive downstream effect and improves performance on multiple code-switched NLP tasks. We explore two alignment strategies and report improvements of up to 7.32%, 0.76% and 1.9% on Hindi-English Sentiment Analysis, Named Entity Recognition and Question Answering tasks compared to a competitive baseline model.

1 Introduction

Large pretrained multilingual models have enabled cross-lingual transfer on a number of downstream natural language understanding (NLU) tasks. Apart from serving as a good starting point to train models for tasks in low-resource languages, multilingual models (Devlin et al., 2018) (Conneau et al., 2019) have also been used to achieve zero-shot cross-lingual transfer on target languages with no task-specific labeled data. However, compared to monolingual inputs, the effectiveness of multilingual models on code-switched inputs—i.e., inputs with two or more languages appearing within or across sentences in a conversation—has not been explored enough.

In this work, we aim at explicitly modifying representations from pretrained multilingual models to be more amenable to code-switched inputs. We do this with the help of parallel text in the two component languages and alignment objectives that explicitly encourage representations to be better aligned across the two languages. We conjecture that modifying multilingual embeddings to be better aligned across the two languages will help the model deal better with tokens switching languages within a code-switched sentence. We start with a pretrained multilingual BERT (mBERT) baseline model (Devlin et al., 2018) and design two alignment objectives to be used with parallel text to align the multilingual embeddings. This “aligned" mBERT model is then further fine-tuned with small amounts of code-switched labeled data in the target task. We find such an aligned model to be more accurate on multiple downstream tasks involving code-switched inputs.

The two main highlights of this work can be summarized as follows:

- We propose two alignment-based objectives to be used with mBERT and parallel text in English and Hindi. The aligned models are fine-tuned and further evaluated on code-switched Hindi-English NER, SA and QA tasks. Compared to the baseline mBERT, we obtain clear improvements on all three downstream tasks.
- We investigate how our model behaves in the following two settings: 1) Using a bilingual lexicon instead of parallel text 2) Using Romanized Hindi instead of the native Devanagari script for Hindi.

We also present visualizations that clearly show that the alignment objective helps bring representations for aligned words in Hindi and English closer together.

2 Methodology

We explore two different objectives to encourage cross-lingual contextual alignment in the mBERT model. For this, we need access to parallel text in the component languages corresponding to the code-switched language of interest. We propose both a sequence-level alignment objective that is contrastive in nature, and a word-level alignment objective that is based on minimizing distances between aligned word embeddings.

Proceedings of the 29th International Conference on Computational Linguistics, pages 4268–4273
October 12–17, 2022.

2.1 Contrastive Loss for Sentence-level Alignments

Contrastive learning has been widely used in computer vision as a self-supervised technique to learn visual representations (Chen et al., 2020). Such contrastive objectives are becoming more popular for text-based tasks as well (Gao et al., 2021). We use a contrastive alignment objective with parallel text to improve cross-lingual alignment and potentially yield improved representations for code-switched text.

Consider a batch consisting of N pairs of parallel sentences $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ extracted from a parallel corpus $\mathcal{C}$, where x_i and y_i denote sequences of words in two different languages. Our aim is to improve the alignment of a multilingual model f with respect to $\mathcal{C}$. Let $f(x_i)$ denote the contextual embedding of the word x_i due to the multilingual model f. The contrastive alignment objective is given by:

$$\begin{aligned} L_c = \frac{1}{2N} \sum_{i=1}^{N} - \log \frac{e^{(\mathrm{S}(f(x_i), f(y_i))/\tau)}}{\sum\limits_{\substack{k=1 \\ k \neq i}}^{N} e^{(\mathrm{S}(f(x_i), f(y_k))/\tau)}} \\ + \frac{1}{2N} \sum_{i=1}^{N} - \log \frac{e^{(\mathrm{S}(f(x_i), f(y_i))/\tau)}}{\sum\limits_{\substack{k=1 \\ k \neq i}}^{N} e^{(\mathrm{S}(f(x_k), f(y_i))/\tau)}} \\ + \eta \sum_{i=1}^{N} R_i(f) \end{aligned} \tag{1}$$

where S is a similarity function, τ is a temperature hyperparameter and $R_i(f)$ is a regularization term with a scaling factor of η that is defined as:

$$R_i(f) = 2 - \mathrm{S}(f(x_i), f_0(x_i)) - \mathrm{S}(f(y_i), f_0(y_i)) \tag{2}$$

Here, f_0 denotes the initial pretrained model prior to alignment.

The contrastive objective in Equation (1) forces positive pairs ((x_i, y_i)) to be closer to each other and negative pairs ($(x_i, y_k), \forall k \neq i$) to be pushed further apart. The regularization term ensures that the aligned embeddings do not deviate too much from their initialization. The alignment algorithm using the contrastive objective is further elaborated in the following steps:

1. $f(x_i)$ is the embedding of the [CLS] token in mBERT after passing the entire sequence x_i as its input. For a given batch of N parallel pairs, the loss in Equation 1 is computed over all positive pairs, (x_i, y_i). There are two loss terms associated with each positive pair (x_i, y_i), each consisting of similarity scores between (x_i, y_k) (excluding y_i) and (x_k, y_i) (excluding x_i), respectively.

2. The similarity function between embeddings, denoted as S, is a cosine similarity function. The similarity scores are further scaled by a positive temperature hyperparameter.

3. The regularization term is composed of one loss term per (x_i, y_i) instance and explicitly penalizes divergences in embeddings from the initial pretrained model f_0.

4. The composite loss per batch is finally normalized by the number of positive instances considered per batch i.e. $2N$ pairs.

2.2 Multilingual Loss for Word-level Alignments

While the contrastive loss operates at the level of sentences, we also consider an alignment objective that operates at the level of individual words. This could be considered a more aggressive alignment technique since it encourages every aligned word in parallel sentences to be close together. For every parallel sentence pair (x_i, y_i), we first use an off-the-shelf alignment tool called `awesome-align` (Dou and Neubig, 2021)[1] to extract word alignments. We further filter the aligned pairs based on an alignment prediction probability (set to 0.9 in our experiments) to ensure that we only use high-quality word alignments. If there are N parallel sentences $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ in a batch and $a(x_i, y_i)$ represents a list of index tuples $\{(1, j), \ldots, (m, n)\}$ denoting the aligned word indices in the parallel sentence pair (x_i, y_i), the alignment objective can be written as:

$$L_m = \frac{1}{B} \sum_{i=1}^{N} \sum_{\substack{(m,n) \in \\ \{a(x_i, y_i)\}}} \mathrm{S}(f(x_{i,m}), f(y_{i,n})) + R_i(f)$$

where $x_{i,m}$, $y_{i,n}$ denotes the m^{th} and n^{th} word in x_i and y_i, respectively, and B denotes the total number of successfully aligned word-pairs in the batch.

[1] `https://github.com/neulab/awesome-align`

$R_i(f)$ is the regularization term defined in Equation (2). Note that $f(x_{i,m})$ in L_m refers to a contextual embedding, while $f(x_i)$ in the contrastive loss L_c is the embedding of the [CLS] token.[2]

3 Experiments and Results

3.1 Experimental Setup

3.1.1 Dataset Details

We evaluated our aligned models on three downstream tasks in code-switched Hindi-English — SA (Sentiment Analysis), NER (Named Entity Recognition) and QA (Question Answering) — from the GLUECoS (Khanuja et al., 2020) benchmark. Tasks in the GLUECoS benchmark can be grouped into two categories, sequence labeling tasks (NER, etc.) and tasks requiring deeper semantic understanding (sentiment analysis, etc.). We evaluated our techniques on three tasks, NER, SA and QA, spanning both categories.[3]

NER and QA datasets contain Hindi in the Romanized form, while the SA evaluation sets use the native Devanagari script for Hindi. As an evaluation metric, we use F1 scores for all three tasks. For the cross-lingual alignment training phase, we used parallel text in English-Hindi from the IIT Bombay English-Hindi Corpus (Kunchukuttan et al., 2017). Alternatively, we also experimented with using a bilingual lexicon, MUSE (Lample et al., 2018), instead of parallel text.

GLUECoS NER is sourced from a Twitter NER corpus (Singh et al., 2018) with 2467/308/307 train/dev/test instances. The sentiment analysis dataset is taken from the ICON 2017 shared task; Sentiment Analysis for Indian Languages (SAIL) (Patra et al., 2018) and has 10080/1260/1260 instances in the train/dev/test splits. The QA dataset (Chandu et al., 2018) includes 259/54 instances in the train/dev sets, respectively.

3.1.2 Model Implementation

We use Multilingal BERT (Devlin et al., 2018) (base) as our baseline pretrained multilingual model, that is also the baseline of choice for the GLUECoS benchmark. Subsequent works reporting results on GLUECoS (e.g., Santy et al. (2021)) also used mBERT as their base model. This motivated us to stick to mBERT so that we could reproduce the baseline numbers and contextualize our improvements better compared to prior work.

We train mBERT with the alignment objectives in two different ways: 1) Train all 12 mBERT layers with the alignment objective and 2) Only train a newly-introduced linear layer on top of mBERT and freezing the remaining mBERT layers. The new linear layer will have the same number of input and output dimensions as the last layer in mBERT (i.e., 768 in mBERT base). For training the linear layer, we use the AdamW optimizer at a learning rate of 0.001 with early stopping (and patience set to 10). For training all the mBERT layers, we choose a smaller learning rate of $5e-5$. For the contrastive objective L_c, we used a validation set to tune the scaling factor for the regularization term η and the temperature values. For the multilingual alignment L_m, we only tuned the scaling factor η for the regularization term.

SA (Devanagari)	dev	test
Baseline	$60.3_{\pm 0.00}$	$64.2_{\pm 0.03}$
L_c (|| Devanagari)	$59.4_{\pm 0.01}$	$66.3_{\pm 0.04}$
L_m (|| Devanagari)	**61.0**$_{\pm 0.01}$	**68.9**$_{\pm 0.03}$
L_c (MUSE Devanagari)	$60.7_{\pm 0.00}$	$67.8_{\pm 0.04}$
L_m (MUSE Devanagari)	$59.6_{\pm 0.01}$	$65.9_{\pm 0.02}$

Table 3.1: F-scores after intermediate pretraining of standard mBERT using various alignment schemes on the GLUECoS SA task. || refers to the use of parallel text, and MUSE is the bilingual lexicon. L_c,L_m refer to the contrastive and multilingual alignment schemes.

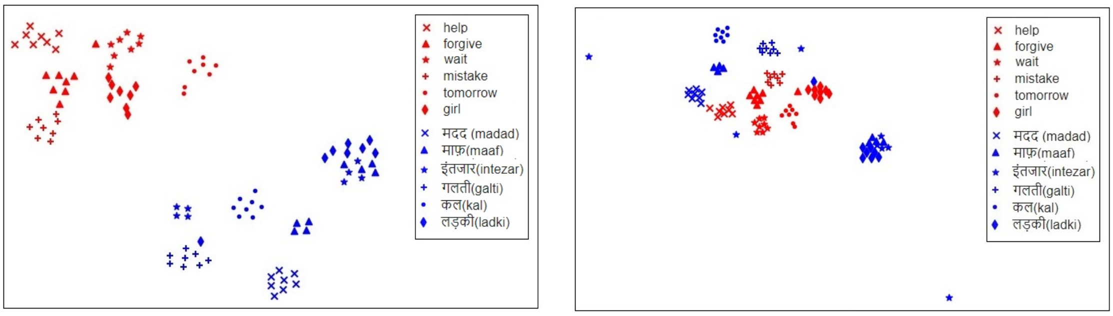
3.2 Results on Downstream Tasks

Table 3.1 lists the F scores on the GLUECoS SA task. The alignment training was done either using parallel text from the IITB Parallel Corpus or the bilingual lexicon from MUSE. This alignment training phase was followed by finetuning on the code-switched Hindi-English SA training data. We see significant improvements in F1 scores for all alignment training schemes. The best F1 score on SA Devanagari is achieved with multilingual alignment over the IITB parallel corpus.

Table 3.2 shows results on both NER and QA. The alignment training is different from SA (in Table 3.1) with only training a newly-added linear

[2] Code and data supporting our work is available at: https://github.com/BarahFazili/AlignmentForCS.

[3] We did not consider the remaining two sequence labeling tasks of GLUECoS — LID-tagging and POS-tagging — since they already yielded fairly high baseline mBERT scores (>95 and >87 for LID and POS, respectively).



(a) Baseline mBERT

(b) Aligned mBERT

Figure 3.1: t-SNE plots using 48 instances of words in each language before and after aligning mBERT using L_m.

layer on top of frozen mBERT layers.[4] Baseline (rand) refers to adding a randomly initialized linear layer on top of the baseline, that is subject to no alignment training and only task-specific finetuning. We observe clear performance improvements on both NER and QA.

Visualizing the alignments. Fig 3.1 visualizes the change in embeddings after the alignment training. We selected 6 pairs of parallel Hindi-English words and created a set of 48 parallel sentences in monolingual Hindi and English; each of the six pairs appear in eight sentences each (in their corresponding scripts). Embeddings for these words across all 96 sentences were extracted from both the baseline mBERT and our multilingual aligned mBERT and plotted in 2D using t-SNE. As seen in Fig 3.1, the parallel words are now closer to each other in the aligned plot irrespective of the underlying language.

System	**NER dev**	**QA dev**
Baseline	$78.7_{\pm 0.01}$	$73.5_{\pm 2.75}$
Baseline (rand)	$78.5_{\pm 0.01}$	$71.8_{\pm 1.67}$
L_m (|| Roman)	$79.3_{\pm 0.00}$	$74.0_{\pm 2.20}$
L_c (|| Roman)	$79.0_{\pm 0.01}$	$74.3_{\pm 2.99}$
L_m (|| Devanagari)	$79.2_{\pm 0.00}$	$72.3_{\pm 0.69}$
L_c (|| Devanagari)	$78.8_{\pm 0.01}$	$74.0_{\pm 1.36}$
L_c (MUSE Roman)	-	$74.9_{\pm 1.44}$
L_m (MUSE Roman)	-	$72.6_{\pm 1.91}$

Table 3.2: F scores after intermediate pretraining of linear layer added on top of frozen standard mBERT using various alignment schemes on the GLUECoS NER,QA and SA tasks

4 Related Work

Large pretrained multilingual models, such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) have achieved state-of-the-art results on monolingual and cross-lingual benchmark tasks. However, their efficacy on code-switched tasks have not been sufficiently explored. (Winata et al., 2021) observed that pretrained multilingual models do not necessarily guarantee effective representations for code-switched text.

Prior work has explored different ways of adapting multilingual pretrained models to be effective for code-switched data. Prasad et al. explore bilingual intermediate pretraining to derive large and consistent performance gains on three different NLP tasks on code-switched text. Santy et al. finetune mBERT with synthetic code switched data generated using random lexical substitution and code-switching constraints based on linguistic theories. Chakravarthy et al. (2020) also pretrain mBERT on code-switched text and adopt other data augmentation techniques to derive performance gains. Aguilar et al. (2021) focus on the role of tokenization and propose a hybrid technique that processes in-vocabulary and out-of-vocabulary tokens differently and observe improvements on three different code-switched NLP tasks. Gupta et al. (2021) use unsupervised self-training to predict pseudolabels on the target task and retain high-confidence predictions as labeled samples that are further used to finetune the model. This leads to a boost in performance on the task of code-switched sentiment analysis.

We note that the L2 alignment technique in Wu and Dredze (2020) is the same as our word-level multilingual alignment objective except for regularizing model parameters rather than the model out-

[4] Backpropagating through all mBERT layers significantly degrades performance for QA. Conversely, training only a linear layer for SA while freezing mBERT layers did not help.

put embeddings (before and after alignment). Wu and Dredze (2020) show two variants of contrastive alignment termed "weak" and "strong". The weak variant strictly uses negative pairs from the other language, while the strong variant uses negative pairs from both within and outside the language. We used the weak variant in our work to avoid overfitting since we did not have a lot of data for alignment training.

Our work departs from prior work on improving NLU for code-switched inputs in that it is the first to explore the use of alignment objectives with parallel text to modify the multilingual representations and make them more suitable for code-switched tasks. Recent work from Deshpande et al. (2021) corroborates our findings and establishes a strong correlation between embedding alignments and downstream performance on cross-lingual transfer. While they present a post-hoc empirical analysis of what factors benefit cross-lingual transfer the most, we explicitly use an alignment-based training for better alignment between languages and improve downstream task performance.

5 Conclusion

In this work, we propose aligning multilingual embeddings using sentence-level (contrastive) and word-level (non-contrastive) objectives. Such an explicit alignment leads to improved performance on three code-switched Hindi-English NLP tasks: SA, NER and QA. Future work will explore the use of alignment objectives in a multi-task framework with the target tasks.

Acknowledgements

The authors thank the anonymous reviewers for their constructive suggestions to improve the draft. The second author gratefully acknowledges support from Google Research India towards research on code-mixing.

References

Gustavo Aguilar, Bryan McCann, Tong Niu, Nazneen Rajani, Nitish Keskar, and Thamar Solorio. 2021. Char2subword: Extending the subword embedding space using robust character compositionality.

Sharanya Chakravarthy, Anjana Umapathy, and Alan W Black. 2020. Detecting entailment in code-mixed Hindi-English conversations. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 165–170, Online. Association for Computational Linguistics.

Khyathi Chandu, Ekaterina Loginova, Vishal Gupta, Josef van Genabith, Günter Neumann, Manoj Chinnakotla, Eric Nyberg, and Alan W. Black. 2018. Code-mixed question answering challenge: Crowd-sourcing data and techniques. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 29–38, Melbourne, Australia. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Ameet Deshpande, Partha Talukdar, and Karthik Narasimhan. 2021. When is bert multilingual? isolating crucial ingredients for cross-lingual transfer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Zi-Yi Dou and Graham Neubig. 2021. awesomealign.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings.

Akshat Gupta, Sargam Menghani, Sai Krishna Rallabandi, and Alan W Black. 2021. Unsupervised self-training for sentiment analysis of code-switched data.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. Gluecos : An evaluation benchmark for code-switched nlp.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. The iit bombay english-hindi parallel corpus.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail$_{c}ode - mixedsharedtask@icon - 2017$.

Archiki Prasad, Mohammad Ali Rehan, Shreya Pathak, and Preethi Jyothi. 2021. The effectiveness of intermediate-task training for code-switched natural language understanding.

Sebastin Santy, Anirudh Srinivasan, and Monojit Choudhury. 2021. Bertologicomix: How does code-mixing interact with multilingual bert? In *AdaptNLP EACL 2021*.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for Hindi-English code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35, Melbourne, Australia. Association for Computational Linguistics.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching?

Shijie Wu and Mark Dredze. 2020. Do explicit alignments robustly improve multilingual encoders? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4471–4482, Online. Association for Computational Linguistics.