Modular Monolingual Adaptation using Pretrained Language Models

Nalin Kumar and Ondřej Dušek
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Prague, Czechia
{nkumar, odusek}@ufal.mff.cuni.cz

Abstract

Building monolingual language models (LMs) for low-resource languages typically relies on adapting pretrained language models (PLMs) by finetuning the whole model on the target language. This approach is widely favored over training from scratch, as it enables effective knowledge transfer. Additionally, prior work has shown that using a language-specific tokenizer can enhance the adaptability. In this work, we hypothesize that full model tuning is often unnecessary and propose a more modular approach. Specifically, we replace the tokens, freeze the corresponding embeddings, and tune the rest of the model. We use Scottish Gaelic, Irish, and Quechua for our experiments, with Quechua being a very low-resource language (8.5k training instances). Evaluation on natural language understanding (NLU) tasks – mask-filling, NER, and POS – shows that our proposed approach improves performance when adapting the models to low-resource languages. Additionally, we provide a comprehensive analysis of the effectiveness of training strategies, the choice of pretrained embeddings, and models.

1 Introduction

There has been a great improvement in the performance of multilingual LMs in recent years, following the advent of large language models. They are not only resource-efficient and enable cross-lingual transfer learning for lower-resource languages, but also improve zero-shot skills. However, due to the *curse of multilinguality* (Conneau et al., 2020), when training a large multilingual model with fixed capacity, the higher resource languages might take up the major share of the parameter space, as models are predominantly trained on datasets consisting of English and other high-resource languages (Li et al., 2024; Achiam et al., 2023; Team et al., 2023; Dubey et al., 2024; Team et al., 2024). Consequently, this limits the downstream performance for underrepresented low-resource languages (Wu and Dredze, 2020).

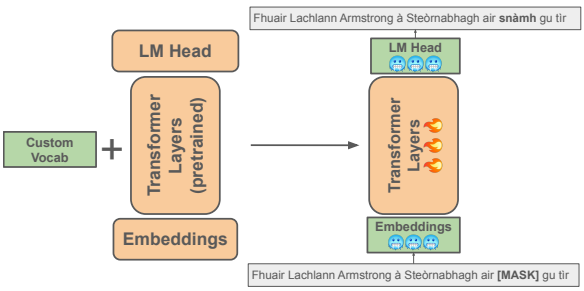

Figure 1: Overview of our proposed method. We first train a custom tokenizer and create corresponding embeddings using FastText. We replace the PLM's input and LM Head embeddings with the created embeddings and freeze them. We tune the rest of the model. As is standard in MLM models, the weights of input and output embeddings are tied.

Previous literature suggests that, while training a model in several languages improves cross-lingual transfer, there might also be negative interference for low-resource languages (Wang et al., 2020; Muller et al., 2021). There have been improvements with vocabulary extension (Chau and Smith, 2021), adapters (Pfeiffer et al., 2020b), and expert ensembles (Blevins et al., 2024). One of the recent and most effective techniques involves language adaptive finetuning (LAFT)(Alabi et al., 2022a), which incorporates training a pretrained multilingual LM (PMLM) on the same pretraining objective, such as masked language modeling (Eisenschlos et al., 2019). However, training the whole model on the available little data might lead to overfitting and be too costly.

In this work, we propose a modular adaptation of PMLMs using a custom tokenizer trained on a low-resource language. We freeze the embeddings, and train only the remaining parameters (see fig 1). This prevents the model parameters from

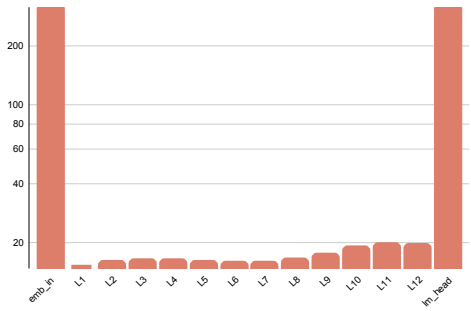

Figure 2: The graph shows the weight differences between each output layers of pretrained and finetuned BERT on Scottish Gaelic. The x-axis represents enumerated layers, while the y-axis represents the Euclidean distance between pretrained and finetuned BERT layers.

overfitting on the training examples. As is evident from Figure 2, the embedding layers are particularly susceptible to overfitting due to a larger matrix size and larger weight updates during backpropagation. Thus, we hypothesize that training the whole model for monolingual adaptation is not always ideal in low-resource settings. Within the context of monolingual adaptation, we address the main research question (**RQ**) of our study: *Does training the whole model give the best performance?*

In particular, we make the following contributions:

- We propose a modular framework for monolingual adaptation of pretrained multilingual language models (PMLMs) for low-resource languages.
- We provide a comprehensive analysis on the role of tokenizers, choice of embedding initialization and effectiveness of multilingual models for monolingual adaptation. We perform experiments across three languages (Scottish Gaelic gd, Irish ga, and Quechua qu) with multiple pretrained models (BERT, Multilingual BERT, Multilingual ModernBERT).

We evaluate our method on the mask-filling, named entity recognition (NER), and parts of speech (POS) tagging tasks. We show that the downstream performance of non-embedding tuning often surpasses the full tuning training strategy. However, the choice of embeddings plays a lesser role than anticipated. In contrast, having a custom tokenizer gives a major improvement over the multilingual tokenizer with a significantly larger vocabulary size. Our experimental code is released on GitHub. [1]

2 Related Works

Role of vocabulary in Language Modeling The choice of vocabulary plays a vital role in training an LM for low-resource language. For instance, reusing the same vocabulary of a PMLM for an unseen language might result in increased token fertility, thereby increasing computational cost (Lundin et al., 2025). To mitigate this, significant work has been done to improve language modeling by scaling vocabulary size. Chau et al. (2020) augment the vocabulary by replacing the [UNK] tokens in mBERT with 99 most frequent wordpiece tokens in the target language. They further train the model on the monolingual corpus of the target language to get better results than the baseline mBERT. Along a similar line of work, Chau and Smith (2021) proposed vocabulary augmentation and transliteration for languages with non-Latin scripts. In one of the recent works, Yamaguchi et al. (2024) experiment with vocabulary expansion in low-resource scenario. To augment the vocabulary more efficiently, Lin et al. (2025) use semantic and frequency-based method. However, Limisiewicz et al. (2023) suggest that vocabulary augmentation might not always help with all the downstream tasks, especially with token-level tasks. Another alternative is to exploit the current models and their embeddings. de Vries and Nissim (2021) adapt English GPT2 model on Italian and Dutch by relearning only the lexical embeddings while freezing the transformer layers. Hong et al. (2024) use a language modeling head tailored to the target language and finetuning it further. In another work, Rust et al. (2021) show that using a monolingual tokenizer works better than its multilingual counterpart. Building upon this work, we also experiment with replacing the pretrained model's vocabulary with the custom target language's tokens.

LAFT Language Adaptive Finetuning (LAFT) method adapts pretrained language models to unseen languages by finetuning them on monolingual texts using the same pretraining objective, thereby improving performance and handling more linguistic nuances. Sani et al. (2025) use LAFT for adapting AfriBERTa to Hausa, an extremely low-resource language. In another work, Alabi et al. (2022b) propose multilingual adaptive fine-

[1] https://github.com/knalin55/MMA-PLM

tuning on several closely related languages. In recent years, Adapter-based finetuning (Pfeiffer et al., 2020a) is one of the most cost-effective ways to train a model on a new language. However, despite having decent performance on target tasks, adapters have a worse generalizability over complete finetuning of models (Shuttleworth et al., 2024).

3 Proposed Approach

We use encoder-based models such as BERT and mBERT (Devlin et al., 2019) for our experiments. To compare the effectiveness of models for monolingual adaptation on the basis of the presence of the target language in the pretraining data, we choose slightly older models. We use Masked Language Modeling (MLM) as our pretraining objective for training our encoder-based models. To address our **RQ**, we propose a more modular approach for monolingual adaptation (Figure 1). We start with building a custom tokenizer for our target language using WordPiece (Devlin et al., 2019), keeping the vocabulary size at 30k. We train the tokenizer on the same text corpus as the language model. This not only reduces the overall number of trainable parameters significantly (25% for mBERT), but also has better sub-word fertility for all models. To create the corresponding embedding matrix, we experiment with three different strategies:

- Reusing base model embedding weights (model): We use the base tokenizer to retokenize all tokens in our newly created vocabulary. In general, most new language-specific tokens will be composed of multiple original model tokens. Therefore, we take the mean of the corresponding embedding weights.

- Static pretrained embeddings (FastText): We use FastText (Bojanowski et al., 2017) to train the embeddings on the tokenized corpus (same corpus used for training tokenizer) with embedding dimension same as the model's.

- Random: To compare the effectiveness of the choice of embeddings, we also experiment with randomly initialized embeddings.

We initialize the model embeddings with the matrix created using one of the above strategies. To give better stability while training the models for low-resource languages, we freeze the input and output embeddings, and we then use the standard MLM objective to train only the non-embedding parameters on the target low-resource language.

4 Experimental Setup

4.1 Datasets and Training Setup

We run our experiments on the CC-100 dataset (Conneau et al., 2020; Wenzek et al., 2020) for the Scottish Gaelic (gd), Irish (ga), and Quechua (qu) languages. The CC-100 dataset provides web-crawled unsupervised corpora for more than 100 languages. We preprocess the data by filtering out extremely small sentences. The resultant training data comprises 8.5k instances for qu, 250k for gd and 500k for ga. We use this data to train our custom tokenizer, the FastText embeddings, and to train the language models with the MLM objective.

We use the base variant for all our models. We train the models for 50 epochs with early stopping based on MLM accuracy. We use dynamic batch size depending on the available GPU. We use the default values for other training hyperparameters using HuggingFace Trainer. We report averaged scores over 3 pretraining runs, along with the standard deviation, for all settings.

4.2 Evaluation Metrics

We evaluate all our models on the mask-filling task (MLM). We also use Named Entity Recognition (NER) and Parts-of-Speech Tagging (POS) as downstream tasks to evaluate our models. We use the WikiAnn dataset (Pan et al., 2017; Rahimi et al., 2019; Lovenia et al., 2024) for the evaluation on the NER task. WikiAnn is a multilingual NER dataset consisting of annotations (LOC, PER, ORG) on Wikipedia articles. The gd and qu datasets comprise 100 instances for each training, development, and test set. The ga dataset consists of 1000 instances each for train, development, and test sets.

Additionally, we use the UD Treebanks (Nivre et al., 2020) as a source of data for the POS task. The gd dataset consists of 3.5k, 656, and 548 instances for train, development, and test sets, respectively, with 17 unique tags from the universal POS tagset (Petrov et al., 2012). The train, development and test sets for ga dataset consist of 4k, 451, and 454 instances. Due to the unavailability of a UD Treebank for the qu language, we do not evaluate the POS tagging task.

We fully finetune the models on downstream tasks, as our goal here is to prioritize task-specific performance rather than generalization. Given that

	tokenization / embeddings	training full	training emb	training non-emb
BERT	model / model	21.8 ±5.2	8.7 ±3.5	23.0 ±2.1
	model / random	23.2 ±0.7	0.0 ±0.0	23.6 ±0.0
	custom / model	32.9 ±1.4	21.8 ±1.4	50.5 ±0.4
	custom / FastText	34.0 ±5.7	1.5 ±0.1	53.4 ±0.2
	custom / random	40.3 ±0.6	0.0 ±0.0	49.3 ±0.3
mBERT	out-of-the-box model		6.9	
	model / model	28.3 ±2.2	24.1 ±0.1	34.0 ±1.0
	model / random	29.7 ±1.1	0.4 ±0.6	21.0 ±0.8
	custom / model	33.4 ±1.0	32.2 ±0.1	54.3 ±0.2
	custom / FastText	42.4 ±0.5	1.3 ±0.1	55.0 ±0.1
	custom / random	38.6 ±0.7	0.0 ±0.1	49.5 ±1.5
mmBERT	model / model	30.9 ±0.6	24.2 ±0.4	41.3 ±0.3
	model / random	11.4 ±1.5	0.1 ±0.1	37.8 ±0.0
	custom / model	41.0 ±1.5	19.8 ±0.7	**57.3** ±0.6
	custom / FastText	20.6 ±18.0	0.1 ±0.0	54.1 ±1.0
	custom / random	36.0 ±1.4	0.1 ±0.1	49.7 ±0.4
rnd. init.	model / random	25.5 ±0.8	0.0 ±0.0	34.5 ±1.0
	custom / FastText	30.3 ±0.2	0.0 ±0.0	52.9 ±0.4
	custom / random	38.6 ±0.1	0.7 ±0.5	46.9 ±0.2

Table 1: Mask filling accuracy for the Scottish Gaelic (gd) language (rnd. init. = randomly initialized model).

	tokenization / embeddings	training full	training emb	training non-emb
BERT	model / model	84.0 ±0.6	80.6 ±1.7	**86.2** ±1.2
	model / random	83.1 ±1.6	65.8 ±3.7	83.1 ±0.9
	custom / model	84.7 ±0.6	82.8 ±0.9	85.0 ±0.3
	custom / FastText	82.4 ±0.6	83.1 ±0.8	75.5 ±2.2
	custom / random	81.6 ±1.3	79.4 ±0.6	80.2 ±1.9
mBERT	model / model	83.9 ±1.1	83.1 ±1.3	82.7 ±0.4
	model / random	81.2 ±3.3	55.9 ±0.6	81.1 ±0.7
	custom / model	85.3 ±0.3	83.5 ±1.5	85.0 ±0.2
	custom / FastText	84.3 ±0.5	71.3 ±0.4	84.2 ±1.2
	custom / random	82.1 ±0.7	61.2 ±5.8	80.7 ±0.6
mmBERT	model / model	63.4 ±2.6	62.8 ±2.4	63.1 ±2.8
	model / random	56.3 ±3.2	54.8 ±0.0	59.6 ±1.3
	custom / model	77.5 ±1.7	81.8 ±2.0	77.7 ±0.9
	custom / FastText	77.1 ±2.5	75.6 ±3.3	75.0 ±1.9
	custom / random	75.9 ±0.9	55.8 ±12.2	76.6 ±0.2
rnd. init.	model / random	82.3 ±1.9	71.5 ±0.6	80.4 ±2.0
	custom / FastText	79.8 ±1.4	74.2 ±1.0	80.0 ±0.1
	custom / random	82.1 ±0.4	76.1 ±1.4	81.3 ±1.0

Table 2: NER accuracy in Scottish Gaelic (gd).

our experiments focus on extremely low-resource languages, we prioritize overall prediction correctness; therefore, we report scores using the accuracy metric across all considered tasks.

4.3 Model Variants

We use multilingual modern BERT (mmBERT) (Marone et al., 2025), multilingual BERT (mBERT), and standard monolingual BERT (Devlin et al., 2019) as our base models. We also compare to a randomly initialized transformer model of the same shape as mBERT.

We experiment with *custom* tokenizers, along with the three choices of embeddings discussed in Section 3 (*model*/*FastText*/*random*). In addition, we compare these setups to re-using the models' own tokenization (*model*), with either models' own or randomly initialized embeddings (*model*/*random*).

We apply the non-embedding training strategy (*non-emb*), as described in Section 3. We compare it to training the full model (*full*) or training the embedding parameters only (*emb*) while keeping the internal transformer layers frozen.

Additionally, we compare our approach with the recent parameter-efficient finetuning method, low-rank adaptation (LoRA) (Hu et al., 2021). The method decomposes each large matrix (i.e., all linear and attention layers of the model), into two small learnable low-rank matrices in the attention layer to reduce the training parameters. We use a LoRA rank of 64 and α = 128.

5 Results

In this section, we provide a detailed analysis of our findings. In Tables 1, 2 & 3, we report accuracy scores for mask-filling, NER and POS tagging tasks, respectively, on gd using different models. We further compare our results with LoRA in Table 6. Table 4 reports the scores for the ga language using mBERT, while Table 5 shows the results from mmBERT models trained for the qu language. Additionally, we report basic system-level efficiency metric values in Table 7, comparing the efficiency of our best setup with full finetuning.

Training the whole model vs. training non-embedding parameters only The non-embedding training strategy consistently outperforms both full finetuning and embedding-only training on the mask-filling (MLM) task across all languages and experimental settings. This trend holds for gd, ga, and qu, and is observed across different models and embedding initialization techniques. We attribute this improvement to the regularizing effect of freezing the input and output embedding layers. Updating the entire embedding matrix, which constitutes almost 25% (for masked language modeling) of the parameter space, can lead to overfitting, especially for low-resource languages. By freezing the embeddings and updating only the transformer layers, the model is able to adapt its higher-level represen-

	tokenization / embeddings	training full	training emb	training non-emb
BERT	model / model	97.4 ±0.1	95.5 ±0.3	97.3 ±0.0
	model / random	96.6 ±0.0	65.1 ±0.2	96.3 ±0.2
	custom / model	96.9 ±0.1	96.1 ±0.1	96.9 ±0.0
	custom / FastText	96.7 ±0.1	94.5 ±0.2	96.7 ±0.1
	custom / random	96.1 ±0.1	95.6 ±0.2	96.0 ±0.1
mBERT	model / model	97.5 ±0.1	97.0 ±0.3	**97.7** ±0.1
	model / random	96.2 ±0.3	72.7 ±0.0	96.3 ±0.1
	custom / model	97.2 ±0.1	96.7 ±0.0	97.2 ±0.1
	custom / FastText	97.0 ±0.1	97.0 ±0.0	93.9 ±0.2
	custom / random	96.4 ±0.1	80.0 ±0.3	96.3 ±0.2
mmBERT	model / model	89.6 ±0.6	90.6 ±0.4	90.6 ±0.3
	model / random	88.1 ±0.4	89.2 ±0.7	89.1 ±0.6
	custom / model	97.1 ±0.1	97.0 ±0.1	97.3 ±0.1
	custom / FastText	96.8 ±0.1	94.2 ±0.3	96.7 ±0.1
	custom / random	96.0 ±0.1	88.3 ±0.4	95.6 ±0.1
rnd. init.	model / random	96.5 ±0.2	72.9 ±0.4	96.0 ±0.2
	custom / FastText	95.8 ±0.2	82.8 ±0.3	96.0 ±0.1
	custom / random	96.1 ±0.1	81.5 ±0.2	96.0 ±0.2

Table 3: POS tagging accuracy in Scottish Gaelic (gd).

tations without excessively altering the embedding space, resulting in better generalization.

For the NER and POS tagging downstream tasks, full finetuning and non-embedding training show comparable results, with both outperforming embedding-only training. This indicates that updating the embedding layer alone is insufficient for learning the task-specific contextual representations.

Effectiveness of using PLM Compared to a randomly initialized model in Table 1, mBERT with FastText embeddings achieves a slightly higher MLM score (55.0) than the randomly initialized transformer (52.9).

Similarly, for the NER and POS tagging tasks, mBERT-based models has better performance. These results demonstrate the benefits of pretrained language models and their ability to transfer knowledge effectively to low-resource settings.

Effectiveness of multilingual PLM We train a BERT model for similar settings and report the scores for the gd language in Tables 1, 2 & 3 for MLM, NER and POS tagging tasks, respectively. Compared to mBERT and mmBERT, BERT achieves slightly lower overall MLM scores. In the default configuration (model/model), multilingual models consistently outperform BERT across all training strategies. However, when the embeddings are randomly initialized, the performance of BERT slightly surpasses mBERT and mmBERT on the MLM task, showing the critical role of pre-

	tokenization / embeddings	training full	training emb	training non-emb
MLM	out-of-the-box model		21.2	
	model / model	28.3 ±1.1	27.1 ±0.2	39.4 ±1.5
	model / random	27.4 ±0.8	00.3 ±0.2	37.4 ±1.3
	custom / model	34.2 ±3.4	38.0 ±0.3	**57.1** ±0.3
	custom / FastText	13.1 ±6.0	00.3 ±0.1	46.8 ±9.2
	custom / random	36.6 ±2.3	00.6 ±0.6	52.4 ±0.8
NER	model / model	93.1 ±0.1	93.1 ±0.1	93.0 ±0.0
	model / random	91.0 ±0.5	72.7 ±1.6	91.1 ±0.3
	custom / model	93.1 ±0.6	92.1 ±0.4	**93.2** ±0.3
	custom / FastText	92.0 ±0.4	88.3 ±0.8	91.7 ±0.4
	custom / random	90.9 ±0.2	71.3 ±0.4	90.3 ±0.0
POS	model / model	**96.7** ±0.0	96.0 ±0.0	96.1 ±0.1
	model / random	95.8 ±0.2	75.2 ±0.2	95.9 ±0.3
	custom / model	96.0 ±0.3	95.2 ±0.2	96.0 ±0.1
	custom / FastText	95.8 ±0.3	95.7 ±0.2	93.3 ±0.8
	custom / random	95.6 ±0.1	83.0 ±0.5	95.4 ±0.2

Table 4: Accuracy for mBERT on Irish (ga) mask filling, POS and NER.

	tokenization / embeddings	training full	training emb	training non-emb
MLM	model / model	09.2 ±0.3	02.3 ±0.2	08.9 ±0.1
	model / random	06.0 ±0.4	00.0 ±0.0	07.8 ±0.3
	custom / model	24.0 ±0.5	01.8 ±0.1	**29.4** ±0.4
	custom / FastText	23.7 ±0.9	00.9 ±0.2	28.5 ±0.5
	custom / random	18.7 ±1.6	00.0 ±0.0	23.0 ±1.3
NER	model / model	63.2 ±3.5	64.0 ±2.7	59.6 ±1.6
	model / random	56.0 ±0.2	57.5 ±0.6	58.0 ±0.3
	custom / model	**84.6** ±0.7	81.9 ±0.1	82.3 ±1.4
	custom / FastText	81.6 ±2.6	78.3 ±1.4	81.7 ±0.9
	custom / random	64.9 ±5.7	55.5 ±3.2	69.7 ±0.7

Table 5: Accuracy for mBERT on Quechua (qu) mask filling and NER.

trained multilingual embeddings in enabling effective cross-lingual knowledge transfer.

For the NER task, BERT performs comparably to or slightly better than mBERT under embedding-only and non-embedding training setups. In contrast, with full finetuning, mBERT achieves higher NER scores than BERT. Meanwhile, mmBERT yields noticeably lower NER performance than BERT across all configurations.

Language inclusion in pretraining data Table 4 provides MLM, NER and POS tagging task scores on mBERT for the ga language. As shown in the table, full finetuning yields only a slight improvement in MLM performance over the out-of-the-box mBERT model, supporting the idea that a language's prior inclusion in the model's multilingual pretraining data facilitates monolingual adaptation to this language. The custom/model setting with

	tokenization / embeddings	task accuracy MLM	NER	POS
mBERT	model / model	23.8 ±1.2	79.1 ±1.9	95.3 ±0.1
	custom / model	**35.1** ±0.9	**80.8** ±0.9	**95.8** ±0.2
	custom / FastText	23.1 ±1.3	70.1 ±0.7	92.5 ±0.3
BERT	model / model	16.3 ±0.3	78.9 ±0.0	94.5 ±0.4
	custom / model	31.9 ±1.0	79.8 ±1.3	95.1 ±0.1
	custom / FastText	30.3 ±0.9	74.0 ±0.3	93.7 ±0.1

Table 6: Mask filling, NER and POS accuracy for LoRA finetuning on Scottish Gaelic (gd).

model / method	# tr. par.	VRAM	tr. time	latency
mBERT / full	178M	1.3G	32H	9.7ms
mBERT / ours	85.6M	0.9G	14H	8.6ms
BERT / full	108M	0.8G	30H	9.0ms
BERT / ours	85.6M	0.9G	15H	8.6ms
mmBERT / full	308M	2.3G	71H	19.2ms
mmBERT / ours	110M	1.0G	21H	13.3ms

Table 7: System-level efficiency metrics on Scottish Gallic (gd), comparing full finetuning with our setup (custom tokenization, model-initialized embeddings, non-embedding finetuning). We report the number of trainable parameters (# tr. par.) in million parameters (M), VRAM usage at training time in gigabytes (G), training time (tr. time) in hours (H), and average inference latency (time to process one input) in milliseconds (ms).

non-embedding training still gives an improvement of more than 2.5× over the base model for the MLM task, regardless of the fact that the data set was already included in the pretraining setup.

Choice of tokenizer and embeddings In line with Rust et al. (2021), our custom tokenizer consistently outperforms the multilingual tokenizer on the mask-filling task. This improvement can be attributed to the improved finetuned subword segmentation on the target language, which reduces fertility and improves the proportion of continued words. In addition to improving performance, the custom tokenizer also substantially reduces model size – particularly for token classification tasks – since the embedding layers occupy nearly 25% of the total parameters. A smaller, language-specific vocabulary, thereby, not only enhances efficiency but also increases training speed due to lower computational and memory requirements.

In contrast, the choice of embedding initialization does not play a major role in the non-embedding training setting. There is only a slight improvement in the MLM scores for FastText embeddings over the other choices (Table 1). Randomly initialized embeddings, interestingly, perform only slightly worse than both model-based and FastText embeddings. This suggests that the transformer layers can relearn and realign to the new embedding space, acting as an effective regularization strategy. Under full finetuning, however, the default embeddings seem to have a negative impact, likely because the model is heavily biased toward non-gd knowledge and has to first unlearn irrelevant representations.

Better encoder model mmBERT achieves slightly better performance than mBERT and BERT on the mask-filling task. This gain can likely be attributed to its larger model capacity and stronger cross-lingual representations, due to training on a more diverse pretraining corpus. Due to its richer embedding space, mmBERT also outperforms the corresponding FastText-based setting. Surprisingly, the performance of mmBERT is lower than mBERT and BERT in some variants for the downstream tasks. We suspect that the reason might be "catastrophic overtraining" (Springer et al., 2025), due to which the model fails to finetune on a downstream task.

Training on very low-resource language Experiments for qu follow a similar trend as for gd. In particular, the non-embedding training strategy outperforms full tuning with default and FastText emebddings. Furthermore, similar to previous trends, models trained with the default tokenizer underperform compared to those using a custom tokenizer. This can be attributed to the minimal representation of qu in the model's pretraining data, resulting in weaker language representations.

In the NER evaluation, models with default tokenization again performs worse than their custom-tokenized counterparts. Among the custom tokenizer variants, FastText and model embeddings have comparable performance, while random embeddings perform slightly worse than both.

Overall, the performance on qu remains low across both evaluation metrics, primarily due to the limited size of the available training data.

Comparison with LoRA Although LoRA scores are comparable to the full finetuning strategy, our proposed training approach consistently outperforms LoRA on all three tasks (Table 6). We suspect that the method's low rank restricts the model's capacity to learn more complex tasks, thereby limiting performance.

System-level metrics As shown in Table 7, our proposed approach is more energy-efficient than model finetuning. Since our method has a smaller vocabulary, the number of training parameters is significantly lower than in full model tuning, thereby using less VRAM and having better latency at inference time. Additionally, due to better embedding initialization and smaller model size, our approach converges faster than the conventional full training method.

6 Conclusion

In this work, we adapt a PLM to a new language. We hypothesize that tuning the entire parameter space is often unnecessary and might even be suboptimal for low-resource language modeling. To address this, we propose a modular monolingual adaptation strategy that replaces the tokenizer with a language-specific one, initializes new embeddings using different strategies, freezes the embedding layers, and trains only the non-embedding parameters using the masked language modeling objective.

We experiment with multiple models (BERT, mBERT, and mmBERT) in multiple languages (gd, ga & qu) and training strategies. Our results consistently show that non-embedding training outperforms or matches full finetuning on the masked language modeling task, while requiring fewer trainable parameters and reducing training cost. We attribute this improvement to better regularization due to frozen embeddings, which mitigates overfitting in low-resource settings. On downstream NER and POS tagging tasks, non-embedding training performs comparably to full finetuning and consistently outperforms embedding-only training, thereby showing the effectiveness of our proposed approach.

We show that multilingual pretrained models provide clear advantages over randomly initialized transformers, demonstrating effective cross-lingual knowledge transfer—even when the target language is absent or minimally represented in pretraining. Furthermore, the inclusion of a related language in pretraining results in effective adaptation, but it does not seem to be a necessary condition. Also, following previous work, we validate the use of a custom tokenizer over multilingual tokenizers. It consistently improves performance over the default multilingual tokenizer and reduces fertility and improves the proportion of continued words, particularly for low-resource languages. In contrast, the choice of embedding initialization (model-based, FastText, or random) has a relatively minor performance improvement for the non-embedding training settings, suggesting that the transformer layers can effectively realign to new lexical spaces.

Overall, our findings demonstrate that full model finetuning is not always the optimal strategy for low-resource monolingual adaptation. A modular approach that combines custom tokenization with selective parameter training provides a simple, parameter-efficient, and effective alternative. For future work, our approach can be extended to decoder-based architectures, more low-resource languages and building mutlilingual models for a language family.

7 Limitations

Our experiments cover only three low-resource languages (Scottish Gaelic, Irish, and Quechua), which limits the extent to which our approach transfers to typologically diverse languages with different scripts or morphological properties. The evaluation is restricted to relatively simple NLU tasks (mask-filling, NER, POS) due to the lack of available gold-standard data for more complex tasks in the target languages. While it is possible to generate synthetic data for such tasks based on English benchmarks, the result will induce noise due to alignment errors and localization/cultural biases. We reserve these experiments for further work.

8 Acknowledgments

This work was funded by the European Union (ERC, NG-NLG, 101039303) and by Charles University projects GAUK 302425 and SVV 260 821. It used resources of the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2023062).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jesujoba Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022a. 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.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022b. 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.

Terra Blevins, Tomasz Limisiewicz, Suchin Gururangan, Margaret Li, Hila Gonen, Noah A. Smith, and Luke Zettlemoyer. 2024. Breaking the curse of multilinguality with cross-lingual expert language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10822–10837, Miami, Florida, USA. Association for Computational Linguistics.

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.

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.

Ethan C. Chau and Noah A. Smith. 2021. Specializing multilingual language models: An empirical study. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 51–61, Punta Cana, Dominican Republic. 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.

Wietse de Vries and Malvina Nissim. 2021. As good as new. how to successfully recycle English GPT-2 to make models for other languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 836–846, 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.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Julian Eisenschlos, Sebastian Ruder, Piotr Czapla, Marcin Kadras, Sylvain Gugger, and Jeremy Howard. 2019. MultiFiT: Efficient multi-lingual language model fine-tuning. 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 5702–5707, Hong Kong, China. Association for Computational Linguistics.

Jimin Hong, Gibbeum Lee, and Jaewoong Cho. 2024. Accelerating multilingual language model for excessively tokenized languages. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11095–11111, Bangkok, Thailand. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ninghao Liu, and Mengnan Du. 2024. Quantifying multilingual performance of large language models across languages. *arXiv preprint arXiv:2404.11553*.

Tomasz Limisiewicz, Jiří Balhar, and David Mareček. 2023. Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5661–5681, Toronto, Canada. Association for Computational Linguistics.

Nankai Lin, Peijian Zeng, Weixiong Zheng, Shengyi Jiang, Dong Zhou, and Aimin Yang. 2025. Rethinking vocabulary augmentation: Addressing the challenges of low-resource languages in multilingual models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2919–2934, Abu Dhabi, UAE. Association for Computational Linguistics.

Holy Lovenia, Rahmad Mahendra, Salsabil Maulana Akbar, Lester James V. Miranda, Jennifer Santoso, Elyanah Aco, Akhdan Fadhilah, Jonibek Mansurov, Joseph Marvin Imperial, Onno P. Kampman, Joel Ruben Antony Moniz, Muhammad Ravi Shulthan Habibi, Frederikus Hudi, Railey Montalan, Ryan Ignatius, Joanito Agili Lopo, William Nixon, Börje F. Karlsson, James Jaya, and 42 others. 2024. Seacrowd: A multilingual multimodal data hub and benchmark suite for southeast asian languages. *arXiv preprint arXiv: 2406.10118*.

Jessica M Lundin, Ada Zhang, Nihal Karim, Hamza Louzan, Victor Wei, David Adelani, and Cody Carroll. 2025. The token tax: Systematic bias in multilingual tokenization. *arXiv preprint arXiv:2509.05486*.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme.

2025. mmbert: A modern multilingual encoder with annealed language learning. *arXiv preprint arXiv:2509.06888*.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. 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.

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.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. 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 3118–3135, Online. Association for Computational Linguistics.

Sani Abdullahi Sani, Shamsuddeen Hassan Muhammad, and Devon Jarvis. 2025. Investigating the impact of language-adaptive fine-tuning on sentiment analysis in Hausa language using AfriBERTa. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 101–111, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. 2024. Lora vs full fine-tuning: An illusion of equivalence. *arXiv preprint arXiv:2410.21228*.

Jacob Mitchell Springer, Sachin Goyal, Kaiyue Wen, Tanishq Kumar, Xiang Yue, Sadhika Malladi, Graham Neubig, and Aditi Raghunathan. 2025. Overtrained language models are harder to fine-tune. In *International Conference on Machine Learning*, pages 56719–56789. PMLR.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

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

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

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.

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. 2024. How can we effectively expand the

vocabulary of llms with 0.01 gb of target language text? *arXiv preprint arXiv:2406.11477*.