UniBridge: A Unified Approach to Cross-Lingual Transfer Learning for Low-Resource Languages

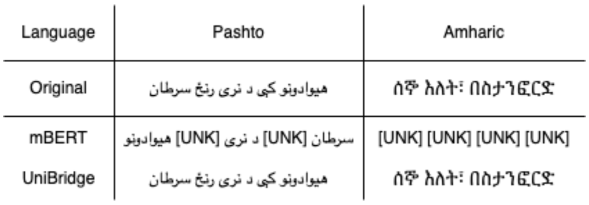
Trinh Pham[1*], **Khoi M. Le**[2*], **Luu Anh Tuan**[3†]
[1]Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City, Vietnam
[2]VinAI Research, Vietnam
[3]Nanyang Technological University, Singapore
phkhanhtrinh23@gmail.com, v.khoilm1@vinai.io, anhtuan.luu@ntu.edu.sg

Abstract

In this paper, we introduce UniBridge (Cross-Lingual Transfer Learning with Optimized Embeddings and Vocabulary), a comprehensive approach developed to improve the effectiveness of Cross-Lingual Transfer Learning, particularly in languages with limited resources. Our approach tackles two essential elements of a language model: the initialization of embeddings and the optimal vocabulary size. Specifically, we propose a novel embedding initialization method that leverages both lexical and semantic alignment for a language. In addition, we present a method for systematically searching for the optimal vocabulary size, ensuring a balance between model complexity and linguistic coverage. Our experiments across multilingual datasets show that our approach greatly improves the F1-Score in several languages. UniBridge is a robust and adaptable solution for cross-lingual systems in various languages, highlighting the significance of initializing embeddings and choosing the right vocabulary size in cross-lingual environments.

1 Introduction

Recently, multilingual pre-trained language models (LMs) have significantly advanced natural language processing (NLP) tasks, narrowing the performance gap between English and various other languages. Multilingual pre-trained models such as XLM-R (Conneau et al., 2020) and mBERT (Devlin et al., 2019) are currently strong models for effectively cross-lingual transfer (Hu et al., 2020; Artetxe et al., 2020; Le et al., 2024). However, these models pose a limitation that they are pre-trained on a limited set of approximately 100 languages, leaving a substantial void for the vast array of the world's nearly 7000 languages (van Esch et al., 2022). The resultant disparity disproportionately affects low-resource languages that are not covered in their pre-trained corpora (Wu and Dredze, 2020; Pfeiffer et al., 2020), impeding their performance compared to their high-resource counterparts.



Language	Pashto	Amharic
Original	هیوادونو کې د نری رنخ سرطان	ሰኞ እለት፡ በስታንፎርድ
mBERT	سرطان [UNK] نری د [UNK] هیوادونو	[UNK] [UNK] [UNK] [UNK]
UniBridge	هیوادونو کې د نری رنخ سرطان	ሰኞ እለት፡ በስታንፎርድ

Figure 1: Some languages/scripts are not covered in the pre-trained corpora. Hence, the pre-trained tokenizer will eventually produce many unknown tokens which corrupts the sentence's meaning and results in poor performance.

Recent efforts propose the use of adapters to mitigate the knowledge gap in low-resource languages prior to transferring knowledge for specific tasks (Pfeiffer et al., 2020; Üstün et al., 2020; Ansell et al., 2021). These methods adapt the pre-trained LMs to a new language by utilizing monolingual data, enabling the model to acquire a robust representation of the target language before receiving knowledge from the source language. Despite enhanced performance in languages not included in the pre-trained corpora, these approaches still exhibit poor performance in languages with unseen scripts (i.e., the scripts that are not presented in the pre-training corpora; see Figure 1). To address the issue of unseen scripts, existing studies (Artetxe et al., 2020; Pfeiffer et al., 2021) propose acquiring a new vocabulary embedding for newly discovered languages. However, these methods heavily rely on manually configuring the vocabulary size and initializing the embedding matrix.

Furthermore, recent Cross-Lingual Transfer Learning studies focus on English due to its abundant pre-trained data and impressive task performance, our experiments reveal that high performance in English tasks does not necessarily guar-

[*] Equal Contribution.
[†]Corresponding author.

antee successful transfer to other languages, particularly low-resource languages. Therefore, we suggest an automated method utilizing the LMs to identify the most suitable set of source languages for knowledge aggregation, leading to notable performance improvements over single-source language transfer.

Our research empirically tested the effectiveness of newly random initialized embeddings and fixed vocabulary size. We then introduce an efficient technique for determining the optimal vocabulary size for new languages, utilizing the syntactic and semantic insights from the pre-trained LMs. In addition, we present an innovative method for transferring knowledge from multiple sources, which allows the model to choose the best combination of source languages to improve the overall performance. Our results contribute to the ongoing discussion about managing linguistic diversity in NLP, particularly for languages with limited resources, emphasizing the importance of a detailed and inclusive strategy in creating multilingual pre-trained LMs.

We evaluate our approach on sequence tagging tasks (e.g. NER, POS) and classification (e.g. NLI) with two strong baselines, mBERT and XLM-R, and observe a significant increase in the F1 and accuracy score [1]. In summary, our contributions are:

- We propose a novel approach to automatic search for a suitable vocabulary size to adapt to a new language.
- We propose a new strategy to initialize the embedding that leverages the syntactic and semantic knowledge encoded in the pre-trained LMs to address the missing tokens when adapting to low-resource languages.
- We propose a method to aggregate multi-source transfer learning to enhance the performance on cross-lingual transfer tasks. We show that multi-source can outperform effective multi-language learning.

2 Methodology

Our proposed framework includes five stages as illustrated in Figure 2. In the following section we will detail each stage of the framework: **1)** Vocabulary size searching, **2)** Language-specific embedding initialization, **3)** Model adaptation to new languages not covered in the pre-training data, **4)** Downstream task training, **5)** Multi-source transfer downstream task inference.

2.1 Vocabulary size searching

Whether training from scratch or starting with a pre-trained language model, every NLP practitioner faces the task of determining the appropriate vocabulary size. Thus, choosing a suitable vocabulary size requires exhaustive searching (i.e., the whole training and testing process is required to determine the best vocabulary size). For UniBridge, the vocabulary is determined by using only CPU and is not time-consuming as it does not require any language model training phases. This is achieved by leveraging the average log probability (ALP, Zheng et al. (2021)). The algorithm for vocabulary size searching is illustrated by Algorithm 1.

Algorithm 1 Vocabulary size searching algorithm.

```
Require: $\mathcal{D}$: monolingual data, contains a list of words sentences, $v_i$: initial vocabulary size, $v_m$: maximum vocabulary size that the system should not exceed, $\delta_v$: increased step of vocabulary size, $\epsilon_s$: a threshold for stopping the algorithm.
$v \leftarrow v_i$
$t \leftarrow$ build tokenizer with vocab size $v$ on $\mathcal{D}$
$s_{prev} \leftarrow ALP(\mathcal{D}, t)$
$\Delta_s = \infty$
while $\Delta_s > \epsilon_s$ do
    $v \leftarrow v + \delta_v$
    if $v > v_m$ then
        $v \leftarrow v_m$
        $t \leftarrow$ build tokenizer with $v$ on $\mathcal{D}$
        Break the loop
    else
        $t \leftarrow$ build tokenizer with $v$ on $\mathcal{D}$
        $s_{curr} \leftarrow ALP(\mathcal{D}, t)$
        $\Delta_s = s_{curr} - s_{prev}$
        $s_{prev} \leftarrow s_{curr}$
    end if
end while
return Tokenizer $t$ with vocab size $v$
```

The concept of Average Log Probability (ALP) was introduced by Zheng et al. (2021), who argue that ALP is related to the effectiveness of subsequent tasks.

[1] https://github.com/TokisakiKurumi2001/UniBridge

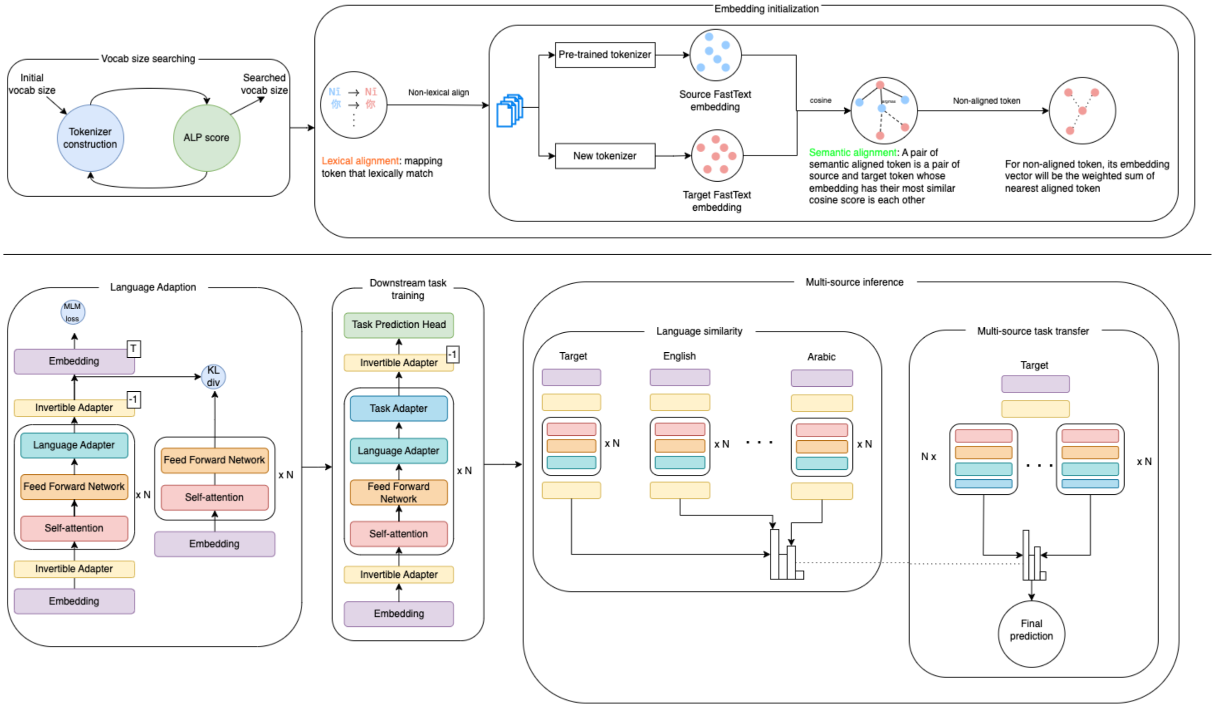


Figure 2: Illustration of UniBridge: UniBridge represents an end-to-end framework for Cross-Lingual Transfer Learning. The framework encompasses various stages, including determining the appropriate vocabulary size, initializing language-specific embedding, adapting the model to new languages, and transferring task knowledge from multiple source languages. This approach aims to harness the power of a multilingual embedding space rather than relying on a single-source transfer language, such as English.

$$ALP(\mathcal{D}, t) = \frac{1}{|t(\mathcal{D})|} \sum_{j=1}^{|t(\mathcal{D})|} \sum_{k=1}^{|s_j|} \text{log} p_{uni}(s_j^k) \quad (1)$$

For more details, readers are advised to refer to the work of Zheng et al. (2021). It is worth noting that although ALP has a high correlation with downstream tasks, the work did not provide a solution to find an optimal vocabulary size. Therefore, in this work, we propose using the 'degree of changes' in the ALP score, e.g., Δ_s. Initially, the starting vocabulary size is chosen and the ALP is calculated. Through a series of increases in the vocab size by δ_v, we can calculate the difference between the current ALP and the previous one. Thus, the algorithm will stop when the difference reaches a specific threshold ϵ_s. This threshold indicates that the optimal vocabulary size has been obtained. Continuing to increase the size will result in similar or worse performance. Therefore, we stop the algorithm to maintain the efficiency of training. Additionally, our method stands out from traditional grid search by using the 'degree of changes' of the ALP score indirectly, rather than directly as in grid search.

2.2 Language-specific embedding initialization

When training for a new language, using a randomly initialized embedding can lead to prolonged training times for optimal performance, especially in low-resource settings with a dataset size of around 10K samples. In such cases, strategically initializing the embedding proves to be more effective than a random approach. While FOCUS (Dobler and de Melo, 2023) demonstrates the use of a pre-trained LM's embedding for initialization, it depends heavily on a simple lexical overlapped alignment for subsequent stages, thus decreasing the downstream task performance. To address this gap, our approach initializes the new embedding by leveraging the pre-trained LMs in both syntactic and semantic aspects. In the initial stage, we obtain the target tokenizer t_T for the new language, with t_S being the source tokenizer of the pre-trained LMs. Representing the vocabulary sets as V^T and V^S for the target and source tokenizers respectively, and embedding matrices as $E^T[\cdot]$ and $E^S[\cdot]$, we copy the source embedding to the target embedding for the overlapping tokens $O^L = V^T \cap V^S$. This method ensures a seamless integration of knowledge from the pre-trained LMs, addressing both syntactic and semantic aspects of the new language's embedding initialization.

$$\forall o \in O^L : E^T[o] = E^S[o] \tag{2}$$

Although the number of lexical overlapping tokens can be substantial when utilizing the same script, such as Latin or Han, this phenomenon does not extend to unseen scripts. To address this challenge, we define the non-lexical alignment set as $A_T^L = V^T \setminus O^L$ and initiate a search for semantically aligned tokens within this set. Despite languages having different scripts, the underlying meanings often converge on similar definitions. To facilitate this alignment, we train two static embeddings—one for the source tokenizer (F^S) and another for the target tokenizer (F^T) —using the monolingual dataset $\mathcal{D}$. These embeddings are denoted as $F^S[\cdot]$ for the source tokenizer and $F^T[\cdot]$ for the target tokenizer. For each token v_i in A_T^L, we calculate the cosine similarity with every token v_j in $A_S^L = V^S \setminus O^L$, resulting in a matrix $S_{i,j} \in \mathbb{R}^{|A_T^L| \times |A_S^L|}$. A pair of semantically aligned tokens (v_i, v_j) is defined as a pair of source and target tokens whose embeddings exhibit the highest cosine similarity score to each other, or:

$$i = \underset{l}{argmax}(S_{l,j}) \quad \text{and} \quad j = \underset{l}{argmax}(S_{i,l}) \tag{3}$$

Refer to Equation 3, we define $S = \{(i,j) | i = \underset{l}{argmax}(S_{l,j}) \text{ and } j = \underset{l}{argmax}(S_{i,l})\}$. Each token that is semantically aligned will have the embedding copied from their counterpart from the source embeddings.

$$\forall (i,j) \in S : E^T[i] = E^S[j] \tag{4}$$

For the remaining non-aligned tokens, $A_T = A_T^L \setminus S_i$ and $A_S = A_S^L \setminus S_j$ where S_i, S_j is the set of semantically aligned token of the target and source vocabulary (i.e. $S_i = \{i | (i,j) \in S\}$, $S_j = \{j | (i,j) \in S\}$), we initialize the target embedding using the weighted sum of the aligned target tokens. We compute the cosine similarity between each non-aligned token $a_T \in A_T$ and the set of aligned target tokens (comprising both lexical and semantically aligned tokens) $o_T \in O^L \cup S_i$.

$$c_{a,o} = \frac{F^T[a_T] F^T[o_T]^\top}{\|F^T[a_T]\| \cdot \|F^T[o_T]\|} \tag{5}$$

To obtain the most similar aligned symbols o_T for a single symbol a_T, we use the same approach in Dobler and de Melo (2023), using sparsemax (Martins and Astudillo, 2016) over c_a, where c_a is a vector containing all similarity scores from $c_{a,o}$. Sparsemax is a variant softmax, but it assigns zero to low-probability element. By this, we can overcome the problem posted by skew distribution where some tokens has only one or two similar tokens while others have more. The weight $w_{a,o}$ for each aligned token o_T as defined in Equation 6.

$$w_{a,o} = \text{sparsemax}_o(c_a) \tag{6}$$

We denote S_a as a set of similar aligned tokens, which contains o_T whose probability is non-zero assigned by sparesemax.

$$S_a = \{o_T \in O^L \cup S_i | w_{a,o} > 0\} \tag{7}$$

Using the set S_a and the weight $w_{a,o}$, the embedding for the non-aligned token a_T is calculated as the weighted sum of its most similar aligned tokens.

$$\forall a_T \in A_T : E^T[a_T] = \sum_{o_T \in S_a} w_{a_T,o_T} E^T[o_T] \tag{8}$$

2.3 Model adaptation to new languages & Downstream task training

Continual pre-training, also known as language adaptation, has proven to be an effective method for enhancing the downstream performance of zero-shot cross-lingual tasks, as demonstrated by studies such as Ke et al. (2023); Alabi et al. (2022); Ebrahimi and Kann (2021). To mitigate the environmental impact and reduce model storage requirements, we opt to pre-train only a portion of the model, aligning with the approach introduced in MAD-X (Pfeiffer et al., 2020).

As in Figure 2, we made some modifications to the MAD-X configuration. Firstly, we initialize a new embedding for UniBridge which is achieved from previous stages and train the embedding together with adapters while still freezing all the pre-trained LMs' parameters. Secondly, we propose using the KL divergence together with the MLM loss (Appendix A). We see that although the frozen parameter in each layer of the pre-trained LMs helps guide the trainable adapters of the new language's embedding representation into the same pre-trained LM's embedding space, MLM is not sufficient as it only enforces the adapter to predict the mask token and this cannot guarantee the new language's representation is the same as multilingual embedding space encoded by the pre-trained LMs. This limitation prohibits the knowledge transferability of task adaptation since task adaptation takes a source language (usually high-resource languages such as English, Chinese, etc) and transfers the task knowledge directly to the target language without any alignment between the two languages. Therefore, we use KL divergence as a regularizer to guide the model not only to learn the language representation well, but also to maintain the same space as the source language in order to achieve better transferability.

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{MLM}(y, \hat{y}) \\ &\quad + \beta D_{KL}(\pi_{UniBridge}(h|x) \| \pi_{PLM}(h|x))\end{aligned} \tag{9}$$

y and $\hat{y}$ are the ground truth and prediction logits of the mask prediction task, respectively. $\pi_{UniBridge}(h|x)$ is the last hidden state of UniBridge, it is the output of the invertible adapter before goes to the linear classification head for masked predicting. $\pi_{PLM}(h|x)$ is the last hidden state of the pre-trained LMs, it is the output of the last Transformer layer, as in Figure 2, and is the input of the linear classification head for mask predicting.

2.4 Multi-source transfer downstream task inference

Instead of using one task adapter from one source language, we propose aggregating the knowledge from multiple source languages to derive a better result. For each target language, we compute the harmony weight or similarity distance between languages. Some libraries such as Lang2Vec (Malaviya et al., 2017) provide a similarity score between languages. However it does not cover all the languages. To overcome this problem, we directly use the language model (that UniBridge produced from previous stages) to measure the similarity between languages. In the Appendix D.2, we will provide a detailed comparison between Lang2Vec and UniBridge. This analysis will highlight the differences and similarities between the two approaches, offering insights into their respective performances and effectiveness.

For each target language, we collect K samples of parallel sentences from datasets such as Tatoeba (Tiedemann, 2020) or FLORES-200 (Guzmán et al., 2019; Goyal et al., 2022; Team et al., 2022) between the target language and a set of N source languages.

We denote $\mathcal{D}^T$ as a monolingual dataset extracted from the parallel dataset on the target side, $\mathcal{D}^{S_i}$ is the monolingual dataset extracted from the parallel dataset on the source side of the i-th source language. Each sentence is fed into the UniBridge with the corresponding language adapters and obtains a set of hidden states (i.e., output from the invertible adapter).

$$H_l = \{\pi^l_{UniBridge}(s) | s \in \mathcal{D}^l\} \tag{10}$$

$\pi^l_{UniBridge}$ is the UniBridge model which use the l adapter; $\mathcal{D}^l$ is $\mathcal{D}^T$ for the target language and $\mathcal{D}^{S_i}$ for the i-th source languages. The inverse L_2 distance between the target hidden state H_t for target language t and source hidden state H_s for source language s will be computed.

$$d_{t,s} = \frac{1}{L_2\text{-norm}(H_t, H_s)} \tag{11}$$

After that, we compute the softmax over the inverse L_2 distance to gain the harmony weight between target language t and set of source languages $S = \{s_i\}_{i=1}^N$.

$$w_t = \text{softmax}_s(d_{t,s}) \tag{12}$$

Using this harmony weight, instead of replacing the task adapter for each different source language during inference like MAD-X, we forward through all the task adapters in parallel. The last logit prediction will be the weighted sum of all the logits predicted by each source language weighted by the harmony weight.

$$\hat{y} = \sum_{s \in S} w_{t,s} \hat{y}_s \tag{13}$$

$\hat{y}_s$ is the logit prediction from source language s.

The intuition behind the harmony weight is that given a pair of parallel sentence, each sentence is encoded by a different language model. The difference between the hidden states produced by this process turns out to be the difference between languages itself since the sentences convey the same meaning. Therefore, inversing the difference and applying softmax will result in the similarity that we can up-weight for languages, and they could be beneficial to the target language on downstream tasks and, at the same time, down-weight the languages that are distant from the target language. Through our experiment, we show that multi-source inference outperforms single-source transfer and multi-language learning settings.

3 Experimental setup

Language set: The set of source languages are: English, Chinese, Russian, Arabic and Japanese. For the target languages, we evaluate ***14*** low-resource languages from WikiANN (Rahimi et al., 2019) whose training set consists of only 100 samples per language, ***9*** low-resource languages from Universal Dependencies (UD) whose training set consists of just few thousands samples per language and ***10*** languages from the AmericasNLI (Ebrahimi et al., 2022).

Monolingual data: For the language adaptation part, we extract from the Wikipedia dataset from HuggingFace [2] 10K samples for simulating the low-resource settings, each sample consists of 128 words, for each target language. For source languages, the number of samples is 50K per language to simulate the rich-resource languages. For languages in AmericasNLI, we use one side of the translation dataset from Mager et al. (2021).

Tokenizer: We use the SentencePiece (Kudo and Richardson, 2018) to learn the token from the monolingual data with the vocab size determined by our Algorithm 1.

Downstream data: ***NER***: We train UniBridge on the train split of WikiANN for all the source language sets and perform inference for the target language on the test split. ***POS***: We train UniBridge from the train split of UD for all the source languages sets. ***NLI***: We train UniBridge from the train split of XNLI (Conneau et al., 2018) for English, Chinese, Arabic and Russia due to the missing Japanese set.

Baseline: We evaluate UniBridge against the MAD-X framework and zero-shot cross-lingual fine-tuning using pre-trained language models (LMs). In the zero-shot approach, we fine-tune the entire pre-trained LM on the combined training data of all source languages and then directly infer on the target languages. With MAD-X, we adhere to its standard setup, training on monolingual data. To perform multi-language training, we combine training data from all source languages to train a "universal" task adapter. For inference, we swap the language adapter for each target language and integrate the "universal" task adapter. For UniBridge, we implement the language adaptation and task training stages as detailed in Section 2.3. During inference, we combine the task adapters from 5 source languages for multi-source transfer and report the F1 score for ***NER*** and accuracy score ***POS***, ***NLI*** on the target language's test split.

The hyperparameters for training, inference as well as the computational resources are given in Appendix C.

4 Results and Analysis

We present the result of our method and the baselines in Table 1 and 2 for NER task and Table 3 and 4 for POS tagging task. We report the NLI results in Table 12 and 13 in Appendix D.1. UniBridge outperforms strong baselines such as whole model fine-tuned (XLM-R, mBERT) and MAD-X framework by a large margin, i.e, for the XLM-R model, we outperform 11 over 14 languages. For POS tagging task, we outperform both baselines with two different backbone models. We also see this trend in NLI task (Appendix D.1). This highlights the effect of leveraging multiple source languages during inference to help make better decisions since each language contributes knowledge that benefits the model at prediction. Meanwhile, multi-training offers a more robust performance but also introducing more difficulties during training. The fact that UniBridge outperforms strong baselines such as whole fine-tuned model indicates that given a small monolingual and lightweight adaptation using adapters, we can significantly improve the cross-lingual tasks for uncovered languages. Compared to MAD-X, our approach differs from the use of a new embedding layer. For unseen languages, using a more specific layer of embedding can remarkably enhance the performance. Even though MAD-X already uses the invertible adapters as a component to adapt embedding layer to unseen languages, these components may not sufficient for rare languages with unseen scripts such as Amharic (**amh**), Khmer (**khm**), Kanada (**kan**). In addition, to evaluate UniBridge with large (decoder-style) Language Models (LLMs), we expanded our experiments beyond XLM-R and mBERT to include mGPT (Shliazhko et al., 2024) and mBART (Liu et al., 2020). This extension provides a more robust assessment of UniBridge's effectiveness across different model types, demonstrating its versatility

[2] https://huggingface.co/datasets/graelo/wikipedia

	amh	**ang**	**cdo**	**crh**	**eml**	**frr**	**khm**	**kan**	**lij**	**pbt**	**san**	**snd**	**sin**	**som**	**Average**
XLM-R	43.31	**52.71**	22.04	44.62	40	44.17	40.69	45.34	40.45	46	**41.28**	43.13	50.03	50.23	43.14
MAD-X (XLM-R)	39.3	46.59	17.32	36.63	33.86	39.51	**50**	45.24	38.13	42.66	19.93	39.06	39.55	49.6	38.38
UniBridge (XLM-R)	**49.6**	43.24	**42.91**	**46.03**	**40.15**	**50.67**	42.67	**48.72**	**45.16**	**46.09**	29.74	**51.32**	**52.86**	**54.22**	**45.95**

Table 1: The results of the F1 Score for every setup with XLM-R as a backbone are showcased in 14 diverse languages of WikiANN. We highlight in **bold** the highest F1 score and underline the second highest of each target language for each backbone model.

	amh	**ang**	**cdo**	**crh**	**eml**	**frr**	**khm**	**kan**	**lij**	**pbt**	**san**	**snd**	**sin**	**som**	**Average**
mBERT	12.87	52.24	19.76	**47.81**	**39.71**	51.3	18.46	42.86	45	25.86	**30.71**	13.61	2.79	46.15	32.08
MAD-X (mBERT)	13.91	51.48	16.22	46.22	39.2	45.76	19.2	31.3	37.35	29.25	22.96	20.31	12.34	37.66	30.23
UniBridge (mBERT)	**15.46**	**53.28**	**30.42**	45.67	36.15	**54.72**	**19.49**	**44.07**	**45.49**	**39.33**	20.55	**42.36**	**13.68**	**62.28**	**37.35**

Table 2: The results of the F1 Score for every setup with mBERT as a backbone showcased in 14 diverse languages of WikiANN. We highlight in **bold** the highest F1 score and underline the second highest of each target language for each backbone model.

and potential in leveraging various LLM architectures for improved language representation. The results are presented in Appendix D.8, showcasing the comparative performance and strengths of UniBridge in diverse settings.

Although UniBridge can successfully improve cross-lingual generalization, there are still some inconsistencies in the performance of a language on different tasks, e.g., Amharic (**amh**), Ligurian (**lij**), and Sanskrit (**san**) on NER and POS tasks. We hypothesize that the inconsistency arises from the misalignment in the subspace between the language adapter and the task adapter. One approach to mitigate this misalignment is to regularize the representation so that the newly learned representation is shared between the source and target languages. UniBridge leverages KL divergence as a regularization approach. This may not be sufficient to completely resolve the inconsistency, but given our constrained resources, KL divergence fits our requirements well. We leave other advanced methods, such as optimal transport or contrastive learning, for future work.

5 Ablation study

5.1 Contribution of each component

We study the contribution of each UniBridge component independently to investigate the critical components of each module. To remove KL divergence, we simply remove the KL loss from equation 9, keeping only MLM loss. To remove the embedding initialization component, we randomly initialize the embedding drawn from the Xavier normal distribution (Glorot and Bengio, 2010). To remove the vocab size search component, we fix the vocab size to 10k for every target language and use SentencePiece (Kudo and Richardson, 2018). To remove the multi-source transfer, we consider English as the single source language transferred due to its wide use in many cross-lingual transfer works.

We report the mean and standard deviation of the F1 scores between 14 languages of 2 backbone models when applying UniBridge and the components removed from UniBridge in Table 5, details of each language can be found in Appendix D.3. Among components, **embedding initialization** plays the most critical role since removing it, we experience performance drops of about 39 and 20 for XLM-R and mBERT, respectively. For **multi-source transfer** component, mBERT experiences a larger drop with 11 F1 drop while XLM-R is down from 45 to 42. However, the standard deviation when removing multi-source transfer is larger than that of UniBridge (XLM-R), indicating that multi-source benefits more languages compared to single language transferred. Although removing **KL divergence** off the XLM-R improves its performance by 1 F1 score, the standard deviation increases by 1 score. Thus, KL divergence benefits languages in maintaining a more stable improvement among languages. On the other hand, removing KL divergence while using mBERT as a backbone model hurts the performance and drops 3 F1 scores. In order to clarify the effectiveness of KL-Divergence in the other model, we conducted experiments in Appendix D.4. **Vocab size searching** with dynamic vocab size significantly improves the performance for mBERT backbone with an improvement of the 7 F1 score. This implies that different languages should be applied differently and dynamically to best adapt to their linguistic features.

	amh	lij	olo	san	snd	sin	tam	tgl	tat	Average
XLM-R	46.02	39.15	60.69	32.9	70.01	**76.25**	**85.53**	67.45	57.89	59.54
MAD-X (XLM-R)	**47.72**	58.28	69.48	36.1	71.2	73.86	83.85	69.01	65.83	63.88
UniBridge (XLM-R)	40.88	**73.75**	**81.45**	**38.94**	**71.37**	63.52	83.5	**72.62**	**81.3**	**67.81**

Table 3: The results of the accuracy for every setup with XLM as a backbone are showcased in 9 diverse languages of UD. We highlight in **bold** the highest accuracy score and underline the second highest of each target language for each backbone model.

	amh	lij	olo	san	snd	sin	tam	tgl	tat	Average
mBERT	8.59	60.66	61.49	9.35	20.39	11.47	72.93	66.3	83.2	37.82
MAD-X (mBERT)	13.31	50.47	59.61	10.88	24.93	25.68	66.61	55.56	74.17	42.36
UniBridge (mBERT)	**29.24**	**65.53**	**70.65**	**12.86**	**66.78**	**52.61**	**75.23**	**70.65**	**84.16**	**58.64**

Table 4: The results of the accuracy for every setup with mBERT as a backbone are showcased in 9 diverse languages of UD. We highlight in **bold** the highest accuracy score and underline the second highest of each target language for each backbone model.

	XLM-R	mBERT
UniBridge	45.95±6.28	37.35±15.38
- KL Divergence	46.87±7.02	34.78±17.48
- Embedding initialization	6.56±6.11	10.21±8.72
- Vocab size searching	45.48±7.54	30.59±14.55
- Multi-source transfer	42.05±9.91	25.66±12.3

Table 5: The performance of UniBridge when removing each component independently. Here, each removed component are indicating by the minus sign (-). For each removed components, other components are remained the same as the default configuration.

5.2 Vocabulary size

In this section, we contrast our approach with a novel technique for vocabulary initialization called EXTEND (Wang et al., 2020). EXTEND operates by initially expanding mBERT's vocabulary to accommodate the new language and then proceeding with pre-training on this language. In our comparison, EXTEND undergoes full fine-tuning for the MLM pre-training task. Subsequently, EXTEND is further fine-tuned using the monolingual data of each target language. Despite its extensive fine-tuning and high computational requirements, EXTEND does not perform satisfactorily on NER in comparison to UniBridge, as illustrated in Table 19 in Appendix D.5. UniBridge offers a much lighter and faster alternative, employing adapters for cross-lingual transfer learning. The lightweight and rapid nature of UniBridge significantly enhances the effectiveness of our method. Furthermore, we present an elaborate Table 20 containing various vocabulary sizes for each target language in the Appendix D.6. Regarding the lexical similarity of subwords in the vocabulary, we offer illustrations of subwords that exhibit similarity in both mBERT and XLM-R. These examples can be found in Figure 4 within Appendix B.

5.3 ALP Threshold

We conducted experiments using different ALP thresholds to identify the most effective one. We tested threshold values such as 5.0, 10.0, and 15.0 during the pre-training process of UniBridge. In essence, raising the threshold led to a decrease in vocabulary size as the algorithm ended prematurely. As a result, we noticed a decrease in the F1-Score of mBERT and XLM-R as the threshold values increased, as illustrated in Figure 3.

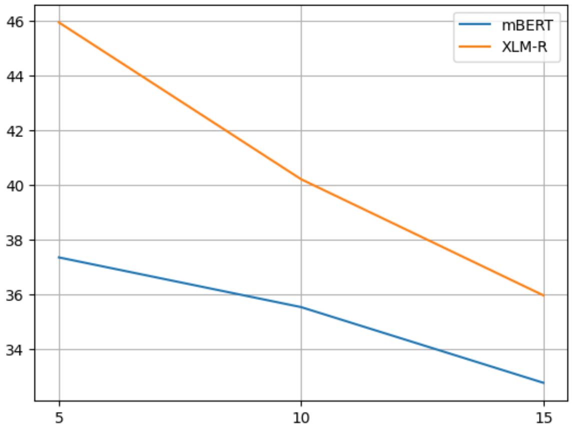


Figure 3: Mean F1-Score across various ALP thresholds.

More experiments and ablation study can be found in Appendix D.

6 Related works

Dynamic vocabulary size: It is common among NLP practitioners that the vocabulary size is considered a hyperparameter and requires manual settings. Algorithms such as BPE (Gage, 1994), Word-

Piece (Wu et al., 2016), SentencePiece (Kudo and Richardson, 2018) focus more on how to build a set of predefined number of tokens that statistically retrieved from the monolingual dataset. Some work such as VoCAP (Zheng et al., 2021), XLM-V (Liang et al., 2023) proposed algorithm to dynamically assign a vocabulary size for each language during the multilingual training. However, in monolingual training, there are some few works concerning this problem such as BPE-dropout (Provilkov et al., 2020), VOLT (Xu et al., 2021) learns to have an optimal vocab size via reducing the original vocab size using optimal transport as in VOLT or randomly removing merge in BPE-dropout.

Initialization: Artetxe et al. (2020) proposed to randomly initialize the new embedding for new language adaptation. Meanwhile, Wang et al. (2020), Chau et al. (2020), Pfeiffer et al. (2021) leverage the lexical similarity between the old vocabulary and the new vocabulary to initialize the embedding. On the other hand, there are works that explore the semantic space for initialization. SMALA (Vernikos and Popescu-Belis, 2021) directly finds the aligned token through the highest cosine similarity score. WECHSEL (Minixhofer et al., 2022) and FOCUS (Dobler and de Melo, 2023) use static embedding to find aligned tokens.

Multi-source transfer: Single-source transfer, especially, English-as-the-source-language receives many attentions. Artetxe et al. (2020), Ansell et al. (2021), Tu et al. (2022) leverages the multilingual backbone model, fine-tune on English downstream task and perform zero-shot transfer on target language test's set. Until recently, researchers have pointed out that using a multilingual training set is more beneficial compared to a single language. DeMuX (Khanuja et al., 2023) investigates the dataset level to accumulate examples that best benefit transferring using active learning. Dossou et al. (2022), Ogunremi et al. (2023) pre-train on the multilingual African dataset before distilling knowledge to downstream tasks.

7 Conclusion

In this paper, we investigate Cross-Lingual Transfer Learning, focusing on languages with constrained resources. Our contribution lies in an algorithm that autonomously determines the optimal vocabulary size for a new language, informed by its monolingual corpus, and an innovative method for initializing a new embedding matrix, drawing from both semantic and lexical facets of the pre-trained language models. Additionally, we introduce a novel technique for aggregating multi-source transfer learning, enhancing the efficacy of cross-lingual transfer tasks. Our empirical tests demonstrate the adaptability of our method across different models, yielding significant enhancements in performance. A thorough investigation of key elements highlights UniBridge's effectiveness in various situations, offering an in-depth understanding of the robustness of our approach.

Limitation

UniBridge is trained on the extracted Wikipedia with some heuristic noise filtering. However, we believe that further pre-processing such as language identification and noise filtering pipeline could further produce higher-quality monolingual data, which potentially improve the language adaptation stage. UniBridge incorporates the use of adapter to perform cross-lingual generalization, while this leverages the modular characteristic of adapter, it also inherited some limitation of the adapter itself (Kunz and Holmström, 2024; Alabi et al., 2024).

References

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jesujoba O. Alabi, Marius Mosbach, Matan Eyal, Dietrich Klakow, and Mor Geva. 2024. The hidden space of transformer language adapters.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. 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.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, Online. Association for Computational Linguistics.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Konstantin Dobler and Gerard de Melo. 2023. Focus: Effective embedding initialization for monolingual specialization of multilingual models.

Bonaventure F. P. Dossou, Atnafu Lambebo Tonja, Oreen Yousuf, Salomey Osei, Abigail Oppong, Iyanuoluwa Shode, Oluwabusayo Olufunke Awoyomi, and Chris Emezue. 2022. AfroLM: A self-active learning-based multilingual pretrained language model for 23 African languages. In *Proceedings of The Third Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 52–64, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. 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 4555–4567, Online. Association for Computational Linguistics.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios, Ivan Vladimir Meza Ruiz, Gustavo Giménez-Lugo, Elisabeth Mager, Graham Neubig, Alexis Palmer, Rolando Coto-Solano, Thang Vu, and Katharina Kann. 2022. AmericasNLI: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, Dublin, Ireland. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. 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 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*.

Simran Khanuja, Srinivas Gowriraj, Lucio Dery, and Graham Neubig. 2023. Demux: Data-efficient multilingual learning.

Taehyeon Kim, Jaehoon Oh, NakYil Kim, Sangwook Cho, and Se-Young Yun. 2021. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jenny Kunz and Oskar Holmström. 2024. The impact of language adapters in cross-lingual transfer for NLU. In *Proceedings of the 1st Workshop on Modular and Open Multilingual NLP (MOOMIN 2024)*, pages 24–43, St Julians, Malta. Association for Computational Linguistics.

Khoi M. Le, Trinh Pham, Tho Quan, and Anh Tuan Luu. 2024. LAMPAT: Low-Rank Adaption for Multilingual Paraphrasing Using Adversarial Training. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pretraining for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Manuel Mager, Arturo Oncevay, Abteen Ebrahimi, John Ortega, Annette Rios, Angela Fan, Ximena Gutierrez-Vasques, Luis Chiruzzo, Gustavo Giménez-Lugo, Ricardo Ramos, Ivan Vladimir Meza Ruiz, Rolando Coto-Solano, Alexis Palmer, Elisabeth Mager-Hois, Vishrav Chaudhary, Graham Neubig, Ngoc Thang Vu, and Katharina Kann. 2021. Findings of the AmericasNLP 2021 shared task on open machine translation for indigenous languages of the Americas. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 202–217, Online. Association for Computational Linguistics.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2535, Copenhagen, Denmark. Association for Computational Linguistics.

André F. T. Martins and Ramón Fernandez Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Tolulope Ogunremi, Dan Jurafsky, and Christopher Manning. 2023. Mini but mighty: Efficient multilingual pretraining with linguistically-informed data selection. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1251–1266, Dubrovnik, Croatia. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Anastasia Kozlova, Vladislav Mikhailov, and Tatiana Shavrina. 2024. mGPT: Few-Shot Learners Go Multilingual. *Transactions of the Association for Computational Linguistics*, 12:58–79.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2888–2913, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang.

2022. No language left behind: Scaling human-centered machine translation.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Lifu Tu, Caiming Xiong, and Yingbo Zhou. 2022. Prompt-tuning can be much better than fine-tuning on cross-lingual understanding with multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5478–5485, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Daan van Esch, Tamar Lucassen, Sebastian Ruder, Isaac Caswell, and Clara Rivera. 2022. Writing system and speaker metadata for 2,800+ language varieties. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5035–5046, Marseille, France. European Language Resources Association.

Giorgos Vernikos and Andrei Popescu-Belis. 2021. Subword mapping and anchoring across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2633–2647, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. 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 7361–7373, Online. Association for Computational Linguistics.

Bo Zheng, Li Dong, Shaohan Huang, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song, and Furu Wei. 2021. Allocating large vocabulary capacity for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3203–3215, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Why KL Divergence and MLM Loss work?

For the KL-Divergence effect, in contrast to MAD-X, UniBridge incorporates a novel training embedding. This results in the pre-task adapter representation being more inclined to reflect the characteristics of the target language compared to solely using the language adapter in MAD-X. Consequently, this introduces a misalignment between the language adapter and the task adapter, as each represents a different language.

In our research, we opt for KL-Divergence to regulate the representation to ensure it is shared across both the source and target languages (Kim et al., 2021). KL-Divergence requires less computational resources compared to other methods like layer-wise regularization or optimal transport.

For the MLM Loss effect, it is extremely effective in training encoder-only LMs because it encourages the model to learn rich contextual representations of language and facilitates effective pre-training.

In MLM, a portion of the input tokens is randomly masked, and the model is trained to predict these masked tokens based on the context provided by the surrounding tokens. This forces the model to learn contextual representations of words and phrases of that target language (Sinha et al., 2021). Moreover, by randomly masking tokens, MLM introduces noise into the training process, which can prevent overfitting and encourage the model to learn more generalizable features of the data.

B Similar tokens between pre-trained LM and UniBridge specific tokenizer

We illustrate the similar tokens between pre-trained LM and UniBridge specific tokenizer in Figure 4.

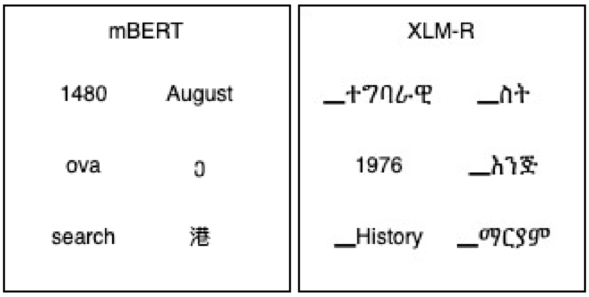


Figure 4: Illustrations of subwords exhibiting similarity in both mBERT and XLM-R.

Hyperparameter	Value
Initial vocab size v_i	7000
Maximum vocab size v_m	60000
Increased step of vocab size δ_v	1000
Threshold for stopping the algorithm ϵ_v	5.0

Table 6: The hyperparameter for vocabulary size searching process.

C Computational resources and hyperparameter for training, inference

All experiments are conducted on T4 machines. Training the UniBridge's language adapter takes approximately 2.5 hours on a single T4 machine with a batch size of 16. Separately, training UniBridge's task adapter, takes about 0.5 hours per source language on a single T4 machine with a batch size of 16.

We present the hyperparameters for training and inference for UniBridge and all the baselines' configurations in Table 6, 7, 8, 9, 10, and 11.

D More experiments and ablation study

D.1 Performance of UniBridge on NLI task

We report the performance of UniBridge on the AmericasNLI dataset in Tables 12 and 13.

D.2 UniBridge v.s. Lang2Vec

Our method's reliance on parallel data enables it to capture typological similarities as well as syntactic and semantic relationships between languages. By utilizing parallel sentences, we can develop more

Hyperparameter	Value
Static embedding model	FastText
Static embedding dimension	300
Number of epoch of training	3

Table 7: The hyperparameter for embedding initialization stage.

Hyperparameter	Value
Invertible adapter reduced factor	2
Language adapter reduced factor	2
KL divergence weight β	1.0
Masked probability	0.15
Number of epochs trained	50
Batch size	32
Learning rate	{5e-5, 2e-4, 5e-4, 1e-3}

Table 8: The hyperparameter for language adaptation training. The adapter dimension is dynamically determined by reducing the Transformer's hidden size by a factor of reduced factor. Each language has a different proportion in the pre-trained LMs' knowledge; therefore, to have an optimal language adaptation, different learning rate for different language is required.

Hyperparameter	Value
Task adapter reduced factor	16
Number of epochs trained	11
Batch size	32
Learning rate	{5e-4, 1e-3}

Table 9: The hyperparameter for task adaptation. Each language has a different proportion in the pre-trained LMs' knowledge; therefore, to have an optimal language adaptation, a different learning rate for different languages is required.

.

Hyperparameter	Value
Number of epochs trained	10
Batch size	32
Learning rate	1e-5

Table 10: The configuration of the pre-trained LMs' fine-tuning on source downstream task and zero-shot transfer to target language.

Hyperparameter	Value
Number of parallel sentences K	10

Table 11: The configuration for multi-source inference.

	aym	**bzd**	**cni**	**grn**	**hch**	**nah**	**oto**	**quy**	**shp**	**tar**	**Average**
XLM-R	36.26	**38.53**	36.4	37.33	**37.33**	39.43	36.89	37.6	35.86	34.66	37.03
MAD-X (XLM-R)	39.46	36.8	38.93	39.73	35.86	40.78	33.42	37.46	**39.06**	**36.53**	37.80
UniBridge (XLM-R)	**52.13**	36.8	**40.26**	**59.59**	35.86	**46.88**	**42.38**	**59.86**	35.6	36.4	**44.58**

Table 12: The results of the accuracy score for every setup with XLM as a backbone showcased in 10 diverse languages of AmericasNLI. We highlight in **bold** the highest accuracy and underline the second highest of each target language for each backbone model.

	aym	**bzd**	**cni**	**grn**	**hch**	**nah**	**oto**	**quy**	**shp**	**tar**	**Average**
mBERT	33.33	**33.33**	33.33	33.33	33.33	33.33	33.2	33.28	33.33	33.33	33.31
MAD-X (mBERT)	33.06	**33.33**	**34.4**	33.46	34	33.33	**33.42**	33.73	32.93	33.2	33.54
UniBridge (mBERT)	**35.73**	**33.33**	33.33	**37.46**	**34.4**	**36.31**	**33.42**	**36.66**	**34.4**	**34.4**	**34.94**

Table 13: The results of the accuracy score for every setup with mBERT as a backbone showcased in 10 diverse languages of AmericasNLI. We highlight in **bold** the highest accuracy and underline the second highest of each target language for each backbone model.

nuanced representations that reflect the intricacies of language structures and meanings.

Moreover, the quality and coverage of typological databases can be inconsistent. Although these databases are available for many languages, they often lack completeness and accuracy. In contrast, parallel corpora, while more challenging to obtain, provide direct evidence of language similarities and differences in real-world contexts. Additionally, our method has shown superior performance compared to Lang2Vec in the experiments conducted on the WikiANN dataset in Table 14 and 15.

D.3 Detail performance of each factor

We present the detailed performance of UniBridge on 14 languages on NER task when removing the contributed components in Table 16 and 17.

D.4 Effectiveness of KL Divergence

In contrast to MAD-X, UniBridge employs a novel training embedding, leading to a pretask adapter representation that better captures the characteristics of the target language than solely using the language adapter in MAD-X. To ensure that the output representation is shared between both source and target languages, we use KL-Divergence. This approach is less computationally intensive than methods such as layer-wise regularization or optimal transport (Section 6).

To assess the effectiveness of using KL-Divergence within UniBridge, we conducted extensive tests on an alternative Language Model, such as mBART, using the WikiANN dataset. The results in Table 18 indicate that KL-Divergence significantly contributes to the overall performance of UniBridge, enhancing its effectiveness considerably.

D.5 UniBridge v.s. EXTEND

We report the result on NER task compared between UniBridge and EXTEND method in Table 19.

D.6 Vocabulary searching result of UniBridge

We report the searched size of the Algorithm 1 for each language in Table 20.

D.7 UniBridge v.s. FOCUS

We compared UniBridge initialization and FOCUS initialization. For UniBridge, the whole pipeline is kept the same as discussed in the paper. For FOCUS (Dobler and de Melo, 2023), we replace step 2 discussed in Section 2.2 with FOCUS initialization pipeline while other steps are kept the same as UniBridge. We report the results on NER task in Table 21 and Table 22.

UniBridge surpasses FOCUS in performance across 10 out of 14 languages and 9 out of 14 languages on WikiANN. Among these languages, approximately 10-15% of the tokens exhibit semantic alignment. We theorize that UniBridge's advantage lies in its ability to leverage these aligned tokens, which facilitates a smoother and quicker convergence during the subsequent MLM training phase compared to FOCUS initialization.

D.8 UniBridge with Large Language Models

To evaluate UniBridge with large (decoder-style) Language Models (LLMs), we extended our experiments to include mGPT and mBART, alongside XLM-R and mBERT. This broader assessment demonstrates UniBridge's versatility and effective-

	amh	**ang**	**cdo**	**crh**	**eml**	**frr**	**khm**	**kan**	**lij**	**pbt**	**san**	**snd**	**sin**	**som**	**Average**
lang2vec (XLM-R)	30.19	**45.51**	36.28	45.8	32.23	41.72	37.75	47.45	31.67	40.05[†]	**49.79**	44.84	48.95	42.17	38.03
UniBridge (XLM-R)	**49.6**	43.24	**42.91**	**46.03**	**40.15**	**50.67**	**42.67**	**48.72**	**45.16**	**46.09**	29.74	**51.32**	**52.86**	**54.22**	**45.96**

Table 14: Comparison between Lang2Vec and UniBridge using the XLM-R backbone on the WikiANN dataset. The highest F1 score for each target language is highlighted in **bold**. The average value for each row is calculated in the last column. [†]: The language Pashto (**pbt**) does not exist in the dictionary of lang2vec thus we set the average weight for it, e.g. 0.2 for every source language.

	amh	**ang**	**cdo**	**crh**	**eml**	**frr**	**khm**	**kan**	**lij**	**pbt**	**san**	**snd**	**sin**	**som**	**Average**
lang2vec (mBERT)	8.76	26.85	**32.25**	34.0	21.23	16.0	**26.85**	37.38	27.51	28.32[†]	12.12	11.34	12.57	35.42	23.61
UniBridge (mBERT)	**15.46**	**53.28**	30.42	**45.67**	**36.15**	**54.72**	19.49	**44.07**	**45.49**	**39.33**	**20.55**	**42.36**	**13.68**	**62.28**	**37.35**

Table 15: Comparison between Lang2Vec and UniBridge using the mBERT backbone on the WikiANN dataset. The highest F1 score for each target language is highlighted in **bold**. The average value for each row is calculated in the last column. [†]: The language Pashto (**pbt**) does not exist in the dictionary of lang2vec thus we set the average weight for it, e.g. 0.2 for every source language.

ness across different model types. The results, presented in Table 23, highlight the strengths of UniBridge in diverse settings.

	am	ang	cdo	crh	eml	frr	km	kn	lij	ps	sa	sd	si	so
UniBridge (XLM-R)	**49.6**	43.24	42.91	46.03	40.15	50.67	**42.67**	48.72	**45.16**	46.09	29.74	**51.32**	**52.86**	54.22
- KL Divergence	47.66	45.61	47.1	45.91	37.78	**58.1**	40	**50**	43.92	**49.61**	**31.91**	50.74	51.1	**56.79**
- Embedding initialization	6.64	1.23	0.59	2.43	1.56	2.49	15.53	11.32	12.32	2.32	1.15	15.38	2.87	15.95
- Vocab size searching	36.13	**57.14**	**47.37**	**47.54**	**42.91**	54.95	39.65	45.76	42.75	46.44	28.06	47.35	47.51	53.11
- Multi-source transfer	40.58	56.13	36.68	45.49	35.96	57.14	32.67	45.53	39.23	33.77	22.93	39.27	47.37	55.97

Table 16: The detailed performance of UniBridge based on backbone model XLM-R when removing contributed components.

	am	ang	cdo	crh	eml	frr	km	kn	lij	ps	sa	sd	si	so
UniBridge (mBERT)	15.46	**53.28**	**30.42**	45.67	**36.15**	54.72	**19.49**	**44.07**	45.49	39.33	20.55	**42.36**	13.68	**62.28**
- KL Divergence	2.42	52.07	25.52	42.97	32.7	**55.56**	19.29	40.69	**46.15**	**40.15**	16.43	40.14	11.58	61.26
- Embedding initialization	6.58	3.59	23.53	12.35	9.84	27.75	2.34	11.06	13.04	7.61	1.54	1.1	1.2	21.36
- Vocab size searching	0.15	43.82	17.78	**48.8**	32.74	47.58	16.74	33.61	34.92	29.06	**23.32**	35.99	**15.18**	48.51
- Multi-source transfer	**25.08**	47.21	15.68	30.72	19.86	41.95	9.33	29.37	29.86	21.73	11.26	22.84	10.69	43.66

Table 17: The detailed performance of UniBridge based on backbone model mBERT when removing contributed components.

	amh	ang	cdo	crh	eml	frr	khm	kan	lij	pbt	san	snd	sin	som	Average
mBART	19.19	15.47	10.46	9.1	14.92	18.86	13.16	15.52	6.22	11.45	19.31	16.68	13.21	14.04	14.11
MAD-X (mBART)	67.03	51.24	56.57	29.73	39.13	51.5	28.79	43.52	**49.72**	45.25	51.85	58.64	**60.33**	51.2	48.89
UniBridge without KL-Divergence (mBART)	53.76	60.7	62.4	**65.67**	66.27	56.08	33.43	39.13	42.67	33.13	59.52	45.91	58.9	52.02	52.11
UniBridge (mBART)	**69.15**	**67.5**	**67.89**	61.91	**67.14**	**57.07**	**41.74**	**48.37**	44.1	**52.47**	**60.99**	**59.12**	59.29	**54.48**	**57.94**

Table 18: Various configurations with the mBART backbone on the WikiANN dataset. We highlight in **bold** the highest F1 score and underline the second highest of each target language for each backbone model.

	amh	ang	cdo	crh	eml	frr	khm	kan	lij	pbt	san	snd	sin	som	Average
mBERT	12.87	52.24	19.76	**47.81**	**39.71**	51.3	18.46	42.86	45	25.86	**30.71**	13.61	2.79	46.15	32.08
EXTEND (mBERT)	10.25	**60.66**	26.95	42.58	30.42	29.71	**22.04**	35.41	**48.63**	21.16	14.27	**49.94**	11.45	50.78	32.45
UniBridge (mBERT)	**15.46**	53.28	**30.42**	45.67	36.15	**54.72**	19.49	**44.07**	45.49	**39.33**	20.55	42.36	**13.68**	**62.28**	**37.35**

Table 19: The results of the F1 Score for every setup with mBERT as a backbone showcased in 14 diverse languages of WikiANN. We highlight in **bold** the highest F1 score and underline the second highest of each target language for each backbone model.

	am	ang	cdo	crh	eml	frr	khm	kan	lij	pbt	san	snd	sin	som
UniBridge	19k	19k	10k	8k	8k	18k	51k	27k	20k	16k	31k	14k	20k	26k

Table 20: The approximate vocabulary sizes of each target language.

	am	ang	cdo	crh	eml	frr	khm	kan	lij	pbt	san	snd	sin	som
FOCUS (XLM-R)	45.72	40.13	**43.03**	**46.03**	**41.53**	45.24	35.12	45.85	40.09	43.24	**30.15**	50.67	51.22	46.89
UniBridge (XLM-R)	**49.6**	**43.24**	42.91	46.03	40.15	**50.67**	**42.67**	**48.72**	**45.16**	**46.09**	29.74	**51.32**	**52.86**	**54.22**

Table 21: FOCUS initialization and UniBridge with XLM-R backbone on WikiANN.

	am	ang	cdo	crh	eml	frr	khm	kan	lij	pbt	san	snd	sin	som
FOCUS (mBERT)	**17.85**	48.82	20.25	**46.05**	**45.01**	**55** 15.83	43.16	43.17	36.55	**21.24**	40.83	11.07	55.85	
UniBridge (mBERT)	15.46	**53.28**	**30.42**	45.67	36.15	54.72	**19.49**	**44.07**	**45.49**	**39.33**	20.55	**42.36**	**13.68**	**62.28**

Table 22: FOCUS initialization and UniBridge with mBERT backbone on WikiANN.

	amh	ang	cdo	crh	eml	frr	khm	kan	lij	pbt	san	snd	sin	som	Average
mGPT	7.49	18.29	17.44	9.41	9.79	5.14	7.85	7.28	14.14	6.35	18.53	11.28	12.69	18.66	11.74
MAD-X (mGPT)	**63.1**	51.15	62.28	55.6	43.79	60.55	60.32	**55.75**	**63.11**	50.03	61.62	**56.66**	64.27	61.36	57.83
UniBridge (mGPT)	61.09	**60.32**	**65.13**	**63.73**	**54.06**	**69.43**	**62.35**	55.38	62.24	**54.28**	**66.07**	54.51	**66.42**	**70.29**	**61.81**
mBART	19.19	15.47	10.46	9.1	14.92	18.86	13.16	15.52	6.22	11.45	19.31	16.68	13.21	14.04	14.11
MAD-X (mBART)	67.03	51.24	56.57	29.73	39.13	51.5	28.79	43.52	**49.72**	45.25	51.85	58.64	**60.33**	51.2	48.89
UniBridge (mBART)	**69.15**	**67.5**	**67.89**	61.91	**67.14**	**57.07**	**41.74**	**48.37**	44.1	**52.47**	**60.99**	**59.12**	59.29	**54.48**	**57.94**

Table 23: Various configurations with the mGPT and mBART backbone on the WikiANN dataset. We highlight in **bold** the highest F1 score and underline the second highest of each target language for each backbone model.