TokAlign: Efficient Vocabulary Adaptation via Token Alignment

Anonymous ACL submission

Abstract

Tokenization serves as a foundational step for Large Language Models (LLMs) to process text. In new domains or languages, the inefficiency of the tokenizer will slow down the training and generation of LLM. The mismatch in vocabulary also hinders deep knowledge transfer between LLMs like token-level distillation. To mitigate this gap, we propose an efficient method named **TokAlign** to replace the vocabulary of LLM from the token co-occurrences view, and further transfer the token-level knowledge between models. It first aligns the source vocabulary to the target one by learning a one-to-one mapping matrix for token IDs. Model parameters, including embeddings, are rearranged and progressively fine-tuned for the new vocabulary. Our method significantly improves multilingual text compression rates and vocabulary initialization for LLMs, decreasing the perplexity from 2.9e^5 of strong baseline methods to 1.2e^2 after initialization. Experimental results on models across multiple parameter scales demonstrate the effectiveness and generalization of TokAlign, which costs as few as 5k steps to restore the performance of the vanilla model. After unifying vocabularies between LLMs, token-level distillation can remarkably boost (+4.4% than sentence-level distillation) the base model, costing only 235M tokens.

1 Introduction

Large language models (Touvron et al., 2023a; OpenAI, 2023; Yang et al., 2024) first tokenize text input into several tokens during inference and training, which compresses text and addresses the out-of-vocabulary problem (Sennrich et al., 2016; Wu et al., 2016; Kudo, 2018). However, the low compression rate of vanilla tokenizers on new languages or domains decelerates the training and inference process. As shown in Figure 1, the compression rate of capable large language models like LLaMA3 (Meta, 2024) on low-resource languages

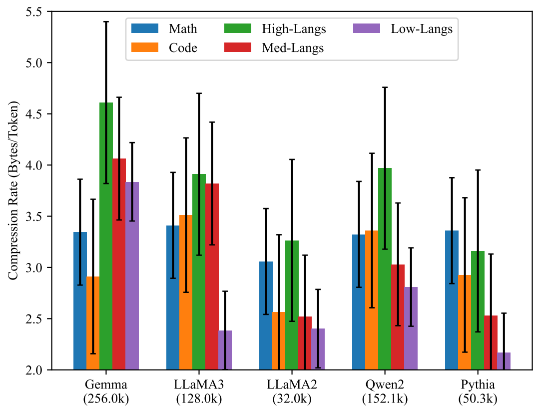


Figure 1: The compression rates of tokenizers across different domains and languages, which are still low in the code domain and low-resource languages for most of tokenizers. Refer to Table 6 in Appendix B.1 for more details.

still largely lags behind the others. For example, Armenian text is 3.95x longer in tokens than English text under the same byte size with the LLaMA3 tokenizer. On the other hand, each LLM has specific strengths and weaknesses, which arise from its pre-training corpus and method. The mismatch in the vocabulary impedes the deep knowledge transfer between them like token-level distillation and ensemble. Considering the huge cost of re-training LLM for a new tokenizer, it is important to investigate efficient vocabulary adaptation methods.

To address the problems above, we introduce a novel method called **TokAlign** for large language models from a view of token-token co-occurrences. It is motivated by the general process of training an LLM: the pre-training corpus is first tokenized into tokens, and then input into the model. Given the same pre-training corpus, different tokenizers result in various sequences of token IDs, while the semantic and syntactic information is preserved in the token-token co-occurrence. Therefore, TokAlign strives to align token IDs from the original vocabulary and the target ones based on the global token-token co-occurrence matrix (Pennington et al., 2014) and learns a token-token alignment matrix. We further propose two metrics to evaluate the performance of the token-token alignment matrix based on text matching and semantic similarity. Given the learned alignment matrix, the new target embedding and language modeling head of LLM ("lm_head" in the Transformers (Wolf, 2019)) are initialized from the parameters of the most similar source token. Further vocabulary adaptation process is divided into a progressive two-stage procedure to improve the stability of convergence.

Given a target multilingual vocabulary for substitution, the model trained on the English corpus obtains a good initialization, decreasing the perplexity from $2.9e^5$ to $1.2e^2$, and improves 29.2% compression rates across 13 languages on average. The training process of TokAlign is 1.92x faster than strong baseline methods, and does not require additional hundreds of GPU hours to train a hypernetwork for embedding initialization (Minixhofer et al., 2024). Experimental results on models across different scales show that as few as 5k steps are needed for our method to recover the performance of vanilla models on the general domain. Moreover, unifying vocabulary between models further facilitates the token-level distillation, which is 4.4% better than the sentence-level distillation on the same corpus. The performance of the 1B model is comparable with the vanilla 7B model after token-level distillation from a capable LLM. In summary, our contributions are as follows:

- We propose an unsupervised method to align token IDs between two vocabularies and replace the vocabulary of LLMs from the token-token co-occurrence view.
- We introduce two metrics to evaluate the performance of the token-level alignment matrix learned, which are proportional to the initial loss of pre-training.
- Experimental results on ten datasets show that our method promotes the cross-lingual knowledge transfer among multiple languages and deep knowledge transfer between models like token-level distillation.

2 Related Works

Our work is related to word representation, large language models, and vocabulary adaption, which will be briefly introduced below.

Word Representation Based on the distributional semantic hypothesis, Bengio et al. (2003) introduced the neural probabilistic language model to learn word representation. Researchers mainly focus on improving the effectiveness during learning word representations (Mikolov et al., 2013a,b; Bojanowski et al., 2017), which provide a good initialization for neural networks like LSTM and GRU (Hochreiter, 1997; Chung et al., 2014). GloVe (Pennington et al., 2014) provides a method to train word representations from a view of global word-word co-occurrence matrix decomposition. It motivates us to train a word representation for each token and align tokens from statistical co-occurrence information in the pre-training corpus.

Large Language Model Through scaling in the parameters and pre-training corpus (Kaplan et al., 2020; Hoffmann et al., 2022), large language models like GPT-4 and LLaMA3 (Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023a,b; Meta, 2024; GLM et al., 2024) demonstrate impressive performance across multiple tasks. However, the mismatch in the vocabulary greatly hinders the deep knowledge transfer between different models. We aim to mitigate this problem by introducing an efficient method to replace the tokenizer of a large language model.

Vocabulary Adaption is investigated mainly in the multilingual domain, especially the cross-lingual knowledge transfer problem (Scao et al., 2023; Muennighoff et al., 2023; Yang et al., 2023; Zhu et al., 2023; Üstün et al., 2024; Li et al., 2024; Liu et al., 2024; Minixhofer et al., 2024; Yamaguchi et al., 2024). It aims to improve the encoding effectiveness of tokenizer on corpora from new languages or domains, and is often implemented by extending the original vocabulary (Tran, 2020; Chau et al., 2020; Minixhofer et al., 2022; Dobler and de Melo, 2023; Downey et al., 2023). Most methods, like Focus (Dobler and de Melo, 2023), rely on the tokens belonging to both source vocabulary and target vocabulary to initialize the other new tokens in the target vocabulary. Our method differs from these studies for the whole replacement of vocabulary and does not rely on the tokens in both source vocabulary and target vocabulary.

The pipeline of TokAlign to adapt vocabulary is similar to WECHSEL(Minixhofer et al., 2022), while the main difference lies in the representation and alignment of tokens. WECHSEL requires a bilingual dictionary and word representation to

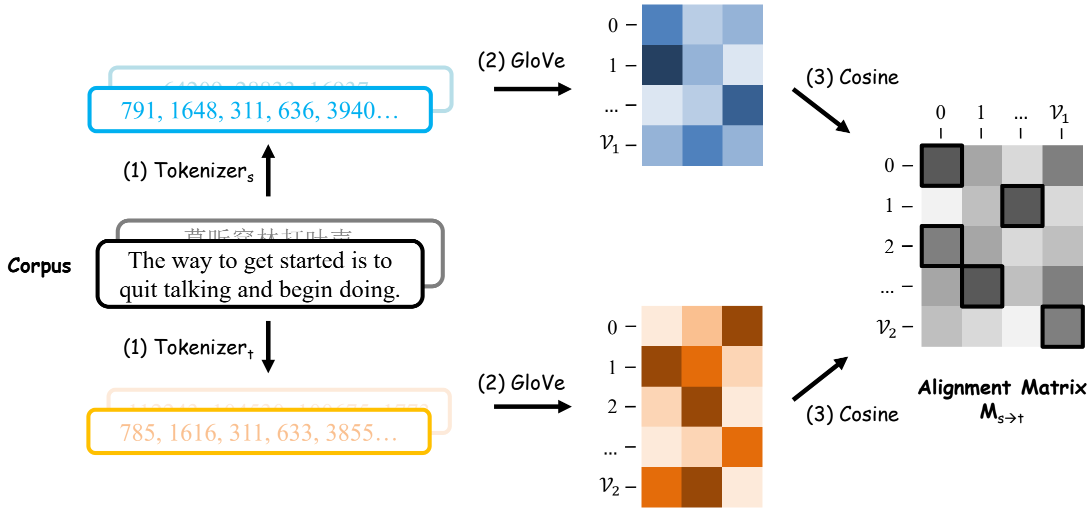


Figure 2: Illustration of TokAlign to align token IDs from different vocabularies. We train token representations on the tokenized corpus, and align token IDs by the cosine similarity. It is noted that the IDs of tokens belonging to both vocabularies are directly replaced without alignment.

align tokens and calculates the similarity between tokens by tokenizing all words in the dictionary and linearly composing word representations. In contrast, TokAlign conducts token representation learning and alignment in an unsupervised way, which can apply to languages without bilingual dictionaries.

3 Method: TokAlign

3.1 Vocabulary Alignment

As shown in Figure 2, there are three steps for TokAlign to align two vocabularies from the token-token co-occurrence information. We denote the source tokenizer as Tokenizer$_s$, which has $\mathcal{V}_s$ tokens, and the target tokenizer as Tokenizer$_t$ with $\mathcal{V}_t$ tokens, correspondingly.

Step 1: Tokenization The comprehensiveness of the pre-training corpus is important to obtain a well-trained token representation. An unbalanced corpus makes it hard to learn the representation of tokens in the tail of vocabulary. Thus, the corpus used in this work is empirically composed of multilingual corpus "CulturaX" [40%] (Nguyen et al., 2024), code corpus "The Stack" [30%] (Kocetkov et al., 2023), and math corpus "Proof-Pile-2" [30%] (Azerbayev et al., 2024). We tokenize the mixed corpus using various tokenizers and obtain multiple sequences of token IDs for the same corpus. The default amount of tokens used in this step is 1B, which is investigated in Appendix B.2.

Step 2: Token Representation Learning We adopt GloVe (Pennington et al., 2014) to learn the representation of tokens from the first step. The main reason is that GloVe considers more global statistical information than those slide window methods like CBOW and FastText (Mikolov et al., 2013a,b; Bojanowski et al., 2017). The details of training settings for GloVe vectors refer to Appendix A.

Step 3: Token Alignment Based on the assumption that token representations capture the semantic information in the token, we align token IDs using the pair-wise cosine similarity of learned token representations. It should be noted that the IDs of tokens belonging to both vocabularies are directly replaced without the need to align. $M_{s\to t}$ denotes the learned token-token alignment matrix, which records the pair-wise similarity of each source token and target token. It can serve as the one-to-one mapping function for each source/target token to find the most similar token from the target/source vocabulary.

3.2 Alignment Evaluation

Figure 3(a) illustrates our metrics to evaluate the performance of alignment matrix $M_{s\to t}$. We first tokenize the test corpus $\mathcal{C}$ using different tokenizers, which results in $\mathcal{C}_s$ and $\mathcal{C}_t$. The token ID corpus $\mathcal{C}_s$ from the source tokenizer is converted to its most similar target token ID by alignment matrix $M_{s\to t}$, and comes to the corpus $\mathcal{C}_t'$. From the view of token ID matching, the higher BLEU-1 score between $\mathcal{C}_t'$ and the corpus $\mathcal{C}_t$ from the Tokenizer$_t$, the better alignment matrix $M_{s\to t}$ is.

We further propose a semantic evaluation met-

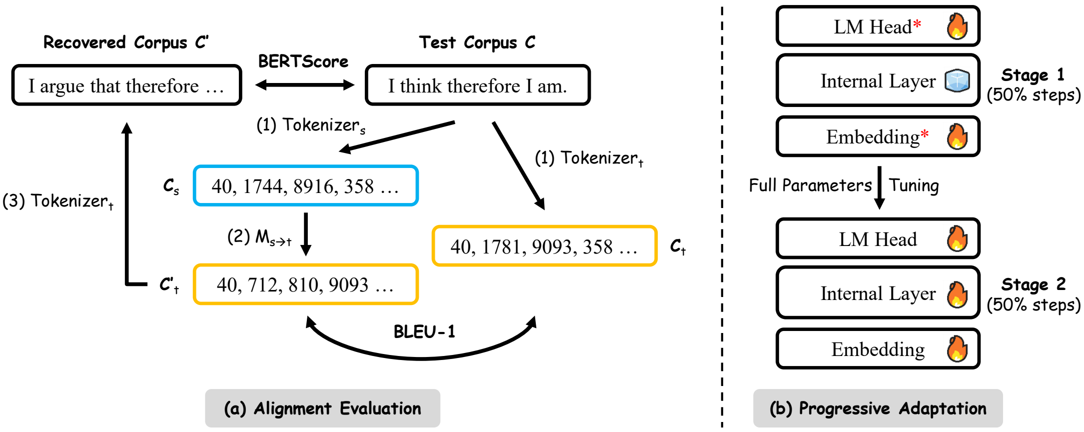


Figure 3: (a) We choose BLEU-1 and BERTScore to evaluate the performance of alignment matrix $M_{s \to t}$ (b) Embedding and lm_head are tuned at the first half part of the process, followed by full parameter tuning. * indicates the parameter of each target token is first initialized from the most similar source token by alignment matrix $M_{s \to t}$.

ric: It de-tokenizes the target token ID corpus $\mathcal{C}_t'$ using Tokenizer$_t$ into the recovered text corpus $\mathcal{C}'$, and evaluates the semantic similarity between $\mathcal{C}'$ and original corpus $\mathcal{C}$ using BERTScore. The better alignment matrix $M_{s \to t}$ learned preserves more semantics in the test corpus $\mathcal{C}$, bringing higher BERTScore of the recovered $\mathcal{C}'$ and $\mathcal{C}$.

3.3 Progressive Adaptation

Given the alignment matrix $M_{s \to t}$, the parameters of each token in the target vocabulary are initialized from the ones of the most similar source token. We find that these re-arranged embeddings and lm_head provide a good initialization for the new model (Section 4.2.1). Figure 3(b) illustrates the two-stage tuning for an LLM to adapt to the new vocabulary. The re-arranged embedding and lm_head are tuned first to avoid loss spike and improve the training stability (Figure 6). The other parameters of internal layers are further tuned together in the last half-part process.

4 Experiments

4.1 Experiments Settings

Large Language Models We adopt the fully open-source language model series Pythia (Biderman et al., 2023) as base models in this work. It is noted that we do not intend to achieve state-of-the-art large language model performance but rather investigate an efficient method to replace the English-centric tokenizer like Pythia. To transfer token-level knowledge from other capable large language models, tokenizers and vocabularies of Gemma (Team et al., 2024), Qwen2 (Yang et al., 2024), LLaMA2 (Touvron et al., 2023b), and LLaMA3 (Meta, 2024) are selected as the target to replace. We report hyper-parameters in Appendix A, and will make codes public after review to promote future research.

Corpus To reduce the risk of distribution shift from the training data, we choose the vanilla pre-training corpus Pile (Gao et al., 2020) of Pythia in the fine-tuning process. We also investigate the robustness of the corpus used in the vocabulary alignment by replacing it with Slimpajama (Soboleva et al., 2023). Corpora of downstream tasks and multiple languages are applied in cross-lingual and cross-model knowledge transfer experiments (Section 4.2.1 and 4.2.2).

Evaluation Tasks Following the common practices to evaluate large language models (Lin et al., 2022; Biderman et al., 2023; Zhang et al., 2024), there are 10 datasets, including commonsense reasoning (Clark et al., 2018; Mihaylov et al., 2018; Zellers et al., 2019; Ponti et al., 2020; Bisk et al., 2020; Sakaguchi et al., 2020) and reading comprehension (Clark et al., 2019) tasks, used in this work. To avoid the randomness from the prompt and evaluation method, we adopt the default prompt from the commonly used language model evaluation harness framework (Gao et al., 2024). Further information about the evaluation tasks is reported in Appendix D.

Baselines We introduce the following vocabulary adaptation methods as baseline methods in this work:

- **Random Initialization** for each token $t \in \{\mathcal{V}_t \setminus (\mathcal{V}_t \cap \mathcal{V}_s)\}$ employs the default initialization method of huggingface Transformers and

Model	High: ar	de	en	ja	zh	Medium: bn	ko	th	uk	vi	Low: ta	te	ur	Avg ↓
Qwen2$_{1.5B}$	4.7	11.1	15.7	6.0	4.6	2.4	3.3	2.6	5.7	3.3	2.8	3.4	4.0	5.3
Pythia$_{1B}$	7.6	15.4	**21.7**	9.9	13.2	3.4	5.6	4.3	6.7	6.3	2.9	3.3	5.8	8.2
w/ Focus Init.	$4.1e^3$	$1.7e^5$	$1.8e^6$	$2.1e^4$	$9.6e^2$	$6.5e^4$	$1.0e^3$	$5.6e^3$	$1.6e^6$	$8.4e^2$	$5.0e^4$	$1.9e^5$	$1.9e^5$	$3.1e^5$
+ LAT	8.3	27.1	59.7	14.0	14.0	3.6	5.9	3.8	7.3	5.9	3.5	3.6	4.3	12.4
w/ TokAlign Init.	$1.2e^2$	$2.2e^2$	$1.0e^2$	$3.6e^2$	$1.2e^2$	46.5	60.1	70.8	$1.5e^2$	49.2	61.0	$1.1e^2$	50.9	$1.2e^2$
+ LAT	**6.3**	**13.9**	23.6	**8.9**	**9.0**	**2.4**	**4.4**	**3.2**	**5.2**	**4.4**	**2.3**	**2.4**	**3.7**	**6.9**
Qwen2$_{7B}$	3.9	8.1	11.8	4.9	3.8	2.1	2.9	2.3	3.8	2.9	2.3	2.6	3.3	4.2
Pythia$_{6.9B}$	5.9	10.8	**16.7**	7.9	9.9	3.0	4.6	3.7	4.9	4.9	2.6	2.9	4.8	6.3
w/ Focus Init.	$6.9e^3$	$1.6e^5$	$1.2e^6$	$2.4e^4$	$1.3e^3$	$2.5e^4$	$7.2e^2$	$3.3e^3$	$1.9e^6$	$7.9e^2$	$1.7e^4$	$1.5e^5$	$1.2e^5$	$2.8e^5$
+ LAT	6.8	17.6	39.3	10.8	11.1	2.5	5.0	3.3	5.2	4.8	2.3	2.5	3.7	8.8
w/ TokAlign Init.	$1.2e^2$	$1.9e^2$	81.4	$3.7e^2$	$1.3e^2$	52.5	53.3	66.2	$1.4e^2$	49.2	46.4	92.1	48.7	$1.1e^2$
+ LAT	**5.2**	**9.9**	17.8	**7.4**	**7.9**	**2.1**	**3.8**	**2.8**	**4.0**	**3.7**	**2.1**	**2.1**	**3.1**	**5.5**
Δ Length (%) ↓	−44.5	−13.1	−0.8	−32.4	−50.0	−22.2	−52.2	−46.1	−15.5	−51.7	−20.3	−2.9	−28.5	−29.2

Table 1: The normalized perplexity on the valid corpus of CulturaX. The perplexity is normalized to the vocabulary of Pythia following Wei et al. (2023). "**High**", "**Medium**", and "**Low**" indicates the available amount of linguistic resources. "w/ xxx Init." denotes the performance of the model after initialization without any tuning steps.

Model	XNLI: en	de	zh	ar	th	vi	ur	PAWS-X: de	en	ja	ko	zh	XCOPA: th	vi	ta	XStoryCloze: en	zh	ar	te	Avg
Pythia$_{1B}$	**51.0**	37.8	42.6	35.9	34.8	37.0	34.7	49.6	49.3	54.8	**54.9**	52.9	54.0	53.2	55.4	**64.3**	48.6	48.0	52.9	48.0
w/ Focus Init.	32.8	32.2	33.6	33.6	33.5	32.0	32.8	44.8	44.9	45.7	44.8	44.7	52.4	48.6	**57.0**	45.9	47.8	**48.8**	46.5	42.2
+ LAT	46.0	35.1	34.9	32.9	32.5	35.4	34.7	50.6	45.5	55.9	53.4	**55.3**	53.8	52.6	55.4	55.8	48.8	47.6	50.4	46.1
w/ TokAlign Init.	49.9	36.6	33.2	31.8	33.2	34.4	34.4	52.4	**52.1**	**56.1**	54.7	**55.3**	53.6	48.0	55.2	61.0	47.6	47.1	51.0	46.7
+ LAT	50.9	**39.3**	**42.7**	**37.4**	**37.4**	**40.3**	**35.7**	**54.6**	50.2	55.9	**54.9**	**55.3**	**55.2**	**53.6**	53.6	64.0	**51.1**	47.8	**53.5**	**49.1**
Pythia$_{6.9B}$	54.4	**39.0**	**46.2**	39.3	39.8	39.3	36.4	43.8	40.2	50.2	54.2	50.2	**56.2**	54.4	52.2	**70.4**	53.9	**50.3**	53.8	48.6
w/ Focus Init.	31.5	31.3	33.0	32.6	33.4	32.2	32.6	44.8	42.4	52.7	45.5	44.7	52.2	48.6	**55.6**	44.5	47.1	47.8	47.1	42.1
+ LAT	52.6	34.9	36.6	35.1	33.6	39.0	34.5	**51.1**	43.8	55.9	55.3	55.4	54.2	52.4	53.8	61.0	48.7	47.7	53.7	47.3
w/ TokAlign Init.	53.3	36.3	35.0	34.6	34.6	33.0	33.8	48.8	44.6	**56.2**	55.7	55.3	54.6	52.2	54.6	66.8	48.6	47.7	50.0	47.1
+ LAT	**55.2**	35.8	43.5	**40.4**	**40.2**	**43.0**	**37.1**	43.2	**45.8**	55.8	**55.8**	**55.5**	54.6	**57.0**	54.6	70.2	**54.4**	49.3	**53.9**	**49.7**

Table 2: Zero-shot in-context learning results of cross-lingual transfer. Refer to Table 8 for few-shot results.

reuses the parameters of token $t \in \{\mathcal{V}_t \cap \mathcal{V}_s\}$, which belongs both vocabularies.

- **Random Permutation** initializes each token $t \in \{\mathcal{V}_t \setminus (\mathcal{V}_t \cap \mathcal{V}_s)\}$ using the parameter of randomly chosen token from the source vocabulary. The parameters of shared tokens are also reused.

- **WECHSEL** (Minixhofer et al., 2022) linearly transfers embeddings of source tokens into target tokens by tokenizing and recomposing additional word embeddings W^s and W^t, which are aligned with a bilingual dictionary.

- **OFA** (Liu et al., 2024) factorizes the embeddings of source model E_s into the primitive embedding P and source coordinate F_s that is further re-composed by multilingual word embedding W to the target coordinate F_t. The assembled primitive embedding P and target coordinate F_t come to the target embedding E_t.

- **Focus** (Dobler and de Melo, 2023) initializes the embedding parameters of token $t \in \{\mathcal{V}_t \setminus (\mathcal{V}_t \cap \mathcal{V}_s)\}$ using the weighted sum of the ones from the token $t \in \{\mathcal{V}_t \cap \mathcal{V}_s\}$. It largely depends on the size of $\|\mathcal{V}_t \cap \mathcal{V}_s\|$, and performs poorly when the overlapping percentage of $\mathcal{V}_t$ and $\mathcal{V}_s$ is low.

- **ZeTT** (Minixhofer et al., 2024) trains an additional hypernetwork H_θ to generate the parameters for each token $t \in \mathcal{V}_t$. The added hypernetwork brings a lot of training costs.

4.2 Main Results

We first report the final results of two applications after replacing vocabulary: cross-lingual transfer (Section 4.2.1) and cross-model knowledge transfer (Section 4.2.2), then show vocabulary adaptation results of methods (Section 4.3).

4.2.1 Cross-lingual Transfer

When applied to new domains or languages, tokenizers with higher compression rates can speed up the learning and inference of large language models. From the view of token co-occurrence, tokens from other languages can be aligned and initialized by the tokens with similar semantics in the source vocabulary, which can boost the cross-lingual knowledge transfer. Therefore, we replace

Model	ARC-E 0	ARC-E 5	BoolQ 0	BoolQ 5	HellaSwag 0	HellaSwag 5	OpenbookQA 0	OpenbookQA 5	PIQA 0	PIQA 5	WinoGrande 0	WinoGrande 5	Avg 0	Avg 5
Pythia_{1B}	56.82	58.71	60.43	57.37	37.68	37.66	18.80	19.00	70.40	71.49	53.20	52.01	49.55	49.37
+ Direct tuning	57.49	55.64	70.70	72.11	41.24	41.60	25.40	28.40	69.04	70.08	54.70	54.78	53.10	53.77
+ Sentence distill	52.27	53.41	67.49	67.06	39.03	39.08	21.80	22.80	66.97	68.99	51.85	52.17	49.90	50.58
w/ Gemma_{7B}	55.39	56.99	67.19	69.69	36.53	37.26	19.00	22.80	68.82	69.21	52.33	53.51	49.88	51.58
w/ Qwen2_{7B}	62.33	63.17	70.18	72.54	41.58	42.21	22.00	**28.20**	**73.01**	73.18	55.01	55.56	54.02	55.81
w/ LLaMA3_{8B}	**64.02**	**64.56**	**73.91**	**74.19**	**42.11**	**42.34**	**24.20**	27.60	72.74	**73.83**	**55.49**	**56.43**	**55.41**	**56.49**
$\text{Pythia}_{6.9B}$	65.99	69.23	62.84	62.02	47.56	47.64	25.00	27.00	74.65	75.41	60.46	62.43	56.08	57.29
+ Direct tuning	66.25	66.20	79.30	78.87	52.21	53.39	33.20	33.00	72.91	74.48	62.90	61.72	61.13	61.28
+ Sentence distill	61.70	65.36	76.64	76.88	48.98	51.33	28.20	30.40	70.18	71.55	58.96	62.19	57.44	59.62
w/ Gemma_{7B}	67.59	68.94	76.06	75.66	47.83	48.36	28.40	31.40	73.78	75.52	59.04	64.17	58.78	60.67
w/ Qwen2_{7B}	**71.72**	**73.27**	**79.85**	**80.00**	**50.78**	**51.12**	**29.20**	**34.00**	**77.26**	**77.91**	**61.33**	**64.56**	**61.69**	**63.48**
w/ LLaMA3_{8B}	67.05	69.78	77.83	78.78	48.83	50.15	26.00	32.00	74.21	76.22	60.22	60.93	59.02	61.31

Table 3: The main results of token-level distillation on six downstream tasks with only 235M tokens. "+Sentence distill" denotes the sentence-level distillation results with Qwen2_{7B}(Yang et al., 2024), which fine-tunes on the output from Qwen2_{7B} given questions as prompt.

the English-centric tokenizer of Pythia with the one of Qwen2 to evaluate the performance on cross-lingual transfer settings.

As shown in Table 1, the perplexity of Pythia initialized using TokAlign ($1.2e^2$) is significantly better than the one of strong baseline method Focus ($2.9e^5$). The length of tokens after text tokenization has reduced by 29.2% on average across these languages. After only 2k steps of **L**anguage **A**daptation **T**uning ("+LAT"), TokAlign improved 14.5% over the vanilla model on average, while Focus still performed worse. It is noted that the performance of Pythia using TokAlign on three low-resource languages even outperforms the ones of Qwen2 with a similar parameter amount.

Table 2 and 8 in Appendix B.5 further report zero-shot and few-shot in-context learning results on four multilingual datasets. We can find that TokAlign brings a better-initialized model than the baseline method Focus (+4.4%), and transfers the knowledge into other languages like Japanese (ja, +2.3%) and Vietnamese (vi, +2.2%).

It is interesting to find that the perplexity of Pythia_{1B} initialized by TokAlign reaches $1.2e^2$, while the in-context learning results are comparable with the ones of Focus after adapting on the multilingual corpus. We argue that it arises from the reserved English ability with TokAlign (54.2%), which significantly outperforms Focus (40.8%).

4.2.2 Cross-model Transfer

Unifying vocabulary with capable LLMs enables token-level distillation and transfers the knowledge learned into smaller models to decrease inference costs. In this section, training samples from downstream tasks and the corpus of Pile are used in the token-level distillation experiments. The logit of each token from the teacher model is taken as the soft label for Pythia to learn. We empirically set the proportion of training samples to 15% to avoid a significant degradation in the performance of language modeling (Wei et al., 2023).

Table 3 reports the results of two baseline methods and token-level distillation from three teacher models using 235M tokens. It can be found that token-level distillation is significantly better than the one of sentence-level distillation. Given the same teacher model Qwen2_{7B}, the improvement of Pythia over the sentence-level distillation result reaches 4.4%. The performance of Pythia_{1B} is even comparable with the vanilla Pythia_{7B} after token-level distillation. It is also noted that the knowledge transfer between models will be constrained in sentence-level distilling without unifying vocabulary, which further demonstrates the importance of unifying tokenizers between models.

4.3 Vocabulary Adaptation Results

We show experimental results of replacing the Pythia vocabulary (50.3k) with the Gemma vocabulary (256.0k) using all methods in Table 4. Given the same amount of tokens to fine-tune, it can be found that TokenAlign performs better than other baseline methods. The average improvement of TokenAlign over the strong baseline method ZeTT reaches 2.4%, and 97.6% performance of the vanilla model is reserved after vocabulary replacement. ZeTT requires more computation to train a hypernetwork for the parameters prediction, e.g., 661.2 GPU hours for $\text{Pythia}_{2.8B}$, while our method only costs less than two hours on a CPU server with 128 cores to train GloVe embeddings and align tokens. Replace the corpus to train the GloVe embedding with 1B SlimPajama (Soboleva et al., 2023) tokens brings comparable results (the "w/ SlimPajama" row). It demonstrates the robust-

Model	#GPU Hour	ARC-E		BoolQ		HellaSwag		OpenbookQA		PIQA		WinoGrande		Avg	
		0	5	0	5	0	5	0	5	0	5	0	5	0	5
Pythia$_{1B}$	–	56.82	58.71	60.43	57.37	37.68	37.66	18.80	19.00	70.40	71.49	53.20	52.01	49.55	49.37
w/ Rand. Init.	99.70	31.36	31.61	37.83	49.11	26.35	26.40	14.00	12.60	54.57	55.33	49.17	49.17	35.55	37.37
w/ Rand. Perm.	99.70	31.69	32.95	37.77	54.80	26.43	26.39	14.00	12.60	55.50	55.98	47.04	50.67	35.40	38.90
w/ OFA	99.70	38.17	37.79	55.14	52.35	28.29	28.62	14.40	12.20	58.43	58.54	49.96	50.99	40.73	40.08
w/ WECHSEL	99.70	43.35	45.33	56.61	54.34	32.53	32.41	14.80	16.20	61.70	62.89	52.01	52.72	43.50	43.98
w/ Focus	99.70	46.55	48.95	56.21	**55.78**	32.27	32.46	19.20	18.00	63.82	64.80	51.70	51.78	44.96	45.29
w/ ZeTT	418.94	47.14	49.03	57.06	53.70	34.06	34.06	18.40	19.40	64.15	65.34	52.09	51.22	45.48	45.46
w/ TokAlign	99.70	**54.46**	**56.86**	58.90	52.26	36.16	36.27	**21.00**	**20.20**	**67.74**	**68.50**	52.25	50.91	48.42	47.50
w/ SlimPajama	99.70	53.54	55.68	57.55	53.85	36.10	35.99	19.40	**20.20**	67.03	67.52	52.09	51.22	47.62	47.41
+ Align Rep.	99.70	54.25	56.65	**59.33**	54.68	**37.08**	**36.91**	20.20	19.40	67.36	68.17	**54.38**	**52.80**	**48.77**	**48.10**
Pythia$_{2.8B}$	–	63.80	67.00	63.91	65.14	45.32	45.04	24.00	25.20	74.05	74.43	58.64	60.77	54.95	56.26
w/ Rand. Init.	194.78	30.47	32.91	38.20	51.07	26.46	26.69	14.40	13.20	55.17	55.06	48.30	50.51	35.50	38.24
w/ Rand. Perm.	194.78	31.48	31.86	37.83	50.46	26.48	26.49	13.60	14.40	54.03	54.95	50.20	48.86	35.60	37.84
w/ OFA	194.78	50.13	54.12	60.89	61.47	36.39	36.88	18.00	19.00	65.18	64.80	54.06	54.85	47.44	48.52
w/ WECHSEL	194.78	52.48	54.92	59.42	56.76	36.79	37.30	19.20	20.80	64.04	64.25	56.43	55.72	48.06	48.29
w/ Focus	194.78	54.29	58.16	61.44	62.84	38.38	39.09	20.00	20.20	68.44	68.28	54.62	56.04	49.53	50.77
w/ ZeTT	855.96	57.15	59.42	61.68	62.05	42.17	42.25	21.80	23.60	71.11	71.16	56.59	59.19	51.75	52.95
w/ TokAlign	194.78	61.62	65.15	63.82	65.47	43.13	43.18	**23.40**	**25.80**	72.14	72.42	**58.17**	**61.17**	53.71	55.53
+ Align Rep.	194.78	**61.66**	**65.66**	**64.56**	**65.66**	**43.97**	**44.09**	22.40	25.00	**73.01**	**73.23**	58.09	60.54	**53.95**	**55.70**

Table 4: The main results of replacing the vocabulary of Pythia to Gemma. The best performance among the eight methods is displayed in **bold**. "+Align Rep." denotes the GloVe embeddings for tokens are converted into relative representations using 300 common tokens in both vocabularies before alignment following (Mosca et al., 2023).

ness of our method on the pre-training corpus for token embedding and alignment matrix. Following Moschella et al. (2023), we also evaluate the method that converts token representations into relative ones using 300 common tokens in both vocabularies as anchors before calculating the alignment matrix $M_{s \to t}$, which brings better performance.

4.4 Analysis

The loss curves of Pythia$_{2.8B}$ with different methods during the first 2.5k steps are shown in Figure 4. We find that TokAlign brings a better initialization and decreases the first-step training loss from 17.8 (Focus) to 9.5. Moreover, the training process with TokAlign is faster than other methods, which reaches 2.75 at the 1.3k step and is 1.92x (2.5/1.3) speed up than Focus.

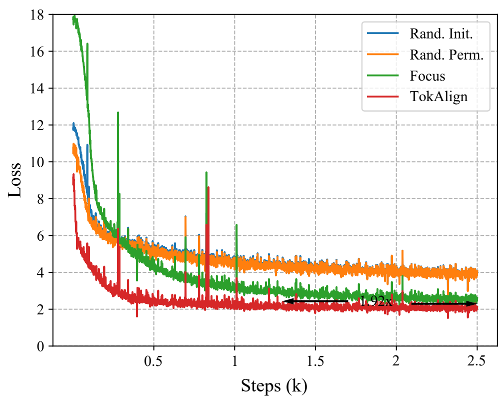
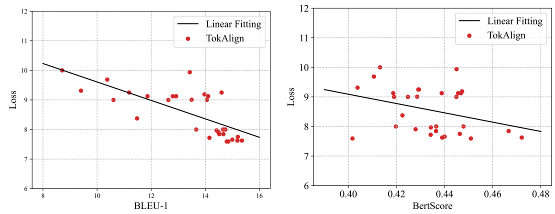


Figure 4: The training loss of Pythia$_{2.8B}$.

Better alignment brings better initialization. We further investigate the impact of the learned alignment matrix $M_{s \to t}$ by changing the hyper-parameters of GloVe. It is noted that different alignment matrices $M_{s \to t}$ bring different initial parameters, and also result in different BLEU-1 scores on the same evaluation corpus. Figure 5(a) illustrates the negative relationship between the first-step training loss and BLEU-1. The sentence embedding model named "all-mpnet-base-v2" (Song et al., 2020) is adopted in the BERTScore evaluation. As shown in Figure 5(b), it also shows a clear negative relationship with the initial training loss. In other words, the higher the BLEU-1 score or BERTScore for the alignment matrix $M_{s \to t}$, the better the initial parameter is.



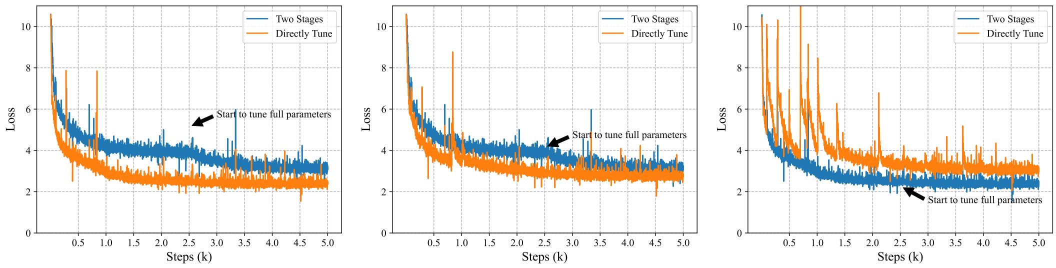
(a) BLEU-1($\mathcal{C}_t, \mathcal{C}_t'$) (b) BERTScore($\mathcal{C}, \mathcal{C}'$)

Figure 5: The relationship between initial training loss and BLEU-1 (a) or BERTScore (b) for Pythia$_{1B}$.

More overlapping comes to faster convergence and higher performance. TokAlign is further applied to the other three target tokenizers: Qwen2, LLaMA2, and LLaMA3. Table 5 reports the performance of models after replacing vocabulary on six datasets. TokAlign recovers 98.0% performance of the base model on average with only 5k steps. Given a target vocabulary with more tokens than the one of Pythia (50.3k), it can be found that a higher overlapping ratio brings a better performance of model replaced (97.6% for Gemma to

Model	$\#\mathcal{V}$ (k)	ARC-E 0	ARC-E 5	BoolQ 0	BoolQ 5	HellaSwag 0	HellaSwag 5	OpenbookQA 0	OpenbookQA 5	PIQA 0	PIQA 5	WinoGrande 0	WinoGrande 5	Avg 0	Avg 5
Pythia$_{1B}$	50.3	56.82	58.71	60.43	57.37	37.68	37.66	18.80	19.00	70.40	71.49	53.20	52.01	49.55	49.37
→ Gemma	256.0	54.46	56.86	**58.90**	52.26	36.16	36.27	**21.00**	20.20	67.74	68.50	52.25	50.91	48.42	47.50
→ Qwen2	152.1	54.46	57.07	54.80	49.79	37.18	37.04	19.20	18.40	68.44	**70.24**	53.35	52.80	47.91	47.56
→ LLaMA2	32.0	49.45	52.02	58.32	55.75	35.38	35.45	18.80	17.80	66.32	66.65	53.91	50.91	47.03	46.43
→ LLaMA3	128.0	**54.63**	**57.28**	55.84	**53.70**	**37.34**	**37.43**	20.20	**20.40**	**69.04**	70.18	**54.46**	**53.43**	**48.59**	**48.74**
Pythia$_{2.8B}$	50.3	63.80	67.00	63.91	65.14	45.32	45.04	24.00	25.20	74.05	74.43	58.64	60.77	54.95	56.26
→ Gemma	256.0	61.62	65.15	63.82	**65.47**	43.13	43.18	23.40	**25.80**	72.14	72.42	58.17	**61.17**	53.71	**55.53**
→ Qwen2	152.1	**62.54**	**66.04**	62.35	63.55	44.46	44.39	23.20	24.60	**73.50**	**73.56**	**59.04**	59.59	54.18	55.29
→ LLaMA3	128.0	61.83	64.60	**64.40**	63.94	**44.62**	**44.59**	**23.80**	25.60	73.45	73.29	57.54	58.72	**54.27**	55.12
Pythia$_{6.9B}$	50.3	65.99	69.23	62.84	62.02	47.56	47.64	25.00	27.00	74.65	75.41	60.46	62.43	56.08	57.29
→ Gemma	256.0	65.40	68.35	62.39	59.57	45.75	45.86	22.00	25.60	73.39	74.10	60.38	61.17	54.89	55.77
→ Qwen2	152.1	65.57	**68.43**	**64.07**	57.61	46.84	46.91	**25.60**	25.40	73.45	74.65	61.17	63.14	56.12	56.02
→ LLaMA3	128.0	**66.46**	68.35	63.79	**60.64**	**47.28**	**47.31**	**25.60**	**28.20**	**74.48**	**75.84**	**61.48**	**63.30**	**56.52**	**57.27**

Table 5: The benchmark results of replacing different tokenizers using TokAlign. The overlapping ratio between the vocabulary of Pythia and other models are 6.23% (Gemma), 26.92% (Qwen2), 28.10% (LLaMA2), 32.85% (LLaMA3).



(a) Learning rate 2e^{-5} (b) Learning rate 8e^{-5} (c) Learning rate 6.4e^{-4}

Figure 6: The loss curve of Pythia$_{1B}$ under two-stage tuning or direct full parameters tuning.

99.1% for LLaMA3). The zero-shot in-context learning results for Pythia$_{6.9B}$ with LLaMA3 vocabulary even surpass the vanilla base model. The results of Pythia$_{1B}$ with LLaMA2 vocabulary are only 94.5%, which is inferior to the average result. We argue that it may come from the missing 75.0M parameters (7.4% for Pythia$_{1B}$) after switching to a 32.0k vocabulary from the 50.3k vocabulary.

Figure 8 in Appendix B.3 shows the training loss curve. The replacing process of the Gemma tokenizer is the slowest, which may come from the only 6.23% overlapping ratio between two vocabularies. It is in line with the result of random initialization in Figure 10. Appendix B.3 reports more quantitative results by shuffling the alignment matrix, which further demonstrates the importance of token alignment.

Two-stage tuning brings a more stable convergence. To replace the tokenizer and keep the performance of the vanilla model, we only fine-tune the vocabulary-related parameters at the first stage. The main reason for two-stage tuning is to take these parameters as the adapters of different tokenizers and avoid the well-trained parameters of the internal layer being distracted by the new initialized parameters.

Figure 6 illustrates that our two-stage tuning method makes the convergence more stable under a high learning rate like 6.4e^{-4}, which comes to better performance after vocabulary adaptation. It is noted that the loss spike also occurs at the first stage, fine-tuning vocabulary-related parameters only, under such a high learning rate like 2.56e^{-3} in Figure 9.

5 Conclusion and Future Work

In this paper, we introduce a method named TokAlign to replace the tokenizer of large language models from a token-token co-occurrence view. Extensive experiments demonstrate that TokAlign restores the performance of vanilla models after vocabulary adaptation, which enables cross-lingual knowledge transfer and deep knowledge transfer between models like token-level distillation.

Beyond replacing the vocabulary of large language models, our method can be extended to replace the vocabulary of multi-modal models by aligning different modal tokens. The other direction is to develop a faster method, e.g., incorporating meta-learning in the two-stage tuning method to speed up the convergence.

Limitations

The first limitation comes from the assumption that the pre-training data distribution is available. We conduct experiments on Pythia with different parameter amounts, which provide public model weights and pre-training corpus. Due to the limited computation resource budget, open-source language models with unknown pre-training corpus like Mistral (Jiang et al., 2023) are not investigated in this work. However, the pre-training corpus distribution of open-weighted large language models can be roughly inferred by the BPE vocabulary (Hayase et al., 2024). It can re-construct a similar pre-training corpus to conduct replacing tokenizer experiments.

Another limitation is the additional 5k steps for vocabulary adaptation to replace a tokenizer. From the loss curve of TokAlign (Figure 8), we find that the start of full parameters tuning can be faster, which may result in a better balance between performance and computational budget. Appendix B.4 reports a preliminary result with only 2k steps, where TokAlign also shows a promising result.

References

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7432–7439.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

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

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.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

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.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359. Curran Associates, Inc.

Konstantin Dobler and Gerard de Melo. 2023. FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13440–13454, Singapore. Association for Computational Linguistics.

C.m. Downey, Terra Blevins, Nora Goldfine, and Shane Steinert-Threlkeld. 2023. Embedding structure matters: Comparing methods to adapt multilingual vocabularies to new languages. In *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL)*, pages 268–281, Singapore. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Jonathan Hayase, Alisa Liu, Yejin Choi, Sewoong Oh, and Noah A Smith. 2024. Data mixture inference: What do bpe tokenizers reveal about their training data? *arXiv preprint arXiv:2407.16607*.

S Hochreiter. 1997. Long short-term memory. *Neural Computation MIT-Press*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Denis Kocetkov, Raymond Li, Loubna Ben allal, Jia LI, Chenghao Mou, Yacine Jernite, Margaret Mitchell, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro Von Werra, and Harm de Vries. 2023. The stack: 3 TB of permissively licensed source code. *Transactions on Machine Learning Research*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 318–327, Singapore. Association for Computational Linguistics.

Chong Li, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2024. Improving in-context learning of multilingual generative language models with cross-lingual alignment. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8058–8076, Mexico City, Mexico. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yihong Liu, Peiqin Lin, Mingyang Wang, and Hinrich Schuetze. 2024. OFA: A framework of initializing unseen subword embeddings for efficient large-scale multilingual continued pretraining. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1067–1097, Mexico City, Mexico. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. *Qwen blog*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2018. Mixed precision training. In *International Conference on Learning Representations*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.

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.

Benjamin Minixhofer, Edoardo Maria Ponti, and Ivan Vulić. 2024. Zero-shot tokenizer transfer. *Preprint*, arXiv:2405.07883.

Edoardo Mosca, Mohamed Hesham Ibrahim Abdalla, Paolo Basso, Margherita Musumeci, and Georg Groh. 2023. Distinguishing fact from fiction: A benchmark dataset for identifying machine-generated scientific papers in the LLM era. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 190–207, Toronto, Canada. Association for Computational Linguistics.

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

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2024. CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4226–4237, Torino, Italia. ELRA and ICCL.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *arXiv preprint arXiv:2309.09400*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI blog*.

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

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740.

BigScience Workshop: Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, and Alexandra Sasha Luccioni et al. 2023. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Alexey Tikhonov and Max Ryabinin. 2021. It's All in the Heads: Using Attention Heads as a Baseline for Cross-Lingual Transfer in Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3534–3546, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ke Tran. 2020. From english to foreign languages: Transferring pre-trained language models. *arXiv preprint arXiv:2002.07306*.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939, Bangkok, Thailand. Association for Computational Linguistics.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, et al. 2023. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*.

T Wolf. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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. *Preprint*, arXiv:1609.08144.

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. 2024. An empirical study on cross-lingual vocabulary adaptation for efficient language model inference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6760–6785, Miami, Florida, USA. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Wen Yang, Chong Li, Jiajun Zhang, and Chengqing Zong. 2023. Bigtranslate: Augmenting large language models with multilingual translation capability over 100 languages. *arXiv preprint arXiv:2305.18098*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. 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 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Extrapolating large language models to non-english by aligning languages. *arXiv preprint arXiv:2308.04948*.

A Hyper-parameters

GloVe Training We empirically train GloVe vectors with 1B tokens, which covers most tokens from Gemma (95.10%), Qwen2 (93.40%), LLaMA2 (99.35%), and LLaMA3 (98.04%). The dimension size is set to 300. The max training iteration and the size of the slide window are 15.

Model Tuning The optimizer adopted in this work is AdamW (Loshchilov and Hutter, 2019), where $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate for baseline methods is set to 5e-5 to reduce the loss spike in Figure 6(b) and Figure 6(c). We adopt bf16 mixed precision training, ZeRO-1, and flash-attention to save GPU memory cost and speed up the training process (Micikevicius et al., 2018; Rasley et al., 2020; Dao et al., 2022). Following Biderman et al. (2023), the batch size is set to 2M tokens and the max sequence length is 2048.

B Additional Results

B.1 Tokenizer Compression Rate

Table 6 reports detailed compression rates of tokenizers across different domains and languages. We randomly sample 10 subsets or languages from vanilla datasets (Azerbayev et al., 2024; Kocetkov et al., 2023) to estimate the compression rate. Following Lai et al. (2023), the division of languages between "**High**", "**Medium**" and "**Low**" is determined by the available amount resource on CommonCrawl.

B.2 GloVe Vectors

We show the effects of different token amounts for the GloVe vectors training in Figure 7. It can be found that 1B tokens used in this work provide a high vocabulary coverage (>90%) and better initialization for Pythia$_{1B}$. Due to the limited computation budget, experiments with more than 1B tokens are not conducted.

B.3 Convergence Analysis

To investigate the effect of overlapping rate between two tokenizers to the convergence of training, we plot Figure 10 for the random initialization baseline method. The convergence of Gemma tokenizer is slower than the other tokenizers and comes to worse results, which are similar to the case in Figure 8.

Moreover, we randomly shuffle the alignment matrix learned in TokAlign to imitate the case that other worse methods rather than cosine similarity to calculate the alignment matrix. Figure 11 shows that the higher percentage of randomly shuffle comes to higher initial training loss and slower convergence.

B.4 Fast Vocabulary Adaptation Results

We further investigate a challenge condition that fine-tunes only 2B tokens to adapt the target vocabulary. To meet the requirement, we reduce the batch size to 1M tokens and set the number of fine-tuning steps to 2k. Table 7 shows the results of adapting to the other 3 tokenizers using TokAlign. It can be found that 95.66% performance of the vanilla model is recovered on average, which further demonstrates the effectiveness of our method.

B.5 In-context Learning Results during Cross-lingual Transfer

Table 2 and 8 report the 0-shot and 5-shot in-context learning results on 4 multilingual datasets. The average improvement over the baseline method Focus is 2.35% after language adaptation pre-training. We can find that the model initialized by TokAlign is comparable to the one of Focus after language adaptation pre-training, which mainly comes from the strong English performance preserved by TokAlign.

Case study of multilingual token alignment. Table 9 provides nine new tokens from three languages with their top 3 tokens in the source vocabulary. In most cases, a clear semantic relationship between two aligned tokens cannot be found. We argue that it may come from the following two reasons:

Domain	Subset / Language	Tokenizer				
		Gemma	LLaMA3	LLaMA2	Qwen2	Pythia
Math (Azerbayev et al., 2024)	*ArXiv*	2.8561	2.7765	2.7040	2.7445	2.8489
	Textbooks	4.0883	4.3270	3.6500	4.2899	3.9464
	Wikipedia	3.1753	3.2049	2.8792	3.0312	3.2898
	ProofWiki	2.7538	2.8115	2.5996	2.7900	2.7363
	StackExchange	3.2062	3.2814	3.0094	3.2107	3.2222
	WebPages	3.9885	4.0655	3.5070	3.8720	4.1136
Code (Kocetkov et al., 2023)	*Python*	3.3401	4.1331	3.0072	4.0339	3.2328
	Java	3.7175	4.4900	3.2193	4.4141	3.4914
	Go	2.9274	3.4797	2.5189	3.3870	2.8542
	VHDL	2.1038	2.4814	1.8724	2.2961	2.1395
	ActionScript	3.3470	3.9717	2.7852	3.9180	3.2949
	Scheme	2.7178	3.3045	2.4586	2.9713	2.9326
	Haml	3.2423	3.8429	2.9588	3.8002	3.1016
	Xbase	2.8739	3.4325	2.3300	3.3475	2.7837
	Mako	3.4387	4.0746	3.1238	4.0311	3.2844
	EmberScript	1.4104	1.9017	1.3819	1.4082	2.1540
High-Langs (Nguyen et al., 2023)	*English*	4.4971	4.6042	3.8647	4.4875	4.4505
	Russian	6.7529	5.8131	4.9275	5.3559	3.5802
	Spanish	4.6068	3.8416	3.4517	3.8330	3.3655
	German	4.4605	3.6314	3.4417	3.6041	3.1096
	French	4.2258	3.7378	3.4445	3.7243	3.3565
	Chinese	3.7378	3.2373	1.8434	3.9859	1.9896
	Italian	4.2211	3.4952	3.3320	3.4573	3.1928
	Portuguese	4.2731	3.6030	3.2031	3.5850	3.2022
	Polish	3.5583	2.8548	2.6639	2.9464	2.4333
	Japanese	5.7640	4.2796	2.4701	4.7059	2.9326
Medium-Langs (Nguyen et al., 2023)	*Czech*	3.3402	3.2875	2.5978	2.4490	2.3884
	Vietnamese	4.5376	4.2766	1.9699	4.2877	2.0382
	Persian	5.6465	5.3015	1.7938	3.1923	2.3707
	Hungarian	3.2337	2.6008	2.6311	2.5500	2.3878
	Greek	4.4691	4.5671	1.8544	2.1225	3.0283
	Romanian	3.5558	3.0566	2.8355	3.0083	2.8981
	Swedish	3.7087	3.1398	2.9214	3.0977	2.9620
	Ukrainian	5.5141	5.5985	4.5904	3.6179	3.0702
	Finnish	3.2659	2.6748	2.4176	2.6473	2.6112
	Korean	3.3556	3.6957	1.5977	3.3330	1.5667
Low-Langs (Nguyen et al., 2023)	*Hebrew*	4.0487	1.8592	1.7875	4.3773	2.0380
	Serbian	4.8596	3.9234	4.2642	3.6267	2.9896
	Tamil	5.6161	2.0279	2.2615	2.4759	1.9765
	Albanian	2.8919	2.6536	2.2945	2.6037	2.3631
	Azerbaijani	2.8585	2.4857	2.0407	2.3797	2.1534
	Kazakh	3.8172	2.9176	3.0869	2.9263	2.3236
	Urdu	4.4364	2.8462	1.7260	2.7174	1.9458
	Georgian	3.8237	1.4828	2.5595	2.6951	2.2077
	Armenian	3.2133	1.1658	1.7000	1.8531	1.3922
	Icelandic	2.7964	2.4860	2.3050	2.4330	2.3185

Table 6: The compression rates (bytes/token) of different tokenizers.

Model	$\#\mathcal{V}$ (k)	ARC-E 0	ARC-E 5	BoolQ 0	BoolQ 5	HellaSwag 0	HellaSwag 5	OpenbookQA 0	OpenbookQA 5	PIQA 0	PIQA 5	WinoGrande 0	WinoGrande 5	Avg 0	Avg 5
Pythia_{1B}	50.3	56.82	58.71	60.43	57.37	37.68	37.66	18.80	19.00	70.40	71.49	53.20	52.01	49.55	49.37
→ Gemma	256.0	51.09	52.44	53.12	52.35	35.00	35.05	20.20	18.60	64.80	65.83	53.12	51.62	46.22	45.98
→ Qwen2	152.1	53.41	55.47	53.52	55.81	36.12	36.38	20.80	18.00	68.50	68.88	54.38	52.80	47.79	47.89
→ LLaMA3	128.0	51.73	55.09	59.05	55.08	36.42	36.52	19.40	19.60	67.68	68.34	53.43	53.75	47.95	48.06

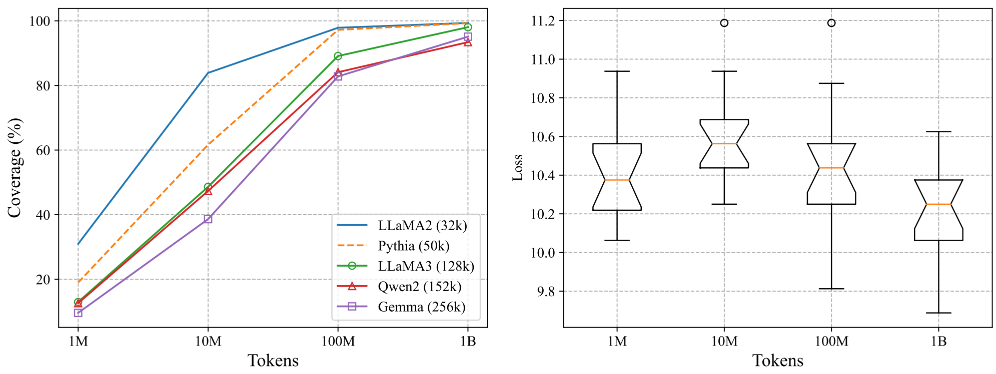
Table 7: The main results of replacing the vocabulary of Pythia for TokAlign using 2B tokens from the Pile corpus.

	XNLI							PAWS-X					XCOPA			XStoryCloze				
Model	**en**	**de**	**zh**	**ar**	**th**	**vi**	**ur**	**de**	**en**	**ja**	**ko**	**zh**	**th**	**vi**	**ta**	**en**	**zh**	**ar**	**te**	**Avg**
Pythia_{1B}	46.2	38.6	**38.9**	**36.9**	35.2	**38.9**	34.9	48.9	48.3	52.9	**53.3**	54.1	53.4	52.6	55.4	**65.3**	48.6	48.2	52.2	**47.5**
w/ Focus Init.	32.8	32.2	33.6	33.6	33.5	32.0	32.8	44.8	46.0	48.9	44.8	44.7	51.4	47.6	**55.6**	45.9	48.6	**48.5**	46.8	42.3
+ LAT	**47.0**	36.7	35.4	34.3	33.5	35.1	33.9	51.5	48.6	53.7	51.2	54.0	**54.4**	51.6	**55.6**	55.8	48.7	47.5	50.4	46.3
w/ TokAlign Init.	44.9	37.4	34.0	32.8	**35.3**	35.2	34.5	50.2	**50.3**	52.0	53.1	**54.4**	**54.4**	50.0	54.4	61.2	48.3	47.6	50.0	46.3
+ LAT	44.4	**39.0**	38.7	35.6	35.1	37.8	**35.5**	**51.9**	49.3	**54.7**	53.1	50.6	54.2	**54.0**	52.8	64.7	**50.8**	48.0	**52.4**	**47.5**
$\text{Pythia}_{6.9B}$	**53.0**	40.7	**41.7**	**38.9**	37.3	41.3	35.1	49.4	47.1	52.9	52.2	52.4	**55.0**	53.6	53.6	**73.1**	**54.6**	**49.9**	**53.9**	49.2
w/ Focus Init.	31.5	31.3	33.0	32.6	33.4	32.2	32.6	44.8	46.4	52.3	51.2	54.5	52.4	47.4	**56.0**	44.9	47.3	48.5	47.6	43.1
+ LAT	45.1	37.7	35.3	33.4	35.0	38.1	33.8	49.5	49.0	52.6	54.5	55.3	52.0	51.2	53.8	61.5	48.3	47.3	53.4	46.7
w/ TokAlign Init.	50.8	39.1	34.4	34.5	33.9	34.6	**35.2**	50.0	47.7	**53.9**	54.3	55.2	53.2	51.2	53.2	68.0	48.5	47.8	50.2	47.1
+ LAT	49.2	**41.5**	37.8	36.9	**38.7**	**41.9**	34.7	**51.2**	**49.5**	53.5	**54.8**	**55.4**	53.4	**59.8**	52.8	73.0	53.9	49.2	53.6	**49.5**

Table 8: Five-shot in-context learning results of cross-lingual transfer.

	French			Chinese			Korean		
Top-3	**dire(speak)**	**aller(go)**	**oui(are)**	**吃(eat)**	**科学(science)**	**智能(intelligence)**	**능(competence)**	**집(house)**	**왜(why)**
					Qwen2 (Target Tokenizer)				
1	ada	Ġsta	Ġsalv	allel	Ġantagon	_{[	Si	ĠBart	bst
2	ays	ĠÃ¨	Ġvas	Ġindicator	Ġign	liquid	uria	ĠPAT	rains
3	Ġ-	Ġdetermin	Ġexplos	Ġbasic	Ġcritic	Layer	ost	ĠEdgar	irc
					Gemma (Target Tokenizer)				
1	Ġj	Cor	Tools	kernel	ĠLed	Ġcommittee	Ġmang	Ġcru	Ġcholesterol
2	Ġdar	Ġequality	directed	sentence	COUNT	ĠUND	ial	Ġcal	Ġmolecule
3	ba	Lex	afx	messages	Ġglycine	Ġfactors	Ġrebut	Ġmalt	apor

Table 9: The case study of new tokens from other languages in the target vocabulary with top-3 source tokens aligned. The language family of French, Chinese, and Korean are Indo-European, Sino-Tibetan, and Koreanic, respectively.



(a) Vocabulary coverage

(b) Initial loss with Gemma tokenizer

Figure 7: The average vocabulary coverage (a) and initial training loss of Pythia$_{1B}$ (b) under different amount tokens to train the GloVe vector.

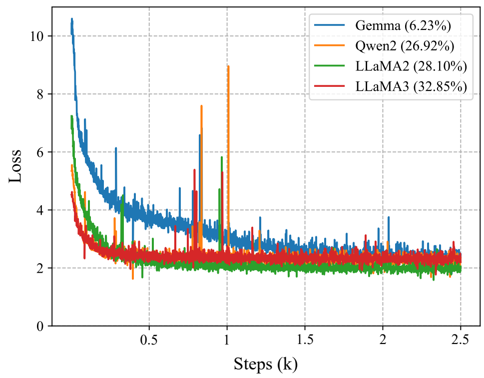


Figure 8: The training loss curve of Pythia$_{1B}$ for different overlapping ratios.

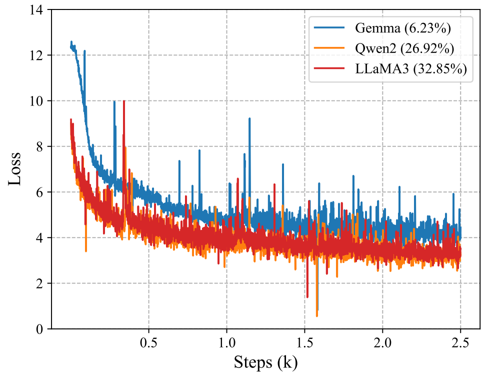


Figure 10: The training loss to different tokenizers using random initialization baseline.

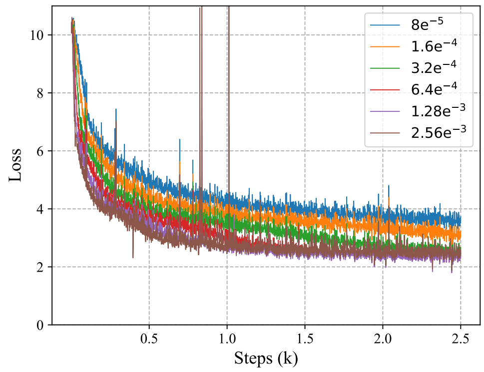


Figure 9: The training loss curve of Pythia$_{1B}$ for learning rate used during replacing to the Gemma tokenizer.

- BPE algorithm (Sennrich et al., 2016) divides words into the sub-word units, also called tokens, from the statistical co-occurrence information. There may be less superficial semantic information in the tokens divided compared with words in the natural language.

- The GloVe vector for each token is obtained from the token-token co-occurrence information. These aligned tokens often appear together, e.g., 科学(science) and “Ġcritic”, 왜(why) and “rains”.

Therefore, it is better to choose a matric to quantify the performance of the alignment matrix learned, for example, the BLEU-1 score or BERTScore in Section 3.2.

C Language Codes

We provide details of languages involved in Table 10. Following Lai et al. (2023), languages are divided by the data ratios in CommomCrawl: High (>1%), Medium (>0.1%), and Low (>0.01%).

D Evaluation Tasks

We report the statistics of evaluation tasks used in Table 11. Here are the descriptions of these evaluation tasks:

Natural Language Inference aims to determine the semantic relationship (Entailment, neural, or contradiction) between the premise and hypothesis (Conneau et al., 2018).

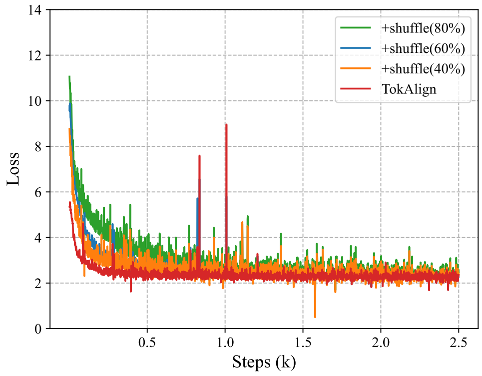


Figure 11: The training loss of Pythia$_{1B}$ when replacing tokenizer to Qwen2 under different percentages of shuffling.

ISO 639-1	Language	Family
AR	Arabic	Afro-Asiatic
BN	Bengali	Indo-European
DE	German	Indo-European
EN	English	Indo-European
JA	Japanese	Japonic
KO	Korean	Koreanic
TA	Tamil	Dravidian
TE	Telugu	Dravidian
TH	Thai	Kra-Dai
UR	Urdu	Indo-European
VI	Vietnamese	Austroasiatic
ZH	Chinese	Sino-Tibetan

Table 10: Details of language codes in this work.

Paraphrase Detection requires the model to evaluate whether the second sentence is a paraphrase of the first sentence in this task (Yang et al., 2019).

Commonsense Reasoning is a task for the model to reason the gold answer based on the semantic coherence and physic rules (Clark et al., 2018; Mihaylov et al., 2018; Zellers et al., 2019; Ponti et al., 2020; Bisk et al., 2020; Sakaguchi et al., 2020; Tikhonov and Ryabinin, 2021).

Reading Comprehension needs the model to infer whether the given passage can answer the query (Clark et al., 2019).

E Licenses of Scientific Artifacts

We follow and report the licenses of scientific artifacts involved in Table 12.

Task	Dataset	#Lang	Data Curation	#Train	#Dev	#Test
Natural Language Inference	XNLI	15	Translation	–	2,490	5,010
Paraphrase Detection	PAWS-X	7	Aligned	–	2,000	2,000
Reasoning	ARC-Easy	1	–	2,251	570	2,376
	HellaSwag	1	–	39,905	10,042	10,003
	OpenbookQA	1	–	4,957	500	500
	PIQA	1	–	16,000	2,000	3,000
	XCOPA	12	Translation	33,810	100	500
	XStoryCloze	11	Translation	361	–	1,511
	WinoGrad	1	–	40,398	1,267	1,767
Reading Comprehension	BoolQ	1	–	9,427	3,270	–

Table 11: Statistic of evaluation datasets used.

Name	License
Transformers	Apache 2.0 license
lm-evaluation-harness	MIT license
matplotlib	PSF license
Focus	MIT license
WECHSEL	MIT license
Pythia	Apache 2.0 license
LLaMA3	Meta LLaMA 3 community license
Qwen2	Tongyi Qianwen license
Gemma	Gemma license
The Pile	MIT license

Table 12: Licenses of scientific artifacts involved in this work.