MoSECroT: Model Stitching with Static Word Embeddings for Crosslingual Zero-shot Transfer

Haotian Ye[1,2,*], **Yihong Liu**[1,2,*], **Chunlan Ma**[1,2,*], **and Hinrich Schütze**[1,2]

[1]Center for Information and Language Processing, LMU Munich
[2]Munich Center for Machine Learning (MCML)
{yehao, yihong, chunlan}@cis.lmu.de

Abstract

Transformer-based pre-trained language models (PLMs) have achieved remarkable performance in various natural language processing (NLP) tasks. However, pre-training such models can take considerable resources that are almost only available to high-resource languages. On the contrary, static word embeddings are easier to train in terms of computing resources and the amount of data required. In this paper, we introduce **MoSECroT** (**Mo**del **S**titching with Static Word **E**mbeddings for **Cro**sslingual Zero-shot **T**ransfer), a novel and challenging task that is especially relevant to low-resource languages for which static word embeddings are available. To tackle the task, we present the first framework that leverages relative representations to construct a common space for the embeddings of a source language PLM and the static word embeddings of a target language. In this way, we can train the PLM on source-language training data and perform zero-shot transfer to the target language by simply swapping the embedding layer. However, through extensive experiments on two classification datasets, we show that although our proposed framework is competitive with weak baselines when addressing MoSECroT, it fails to achieve competitive results compared with some strong baselines. In this paper, we attempt to explain this negative result and provide several thoughts on possible improvement.

1 Introduction

The emergence of PLMs and their multilingual counterparts (mPLMs) (Devlin et al., 2019; Conneau et al., 2020) have proven effective for various NLP tasks (Artetxe et al., 2020; ImaniGooghari et al., 2023). However, such models are mostly limited to no more than a hundred languages, as the pre-training requires considerable data that is only available to these languages, leaving the majority of the world's low-resource languages uncovered. In this work, we explore the possibility of leveraging (1) a PLM in a source language, (2) static word embeddings in a target language, which are readily available for many low-resource languages and are much easier to train, and (3) a technique called model stitching, to enable zero-shot on the target language without the need to pre-train.

Our contribution is summarized as follows: (i) we introduce **MoSECroT**, a novel and challenging task for (especially low-resource) languages where static word embeddings are available. (ii) We propose a solution that leverages relative representations to construct a common space for source (English in our case) and target languages and that allows zero-shot transfer for the target languages.

2 Related Work

Aligned crosslingual word embeddings enable transfer learning by benefiting from a shared representation space for the source and target languages. Such embedding pairs are typically either trained jointly (Hermann and Blunsom, 2014; Vulic and Moens, 2016) or obtained through post-alignment (Lample et al., 2018; Artetxe et al., 2018). Our work applies a transformation in the manner of the latter to align two embedding spaces where the source embeddings are derived from a PLM and target embeddings are static word embeddings.

Based on a recent consensus that similar inner representations are learned by neural networks regardless of their architecture or domain (Kornblith et al., 2019; Vulić et al., 2020), Moschella et al. (2023) propose an approach to align latent spaces with respect to a set of samples, called parallel anchors. They transform the original, absolute space to one defined by relative coordinates of the parallel anchors, and denote all the transformed samples in the relative coordinates as relative representations.

Model stitching was proposed as a way to com-

[*]Equal contribution.

bine (stitch together) components of different neural models. Trainable stitching layers are first introduced by Lenc and Vedaldi (2015), with a series of subsequent works demonstrating the effectiveness of the approach (Bianchi et al., 2020; Bansal et al., 2021).

3 MoSECroT Task Setting

The task setting is straightforward: given a PLM of a high-resource language (regarded as the source language) and static word embeddings of another language (possibly low-resource and regarded as the target language), the goal is to achieve zero-shot transfer by using the target language embeddings directly with the source language model via embedding layer stitching. This can be done by first applying an alignment between the source and target embedding spaces and subsequently swapping the embedding matrices of the PLM.

We propose a novel method that leverages relative representations for embedding space mapping. In the following, we describe our methodology in more detail.

4 Methodology

Parallel anchor selection We first extract bilingual parallel lexica between the source and the target language. For most high-resource languages, large bilingual lexica are available from MUSE[1]. For low-resource languages, we crawl translations of source language vocabulary from PanLex[2] and Google Translate[3]. Then we derive a subset of the lexica as the parallel anchors A for our method: we only keep those parallel lexica which exist in the embeddings of source and target languages[4].

Relative representations Following Moschella et al. (2023), we build relative representations (RRs) for each token in the embedding space based on their similarities with anchor tokens in the respective language. Specifically, we compute the cosine similarity of the embedding of each token with the embedding of each anchor token. For example, in the source language, the similarity between token x_i and anchor a_j is calculated as follows:

$$r^s_{(i,j)} = \text{cos-sim}(\boldsymbol{E}^s_{\{x_i\}}, \boldsymbol{E}^s_{\{a_j\}})$$

where $\boldsymbol{E}^s_{\{x_i\}}$, $\boldsymbol{E}^s_{\{a_j\}}$ are the word embedding of x_i and a_j in the source PLM embeddings $\boldsymbol{E}^s$. The relative representation of token x_i from the source language is then defined as follows:

$$\boldsymbol{R}^s_{\{x_i\}} = [r^s_{(i,1)}, r^s_{(i,2)}, r^s_{(i,3)}, \cdots, r^s_{(i,|A|)}]$$

Note that the relative representation is sensitive to the order of the anchors, so the relative representation for each token is computed with the anchors in the same order. This computation results in a matrix $\boldsymbol{R}^s \in \mathbb{R}^{|V^s|\times|A|}$ for the source language and a matrix $\boldsymbol{R}^t \in \mathbb{R}^{|V^t|\times|A|}$ for the target language, where $|V^s|$ (resp. $|V^t|$) is the source-language (resp. target-language) vocabulary size and $|A|$ is the number of parallel anchors.

Embedding mapping The obtained relative representations are vectors in $\mathbb{R}^{|A|}$ for both source and target languages. This dimension does not suit the hidden dimension of the Transformer body of the source PLM. Therefore, we propose to map the relative representations of both source and target languages back to $\mathbb{R}^D$. Given $\boldsymbol{E}^s$ and $\boldsymbol{R}^s$ for source language (resp. $\boldsymbol{E}^t$ and $\boldsymbol{R}^t$ for target language), we compute the transformed embedding of any token x_i from the source language (resp. any token y_i from the target language) as follows:

$$\boldsymbol{F}^s_{\{x_i\}} = \frac{\sum_{n\in\mathbb{N}(x_i)}(\boldsymbol{R}^s_{\{x_i\},n}/\tau \cdot \boldsymbol{E}^s_{\{n\}})}{\sum_{n\in\mathbb{N}(x_i)} \boldsymbol{R}^s_{\{x_i\},n}/\tau}$$

$$\boldsymbol{F}^t_{\{y_i\}} = \frac{\sum_{n\in\mathbb{N}(y_i)}(\boldsymbol{R}^t_{\{y_i\},n}/\tau \cdot \boldsymbol{E}^s_{\{n\}})}{\sum_{n\in\mathbb{N}(y_i)} \boldsymbol{R}^t_{\{y_i\},n}/\tau}$$

where $\mathbb{N}(x_i)$ (resp. $\mathbb{N}(y_i)$) is the set of top-k closest anchors in terms of the cosine similarity recorded in $\boldsymbol{R}^s_{x_i}$ (resp. $\boldsymbol{R}^t_{y_i}$), $\boldsymbol{R}^s_{\{x_i\},n}$ (resp. $\boldsymbol{R}^s_{\{y_i\},n}$) is the cosine similarity between $\boldsymbol{E}^s_{\{x_i\}}$ (resp. $\boldsymbol{E}^t_{\{y_i\}}$) and $\boldsymbol{E}^s_{\{n\}}$ (resp. $\boldsymbol{E}^t_{\{n\}}$), and τ is the temperature. Note that both the resulting transformed embeddings $\boldsymbol{F}^s_{\{x_i\}}$ and $\boldsymbol{F}^t_{\{y_i\}}$ are in $\mathbb{R}^D$, because it is a weighted sum of the anchor embedding in the **source language**, i.e., $\boldsymbol{E}^s_{\{n\}}$. A simple summary of the process is to represent any token, no matter whether it is from the source or target language, as a weighted sum of the embeddings of some parallel anchors in the source-language embedding space.

[1] https://github.com/facebookresearch/MUSE

[2] https://panlex.org

[3] https://translate.google.com

[4] The source language is always English and its embeddings are extracted from English BERT's (Devlin et al., 2019) token embeddings. For target languages, embeddings are static word embeddings from fastText (Bojanowski et al., 2017).

Zero-shot stitching By far we project the target-language embeddings to $\mathbb{R}^D$ which suits the hidden dimension of the Transformer body of the source language. We can simply fine-tune the model ($\boldsymbol{F}^s$ and the Transformer body) on the source-language train set of a downstream task and then assemble a target-language model for zero-shot transfer, without training on the target language. To do this, we only need to swap the source-language embeddings $\boldsymbol{F}^s$ with target-language embeddings $\boldsymbol{F}^t$.

5 Experiments

5.1 Setup

We use the cased version of the English BERT model (`bert-base-cased`) as the source language PLM and consider eight target languages. Three of the target languages are high-resource: German (de), Spanish (es), and Chinese (zh), and the rest are low-resource: Faroese (fo), Maltese (mt), Eastern Low German (nds), Sakha (sah), and Tatar (tt). Pretrained static embeddings for all target languages are available from fastText[5], except for Eastern Low German, for which we download fastText embeddings from Huggingface[6].

We evaluate the proposed method on two text classification datasets: Multilingual Amazon Reviews Corpus (Keung et al., 2020) and Taxi1500 (Ma et al., 2023). See Appendix C for details.

Apart from the standard weighting scheme illustrated in Section 4, we propose two more settings: one where we apply softmax over relative representation weights (in Embedding mapping step), and another using sparsemax (Martins and Astudillo, 2016). Compared to softmax, sparsemax produces sparse weight distributions, meaning more similarities are concentrated on fewer anchors. We conduct preliminary experiments to identify the optimal top-k closest anchors $\in \{1, 10, 50, 100\}$ and find that the results are best when using the top 50 anchors.

5.2 Baselines

We compare our method against three baselines:

Logistic Regression (LG) We train a simple target language logistic regression classifier using the average of static word embeddings of the input sentences. This approach does not require expensive training of a language model but assumes we have sufficient target language training data for a specific downstream task, which is hardly the case for most low-resource languages in real scenarios.

mBERT We fine-tune multilingual BERT (mBERT) (Devlin et al., 2019), using the English training data, and perform zero-shot predictions directly on the target language test data.

Least squares projection (LS) We propose a straightforward approach to project target language embeddings into the same space as the English PLM embeddings. Specifically, we learn a transformation matrix $\boldsymbol{W} \in \mathbb{R}^{D^t \times D}$ by minimizing $||\boldsymbol{A}^t\boldsymbol{W} - \boldsymbol{A}^s||_F^2$, where $\boldsymbol{A}^t \in \mathbb{R}^{|A| \times D^t}$ is the embeddings of anchors in the target language and $\boldsymbol{A}^s \in \mathbb{R}^{|A| \times D}$ is the embeddings of anchors from the English PLM. We then project all target language embeddings using $\boldsymbol{W}$ and replace the BERT embedding layer with the resulting matrix.

	de	es	zh
LR	**0.61**	**0.65**	**0.51**
mBERT	**0.61**	**0.65**	**0.51**
LS	0.46	0.46	0.30
RRs standard top-50	0.53	0.51	0.38
RRs softmax top-50	0.50	0.53	0.38
RRs sparsemax top-50	0.56	0.57	0.24

Table 1: Evaluation results on the Amazon Reviews Corpus. We report macro F_1 scores on the test sets of three high-resource target languages. **Bold**: highest score per column.

5.3 Results

We present evaluation results of RRs with the proposed settings (5.1) and compare them with the baselines in Tables 1 and 2. Macro F_1 is used due to class imbalance in both datasets. We notice that the naive LS baseline is almost always beaten by the proposed method under multiple RR settings. The only exception is nds, where both LS and RRs perform badly. This observation is a strong indicator that RRs can better leverage the semantic similarity encoded in different types of embeddings than LS. Not very surprisingly, zero-shot with mBERT is effective for high-resource languages in both datasets but underperforms LS with large gaps on low-resource languages in Taxi1500, which indicates mBERT is not well-aligned across low-resource languages due to data sparsity. In contrast, for all five low-resource languages not

[5] https://fasttext.cc/docs/en/pretrained-vectors.html

[6] https://huggingface.co/facebook/fasttext-nds-vectors

	de	es	zh	mt	sah	fo	nds	tt
LR	**0.30**	0.32	0.56	**0.38**	**0.48**	**0.47**	**0.18**	**0.43**
mBERT	0.24	**0.60**	**0.62**	0.08	0.07	0.18	0.12	0.18
LS	0.14	0.26	0.24	0.08	0.12	0.06	0.08	0.07
RRs standard top50	0.20	0.44	0.28	0.14	0.16	0.16	0.06	0.14
RRs softmax top50	0.20	0.48	0.28	0.15	0.19	0.16	0.06	0.17
RRs sparsemax top50	0.24	0.37	0.13	0.15	0.18	0.20	0.13	0.21

Table 2: Evaluation results on the Taxi1500 dataset. Reported metrics are macro F_1 scores on the test sets of eight target languages. Scores are averaged over five runs with different random seeds. **Bold**: highest score per column.

seen by mBERT, RRs surpass mBERT, although the margin varies (ranging from +0.12 for sah to +0.01 for nds). On high-resource languages, none of the RR settings outperforms mBERT.

6 Analysis

In this section, we want to propose possible reasons for the suboptimal results obtained by our framework tackling the MoSECroT task.

Anchor selection The quality of the parallel anchors largely relies on the quality of the bilingual lexica, which may contain, among others, polysemous words, that may influence the alignment quality. Normalization can also be a source of ambiguity. For example, MUSE converts all words into lowercase, so the word `sie` can have three meanings in the German-English lexicon: `you`, `she`, and `they`. We (1) only consider one translation (if there are multiple) for each target language word, which may not be the most accurate one; and (2) treat all target language words whose translations are in the source language vocabulary as anchors, which increases the frequency of noisy translation pairs.

We try to decrease the influence of potentially noisy anchor pairs by reducing the number of anchors to 3000 and 500[7] through random sampling, following the observation by Moschella et al. (2023) that uniform selection from an anchor set is both straightforward and has good performance. We also remove stop words, whose translations are more unstable, from the anchor set. Neither of the two modifications shows an improvement over the full anchor set. One possible explanation is that the translation qualities vary across anchors and thus we cannot predict the quality of sampled anchors.

Translation quality We find that a large portion of translations retrieved from PanLex are of low quality. This is partly due to PanLex using intermediate languages when direct translation is unavailable for the language pair. We filter the translations by empirically setting a threshold to the translation quality scores, available through the API for every translation. Nevertheless, we note that a high translation quality score does not guarantee the translation is perfect, and many translations are good despite having low translation quality scores. We believe the lack of high-quality parallel lexica is a possible reason that RRs do not reach their full potential on low-resource languages.

Reinitialized embedding space Our method requires swapping the original PLM embeddings with the transformed English RRs before fine-tuning on English data, whereas the embedding space of RRs might diverge substantially from the original embedding space. As a result, it is unclear whether the rest of the model parameters can be adapted to the new embeddings during fine-tuning, especially on smaller datasets like Taxi1500. We thus suggest the alteration of the embedding space through reinitialization with RRs as a likely factor as to why we do not achieve good performance.

7 Conclusion

In this work, we introduce MoSECroT, a novel and challenging task that is relevant for, in particular, low-resource languages for which static word embeddings are available but few resources exist. In addition, we propose for the first time a method that leverages relative representations for embedding space mapping and enables zero-shot transfer. Specifically, we fine-tune a monolingual English language model using only English data, swap the embeddings with target language embeddings aligned using RRs, and apply zero-shot evaluation on the target language. We show that the proposed method is promising compared with mBERT on un-

[7] Original set contains about 6000 anchors.

seen languages but only modest improvements are achieved. We provide several possible reasons and leave improvement possibilities for future research.

Limitations

In this work, we propose the task of MoSECroT and a solution to leverage available static pre-trained embeddings and tackle downstream tasks for low-resource languages. Our work has a few limitations open to future research. First, we only experiment with one model architecture (BERT). Although many language-specific BERT models exist and thus our method is applicable to a wide range of high-resource source languages, it would nevertheless be interesting to compare performance across different model architectures. Second, the explored tasks are exclusively text classification tasks. We expect that the robustness of our method can be much better studied by applying it to a more diverse set of tasks.

Acknowledgements

This work was funded by the European Research Council (grant #740516).

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. 2021. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 225–236.

Federico Bianchi, Jacopo Tagliabue, Bingqing Yu, Luca Bigon, and Ciro Greco. 2020. Fantastic embeddings and how to align them: Zero-shot inference in a multi-shop scenario. In *Proceedings of the SIGIR 2020 eCom workshop, July 2020, Virtual Event, published at http://ceur-ws.org (to appear)*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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.

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.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland. Association for Computational Linguistics.

Ayyoob ImaniGooghari, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual Amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, Online. Association for Computational Linguistics.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Karel Lenc and Andrea Vedaldi. 2015. Understanding image representations by measuring their equivari-

ance and equivalence. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 991–999. IEEE Computer Society.

Chunlan Ma, Ayyoob ImaniGooghari, Haotian Ye, Ehsaneddin Asgari, and Hinrich Schütze. 2023. Taxi1500: A multilingual dataset for text classification in 1500 languages.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1614–1623, New York, New York, USA. PMLR.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. 2023. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Ivan Vulic and Marie-Francine Moens. 2016. Bilingual distributed word representations from document-aligned comparable data. *J. Artif. Intell. Res.*, 55:953–994.

Ivan Vulić, Sebastian Ruder, and Anders Søgaard. 2020. Are all good word vector spaces isomorphic? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3178–3192, Online. Association for Computational Linguistics.

A Number of closest anchors

In addition to using all (6731) parallel anchors, we consider only the top-k ($k \in \{1, 10, 50, 100\}$) closest anchors of each word. We identify the optimal value for k closest anchors based on zero-shot performance on German and Chinese portions of the Amazon Reviews Corpus (C.1). Table 3 shows results for different k values.

k	de	zh
1	0.44	0.41
10	0.51	0.38
50	0.50	0.40
100	0.51	0.38
6731	0.44	0.21

Table 3: Number of closest parallel anchors (k) and the corresponding zero-shot performance on de and zh portions of the Amazon Reviews Corpus.

B Total number of anchors

Following Moschella et al. (2023), we randomly sample a subset of the parallel anchors ($|\boldsymbol{A}| \in \{500, 3000\}$), and exclude stop words from the anchor set. Table 4 shows zero-shot performance on German and Chinese portions of the Amazon Reviews Corpus (C.1).

$\lvert\boldsymbol{A}\rvert$	de	zh
500	0.39	0.19
3000	0.19	0.19
6731	0.44	0.21

Table 4: The total number of parallel anchors and the corresponding zero-shot performance on de and zh portions of the Amazon Reviews Corpus.

C Evaluation datasets

C.1 Multilingual Amazon Reviews Corpus

Presented by Keung et al. (2020) and containing product reviews in six languages, the original dataset uses five labels corresponding to star ratings, which we aggregate into three classes: positive, neutral, and negative. We evaluate the three high-resource target languages (de, es, zh) on this dataset.

C.2 Taxi1500

Taxi1500 (Ma et al., 2023) is a classification dataset containing six classes for more than 1500 languages, including all of our target languages. We follow the authors' original training procedure and hyperparameters and use a learning rate of 1e-5 instead of 2e-5, which we find works better for our settings.

D Computational resources

Training can be completed in under three hours on eight NVIDIA GeForce GTX 1080 Ti GPUs for the Multilingual Amazon Reviews Corpus or about half an hour on a single NVIDIA GeForce GTX 1080 Ti GPU for Taxi1500.