Comparative Analysis of Cross-lingual Contextualized Word Embeddings

Hossain Shaikh Saadi[1], **Viktor Hangya**[2,3], **Tobias Eder**[1] and **Alexander Fraser**[2,3]
[1]Technical University of Munich, Germany
[2]Center for Information and Language Processing, LMU Munich, Germany
[3]Munich Center for Machine Learning
shaikh.saadi@tum.de, tobias.eder@in.tum.de,
{hangyav,fraser}@cis.lmu.de

Abstract

Contextualized word embeddings have emerged as the most important tool for performing NLP tasks in a large variety of languages. In order to improve the cross-lingual representation and transfer learning quality, contextualized embedding alignment techniques, such as mapping and model fine-tuning, are employed. Existing techniques however are time-, data- and computational resource-intensive. In this paper we analyze these techniques by utilizing three tasks: bilingual lexicon induction (BLI), word retrieval and cross-lingual natural language inference (XNLI) for a high resource (German-English) and a low resource (Bengali-English) language pair. In contrast to previous works which focus only on a few popular models, we compare five multilingual and seven monolingual language models and investigate the effect of various aspects on their performance, such as vocabulary size, number of languages used for training and number of parameters. Additionally, we propose a parameter-, data- and runtime-efficient technique which can be trained with 10% of the data, less than 10% of the time and have less than 5% of the trainable parameters compared to model fine-tuning. We show that our proposed method is competitive with resource heavy models, even outperforming them in some cases, even though it relies on less resources.

1 Introduction

Contextualized word representations generated from pre-trained language models have outperformed previously standard static embeddings. Static distributional word representations offer a single representation for a word regardless of its current context (Mikolov et al., 2013a; Bojanowski et al., 2017). Contrarily, a word's contextual representation is influenced by the context in which it is used. Contextualized embeddings have demonstrated ground-breaking performance across several NLP tasks and languages, and accommodate many semantic and syntactic aspects of words (Devlin et al., 2019; Conneau et al., 2020; Brown et al., 2020). From the introduction of ELMo (Peters et al., 2018) and ULMFiT (Howard and Ruder, 2018) to the present, different types of language models have been proposed (Devlin et al., 2019; Lan et al., 2020; Clark et al., 2020; Conneau et al., 2020; Sanh et al., 2019; Radford et al., 2019; Brown et al., 2020) of which the most influential is BERT (Devlin et al., 2019) which initiated an era of Transformer (Vaswani et al., 2017) based language models.

Multilingual Language Models (MLMs) can perform various tasks across different languages. Previous works (Cao et al., 2020; Liu et al., 2019) have showed that the MLM's performance in different transfer learning tasks can further be improved by alignment. The idea of aligning contextualized embeddings is to move the representations of words with similar meaning from different languages closer to each other. There are several ways to perform alignment on contextualized embeddings, such as anchor mapping (Liu et al., 2019) and full model fine-tuning (Cao et al., 2020). However, all of these methods have several shortcomings. It is (1) time-consuming, taking about 24 hours to perform mapping. In contrast to static embeddings, in case of contextualized embeddings the generation of anchor embeddings is required to be able to perform mapping which is the majority of the required time (Liu et al., 2019). Similarly, it takes about 8 hours to perform model fine-tuning (Cao et al., 2020) on mBERT. It is also (2) resource-intensive requiring a lot of GPU memory due to model size and (3) data-intensive requiring a huge collection of monolingual sentences for anchor generation, while fine-tuning requires around 250K pairs of parallel sentences to produce the best-reported alignment (Cao et al., 2020). As a result of these limitations anchor embeddings map-

ping and fine-tuning are often difficult or expensive to perform, deploy and use in real-world scenarios.

To the best of our knowledge there is no study available until now where different model architectures and alignment techniques on various downstream tasks are systematically compared other than on the most popular models such as mBERT and XLM-RoBERTa (Kulshreshtha et al., 2020; Cao et al., 2020; Libovický et al., 2020). In this paper our main goal is to fill this gap. We have compared five multilingual and seven monolingual models with three current alignment techniques (VecMap (Artetxe et al., 2016), RCSLS (Joulin et al., 2018) and model fine-tuning (Cao et al., 2020)) from different perspectives such as multilingual vs. monolingual, big vs. small models and the effect of vocabulary. To assess the models and alignment techniques from different perspectives we used three different tasks: bilingual lexicon induction (BLI), word retrieval (Cao et al., 2020) and zero-shot cross-lingual natural language inference (XNLI) on two language pairs: high-resource German-English and low-resource Bengali-English.

Motivated by the shortcomings of current alignment methods discussed above, and inspired by the fine-tuning based alignment technique of Cao et al. (2020), in addition to the comparative analysis we propose a parameter, data and time efficient alignment technique which requires 10% of the data, runs within less than 10% of the time and uses the amount of less than 5% of trainable parameters compared to model fine-tuning (Cao et al., 2020). An overview of our proposed approach is given in Figure 1.

The findings of our experiments demonstrate that 1) multilinguality always leads to better performance in cross-lingual transfer tasks. 2) We should choose bigger models over smaller models when the resources (computational and data) are available but 3) in case of unattainable resources smaller but specialized multilingual models, such as indic-bert (Kakwani et al., 2020), should be chosen, since they are capable of outperforming or performing similar to the big multilingual models, such as XLM-RoBERTa (Conneau et al., 2020), on a language the model is specialized for. 4) Having a large vocabulary and language support is not an advantage of itself, instead the number of tokens allocated for a given language/script plays a more important role. 5) Big language models are sensitive to batch size and learning rate. 6) Model fine-tuning based alignment (Cao et al., 2020) strengthens the quality of MLM's contextualized embeddings and 7) our proposed method is competitive with resource heavy models, even outperforming them in some cases despite having a significantly lower number of trainable parameters. Our work shows that in specific cases (such as for Bengali on XNLI task) less resource intensive but more targeted solutions (e.g. indic-bert) can also be successfully employed.

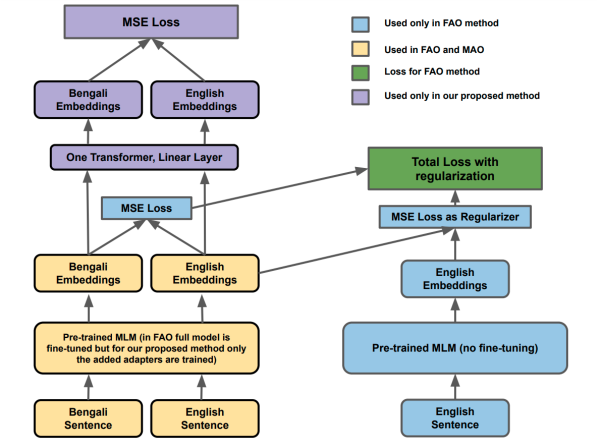

Figure 1: Overview of the fine-tuning based alignment technique (FAO) and our proposed technique (MAO). Small colored square boxes in the upper right corner indicate which modules are used in which method (FAO or MAO).

The paper is structured as the following: the related work is discussed in Section 2. Then Section 3 contains required background knowledge followed by the explanation of our proposed approach in Section 4. Following that, Section 5 contains all the information regarding the tasks, data, different pipelines of our experiments, training procedures and hyperparameters. In Section 6 we discuss the results of different tasks and experiments. Finally, we conclude our work in Section 7.

2 Related Work

By pre-training language models on texts involving multiple languages their representation can be leveraged for cross-lingual applications (Devlin et al., 2019; Conneau et al., 2020). Cross-lingual representation quality can be improved using several alignment approaches. Aldarmaki and Diab (2019) build an orthogonal mapping of contextual ELMo (Peters et al., 2018) embeddings and used these mapping for word and sentence translation

retrieval. Schuster et al. (2019) also employed a mapping approach to align ELMo embeddings, first they acquired context-independent anchors by factorizing the contextualized embedding space into two parts (context-independent and context-dependent) then they applied the mapping approach to the independent part and tested their proposed mapping approach on zero-shot dependency parsing. Similarly, Wang et al. (2019) learned a linear mapping directly using the contextual embeddings generated from BERT and XLM (Conneau and Lample, 2019), while Liu et al. (2019) aligned anchors of contextual mBERT embeddings. Cao et al. (2020) proposed a model fine-tuning based alignment technique using parallel corpora and proposed the word retrieval task to assess its performance. In a similar work to ours, Kulshreshtha et al. (2020) compared different rotation and fine-tuning based alignments on various downstream tasks. However, all previous work focused on improving state-of-the-art cross-lingual performance and tested their proposed approaches only on a few mainstream MLMs, such as BERT or XLM. In contrast, our main goal is to analyse which model and parameters fit certain data and computational resource scenarios the best, thus we investigate applying different types of alignment approaches to different types of multilingual and monolingual models including various architectures and sizes, trained on either monolingual or multilingual data.

Additionally, alignment approaches are resource intensive. Performing anchor generation for mapping takes the majority of the required time (Liu et al., 2019; Kulshreshtha et al., 2020). Likewise, fine-tuning mBERT takes more than 8 hours (Cao et al., 2020), and for XLM-RoBERTa it is even longer. Due to model size, they require a lot of GPU memory. Also, they are data-intensive requiring a huge collection of monolingual sentences (Liu et al., 2019) for anchor generation and during fine-tuning, around 250K pairs of parallel sentences are required to produce an alignment of good quality. Focusing on these shortcomings we propose a parameter, data and time efficient alignment approach to tackle these issues. Our proposed approach is lightweight compared to full model fine-tuning based alignment, as well as more time and data efficient than fine-tuning and anchor based alignment.

3 Background

3.1 Mapping

In this section, we will discuss mapping techniques using contextualized embeddings. The contextualized embeddings mapping process follows a similar principle as static embeddings mapping. Given a seed dictionary of source-target word pairs and their embeddings, a linear projection of the source embeddings to the target space is learned (Mikolov et al., 2013b). Suppose x_i and z_i are source and target word embeddings respectively of the i^{th} word pair in the dictionary. The primary aim is to find a transformation matrix W such that Wx_i is similar to z_i. This can be expressed as the following optimization problem:

$$\underset{W}{arg\_min} \sum_{i=1}^{n} ||Wx_i - z_i||^2$$

Anchor Generation: Many approaches rely on anchors as context independent word representations to generate mapping for contextualized embeddings (Liu et al., 2019; Kulshreshtha et al., 2020). We generate anchors for each of the words by following the procedures of (Liu et al., 2019). For a selected word 1000 sentences where the word is present are selected followed by the generation of contextualized embeddings of each occurrence which are average pooled resulting in the anchor representation. For efficiency, we used 100 sentences instead of 1000 in our systems. In case a word is split into subwords we consider only the embedding of the last subword following (Cao et al., 2020). Additionally, we only considered the output embeddings of the last layer, instead of averaging all layers, since semantic features are manifested in higher layers (de Vries et al., 2020).

3.2 Model Fine-Tuning

In order to improve the alignment of the language model using a parallel corpus Cao et al. (2020) proposed a fine-tuning based alignment method. The intuition of this method is to tune the source language embeddings to be closer to the target language embeddings in the vector space. To bring this intuition into practice a simple but effective loss function was introduced:

$$L(f, C) = - \sum_{(s,t) \in C} \sum_{(i,j) \in a(s,t)} sim(f(i,s), f(j,t)) \quad (1)$$

where (s, t) is a parallel sentence pair of the source and target languages in the parallel corpus C, $a(s, t)$ indicates the word alignments for (s, t), $f(i, s)$ is the contextualized representation of the word at index i in sentence s given by the used MLM and $sim(f(i, s), f(j, t))$ indicates the similarity of the indicated word embeddings defined by:

$$sim(f(i,s), f(j,t)) = -||f(i,s) - f(j,t)||_2^2 \quad (2)$$

However, minimizing (1) could lead to a degenerative solution where all tokens are represented in the same point mass. To avoid this case, the authors proposed a regularizer preventing the target representations from deviating from the initial value significantly. Let f_0 indicate the initial pretrained model before the alignment, then:

$$R(f, C) = \sum_{(s,t)\in C} \sum_{(i,j)\in a(s,t)} ||f(i,t) - f_0(i,t)||_2^2 \quad (3)$$

The regularizer is only applied to the target language representations. The final loss function for the model fine-tuning is the sum of (1) and (3). Note however that due to f_0, two copies of the model have to be kept in memory. Additionally, the model can be fine-tuned using multiple language pairs, by training on the concatenation of their parallel corpora.

In this work we refer to this technique as FAO (fine-tuning based alignment objective) which we also depict in Figure 1.

3.3 Cross-Lingual Evaluation

Word Retrieval For intrinsic evaluation of MLMs Cao et al. (2020) proposed the word retrieval task. Given parallel data, the task is for each source word to retrieve its translation, i.e., find the parallel sentence pair of the source sentence containing the input word and select the right word in it. First, all the source and target language sentences are passed through the language model to build word representations for each word. Note that since a given word type is contained in multiple sentences, it has a contextualized representation for each occurrence. For each of the source words, the most similar word from the target set is taken as its translation pair by calculating their CSLS similarity (Lample et al., 2018). We report the accuracy score for this task. Here the accuracy is defined as the percentage of exact matches between source and target words throughout the whole parallel corpus, similar to Cao et al. (2020).

BLI Given a dictionary of source and target language word pairs, bilingual lexicon induction is the task of translating a source language word to a target language word (Irvine and Callison-Burch, 2017; Shi et al., 2021). In this task, the target word with the highest similarity score is chosen as the translation of the source word by computing the cosine similarity between the anchored embeddings of the source word and the target words. For this task, we report P@1 and P@5. Here, P@1 indicates the percentage of source words where the target word with the highest similarity score is the gold translation. P@5 is the percentage of source words where the gold translation falls among the five target words with the highest similarity scores.

XNLI Cross-lingual natural language inference is a sentence pair classification task using the corpus of (Conneau et al., 2018). It consists of three classes (neutral, entailment and contradiction) and is used to evaluate cross-lingual transfer learning systems. It covers 15 languages, including two low-resource languages (Swahili and Urdu) (Conneau et al., 2018). We report the accuracy score for this task.

4 Proposed Approach

FAO is data-intensive requiring 250K parallel sentences, it is time-consuming and resource-intensive. Similarly, applying simple but efficient alignment techniques like Vecmap and RCSLS is too time-consuming and resource intensive in the case of contextualized embeddings. Inspired by these issues we propose a small alignment architecture which can be trained swiftly (less than 10% of the time required for fine-tuning the whole model) with a few thousand parallel sentences (10% of the data required for fine-tuning the whole model) and trainable parameters for all the proposed architectures are less than 3% of the language model's parameters. To achieve this we add small trainable modules to MLMs and keep the rest of the network frozen.

Linear or Transformer Layer on Top We add a single linear or transformer layer on top of the used MLM. An overview of our proposed method is provided in Figure 1. First, the sentences are fed into the language model then we extract the embeddings

of all the words (we only take the embedding of the last subword following Cao et al. (2020) in case a word is split). These embeddings are then fed to the proposed linear or transformer layer, which outputs embeddings of the same size as the MLM. As mentioned, we train only the added layer and keep the MLM frozen. This way the number of parameters to be trained and the required time are significantly reduced compared to FAO. Additionally, unlike FAO we do not use the regularizer loss which reduces computation and memory use since the initial model (f_0) is unnecessary. The rest of the procedure is the same as described in Section 3. We named our method modified alignment objective (MAO).

Adapters Additionally, we leverage adapters (Pfeiffer et al., 2020) in each of the MLM layers together with a transformer layer on top of the models. Similarly as above, we only trained the transformer and the adapter parameters and kept the language model parameters frozen.

5 Experimental Setup

5.1 Data

We have used three different downstream tasks and for each of the tasks we have different data sources. This section will provide an overview of the data sources across the tasks.

Word Retrieval For the word retrieval task we used German-English (Koehn et al., 2005) and Bengali-English (Hasan et al., 2020) parallel data, and we have followed all the procedures proposed in (Cao et al., 2020). To generate 1-to-1 word alignments we used FastAlign (Dyer et al., 2013).

BLI For the bilingual lexicon induction task we have used MUSE (Lample et al., 2018) train and test dictionaries. As monolingual data for anchor generation needed for VecMap and RCSLS we used WikiDumps[1] for all the three languages. To extract sentences we have used WikiExtractor[2]. We generated anchors for the most frequent 50k words.

XNLI For the XNLI task, we have used English train, validation and test sets, the German test set from (Conneau et al., 2018) and the test data proposed in (Bhattacharjee et al., 2021) for Bengali.

[1] https://dumps.wikimedia.org/

[2] https://github.com/attardi/wikiextractor

5.2 Compared Language Models

We compared five multilingual and seven monolingual language models of different types and sizes. We used multilingual models for all three tasks, however, we tested monolingual models only for BLI. Since BLI is a word-level task not a transfer learning task we wanted to know how much difference different types of monolingual models can make compared to the multilingual models. We have tried monolingual models also for the word retrieval task but their performance was not satisfactory. For this reason, we have excluded monolingual models for the other two tasks (word retrieval and XNLI) to save resources, costs and time. All the used language model names as can be found on *Huggingface Hub*, their architectures, vocabulary size and other information are provided in the appendix in Table 5. Our goal was to select a diverse set of models in terms of architecture (mBERT follows BERT (Devlin et al., 2019), indic-bert (Kakwani et al., 2020) follows ALBERT (Lan et al., 2020) architecture), training data (mBERT uses Wikipedia, XLM-RoBERTa (Conneau et al., 2020) uses CommonCrawl), pretraining tasks (mBERT uses the masked language modeling (MLM) and next sentence prediction tasks, indic-bert uses MLM and sentence order prediction task), number of parameters (indic-bert has only 33M parameters and XLM-RoBERTa has 270M parameters) and vocabulary sizes (mBERT and dBERT has 119k tokens in vocabulary whereas XLM-RoBERTa has 250k tokens). In this work, we want to establish a clear and concise comparison between these language models.

5.3 Pipelines

We have several pipelines and setups for the model alignments and each of the three tasks. We briefly describe these next. For all of our experiments we have used NVIDIA TITAN X GPU with 12 GB RAM.

Alignment Following Cao et al. (2020) we fine-tuned a single multilingual model for both test language pairs (de-en and bn-en) by simultaneously using German-English and Bengali-English parallel sentences in case of both FAO and MAO. Since indic-bert does not support the German language, it was fine-tuned only with Bengali-English sentence pairs. In case of FAO we used 250K parallel sentences pairs for each of the language pairs as in (Cao et al., 2020), while for MAO we used only

25K, except for indic-bert which resulted the best performance with only 7K pairs. We selected these parameters by training the models on different numbers of sentences and testing it on the validation set. We fine-tuned the multilingual models for one epoch following Cao et al. (2020). We report the rest of the used hyperparameters in Table 6 of the appendix. Additionally, we note that adapters could only be used for three multilingual models because at the time of implementation the used AdapterHub toolkit (Pfeiffer et al., 2020) supported only mBERT, dBERT and XLM-RoBERTa but not indic-bert.

Pipelines for Word Retrieval In the word retrieval task as baseline we use language models without any fine-tuning. In the second setup, we fine-tune the multilingual language models using FAO and use it for the word retrieval task. In the third setup, we train our proposed linear and transformer layer with or without adapters.

Pipelines for BLI As baseline for BLI we use language models without any fine-tuning to generate anchors for mapping. In the second and third setups we fine-tune the multilingual language models using either FAO or MAO and use it to generate anchors and perform mapping. We map the anchors using two alignment techniques VecMap (Artetxe et al., 2016) or RCSLS (Liu et al., 2019). We perform the mapping on two language pairs Bengali-English and German-English. We use the mapping for XNLI task as well as described below.

Pipelines for XNLI As baseline for XNLI task we fine-tune the language model and a dedicated classifier layer on the English XNLI data and test them on German and Bengali data. In the second setup we fine-tune the language models using FAO first and then use this fine-tuned model in the same way as the baseline, i.e., we add an additional XNLI specific classification layer. In the third setup we train our proposed models with MAO by adding the trained alignment layers optionally together with adapters between the language model and the classifier layer. We only train the core LM and classifier on XNLI but keep the alignment layer and the adapter frozen. In the last setup, we use mapping matrices built by either VecMap or RCSLS as described above and initialize a linear layer added between the language model and the classifier layer. We do not train this linear layer when training on XNLI. We trained our models for three epochs with batch size 8 or 4 (when trained with mBERT or XLM-RoBERTa) and used $1e^{-6}$ as learning rate.

Models	de-en	bn-en	Minutes
mBERT-cased	28.45	14.55	-
mBERT-cased + FAO	39.64	43.00	500.0
mBERT-cased + lin + MAO	45.84	26.93	29.0
mBERT-cased + trans + MAO	46.73	24.27	30.5
mBERT-cased + ada + transformer + MAO	48.02	24.55	32.5
dBERT	20.71	9.71	-
dBERT + FAO	35.28	39.72	293.0
dBERT + linear + MAO	29.50	14.41	17.5
dBERT + transformer + MAO	32.21	12.58	19.0
dBERT + adapter + transformer + MAO	31.48	12.60	19.5
XLM-RoBERTa	4.33	6.40	-
XLM-RoBERTa + FAO	7.58	6.40	1893.0
XLM-RoBERTa + transformer + MAO	22.54	14.41	31.0
indic-bert	-	12.45	-
indic-bert + FAO	-	29.22	221.0
indic-bert + linear + MAO	-	15.36	4.0
indic-bert + transformer + MAO	-	13.28	4.3

Table 1: Accuracy for word retrieval task for different multilingual models for **bn-en** and **de-en**. Here bn = Bengali, de = German, en = English, trans = transformer, ada = adapter. **Minutes** column indicated the number of minutes it takes to train the model

6 Results & Discussion

We show results for our word retrieval task in Table 1. Results for BLI task is shown in Table 2, while Table 3 shows the results for the XNLI task. The results shown in these tables are the outcome of a single model per setup. We did not average the results across runs or seeds in order to reduce the required computational resources. Next we discuss the comparison of various aspects of the selected models.

Big vs. Small Models From all the results across all the task and languages we observe that big models outperformed smaller models often by a significant margin. In Table 3 for the XNLI task the zero-shot accuracy score on de test set for mBERT is 66.79, for XLM-RoBERTa it is 71.74 whereas for dBERT is 61.74 (dBERT $<$ mBERT $<$ XLM-RoBERTa). In Table 1 for Word Retrieval task accuracy score in the de-en direction for mBERT and dBERT is 28.45 and 20.71 respectively, even after model fine-tuning the scores are 39.64 and 35.28 respectively. We see this pattern for the BLI task as well, in Table 2. We should always choose big models over smaller models when we have available resources (computational, data and time).

Multilingual vs. Monolingual Models From the results of the BLI task in Table 2 it is clear that multilingual models showed far superior performance than monolingual models. In Table 2 the

Models	de-en		bn-en	
	p@1	p@5	p@1	p@5
mBERT-uncased + vec	56.84	71.50	12.33	26.54
mBERT-uncased + rcs	59.79	74.37	12.26	27.27
mBERT-cased + vec	50.95	62.29	7.43	19.43
mBERT-cased + rcs	51.54	67.47	8.71	20.50
mBERT-cased + FAO + vec	57.29	57.58	15.08	29.89
mBERT-cased + FAO + rcs	57.58	70.91	16.68	32.23
mBERT-cased + lin + MAO + vec	50.81	63.69	9.04	20.24
mBERT-cased + lin + MAO + rcs	51.47	64.43	9.45	21.47
mBERT-cased + trans + MAO + vec	51.47	62.15	7.57	19.30
mBERT-cased + trans + MAO + rcs	52.06	63.62	8.84	20.91
mBERT-cased + ada + trans + MAO + vec	50.88	62.51	7.90	18.29
mBERT-cased + ada + trans + MAO + rcs	51.25	63.62	8.51	19.97
dBERT + vec	42.70	49.70	4.15	9.98
dBERT + rcs	43.74	52.28	5.16	13.20
dBERT + FAO + vec	53.46	66.12	11.39	25.06
dBERT + FAO + rcs	53.60	66.86	13.13	27.88
dBERT + lin + MAO + vec	43.37	52.87	4.69	10.52
dBERT + lin + MAO + rcs	43.88	53.97	5.49	11.79
dBERT + trans + MAO + vec	43.00	50.44	4.15	10.18
dBERT + trans + MAO + RCSLS	44.10	52.79	5.42	12.60
dBERT + ada + trans + MAO + vec	43.22	50.14	4.75	10.53
dBERT + ada + trans + MAO + rcs	44.25	52.65	5.63	11.99
XLM-RoBERTa + vec	48.82	60.60	10.32	20.17
XLM-RoBERTa + rcs	58.54	73.49	13.67	28.21
XLM-RoBERTa + FAO + vec	50.88	61.63	6.09	12.13
XLM-RoBERTa + FAO + rcs	54.93	68.85	12.33	24.46
XLM-RoBERTa + trans + MAO + vec	50.88	61.63	14.00	29.42
XLM-RoBERTa + trans + MAO + rcs	59.35	75.03	16.28	32.90
indic-bert + vec	-	-	12.13	21.24
indic-bert + rcs	-	-	12.33	23.99
indic-bert + FAO + vec	-	-	13.73	23.72
indic-bert + FAO + rcs	-	-	15.41	26.27
indic-bert + lin + MAO + vec	-	-	13.60	23.65
indic-bert + lin + MAO + rcs	-	-	14.14	24.59
indic-bert + trans + MAO + vec	-	-	11.59	21.17
indic-bert + trans + MAO + rcs	-	-	12.53	23.72
De BERT + En BERT + vec	43.00	62.44	-	-
De BERT + En BERT + rcs	44.77	63.91	-	-
De dBERT + En dBERT + vec	25.47	43.96	-	-
De dBERT + En dBERT + rcs	27.46	46.53	-	-
De Electra + En Electra + vec	1.62	4.12	-	-
De Electra + En Electra + rcs	3.24	9.71	-	-
Bn BERT + En BERT + vec	-	-	5.16	11.86
Bn BERT + En BERT + rcs	-	-	5.29	12.66

Table 2: P@1 and P@5 scores in BLI task for different models in **de-en** and **bn-en** direction. For de-en and bn-en direction, the coverage for MUSE test set is 90.53% and 99.73% respectively. Coverage is the percentage of word pairs where both source and target word embeddings are present in our embeddings matrices. Here bn = Bengali, de = german, en = english trans = transformer, ada = adapter, vec = VecMap, rcs = RCSLS.

P@1 score for mBERT-cased using VecMap mapping approach in de-en direction is 50.95 but when we used monolingual BERT for both the German and English language the P@1 score decreased to 43.00. We see this performance decrement issue for monolingual models in the bn-en direction and for other models (dBERT) as well in Table 2. Fine-tuned mBERT-cased accompanied by RCSLS outperformed all the models in the de-en and bn-en direction. Monolingual models exhibited significantly poor performance for this word level BLI task, which we did not anticipate.

Models	en	de	bn
mBERT	79.42	66.79	55.21
mBERT + align-matrix	-	67.60	55.04
mBERT + FAO	78.48	68.76	60.92
mBERT + linear + MAO	79.52	67.72	55.51
mBERT + transformer + MAO	80.04	68.04	55.01
mBERT + adapter + transformer + MAO	80.14	69.30	54.69
dBERT	75.51	61.74	50.84
dBERT + align-matrix	-	62.44	49.74
dBERT + FAO	75.11	62.51	53.77
dBERT + linear + MAO	75.77	62.81	53.37
dBERT + transformer + MAO	74.89	62.40	52.10
dBERT + adapter + transformer + MAO	76.43	65.01	50.90
XLM-RoBERTa	80.18	71.74	67.94
XLM-RoBERTa + FAO	78.88	70.28	66.47
XLM-RoBERTa + transformer + MAO	80.52	73.05	68.14
indic-bert	75.93	-	65.59
indic-bert + align-matrix	-	-	67.58
indic-bert + FAO	76.11	-	59.80
indic-bert + linear + MAO	75.57	-	65.97
indic-bert + transformer + MAO	75.81	-	66.85

Table 3: Accuracy scores for XNLI Task for different multilingual models for three different languages **en, de and bn**. Here bn = Bengali, de = German, en = English, trans = transformer, ada = adapter, align-matrix = mapping matrix generated in BLI task using RCSLS for the corresponding language model and language.

Effect of Vocabulary Size and Language Support On the sentence level task of XNLI shown in Table 3, indic-bert outperformed mBERT on bn test set in terms of accuracy score by a large margin (indic-bert achieved accuracy score 65.59 whereas mBERT-cased achieved 55.21), it even performed on par with XLM-RoBERTa on bn (accuracy score for XLM-RoBERTa is 67.94). For low resource languages, big multilingual models mostly split the words into multiple subwords because of the small number of tokens in the vocabulary for that language. But due to parameter sharing and positive interference of high resource languages on the low resource languages (Wang et al., 2020) bigger multilingual models accomplish good performance in different tasks. indic-bert which is trained on 12 Indian subcontinent languages and English has 200k tokens in its vocabulary (though it is smaller than XLM-RoBERTa which has 250K tokens from 100 languages and mBERT has 119K tokens from 104 languages) so it does not split most of the Bengali words into subwords and can capture the context of the Bengali sentence on par with XLM-RoBERTa. Increasing the number of languages and vocabulary does not always lead to better performance.

VecMap vs. RCSLS In Table 2 for all models we observe that RCSLS mapping always outperformed VecMap for BLI task. P@1 scores in de-en and bn-

en direction for mBERT-cased using VecMap are 50.95 and 7.43 respectively while on the contrary for RCSLS P@1 scores are 51.94 and 8.71 respectively. We have also used the align-matrix generated for each of the language models and languages during the zero-shot testing in XNLI task (please refer to Table 3). We have seen that for mBERT, dBERT and XLM-RoBERTa scores increased by a small margin only for the de test set whereas for bn the scores decreased. However, for indic-bert when align-matrix was used the scores increased for bn. VecMap solves a least-square regression problem to learn a mapping. However, RCSLS proposes a unified approach where they directly optimize a retrieval criterion (Joulin et al., 2018). Therefore, RCSLS performs better than VecMap.

Model Fine-tuning Fine-tuning a multilingual model with FAO strengthen its contextualized embeddings quality. Results shown in Table 1, Table 2 and Table 3 indicate that model fine-tuning significantly improved the performance across all tasks and models. In Table 1 accuracy scores for fine-tuned mBERT in word retrieval task for de-en and bn-en direction are 39.64 and 43.00 respectively over the vanilla mBERT's accuracy scores which are 28.45 and 14.55 respectively. In Table 3, on XNLI de and bn test set fine-tuned mBERT achieved accuracy scores 68.76 and 60.92 respectively whereas vanilla mBERT achieved 66.79 and 55.21 respectively. There are some exceptions in the case of XNLI task, where fine-tuned XLM-RoBERTa and indic-bert's performance decreased. Due to constraints in computing resources, we had to fine-tune XLM-RoBERTa with a small batch size; for this reason the performance decreased for XLM-RoBERTa. We have used the same learning rate for all the models during fine-tuning the language model and classifier training for the XNLI task. That might affect fine-tuned indic-bert's performance. We believe rigorous hyperparameter tuning for model fine-tuning and training would improve the model's performance significantly but would lead to higher costs as well.

Proposed Alignment Approach From the accuracy scores reported in Table 1, our proposed alignment approach outperformed fine-tuned mBERT in the de-en direction and XLM-RoBERTa in bn-en direction for word retrieval task. Our alignment approach takes significantly less time than model fine-tuning (see **Minutes** column of Table 1).

bn-en		
Models	**trilingual**	**bilingual**
mBERT-cased + FAO	43.00	40.80
mBERT-cased + lin + MAO	26.93	27.22
mBERT-cased + trans + MAO	24.27	24.42
mBERT-cased + ada + trans + MAO	24.55	24.27
de-en		
Models	**trilingual**	**bilingual**
mBERT-cased + FAO	39.64	40.35
mBERT-cased + lin + MA0	45.84	45.47
mBERT-cased + trans + MA0	47.73	46.80
mBERT-cased + ada + trans + MAO	48.02	48.04

Table 4: Accuracy scores for word retrieval task in bilinguality vs. trilinguality study using mBERT-cased. Here bn = bengali, de = german, en = english trans = transformer, ada = adapter, lin = linear, **bn-en** and **de-en** = following scores are reported for only bn-en and de-en directions respectively, **trilingual** = the models are trained with both bn-en and de-en parallel data, **bilingual** = the models are trained with only bn-en parallel data in case of **bn-en** direction and similarly for **de-en** direction de-en parallel data is used for all model training.

This simple and smaller approach outperformed fine-tuned mBERT, dBERT on the German test set and indic-bert in the Bengali test set in the XNLI task. For the BLI task our proposed approach with XLM-RoBERTa and RCSLS outperformed all the other models for both de-en and bn-en directions by achieving P@5 scores 75.03 and 32.90 for de-en and bn-en directions respectively.

Bilinguality vs. Trilinguality We wanted to study the effect of training our proposed approaches using only a single language pair (German-English or Bengali-English) using FAO and MAO instead of using both of the language pairs simultaneously. In Table 4, trilingual column indicates the accuracy scores when the model is trained on both the German-English and Bengali-English language pairs simultaneously and the bilingual column implies the scores when the model is trained with only one of the language pairs. From Table 4 we observe that for the bn-en direction when we fine-tuned the model using FAO only with Bengali-English data the scores decreased by a small margin, the score was 43.00 (reported in the trilingual column) but it dropped to 40.80 (reported in the bilingual column). Whereas for the de-en direction when we fine-tuned the model with only German-English data the opposite occurred, the accuracy score slightly increased from 39.64 to 40.35. Hence, Bengali has minimal negative inter-

ference on German and German has minimal positive interference on Bengali in the fine-tuning process. However, in case of our proposed approach (MAO) trained with only German-English data, performance on the de-en direction of the linear and transformer model decreased. Only the score of the adapter method increased. Nevertheless, these increments and decrements were by a tiny margin. While on the contrary, when we trained the method with Bengali-English data the performance for the bn-en direction decreased for the adapter method but increased for the other two methods. Therefore, it is unclear whether bilinguality or trilinguality is advantageous over each other in the case of our proposed method.

7 Conclusion

In this paper we have compared currently popular alignment techniques using multilingual and monolingual models of various architectures from different aspects by utilizing two word level tasks (BLI and word retrieval) and one sentence level task XNLI with one low resource (Bengali-English) and one high resource language pair (German-English). We also have proposed a time, data and parameter efficient alignment technique. Our experimental results demonstrate that multilinguality always lead to better performance in cross-lingual transfer tasks. When the resources (computational and data) are available, bigger models are always preferred over smaller models, but when the resources are not accessible, smaller but specialized multilingual models should be chosen, since they are capable of performing similarly to or better than the large multilingual models on the languages the model is specialized for. A large set of supported languages and a large vocabulary does not always assist in all types of tasks in contrast to models specifically trained for a limited number of target languages. Large language models are sensitive regarding batch size and learning rate. Finally, high resource languages and large multilingual models perform well with our proposed approach. In future work we aim to develop alignment techniques capable of performing well even on low resource unseen languages.

Limitations

In case of monolingual language models, the performance of our proposed approach is significantly worse compared to multilingual models. The representations produced by the language specific monolingual models are independent from each other, while in case multilingual models they are to some extent aligned. Using the representations from monolingual models and the simple objective function of our approach, it is more difficult to obtain the same quality alignment as in case of multilingual models which needs further development.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback. The work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (No. 640550) and by the German Research Foundation (DFG; grant FR 2829/4-1).

References

Hanan Aldarmaki and Mona Diab. 2019. Context-aware cross-lingual mapping. 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 3906–3911, Minneapolis, Minnesota. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Kazi Samin, Md Saiful Islam, M. Sohel Rahman, Anindya Iqbal, and Rifat Shahriyar. 2021. Banglabert: Combating embedding barrier in multilingual models for low-resource language understanding.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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.

Wietse de Vries, Andreas van Cranenburgh, and Malvina Nissim. 2020. What's so special about BERT's layers? a closer look at the NLP pipeline in monolingual and multilingual models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4339–4350, Online. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *NAACL*.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for bengali-english machine translation.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Ann Irvine and Chris Callison-Burch. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Philipp Koehn et al. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Saurabh Kulshreshtha, Jose Luis Redondo Garcia, and Ching-Yun Chang. 2020. Cross-lingual alignment methods for multilingual BERT: A comparative study. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 933–942, Online. Association for Computational Linguistics.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1663–1674, Online. Association for Computational Linguistics.

Qianchu Liu, Diana McCarthy, Ivan Vulić, and Anna Korhonen. 2019. Investigating cross-lingual alignment methods for contextualized embeddings with token-level evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 33–43, Hong Kong, China. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space.

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. 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 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Haoyue Shi, Luke Zettlemoyer, and Sida I. Wang. 2021. Bilingual lexicon induction via unsupervised bitext construction and word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 813–826, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019. Cross-lingual BERT transformation for zero-shot dependency parsing. 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 5721–5727, Hong Kong, China. Association for Computational Linguistics.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

A Appendix

For our three different tasks, we have utilized seven monolingual models and five multilingual models. Information on the language models, including the number of parameters, model type, supported languages and vocabulary size is reported in Table 5. Hyperparameters utilized for each experiment in our word retrieval task are mentioned in Table 6. Table 4 contains the results of our bilingual and trilingual training setups.

Model	Param.	Vocab.	Type	Languages
mBERT-uncased	168M	105K	BERT	104 languages
mBERT-cased	179M	119K	BERT	104 languages
dBERT	134M	119K	Distil BERT	104 languages
XLM-RoBERTa	270M	250K	BERT	100 languages
indic-bert	33M	200K	ALBERT	13 languages
bert-base-cased	-	-	BERT	English
distilbert-base-cased	-	-	Distil BERT	English
google/electra-base-generator	-	-	Electra	English
dbmdz/bert-base-german-cased	-	-	BERT	German
distilbert-base-german-cased	-	-	Distil BERT	German
dbmdz/electra-base-german-europeana-cased-discriminator	-	-	Electra	German
sagorsarker/bangla-bert-base	-	-	BERT	Bengali

Table 5: Language models used for our experiments.

Models	Batch Size	Learning rate	Attention Head	Reduction Factor
mBERT+FAO	4	$5e^{-5}$	-	-
mBERT+lin+MAO	16	$1e^{-5}$	-	-
mBERT+transr+MAO	32	$5e^{-8}$	8	-
mBERT+ada +trans +MAO	32	$1e^{-7}$	8	8
dBERT+FAO	4	$5e^{-5}$	-	-
dBERT+lin +MAO	32	$1e^{-5}$	-	-
dBERT+trans +MAO	32	$1e^{-7}$	8	-
dBERT+ ada + trans+MAO	32	$1e^{-7}$	8	8
XLM-RoBERTa+FAO	1	$5e^{-5}$	-	-
XLM-RoBERTa+trans+MAO	32	$5e^{-5}$	8	-
indic-bert+FAO	4	$5e^{-5}$	-	-
indic-bert+lin+MAO	32	$5e^{-5}$	-	-
indic-bert+trans+MAO	32	$1e^{-8}$	8	-

Table 6: Hyperparameters used for different models for the word retrieval task. Here (-) indicates not applicable for this model, trans = transformer, ada = adapter, lin = linear.