Self-Distillation for Model Stacking Unlocks Cross-Lingual NLU in 200+ Languages

Anonymous ACL submission

Abstract

LLMs have become a go-to solution not just for text generation, but also for natural language understanding (NLU) tasks. Acquiring extensive knowledge through language modeling on web-scale corpora, they excel on English NLU, yet struggle to extend their NLU capabilities to underrepresented languages. In contrast, machine translation models (MT) produce excellent multilingual representations, resulting in strong translation performance even for low-resource languages. MT encoders, however, lack the knowledge necessary for comprehensive NLU that LLMs obtain through language modeling training on immense corpora. In this work, we get the best both worlds by integrating MT encoders directly into LLM backbones via sample-efficient self-distillation. The resulting MT-LLMs preserve the inherent multilingual representational alignment from the MT encoder, allowing lower-resource languages to tap into the rich knowledge embedded in English-centric LLMs. Merging the MT encoder and LLM in a single model, we mitigate the propagation of translation errors and inference overhead of MT decoding inherent to discrete translation-based cross-lingual transfer (e.g., translate-test). Evaluation spanning three prominent NLU tasks and 127 predominantly low-resource languages renders MT-LLMs highly effective in cross-lingual transfer. MT-LLMs substantially and consistently outperform translation-test based on the same MT model, showing that we truly unlock multilingual language understanding for LLMs.

1 Introduction

Large Language Models (LLMs) have become the swiss-army knife for natural language understanding (NLU) in English. When pretrained with language modelling on trillions of tokens, LLMs excel at complex NLU tasks with minimal or no labeled data (Brown et al., 2020; Touvron et al., 2023; AI@Meta, 2024). Although these models are predominantly trained on English texts, typically comprising more than 80% of their training data (Touvron et al., 2023; AI@Meta, 2024; Aryabumi et al., 2024), they show strong NLU capabilities also in other high-resource languages (Blevins and Zettlemoyer, 2022; Zhu et al., 2023).[1] However, LLM performance degrades in cross-lingual transfer to languages that are typologically distant from English or virtually *unseen* at pretraining (Ojo et al., 2024; Holtermann et al., 2024; Razumovskaia et al., 2024). This performance degradation restricts the effectiveness of LLMs primarily to English and a tiny subset of high-resource languages and underscores shortcomings in their adaptability to underrepresented low-resource languages, thereby amplifying the cross-lingual language technology gap (Joshi et al., 2020).

In contrast, publicly available *machine translation* models like NLLB (Team et al., 2022) and MADLAD-400 (Kudugunta et al., 2023) are by design oriented towards and showcase ever more inclusiveness; they provide some machine translation capabilities between more than 200 and 400 languages, respectively, in any language direction. Unlike LLMs, machine translation (MT) models, and specifically MT encoders, are designed to semantically align textual representations in a unified embedding space, as demonstrated by their sentence retrieval performance on the FLORES200 dataset (cf. Figure 1).

However, the MT models lack various types of knowledge (e.g., world knowledge, commonsense knowledge), commonly acquired through large-scale language modeling pre-training. Consequently, in multilingual NLU tasks and respective cross-lingual transfer they underperform even smaller multilingual encoders like XLM-R (Conneau et al., 2020) (see Appendix A.2 for an em-

[1] For instance, 5% of the 15T pretraining dataset of Llama 3 comprise non-English data spanning over 30 languages. Aya is tailored for NLU across 23 high-resource languages.

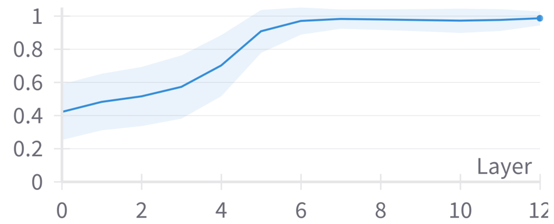

Figure 1: Mean & std. dev. of sentence translation retrieval accuracy with `BERTScore` for NLLB encoder outputs on pooled `dev` & `devtest` sets of Flores200 by layer (Team et al., 2022).

pirical comparison). Because of this, MT models are typically used fully downstream, to translate training and/or test data from the source to the target languages, extending the wide availability of task-annotated English corpora to the target languages (Ruder et al., 2021; Artetxe et al., 2023; Ebing and Glavaš, 2023). Translating training data (TTrain) involves substantial computational resources but yields strong XLT performance. TTrain nevertheless requires LLMs to support the target languages, which does not hold true for low-resource languages (Ojo et al., 2024). Translating test data, on the other hand, enables ZS-XLT with monolingual LMs, but it incurs an additional inference overhead from MT and generally offers performance that is slightly inferior to TTrain. Both TTrain and TTest aim to align the input to accommodate the shortcomings of the LLM representation space, resorting for this to discrete natural language translations coming from the MT decoder. These methods fail to preserve the rich latent representations from the MT encoder and propagate translation errors to LLMs, thereby reducing downstream performance (Ponti et al., 2021).

In this work, we thus propose to merge MT encoders directly with LLMs, creating a unified multilingual LLM for enhanced cross-lingual NLU, termed MT-LLM. The merger of the two models unlocks the potential to combine **1)** the general knowledge available in the original LLM for English and a handful of high-resource languages and **2)** powerful multilingual representations and their cross-lingual semantic alignment available in the MT encoder (see Figure 1). The key idea involves enabling the LLM to directly integrate the output representations from MT encoders, this way extending its NLU performance to virtually all languages supported by the MT encoder.

We align MT encoders with LLMs via self-distillation in two steps. The objective in the first, self-supervised adaptation step is sequence-level alignment between the original LLM and the MT-equipped LLM (MT-LLM). Second, we then address the distributional shifts inherent to adaptation from general-purpose data to downstream task data through task-specific self-distillation. We fine-tune the LLM on labeled task data, then transfer this task knowledge to the MT-LLM by aligning the task-specific output representations.

Contributions. 1) To the best of our knowledge, we are the first to successfully integrate MT encoders into language model backbones for XLT, thereby enabling ZS-XLT to all languages supported by the MT encoder. This integration yields two key benefits: ZS-XLT performance consistently improves over TTest, while simultaneously reducing inference cost by eliminating the need to translate test instances. In turn, we show that the integration is highly efficient and only requires a few self-supervised adaptation steps to yield performance improvements over the LLM backbone. **2)** We empirically show that our approach is agnostic to different types of LLM backbones, i.e., it improves the ZS-XLT capabilities of both decoder-only or encoder-only models. **3)** We compare ZS-XLT and TTest extensively and *fairly* across all supported language on a range of tasks and a wide spectrum of languages.[2] Unlike existing work, we make sure that both cross-lingual transfer approaches—latent with MT-LLM and discrete with TTest—are evaluated on an equal footing. Our results demonstrate that ZS-XLT with MT-LLM surpasses TTest on NLU tasks when both rely on the same MT model.

2 Related Work

Translation-based XLT is a strong XLT baseline (Ruder et al., 2021; Ebrahimi et al., 2022; Aggarwal et al., 2022). Previous studies have explored various techniques for leveraging translated training data in XLT (TTrain): these include training on translated data in a single target language (Ebrahimi et al., 2022), using concatenated data from all target languages (Ruder et al., 2021), sequential training starting with the source language followed by the translated target language (Aggar-

[2] Our unified MT-LLM approach integrates additional MT encoder parameters, while TTest utilizes both the MT encoder and decoder for translating test instances into English. Additionally, ZS-XLT is commonly evaluated on languages unsupported by the LLM, where MT models are employed to bridge this gap in both TTest and TTrain.

wal et al., 2022), and jointly training on both combined (Chen et al., 2023). Recent studies have also benchmarked translating test data (TTEST) (Hu et al., 2020; Isbister et al., 2021), which enables ZS-XLT without the need for extensive fine-tuning for each target language, as in the case of TTRAIN. Moreover, both paradigms can be combined by training on round-trip translated noisy source data (translating source-language data to the target language and back) and evaluating on target language test data translated to the source (Oh et al., 2022; Artetxe et al., 2023; Ebing and Glavaš, 2023). Translating training or test data is essentially a *discrete* approach for adjusting the input (i.e., its language) to the LLM (i.e., language that the LLM is proficient in). In contrast, we propose to align latent representation of input, produced by the MT encoders, to the representation space of the LLM backbone via self-distillation, effectively bypassing translation errors that arise from the discrete translation, output of the MT decoder. By retaining continuous MT encoder representations and avoiding their discretization in the MT decoder, our approach also reduces the time and cost of inference vis-a-vis TTEST. This also means that the MT-LLMs (unlike English-centric LLMs) can also reap further gains from TTRAIN, particularly for low-resource languages unseen during pretraining.

While few studies investigated the integration of rich MT representations into LMs, these efforts have generally focused on task-specific integration, without achieving a global representation alignment between the MT encoder and the (large) language model (Ponti et al., 2021; Unanue et al., 2023). Our approach addresses this limitation by achieving task-agnostic representation alignment between MT and LM before task specialization.

Cross-lingual Transfer with LLMs. Widely used LLMs are predominantly trained on English data with English text accounting for 80-90% of their pretraining corpora (Touvron et al., 2023; AI@Meta, 2024). Despite this imbalance, LLMs demonstrate a surprisingly strong performance in (high-resource) languages, which account for only a small fraction of their pretraining corpora (Blevins and Zettlemoyer, 2022). The pretraining focus on English limits the NLU capabilities of LLMs in many low(er)-resource languages, and languages linguistically distant from English (Ojo et al., 2024). Various methods adapt LLMs to languages not covered during pretraining, including continued pretraining (Shliazhko et al., 2023; Fujii et al., 2024), self-instruction (Wei et al., 2023), and vocabulary extension (Zhao et al., 2024). These methods yield gains in model's target language generation capabilities; however, recent work shows that better generation does not translate to stronger NLU performance (Razumovskaia et al., 2024).

3 Methodology

Idea in a Nutshell. Moving beyond translation-based XLT at the discrete (input data) level, we propose a method that merges a base LLM and an MT encoder into a massively multilingual *'MT-LLM'*. This integration enables the MT-LLM model to perform ZS-XLT to any language supported by the MT encoder by leveraging its multilingual language alignment capabilities that substantially extend those of the base LLM (see again Figure 1).

More concretely, we hypothesize that by fine-tuning additional modular parameters on top of the base LLM, we can align the output representations of the merged MT-LLM with the original output representations of the base LLM. In other words, we learn the MT-LLM alignment via distillation with the LLM itself as the teacher. As a result, the MT encoder representations, which are of high-quality for many languages, act as input for the merged MT-LLM. The MT-LLM merge happens in two stages: **1)** self-supervised (general, task-agnostic) adaptation and **2)** task-specific distillation; we describe both in what follows.

Stage 1: Self-Supervised General Adaptation. Figure 2 illustrates the approach in the first stage. We 'vertically' fuse a multilingual MT encoder E and an LLM M into a multilingual MT-LLM $E \times M$. To this end, we introduce two sets of new, trainable parameters Θ: We first initialize a projection $U \in \mathbb{R}^{d_E \times d_M}$ that maps the output representation space $\mathbb{R}^{d_E}$ of the MT encoder E into the input embedding space $\mathbb{R}^{d_M}$ of the LLM M. We then insert low-rank adapters (LoRAs) (Hu et al., 2022) $\Delta W_{i=1}^{|W|}$ into the linear layers W of the LLM M. All other parameters of the MT encoder E and the LLM M are frozen.

The principal idea is to train the new modular parameters $\Theta = \{U, \Delta W_{i=1}^{|W|}\}$ to enable the LLM backbone M to 'understand' output token embeddings of the massively multilingual MT encoder. To this end, we utilize the original LLM as a teacher, which guides the self-supervised training process of our stacked MT-LLM. In the initial pass,

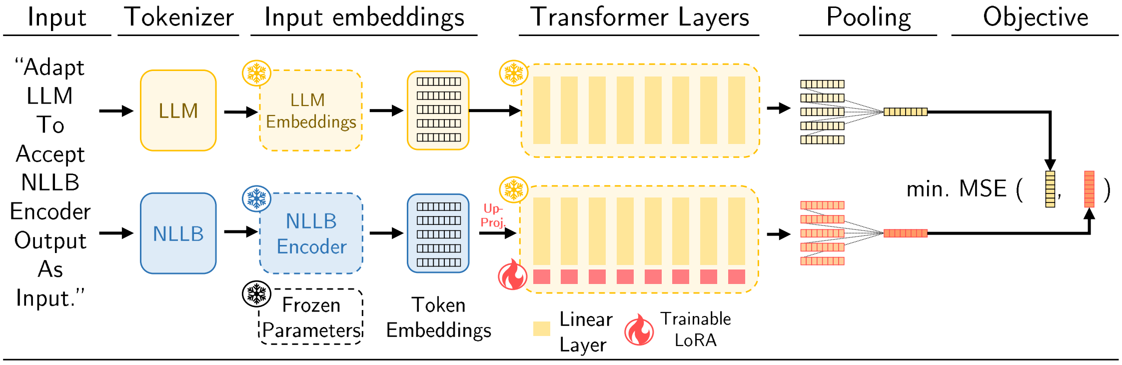

Figure 2: Overview of Stage 1: we merge an MT encoder (NLLB, as a representative MT model) with an LLM (obtaining MT-LLM). We train, in a self-supervised distillation setup, the up-projection and LoRA adapters of the MT-LLM by forcing its output to match (via mean-squared error) the output of the LLM itself.

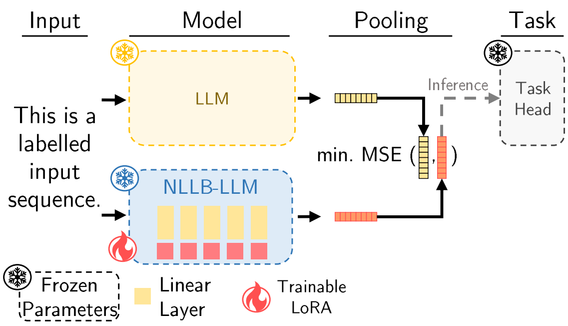

Figure 3: Overview of the architecture in Stage 2: task-specific distillation (again assuming the use of NLLB).

we feed the input sequence S through the original LLM M (i.e., without U and $\Delta W_{i=1}^{|W|}$), and pool the output representations $\{\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{|T_M|}\}$ of tokens $\{t_1, \ldots, t_{|T_M|}\}$ to a sequence embedding $\phi\left(\{x_t\}_{t=1}^{T_M}\right) = \bar{\mathbf{x}}^S$. In the subsequent step, we first re-embed the sequence S with the MT encoder E. We then input the MT encoder output embeddings into the LLM M, now including U and $\Delta W_{i=1}^{L}$, and again pool the resulting output representations $\{\mathbf{z}_1, \ldots, \mathbf{z}_{|T_E|}\}$ of tokens $\{t_1, \ldots, t_{|T_E|}\}$ to a sequence embedding $\phi\left(\{\mathbf{z}_t\}_{t=1}^{T_E}\right) = \bar{\mathbf{z}}^S$. The appropriate pooling function ϕ depends on the prior training regime of the LLM. Common choices include BOS-pooling $\phi\left(\{\mathbf{x}_t\}_{t=1}^{T}\right) = \mathbf{x}_1$ or mean-pooling $\phi\left(\{\mathbf{x}_t\}_{t=1}^{T}\right) = \frac{1}{T}\sum_{i=1}^{T}\mathbf{x}_i$ for encoders, as well as EOS-pooling $\phi\left(\{\mathbf{x}_t\}_{t=1}^{T}\right) = \mathbf{x}_T$ for decoders. We train the parameters Θ (i.e, U and $\Delta W_{i=1}^{L}$) to minimize the mean-squared error $\mathtt{MSE}(\bar{\mathbf{x}}^S, \bar{\mathbf{z}}^S)$.

Stage 2: Task-Specific Distillation. The second stage is designed to bridge the remaining misalignment between the MT encoder and LLM in our fused MT-LLM $E \times M$ in task-specific fine-tuning. We hypothesize that optimizing MT-LLM's representation alignment on general-purpose data is less sample-efficient than task-specific alignment. Figure 3 depicts the task-specific distillation process.

We first fine-tune the base LLM with a classification head $H \in \mathbb{R}^{d_m \times |C|}$ on the labeled task training data. Task fine-tuning reduces the complexity of the LLM's output representations, reducing them to encoding only task-specific features: this, in turn, facilitates task-specific representational alignment for our MT-LLM. Similar to the previous, adaptation stage, we then again fine-tune only the parameters of the LoRA adapters Θ: we continue training the LoRA adapters obtained in Stage 1 (i.e., task-agnostic adaptation).[3] We again minimize the mean-squared error $\mathtt{MSE}(\bar{\mathbf{x}}^S, \bar{\mathbf{z}}^S)$. During inference, we classify instances with the task head H as trained in the initial LLM task fine-tuning. This way we improve the model's ability to generalize on the task, as the MT-LLM is trained to match the output of the knowledge-rich task-specific representations of the fine-tuned LLM.

Both alignment steps together ensure that the latent translations from the MT encoder seamlessly integrate as input representations into the LLM backbone. And this integration extends the access to the knowledge embedded in the LLM to all languages supported by the MT model.

4 Experimental Setup

4.1 Tasks and Languages

We evaluate on two established classification tasks and one multiple-choice machine reading compre-

[3] In our preliminary experiments, merging the LoRAs of the adaptation phase with new, fine-tuning LoRA adapters led to numerical instabilities of weights due to quantization, which severely degraded final task performance.

hension (MRC) task, which all require nuanced NLU capabilities. For each task-dataset combination, we evaluate on all languages supported by the selected underlying MT model.[4]

Natural Language Inference (NLI). We evaluate on XNLI (Conneau et al., 2018), AmericasNLI (AmNLI) (Ebrahimi et al., 2022), and the NLI data of Kardeş-NLU (Senel et al., 2024). We fine-tune models on the training portion of MNLI (Williams et al., 2018). We feed the mean-pooled token representations of the jointly embedded hypothesis-premise sentence-pair into the classifier.

Sentiment Classification is evaluated on NusaX (Winata et al., 2023), which encompasses 10 Indonesian languages.[5] We use the English training, and validation splits with 500 and 100 instances, respectively, as source-language data. We feed the mean-pooled token embeddings of the input text into the classifier.

Multiple-Choice MRC. Belebele is a multiple-choice MRC benchmark encompassing 122 typologically diverse language variants (Bandarkar et al., 2023). We train models on the English training data provided by Bandarkar et al. (2023). We jointly embed the paragraph, question, and answers. For each choice $c_i \in C$, we then average the token embeddings and regress the resulting representation via head $H^{d_M \times 1}$ to a logit $\mathbf{l}_{\mathbf{c_i}}$. We then minimize the cross-entropy between the concatenated choice logits $\{\mathbf{l}_{\mathbf{c_i}}\}_{i=1}^{|C|}$ and the true label.

4.2 Cross-Lingual Transfer Setups

We evaluate XLT abilities of LLMs in two standard setups, ZS-XLT and TTEST. Both paradigms enable XLT without requiring further annotation or prolonged training for any target language. We do not evaluate TTRAIN or involved strategies based on back-translations of source-language training data (Artetxe et al., 2023; Ebing and Glavaš, 2023) as they require computationally intensive task-specific fine-tuning, independently for *each* target language; these variants also require sufficient target language 'understanding' ability from the LLM, which is not there for low-resource languages.[6]

ZS-XLT. In ZS-XLT, the model is first trained on source-language training data. Since the model is multilingual, XLT is inherently supported: we simply run inference on target-language instances. Since LLMs are not sufficiently pretrained multilingually, we align them with an MT encoder with our self-distillation procedure (cf. Stage 1 in §3).

TTEST. In TTEST, the model is initially trained on labeled source-language instances. During inference, the target-language instances are translated to the source language prior to prediction with a dedicated MT model. This enables XLT with monolingual (L)LM backbones.

4.3 Models and Training Setup

Translation Models. We use the NLLB 600M parameter model as our primary MT encoder backbone (Team et al., 2022) for MT-LLM distillation. For TTEST, we translate validation and test datasets with both the 600M NLLB model as well as with the larger, 3.3B parameter variant. We use greedy decoding as Ebing and Glavaš (2023) showed that more sophisticated decoding strategies yield no downstream improvements in XLT.

LLMs. We base our experiments on the Llama 3-8B variant (AI@Meta, 2024) that underwent the 'LLM2Vec process' (BehnamGhader et al., 2024). LLM2Vec is a recipe that converts decoder-only LLMs into powerful sequence encoders by (i) enabling bidirectional attention and continuing training on both (ii) self-supervised masked next-token prediction, and (iii) SimCSE (Gao et al., 2021).[7] We refer to the model that fuses the NLLB 600M encoder with LLM2Vec as NLLB-LLM2Vec. We then adapt to downstream tasks by performing either direct fine-tuning on labeled instances or task-specific self-distillation (cf. Stage 2 in §3).

Training Details. We train all models using LoRAs with rank r=16, alpha α=32, and LoRA dropout of 0.05 inserted into all linear layers. We further train models with the 8-bit AdamW (Loshchilov and Hutter, 2019; Dettmers et al., 2021), 4-bit QLoRA-style quantization (Dettmers et al., 2023), weight decay of 0.01, and with 10% linear warm-up and then linear decay. Experimental results are averaged across three random seeds.[8]

***Stage 1: Setup*.** We train for 10K steps on the

[4]Appendix A.4 lists the full details.

[5]In our experiments on Buginese, as the only outlier, ZS-XLT performance progressively deteriorated with more distillation (cf. Table 12). This is in line with unusual behavior for Buginese reported in other work using NLLB (Ebing and Glavaš, 2023). We thus exclude Buginese results from the main discussion.

[6]Unlike the standalone LLM, our MT-LLM supports TTRAIN to any target language supported by the MT encoder.

[7]We refer the reader to the original LLM2Vec work for further technical details.

[8]For NusaX, we repeat experiments with 5 random seeds due to the smaller dataset size.

10B tokens subsampled from the FineWeb corpus (Penedo et al., 2024). While our approach supports simultaneous adaptation on all languages supported by both the MT model and the LLM, we adapt the LLM only on English text.[9] We set the effective batch size to 256. Learning rate is $2e^{-4}$.

Baselines and Stage 2: Setup. We set the learning rate to $1e^{-4}$ for downstream task experiments. We fine-tune models with an effective batch size of 32, for 3 epochs on NLI, for 5 epochs on Belebele, and for 20 epochs on NusaX. We validate models at every 10% of total training steps. We validate and test on all languages that are supported by our MT model. We start task-specific self-distillation from model snapshots that performed best on source-language validation instances.

5 Results and Discussion

Table 1 summarizes the results for each task, dataset, and model configuration. We then analyze the results per each of these dimensions.

ZS-XLT. Following prior work (Schmidt et al., 2023), we report final XLT test performance for model checkpoints that maximize performance on the source-language (S-DEV) and target-language (T-DEV) validation splits, respectively, in order to estimate the bounds of both expected (S-DEV) and ideal ZS-XLT performance (T-DEV). T-DEV also absorbs fluctuation in transfer performance stemming from sub-optimal hyperparameters (Keung et al., 2020; Schmidt et al., 2023).

LLM2Vec. Despite its strong English performance (as demonstrated in Table 2), the English-centric `LLM2Vec` model based on Llama 3 8B underperforms all other models in ZS-XLT. For instance, the considerably smaller NLLB-Encoder (413M parameters) alone outperforms `LLM2Vec` on both the NLI and NusaX tasks. This confirms that LLMs generally underperform in NLU tasks for languages other than English. Notably, TTEST significantly improves upon the ZS-XLT performance of `LLM2Vec`, especially on datasets that predominantly feature lower-resource languages (AmNLI, Kardeş-NLU). Expectedly, and consistent with findings from related work (Ansell et al., 2023), the larger MT model (NLLB 3B) improves the TTEST performance on all tasks.

NLLB-LLM2Vec. The NLLB encoder alone shows strong ZS-XLT performance on sentiment classification (NusaX) but performs worse on NLI and degrades on MRC (Belebele), as the more intricate NLU task. This suggests that MT encoders indeed lack language understanding abilities and knowledge typically acquired with LM objectives on large-scale corpora. Our integrated NLLB-`LLM2Vec` variants substantially outperform both LLM2Vec and NLLB-Encoder on all NLU tasks, with performance gains on Belebele of 12% and 30%, respectively. Specifically, fine-tuning the NLLB-`LLM2Vec` adapted only in the task-agnostic manner (S1+FT) is already competitive with the more computationally involved TTEST. Our secondary task-specific distillation, i.e., NLLB-`LLM2Vec` S1+S2, further substantially and consistently improves the performance compared to direct fine-tuning (NLLB-`LLM2Vec` S1+FT). The gains are particularly prominent on NusaX ($+9.4\%$), which has the smallest training set. Our full NLLB-`LLM2Vec` S1+S2 consistently beats TTEST based on the same NLLB 600M model by sizable margins ($3 - 11\%$). What is more, NLLB-`LLM2Vec` S1+S2 frequently performs on par or better than TTEST that uses higher-quality translations from the larger NLLB 3B: MT improvements do propagate to ZS-XLT because of favorable model selection on T-DEV. These results show that NLLB-`LLM2Vec` (S1+S2) boosts ZS-XLT by allowing the NLU abilities of the LLM to propagate to many languages via the high-quality multilingual representation space of the NLLB encoder.

In sum, our alignment procedure effectively merges NLLB into `LLM2Vec` to enable ZS-XLT that both outperforms TTEST and reduces inference cost by avoiding decoding in the MT model. Consistent improvements of TTEST with NLLB 3B over NLLB 600M suggests that further ZS-XLT gains can be seized by integrating larger MT models into `LLM2Vec`. Moreover, since NLLB-`LLM2Vec` is inherently multilingual, NLLB-`LLM2Vec` can further benefit from training on translated training data (i.e., TTRAIN): NLLB-`LLM2Vec` is poised to robustly encode noisily translated training data, as it was trained both on MT and denoising autoencoding (Team et al., 2022).

English. Table 2 shows the in-language (i.e., no XLT) English performance by task for our models.

LLM2Vec. Pre-trained on English-dominated web-scale corpora, `LLM2Vec` demonstrates strong perfor-

[9] By unlocking ZS-XLT via self-supervised distillation on English text alone, we demonstrate that any monolingual language model can be equipped with an MT encoder.

	XNLI		AmNLI		Kardeş-NLU		NusaX		Belebele
	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV
Zero-Shot Cross-Lingual Transfer: Fine-tune multilingual model on English training set									
LLM2Vec	68.9$_{\pm2.0}$	71.1$_{\pm2.4}$	40.9$_{\pm2.0}$	43.2$_{\pm1.6}$	46.7$_{\pm1.7}$	51.1$_{\pm13.2}$	54.5$_{\pm13.7}$	58.9$_{\pm10.9}$	48.2$_{\pm3.2}$
NLLB-Encoder	71.6$_{\pm0.2}$	71.8$_{\pm0.3}$	55.3$_{\pm0.6}$	56.4$_{\pm0.3}$	74.9$_{\pm0.5}$	75.2$_{\pm0.6}$	80.7$_{\pm0.2}$	81.7$_{\pm0.1}$	30.4$_{\pm0.4}$
NLLB-LLM2Vec S1+FT	80.0$_{\pm0.9}$	80.4$_{\pm0.4}$	63.0$_{\pm1.9}$	64.3$_{\pm1.2}$	81.5$_{\pm1.3}$	81.3$_{\pm0.4}$	72.7$_{\pm4.4}$	77.5$_{\pm2.4}$	60.2$_{\pm0.5}$
NLLB-LLM2Vec S1+S2	**81.4$_{\pm0.6}$**	**81.7$_{\pm0.5}$**	**64.0$_{\pm0.3}$**	**64.6$_{\pm0.7}$**	**82.2$_{\pm0.5}$**	**82.1$_{\pm0.5}$**	82.1$_{\pm2.6}$	**82.6$_{\pm2.4}$**	62.6$_{\pm0.5}$
Translate-Test: Translate test data to English									
LLM2Vec NLLB-600M	78.7$_{\pm0.7}$	78.6$_{\pm0.9}$	52.0$_{\pm0.7}$	52.7$_{\pm0.6}$	78.8$_{\pm0.8}$	78.4$_{\pm1.0}$	78.3$_{\pm0.9}$	78.8$_{\pm1.2}$	60.7$_{\pm0.7}$
LLM2Vec NLLB-3B	80.2$_{\pm0.6}$	80.2$_{\pm0.8}$	50.9$_{\pm0.4}$	51.2$_{\pm1.7}$	79.9$_{\pm0.9}$	79.9$_{\pm1.0}$	**82.4$_{\pm0.6}$**	**82.6$_{\pm0.5}$**	**64.2$_{\pm0.7}$**

Table 1: **ZS-XLT vs. TTest.** We benchmark models on ZS-XLT against TTest on non-English NLU test sets (cf. §4). S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on source-language (S-DEV) and per target-language (T-DEV) validation splits. Subscripts denote std. deviation. Metrics: accuracy for NLI and Belebele, macro-F1 for NusaX. Best model per column is in **bold**.

	XNLI	NusaX	Belebele
LLM2Vec	**92.5$_{\pm0.3}$**	91.3$_{\pm0.5}$	**94.0$_{\pm0.4}$**
NLLB-Encoder	80.4$_{\pm0.2}$	86.9$_{\pm0.2}$	33.6$_{\pm0.1}$
NLLB-LLM2Vec S1+FT	90.0$_{\pm0.7}$	90.8$_{\pm0.6}$	91.0$_{\pm1.0}$
NLLB-LLM2Vec S1+S2	91.4$_{\pm0.2}$	**92.2$_{\pm0.5}$**	92.4$_{\pm0.7}$

Table 2: **English performance.** We benchmark LLM2Vec, the NLLB encoder, and our fused NLLB-LLM2Vec on English test sets of various NLU benchmarks (cf. §4). See Table 1 for further details.

mance on all tasks. The comparison of in-language performance in Table 2 with ZS-XLT performance in Table 1 shows the scale of performance drop for LLM2Vec in XLT. This means that LLMs require either extensive multilingual pre-training or post-hoc language adaptations for effective XLT.

NLLB-LLM2Vec. The results for NLLB variants, the NLLB encoder alone and our NLLB-LLM2Vec, provide more context for the ZS-XLT results from Table 1. We observe that ZS-XLT performance of NLLB variants is correlated with their in-language English performance. While the NLLB Encoder performs fairly on NusaX (and to some extent also on NLI), it lacks language understanding abilities to that match more complex NLU tasks like Belebele. Our NLLB-LLM2Vec variants, on the other hand, successfully exploit the knowledge of Llama to materially increase English performance over the NLLB Encoder (+58% on Belebele). Our task-agnostic NLLB-LLM2Vec alignment (S1+FT) still lags somewhat behind LLM2Vec after fine-tuning on labeled task data. We manage to narrow this gap for Belebele and NLI with task-specific distillation (S1+S2) (cf. §3) and even surpass the English performance of the LLM for NusaX. This suggests that the task-specific distillation guides NLLB-LLM2Vec to better leverage the knowledge embedded in the weights of LLM2Vec, and shape it specifically for the task.

The results indicate that compositional alignment on the word- or span-level, as introduced in the task-distillation on the Belebele dataset, further improves representational alignment in the MT-LLM. As evident from the comparison of English results in Table 2 and XLT results in Table 1, better global (i.e., task-agnostic) alignment, in turn, directly transfers to closing the 'English knowledge gap', i.e., to better ZS-XLT performance.

5.1 Further Analyses and Discussion

Importance of Adaptation. Figure 4 shows both English and ZS-XLT performance by task for task-specific self-distillation, NLLB-LLM2Vec (S1+S2), after $K \in \{0, 3, 6, 10\}$ steps of task-agnostic alignment (see §3), respectively. The figure points to the importance of task-agnostic adaptation both for English and ZS-XLT performance. The results furthermore highlight that Stage 1 of our alignment is sample-efficient: the largest relative ZS-XLT gains are obtained after only 3K training steps (e.g., +10.5% for Belebele) and then marginalize with further training. We observe the same trends for the English performance (e.g., +15% on Belebele from 3K steps of alignment). These results show that we can effectively tie LLMs and MT encoders into a unified multilingual MT-LLM at computational cost that is negligible w.r.t. both LLM and MT (pre-)training. On NusaX, we observe that while prolonged task-agnostic adaptation benefits the in-language English performance, it does not improve ZS-XLT results. The explanation, we believe, is in the simplicity of the task: ZS-XLT

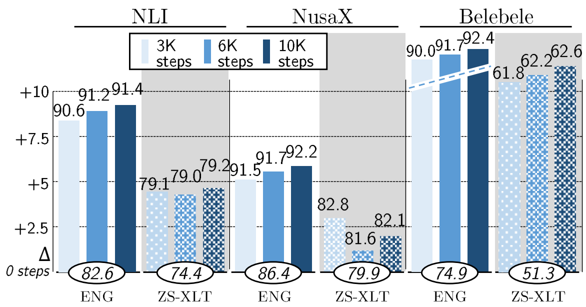

Figure 4: **Impact of Adaptation.** We evaluate our 2-step alignment procedure by $\{0, 3, 6, 10\}$K general adaptation steps (Stage 1) (cf. §3) on English (ENG) and non-English (ZS-XLT) test portions of various NLU benchmarks. Model selection on S-DEV.

	XNLI			AmNLI		KNLU	
GPT-2	ENG	S-D	T-D	S-D	T-D	S-D	T-D
Zero-Shot Cross-Lingual Transfer							
NLLB Enc.	80.4	71.6	71.8	55.3	56.4	74.9	75.2
NLLB FT	82.2	74.7	74.8	**62.4**	**63.2**	**76.2**	**76.5**
Translate-Test							
NLLB-600M	**85.0**	73.6	74.0	54.1	55.2	74.7	75.3
NLLB-3B	**85.0**	**75.1**	**75.4**	52.6	54.5	75.4	75.9

Table 3: **Adaptation on GPT-2.** We perform adaptation (stage 1) with GPT-2 and benchmark NLLB-GPT-2 against GPT-2 in English and non-English test portions of NLI benchmarks. We repeat NLLB Encoder results in gray as a reference. See Table 1 for further details.

performance on NusaX saturates quickly because the NLLB encoder already solves the task well (see Table 1) and thus requires little additional knowledge from LLM2Vec, to which it gets access through the alignment.

The results indicate that the gap in English performance between LLM2Vec and NLLB-LLM2Vec (cf. Table 2) can eventually be closed with longer alignment. We also hypothesize that explicit token or span alignment objectives would improve the generalization: this would be facilitated by the significant overlap between the vocabularies of Llama 3 and NLLB tokenizers.

Fusing Decoder Models with MT Encoders. We additionally test the integration of MT encoders into a decoder LM: we align the NLLB 600M encoder to the GPT-2 medium (354M parameters) and evaluate on NLI.[10] Due to the absence of the EOS token in the pretraining of GPT-2, we perform task-agnostic self-distillation (Stage 1, on the FineWeb corpus) using mean-pooled token representations (cf. §3). Subsequently, we fine-tune the NLLB-GPT-2 directly on MNLI, feeding the EOS-pooled representations into the classifier.[11] We increase the learning rate to $3e^{-4}$ and leave other hyper-parameters unchanged. Like in our main experiments, we compare NLLB-GPT-2 against TTEST with NLLB-600M and NLLB-3.3B, respectively.

The results in Table 3 show that NLLB-GPT-2 successfully taps into GPT-2's 'knowledge' to outperform both the standalone NLLB encoder and the *fair* TTEST baseline using the same NLLB-600M model in ZS-XLT across all datasets. Moreover, NLLB-GPT-2 even surpasses TTEST on GPT-2 with the larger NLLB-3.3B model on AmNLI and KARDEŞ-NLU. These results hold despite the sub-optimal alignment, as indicated by the notable gap in performance to the fine-tuned GPT-2 on the English test portion of XNLI. The discrepancy likely stems from challenges in converting encoders into decoders, as observed in prior work (Wang et al., 2022). We believe that prolonged adaptation and explicit token-level alignment objectives would further improve both sample-efficiency and quality of alignment, reducing the 'knowledge' gap.

6 Conclusion

LLMs quickly emerged as the catch-all solution to NLU in English. However, LLMs still cannot extend their NLU abilities to languages typologically distant from English or virtually unseen at pretraining. In this work, we propose a novel approach to fuse MT encoders with LLM backbones via self-distillation to compile a massively multilingual MT-LLM. The MT-LLM not only strongly improves ZS-XLT performance over TTEST but also removes the overhead of MT decoding at inference. We demonstrate that our distillation procedure is highly efficient and requires only a few thousand steps to convert LLMs into multilingual MT-LLMs, enabling NLU in all languages supported by the MT encoder. We further show that our MT-LLM alignment benefits both encoder and decoder LLMs. In future work we will seek to (1) further improve generalization of MT-LLM by incorporating token-level alignment objectives and (2) extend the MT-LLM to support further languages by post-hoc adaptation of the MT encoder.

[10] We exclude NusaX and Belebele for this ablation: (1) the NLLB encoder performs better than LLM2Vec in the TTEST setting on NusaX; (2) For Belebele, the limited context length of GPT-2 hinders a fair comparison.

[11] We omit task-specific distillation because it performed slightly worse on English in XNLI compared to directly fine-tuning of NLLB-GPT-2 with larger learning rates.

7 Limitations

Our experimental results are based on using Llama 3 and GPT-2 as the LLM backbones and NLLB-600M as the MT encoder in our MT-LLM approach. Expanding our experimental setup to include a wider range of MT encoders and additional LLM backbones would not only validate its applicability across various model families and architectures but also enrich our findings. The Llama 3 backbone of `LLM2Vec` underwent instruction tuning. This means that `LLM2Vec` might have seen labelled data for tasks we experiment on in our work. We strongly believe this does not constitute an issue to evaluate *cross-lingual* transfer of our model configurations. If there was serious leakage of labelled instances, TTEST variants should benefit more strongly as the data is presented in the language Llama was trained on. Our method would gain further support, if our approach extended to generative language modelling. However, sequence-level alignment objectives do not sufficiently align the MT and LLM backbones. The MT and LLM backbones therefore would require either matching or largely overlapping vocabularies to appropriately learn how to fuse the models on the token level (cf. §3). Another non-negligible consideration in our evaluations is that our limited compute budget does not allow for is hyperparameter tuning. We nevertheless believe our main evaluations put model variants on equal footing and hence reliably measure expected ZS-XLT. We further counteract this issue in two ways. First, prior work shows that LoRAs are generally more robust to varying hyperparameters. Second, we report transfer performance both on when selecting models on source-language and per target-language validation. The latter remedies oscillation in ZS-XLT performance (Keung et al., 2020; Schmidt et al., 2023).

References

Divyanshu Aggarwal, Vivek Gupta, and Anoop Kunchukuttan. 2022. IndicXNLI: Evaluating multilingual inference for Indian languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10994–11006, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

AI@Meta. 2024. Llama 3 model card.

Alan Ansell, Marinela Parović, Ivan Vulić, Anna Korhonen, and Edoardo Ponti. 2023. Unifying cross-lingual transfer across scenarios of resource scarcity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3980–3995, Singapore. Association for Computational Linguistics.

Mikel Artetxe, Vedanuj Goswami, Shruti Bhosale, Angela Fan, and Luke Zettlemoyer. 2023. Revisiting machine translation for cross-lingual classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6489–6499, Singapore. Association for Computational Linguistics.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Kelly Marchisio, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress. *Preprint*, arXiv:2405.15032.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. *arXiv preprint arXiv:2308.16884*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *Preprint*, arXiv:2404.05961.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explain the cross-lingual capabilities of english pretrained models. *Preprint*, arXiv:2204.08110.

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.

Yang Chen, Chao Jiang, Alan Ritter, and Wei Xu. 2023. Frustratingly easy label projection for cross-lingual transfer. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5775–5796, Toronto, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2021. 8-bit optimizers via block-wise quantization. *CoRR*, abs/2110.02861.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Benedikt Ebing and Goran Glavaš. 2023. To translate or not to translate: A systematic investigation of translation-based cross-lingual transfer to low-resource languages. *Preprint*, arXiv:2311.09404.

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

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities. *Preprint*, arXiv:2404.17790.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Carolin Holtermann, Paul Röttger, Timm Dill, and Anne Lauscher. 2024. Evaluating the elementary multilingual capabilities of large language models with multiq. *Preprint*, arXiv:2403.03814.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multitask benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Tim Isbister, Fredrik Carlsson, and Magnus Sahlgren. 2021. Should we stop training more monolingual models, and simply use machine translation instead? In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 385–390, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Phillip Keung, Yichao Lu, Julian Salazar, and Vikas Bhardwaj. 2020. Don't use English dev: On the zero-shot cross-lingual evaluation of contextual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 549–554, Online. Association for Computational Linguistics.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Christopher A. Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. Madlad-400: A multilingual and document-level large audited dataset. *Preprint*, arXiv:2309.04662.

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

Jaehoon Oh, Jongwoo Ko, and Se-Young Yun. 2022. Synergy with translation artifacts for training and inference in multilingual tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6747–6754, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jessica Ojo, Kelechi Ogueji, Pontus Stenetorp, and David Ifeoluwa Adelani. 2024. How good are large language models on african languages? *Preprint*, arXiv:2311.07978.

Guilherme Penedo, Hynek Kydlíček, Leandro von Werra, and Thomas Wolf. 2024. Fineweb.

Edoardo Maria Ponti, Julia Kreutzer, Ivan Vulić, and Siva Reddy. 2021. Modelling latent translations for cross-lingual transfer. *Preprint*, arXiv:2107.11353.

Evgeniia Razumovskaia, Ivan Vulić, and Anna Korhonen. 2024. Analyzing and adapting large language models for few-shot multilingual nlu: Are we there yet? *Preprint*, arXiv:2403.01929.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fabian David Schmidt, Ivan Vulić, and Goran Glavaš. 2023. Free lunch: Robust cross-lingual transfer via model checkpoint averaging. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5712–5730, Toronto, Canada. Association for Computational Linguistics.

Lütfi Kerem Senel, Benedikt Ebing, Konul Baghirova, Hinrich Schuetze, and Goran Glavaš. 2024. Kardeş-NLU: Transfer to low-resource languages with the help of a high-resource cousin – a benchmark and evaluation for Turkic languages. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1672–1688, St. Julian's, Malta. Association for Computational Linguistics.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2023. mgpt: Few-shot learners go multilingual. *Preprint*, arXiv:2204.07580.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Inigo Jauregi Unanue, Gholamreza Haffari, and Massimo Piccardi. 2023. T3L: Translate-and-test transfer learning for cross-lingual text classification. *Transactions of the Association for Computational Linguistics*, 11:1147–1161.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective works best for zero-shot generalization? In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22964–22984. PMLR.

Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, Tianxiang Hu, Shangjie Li, Binyuan Hui, Bowen Yu, Dayiheng Liu, Baosong Yang, Fei Huang, and Jun Xie. 2023. Polylm: An open source polyglot large language model. *Preprint*, arXiv:2307.06018.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Genta Indra Winata, Alham Fikri Aji, Samuel Cahyawijaya, Rahmad Mahendra, Fajri Koto, Ade Romadhony, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Pascale Fung, Timothy Baldwin, Jey Han Lau, Rico Sennrich, and Sebastian Ruder. 2023. NusaX: Multilingual parallel sentiment dataset for 10 Indonesian local languages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 815–834, Dubrovnik, Croatia. Association for Computational Linguistics.

Jun Zhao, Zhihao Zhang, Luhui Gao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llama beyond english: An empirical study on language capability transfer. *Preprint*, arXiv:2401.01055.

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. *Preprint*, arXiv:2308.04948.

A Appendix

A.1 Reproducibility details

Compute Requirements. We perform general-purpose adaptation (i.e., stage 1, cf. §3) on 8 A100 80GB, which requires about 22 hours of runtime. All downstream experiments were executed on A100 40GB. Downstream fine-tuning and distillation required for each one of three seeds ca. 20 hours of runtime for NLI, ca. 30 hours of runtime of Belebele, and ca. 20 minutes of runtime per NusaX. We execute these experiments for `LLM2Vec` fine-tuning, NLLB-`LLM2Vec` S1+FT, and NLLB-`LLM2Vec` S1+S2 (cf. §3). The compute required for downstream fine-tuning therefore sums roughly to 450 GPU hours. Subsequent evaluations required, per each of ten evaluated checkpoints, about 3 hours on XNLI, AmNLI, and Kardeş-NLU combined, 5 hours on Belebele, and 10 minutes on NusaX. We estimate that inference therefore requires 725 hours of GPU runtime. In conclusion, our experiments in total required between 50 to 60 days of A100 runtime.

Code. We will make the code publicly available at `github.com/$USER/$PROJECT`. For the preprint under review, the code is available as an attachment to the submission.

Translations. We will make the translated datasets for all tasks available. For the preprint under review, we have attached the translations of validation and tests splits for AmNLI, Kardeş-NLU, NusaX, and Belebele.

Additional Details on Experimental Setup.

Please refer to Table A.3 for details on the number of languages and instances by dataset and split. In what follows, we outline how we accessed the datasets.

MNLI. We access the training portion of the MNLI dataset via Hugging Face at `https://huggingface.co/datasets/nyu-mll/glue`.

XNLI. We access the training portion of the MNLI dataset via Hugging Face at `https://huggingface.co/datasets/nyu-mll/glue`.

AmNLI. We access the training portion of the MNLI dataset via Hugging Face at `https://huggingface.co/datasets/nala-cub/americas_nli`.

Kardeş-NLU. Our code includes a script to access the dataset via the Hugging Face `datasets` framework. The original dataset is available at: `https://github.com/lksenel/Kardes-NLU`.

NusaX. We access the dataset via Hugging Face at `https://huggingface.co/datasets/indonlp/NusaX-senti`.

Belebele. The training dataset is available at `https://github.com/facebookresearch/belebele`. We access the dataset via Hugging Face at `https://huggingface.co/datasets/facebook/belebele`.

LLM2Vec. We use `LLM2Vec` that has been trained without supervision which is available on the Hugging Face hub: here.

NLLB. The distilled 600M parameters variant of NLLB is available at `https://huggingface.co/facebook/nllb-200-distilled-600M`. NLLB 3B can be accessed via `https://huggingface.co/facebook/nllb-200-3.3B`.

A.2 NLLB vs. XLM-R on XNLI

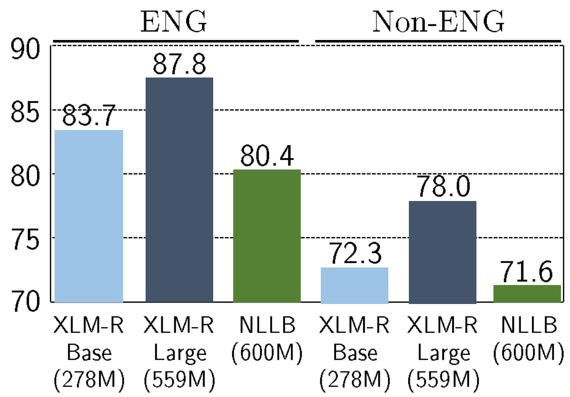

Figure 5: Performance on XNLI English and non-English test portions when selecting the model on English validation instances (cf. §4) for XLM-R-Base, XLM-R-Large, and the NLLB-Encoder (Conneau et al., 2020; Team et al., 2022)

.

A.3 Datasets

	MNLI	XNLI	AmNLI	Kardeş-NLU	NusaX	Belebele
No. of Languages	1	15	3	4	9	117
Training	392,702	−−	−−	−−	500	67,541
Validation	−−	2,490	743	1,000	100	3,773
Test	−−	5,010	750	2,000	400	900

Table 4: Number of instances by split per dataset. For NLI, we train on the training split of MNLI (Williams et al., 2018). Number of languages denominates the number of languages supported by NLLB. For Belebele, we construct training and validation datasets with the scripts provided by (Bandarkar et al., 2023) (cf. Appendix A).

A.4 Full results

Main Results.

	EN	AR	BG	DE	EL	ES	FR	HI	RU	SW	TH	TR	UR	VI	ZH	AVG
Zero-Shot Cross-Lingual Transfer: Fine-tune multilingual model on English training set																
LLM2Vec	$92.5_{\pm 0.3}$	$63.6_{\pm 5.1}$	$68.5_{\pm 3.8}$	$79.8_{\pm 1.3}$	$73.0_{\pm 1.2}$	$84.5_{\pm 0.8}$	$83.0_{\pm 0.9}$	$55.8_{\pm 8.3}$	$78.0_{\pm 1.4}$	$43.0_{\pm 3.6}$	$66.6_{\pm 0.9}$	$67.3_{\pm 2.4}$	$45.2_{\pm 4.1}$	$78.0_{\pm 1.3}$	$78.7_{\pm 0.9}$	$68.9_{\pm 2.0}$
NLLB-Encoder	$80.4_{\pm 0.2}$	$71.3_{\pm 0.8}$	$75.2_{\pm 0.1}$	$74.9_{\pm 0.4}$	$74.5_{\pm 0.0}$	$75.5_{\pm 0.4}$	$75.3_{\pm 0.4}$	$68.6_{\pm 0.8}$	$72.7_{\pm 0.2}$	$69.5_{\pm 0.3}$	$65.7_{\pm 1.0}$	$72.6_{\pm 0.4}$	$65.1_{\pm 0.7}$	$72.8_{\pm 0.2}$	$69.1_{\pm 0.6}$	$71.6_{\pm 0.2}$
NLLB-LLM2Vec S1+FT	$90.7_{\pm 0.4}$	$80.9_{\pm 1.2}$	$85.6_{\pm 0.9}$	$84.9_{\pm 0.8}$	$73.8_{\pm 2.0}$	$86.2_{\pm 0.8}$	$85.8_{\pm 0.5}$	$75.1_{\pm 1.6}$	$82.8_{\pm 0.6}$	$79.3_{\pm 1.1}$	$76.0_{\pm 1.7}$	$78.8_{\pm 1.5}$	$70.7_{\pm 1.8}$	$82.1_{\pm 0.7}$	$78.6_{\pm 1.4}$	$80.0_{\pm 0.9}$
NLLB-LLM2Vec S1+S2	$91.4_{\pm 0.2}$	$81.9_{\pm 0.4}$	$86.5_{\pm 0.5}$	$85.8_{\pm 0.3}$	$79.3_{\pm 6.4}$	$87.4_{\pm 0.3}$	$86.7_{\pm 0.4}$	$76.8_{\pm 0.6}$	$83.6_{\pm 0.5}$	$79.6_{\pm 0.4}$	$77.4_{\pm 0.7}$	$80.0_{\pm 0.3}$	$72.1_{\pm 0.5}$	$83.2_{\pm 0.3}$	$80.0_{\pm 0.9}$	$81.4_{\pm 0.6}$
Translate-Test: Translate test data to English																
LLM2Vec NLLB-600M	$92.5_{\pm 0.3}$	$78.2_{\pm 0.7}$	$83.3_{\pm 0.5}$	$83.8_{\pm 0.8}$	$82.8_{\pm 0.8}$	$85.8_{\pm 0.8}$	$84.1_{\pm 0.9}$	$75.7_{\pm 0.6}$	$78.9_{\pm 0.5}$	$73.2_{\pm 0.9}$	$72.7_{\pm 0.7}$	$79.2_{\pm 0.7}$	$69.5_{\pm 0.8}$	$79.7_{\pm 0.5}$	$74.9_{\pm 0.5}$	$78.7_{\pm 0.7}$
LLM2Vec NLLB-3B	$92.5_{\pm 0.3}$	$80.0_{\pm 0.8}$	$85.1_{\pm 0.5}$	$85.3_{\pm 0.7}$	$83.9_{\pm 0.5}$	$86.5_{\pm 0.8}$	$85.0_{\pm 0.6}$	$76.9_{\pm 0.6}$	$81.2_{\pm 0.4}$	$74.4_{\pm 1.0}$	$74.5_{\pm 0.9}$	$81.2_{\pm 0.7}$	$70.0_{\pm 0.4}$	$81.3_{\pm 0.6}$	$77.7_{\pm 0.4}$	$80.2_{\pm 0.6}$

Table 5: **XNLI (1/2).** We benchmark models on test portions of XNLI (cf. §4). S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on **source-language (S-DEV) validation splits**. Subscripts denote std. deviation. Metric: accuracy.

	EN	AR	BG	DE	EL	ES	FR	HI	RU	SW	TH	TR	UR	VI	ZH	AVG
Zero-Shot Cross-Lingual Transfer: Fine-tune multilingual model on English training set																
LLM2Vec	$92.5_{\pm 0.3}$	$66.3_{\pm 5.1}$	$70.8_{\pm 3.5}$	$81.4_{\pm 3.0}$	$73.8_{\pm 3.5}$	$84.8_{\pm 1.2}$	$83.5_{\pm 1.2}$	$61.8_{\pm 8.0}$	$78.3_{\pm 2.4}$	$48.5_{\pm 3.9}$	$67.1_{\pm 2.9}$	$70.2_{\pm 1.2}$	$51.4_{\pm 5.6}$	$78.0_{\pm 1.3}$	$79.0_{\pm 1.3}$	$71.1_{\pm 2.4}$
NLLB-Encoder	$80.4_{\pm 0.2}$	$71.3_{\pm 0.8}$	$75.6_{\pm 0.5}$	$75.0_{\pm 0.4}$	$74.7_{\pm 0.1}$	$75.6_{\pm 0.6}$	$75.6_{\pm 0.1}$	$68.7_{\pm 0.8}$	$72.8_{\pm 0.2}$	$69.5_{\pm 0.2}$	$66.0_{\pm 0.6}$	$72.9_{\pm 0.3}$	$65.2_{\pm 0.8}$	$72.8_{\pm 0.2}$	$69.2_{\pm 0.6}$	$71.8_{\pm 0.3}$
NLLB-LLM2Vec S1+FT	$90.7_{\pm 0.4}$	$81.2_{\pm 0.6}$	$85.5_{\pm 0.9}$	$85.0_{\pm 0.7}$	$75.0_{\pm 2.2}$	$86.5_{\pm 0.5}$	$85.8_{\pm 0.5}$	$75.8_{\pm 0.9}$	$82.8_{\pm 0.6}$	$79.4_{\pm 0.9}$	$77.0_{\pm 0.6}$	$79.1_{\pm 1.4}$	$71.1_{\pm 1.6}$	$82.4_{\pm 0.4}$	$79.5_{\pm 0.4}$	$80.4_{\pm 0.4}$
NLLB-LLM2Vec S1+S2	$91.4_{\pm 0.2}$	$82.2_{\pm 0.8}$	$86.2_{\pm 0.4}$	$85.9_{\pm 0.4}$	$79.8_{\pm 5.8}$	$87.5_{\pm 0.7}$	$86.8_{\pm 0.6}$	$76.9_{\pm 0.7}$	$84.0_{\pm 0.3}$	$79.9_{\pm 1.0}$	$77.8_{\pm 1.0}$	$80.0_{\pm 0.4}$	$72.2_{\pm 1.1}$	$83.4_{\pm 0.4}$	$80.8_{\pm 0.4}$	$81.7_{\pm 0.5}$
Translate-Test: Translate test data to English																
LLM2Vec NLLB-600M	$92.5_{\pm 0.3}$	$77.9_{\pm 1.5}$	$82.1_{\pm 0.4}$	$83.6_{\pm 0.9}$	$82.2_{\pm 0.4}$	$86.2_{\pm 0.5}$	$84.2_{\pm 1.0}$	$74.9_{\pm 0.7}$	$78.7_{\pm 1.5}$	$73.7_{\pm 1.8}$	$72.8_{\pm 0.8}$	$79.2_{\pm 1.6}$	$69.9_{\pm 2.2}$	$79.7_{\pm 0.5}$	$75.1_{\pm 0.5}$	$78.6_{\pm 0.9}$
LLM2Vec NLLB-3B	$92.5_{\pm 0.3}$	$79.9_{\pm 1.4}$	$84.0_{\pm 0.1}$	$85.1_{\pm 1.1}$	$83.5_{\pm 0.5}$	$87.0_{\pm 0.9}$	$85.3_{\pm 1.1}$	$76.3_{\pm 0.2}$	$80.9_{\pm 1.5}$	$75.0_{\pm 2.0}$	$74.6_{\pm 0.9}$	$81.7_{\pm 1.5}$	$70.0_{\pm 1.7}$	$81.3_{\pm 0.6}$	$77.7_{\pm 0.3}$	$80.2_{\pm 0.8}$

Table 6: **XNLI (2/2).** We benchmark models on test portions of XNLI (cf. §4). S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on **per target-language (T-DEV) validation splits**. Subscripts denote std. deviation. Metric: accuracy.

	AYM		GN		QUY		AVG	
	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV
Zero-Shot Cross-Lingual Transfer: Fine-tune multilingual model on English training set								
LLM2Vec	$40.9_{\pm 3.2}$	$41.8_{\pm 2.0}$	$42.3_{\pm 1.7}$	$44.8_{\pm 1.8}$	$39.6_{\pm 1.4}$	$42.8_{\pm 1.0}$	$40.9_{\pm 2.0}$	$43.2_{\pm 1.6}$
NLLB-Encoder	$62.3_{\pm 0.5}$	$63.4_{\pm 0.1}$	$62.8_{\pm 0.9}$	$64.2_{\pm 0.2}$	$40.8_{\pm 1.0}$	$41.5_{\pm 1.0}$	$55.3_{\pm 0.6}$	$56.4_{\pm 0.3}$
NLLB-LLM2Vec S1+FT	$60.4_{\pm 2.2}$	$62.5_{\pm 1.6}$	$68.8_{\pm 2.4}$	$69.2_{\pm 1.4}$	$59.9_{\pm 1.9}$	$61.2_{\pm 1.4}$	$63.0_{\pm 1.9}$	$64.3_{\pm 1.2}$
NLLB-LLM2Vec S1+S2	$61.0_{\pm 0.6}$	$61.8_{\pm 1.0}$	$69.6_{\pm 1.3}$	$69.8_{\pm 1.4}$	$61.4_{\pm 1.8}$	$62.3_{\pm 1.1}$	$64.0_{\pm 0.3}$	$64.6_{\pm 0.7}$
Translate-Test: Translate test data to English								
LLM2Vec NLLB-600M	$50.7_{\pm 1.9}$	$51.2_{\pm 1.9}$	$55.0_{\pm 0.2}$	$56.8_{\pm 1.7}$	$50.5_{\pm 0.3}$	$50.0_{\pm 1.3}$	$52.0_{\pm 0.7}$	$52.7_{\pm 0.6}$
LLM2Vec NLLB-3B	$45.2_{\pm 0.4}$	$45.0_{\pm 3.3}$	$58.0_{\pm 0.9}$	$59.3_{\pm 2.1}$	$49.6_{\pm 1.0}$	$49.3_{\pm 0.5}$	$50.9_{\pm 0.4}$	$51.2_{\pm 0.7}$

	AZ		KK		KY		UZ		AVG	
	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV
Zero-Shot Cross-Lingual Transfer: Fine-tune multilingual model on English training set										
LLM2Vec	$54.0_{\pm 0.4}$	$61.1_{\pm 0.9}$	$43.5_{\pm 2.3}$	$46.3_{\pm 1.4}$	$41.8_{\pm 1.4}$	$43.1_{\pm 0.1}$	$47.5_{\pm 3.3}$	$55.9_{\pm 2.0}$	$46.7_{\pm 1.7}$	$51.6_{\pm 0.3}$
NLLB-Encoder	$75.8_{\pm 0.4}$	$76.1_{\pm 0.2}$	$74.2_{\pm 0.5}$	$74.3_{\pm 0.6}$	$74.8_{\pm 0.5}$	$75.2_{\pm 0.5}$	$74.8_{\pm 0.8}$	$75.1_{\pm 1.0}$	$74.9_{\pm 0.5}$	$75.2_{\pm 0.6}$
NLLB-LLM2Vec S1+FT	$82.8_{\pm 1.0}$	$82.3_{\pm 0.8}$	$81.1_{\pm 1.2}$	$80.0_{\pm 0.8}$	$80.3_{\pm 1.9}$	$81.3_{\pm 0.7}$	$81.9_{\pm 1.4}$	$81.5_{\pm 0.7}$	$81.5_{\pm 1.3}$	$81.3_{\pm 0.4}$
NLLB-LLM2Vec S1+S2	$83.4_{\pm 1.2}$	$83.0_{\pm 0.9}$	$81.9_{\pm 0.5}$	$81.7_{\pm 0.7}$	$81.2_{\pm 0.4}$	$81.6_{\pm 0.9}$	$82.1_{\pm 0.2}$	$82.3_{\pm 0.5}$	$82.5_{\pm 0.5}$	$82.1_{\pm 0.5}$
Translate-Test: Translate test data to English										
LLM2Vec NLLB-600M	$82.2_{\pm 1.2}$	$81.7_{\pm 1.1}$	$77.2_{\pm 0.4}$	$77.1_{\pm 0.4}$	$76.5_{\pm 1.0}$	$75.8_{\pm 1.9}$	$79.2_{\pm 0.8}$	$79.1_{\pm 0.8}$	$78.8_{\pm 0.8}$	$78.4_{\pm 1.0}$
LLM2Vec NLLB-3B	$84.9_{\pm 0.8}$	$84.9_{\pm 0.9}$	$78.5_{\pm 1.1}$	$78.9_{\pm 0.9}$	$75.8_{\pm 0.5}$	$75.1_{\pm 1.2}$	$80.4_{\pm 1.1}$	$80.8_{\pm 1.1}$	$79.9_{\pm 0.9}$	$79.9_{\pm 1.0}$

Table 7: **AmNLI & Kardeş-NLU.** We benchmark models on test portions of AmNLI and Kardeş-NLU (cf. §4). S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on source-language (S-DEV) and per target-language (T-DEV) validation splits. Subscripts denote std. deviation. Metric: accuracy.

	ENG	ACE	BAN	BJN	BUG	IND	JAV	MIN	SUN	AVG
Zero-Shot Cross-Lingual Transfer: Fine-tune multilingual model on English training set										
LLM2Vec	$91.3_{\pm 0.5}$	$41.6_{\pm 14.5}$	$45.1_{\pm 17.6}$	$56.9_{\pm 13.9}$	$30.5_{\pm 16.7}$	$83.3_{\pm 2.4}$	$54.4_{\pm 14.8}$	$56.5_{\pm 16.8}$	$43.6_{\pm 17.9}$	$51.5_{\pm 14.0}$
NLLB-Encoder	$86.9_{\pm 2.1}$	$80.3_{\pm 0.8}$	$76.9_{\pm 2.0}$	$83.8_{\pm 1.0}$	$67.4_{\pm 2.7}$	$86.4_{\pm 0.7}$	$83.6_{\pm 0.8}$	$80.1_{\pm 0.5}$	$80.7_{\pm 0.4}$	$79.9_{\pm 0.3}$
NLLB-LLM2Vec S1+FT	$90.8_{\pm 0.6}$	$73.9_{\pm 4.3}$	$70.6_{\pm 2.4}$	$79.1_{\pm 2.5}$	$53.9_{\pm 9.3}$	$86.7_{\pm 2.3}$	$81.0_{\pm 2.4}$	$72.6_{\pm 6.3}$	$78.4_{\pm 2.7}$	$74.5_{\pm 3.2}$
NLLB-LLM2Vec S1+S2	$92.2_{\pm 0.5}$	$81.5_{\pm 2.5}$	$74.8_{\pm 4.1}$	$82.3_{\pm 2.6}$	$67.1_{\pm 1.7}$	$89.3_{\pm 0.6}$	$86.4_{\pm 1.8}$	$80.6_{\pm 3.3}$	$83.1_{\pm 3.1}$	$80.6_{\pm 2.3}$
Translate-Test: Translate test data to English										
LLM2Vec NLLB-600M	$91.3_{\pm 0.5}$	$74.2_{\pm 2.2}$	$72.1_{\pm 1.6}$	$79.1_{\pm 1.8}$	$71.3_{\pm 3.7}$	$86.7_{\pm 1.2}$	$79.4_{\pm 1.9}$	$78.7_{\pm 1.6}$	$81.8_{\pm 1.2}$	$77.9_{\pm 1.4}$
LLM2Vec NLLB-3B	$91.3_{\pm 0.5}$	$77.7_{\pm 1.4}$	$75.6_{\pm 1.3}$	$83.8_{\pm 1.5}$	$71.5_{\pm 4.3}$	$88.7_{\pm 0.6}$	$84.3_{\pm 1.3}$	$82.1_{\pm 0.4}$	$86.1_{\pm 1.0}$	$81.2_{\pm 0.8}$

Table 8: **NusaX** We benchmark models on test portions of NusaX (cf. §4). S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on **per target-language (T-DEV) validation splits**. Subscripts denote std. deviation. Metric: macro-F1.

	Zero-Shot Cross-Lingual Transfer				Translate-Test	
	LLM2Vec	NLLB-Encoder	NLLB-LLM2Vec S1+FT	NLLB-LLM2Vec S1+S2	LLM2Vec NLLB-600M	LLM2Vec NLLB-3B
eng_Latn	$92.5_{\pm0.3}$	$80.4_{\pm0.2}$	$90.0_{\pm0.7}$	$91.4_{\pm0.2}$	$92.5_{\pm0.3}$	$92.5_{\pm0.3}$
AVG	$48.2_{\pm3.2}$	$30.4_{\pm0.4}$	$60.2_{\pm0.5}$	$62.6_{\pm0.5}$	$60.7_{\pm0.8}$	$64.2_{\pm0.7}$
acm_Arab	$52.3_{\pm4.1}$	$30.1_{\pm0.2}$	$55.8_{\pm1.2}$	$56.7_{\pm0.6}$	$62.9_{\pm0.6}$	$60.2_{\pm0.3}$
afr_Latn	$66.9_{\pm8.5}$	$33.3_{\pm0.2}$	$76.3_{\pm1.4}$	$80.1_{\pm1.1}$	$79.4_{\pm0.5}$	$79.8_{\pm0.2}$
als_Latn	$47.2_{\pm7.5}$	$28.9_{\pm0.4}$	$70.8_{\pm2.5}$	$72.9_{\pm1.0}$	$71.1_{\pm1.0}$	$75.7_{\pm1.3}$
amh_Ethi	$27.7_{\pm0.9}$	$30.8_{\pm1.1}$	$50.0_{\pm0.6}$	$50.4_{\pm1.5}$	$50.9_{\pm1.4}$	$59.0_{\pm1.0}$
apc_Arab	$52.1_{\pm2.6}$	$30.6_{\pm1.6}$	$58.8_{\pm0.9}$	$60.7_{\pm0.6}$	$66.3_{\pm0.8}$	$65.1_{\pm0.6}$
arb_Arab	$68.5_{\pm6.3}$	$28.5_{\pm1.1}$	$65.7_{\pm0.9}$	$68.1_{\pm1.5}$	$75.2_{\pm1.0}$	$75.9_{\pm0.6}$
ars_Arab	$55.2_{\pm4.8}$	$29.9_{\pm1.6}$	$58.4_{\pm0.5}$	$59.6_{\pm1.1}$	$66.9_{\pm1.3}$	$61.3_{\pm0.2}$
ary_Arab	$45.2_{\pm3.2}$	$31.0_{\pm0.5}$	$46.5_{\pm1.1}$	$48.3_{\pm0.8}$	$51.4_{\pm2.3}$	$54.9_{\pm0.8}$
arz_Arab	$50.1_{\pm6.5}$	$31.2_{\pm1.6}$	$56.8_{\pm0.7}$	$59.7_{\pm0.8}$	$68.6_{\pm1.4}$	$67.4_{\pm0.4}$
asm_Beng	$29.1_{\pm1.8}$	$29.1_{\pm0.5}$	$50.9_{\pm2.5}$	$54.0_{\pm1.1}$	$52.5_{\pm1.3}$	$60.3_{\pm0.2}$
azj_Latn	$44.5_{\pm7.0}$	$29.7_{\pm0.6}$	$53.7_{\pm0.3}$	$56.7_{\pm0.8}$	$64.2_{\pm0.8}$	$66.7_{\pm1.0}$
bam_Latn	$32.1_{\pm1.4}$	$29.3_{\pm0.2}$	$41.1_{\pm0.9}$	$39.5_{\pm1.2}$	$36.0_{\pm0.8}$	$37.3_{\pm1.3}$
ben_Beng	$31.9_{\pm0.8}$	$28.5_{\pm0.8}$	$58.1_{\pm2.4}$	$62.5_{\pm0.8}$	$63.5_{\pm0.8}$	$65.9_{\pm0.7}$
ben_Latn	$34.9_{\pm0.7}$	$27.7_{\pm0.1}$	$27.0_{\pm0.9}$	$28.2_{\pm1.2}$	$29.6_{\pm0.7}$	$25.5_{\pm1.5}$
bod_Tibt	$26.7_{\pm1.9}$	$28.0_{\pm1.5}$	$33.5_{\pm1.3}$	$33.9_{\pm1.1}$	$29.7_{\pm1.7}$	$35.9_{\pm1.9}$
bul_Cyrl	$73.4_{\pm7.5}$	$30.8_{\pm1.2}$	$75.3_{\pm1.2}$	$77.9_{\pm0.1}$	$70.0_{\pm1.7}$	$77.9_{\pm0.5}$
cat_Latn	$77.6_{\pm6.0}$	$33.7_{\pm1.0}$	$79.1_{\pm1.4}$	$82.3_{\pm0.9}$	$74.3_{\pm1.5}$	$79.5_{\pm0.8}$
ceb_Latn	$43.8_{\pm3.7}$	$29.1_{\pm1.2}$	$67.0_{\pm1.6}$	$70.7_{\pm0.7}$	$66.9_{\pm1.9}$	$73.2_{\pm1.5}$
ces_Latn	$73.7_{\pm7.6}$	$30.2_{\pm1.4}$	$71.3_{\pm0.6}$	$75.9_{\pm1.5}$	$70.8_{\pm0.9}$	$76.8_{\pm1.0}$
ckb_Arab	$33.2_{\pm0.9}$	$28.7_{\pm0.6}$	$58.3_{\pm0.5}$	$59.8_{\pm1.2}$	$62.7_{\pm1.3}$	$65.1_{\pm1.0}$
dan_Latn	$73.0_{\pm7.3}$	$32.2_{\pm0.7}$	$81.0_{\pm0.8}$	$83.5_{\pm0.7}$	$73.8_{\pm1.7}$	$79.2_{\pm1.3}$
deu_Latn	$85.1_{\pm2.6}$	$33.8_{\pm0.7}$	$76.0_{\pm0.7}$	$78.1_{\pm0.9}$	$76.1_{\pm1.0}$	$80.1_{\pm0.2}$
ell_Grek	$74.8_{\pm6.7}$	$28.7_{\pm0.9}$	$62.2_{\pm1.6}$	$67.5_{\pm0.2}$	$70.7_{\pm0.3}$	$76.5_{\pm0.8}$
est_Latn	$46.1_{\pm8.0}$	$30.6_{\pm0.3}$	$66.0_{\pm2.0}$	$70.1_{\pm1.0}$	$64.2_{\pm1.4}$	$71.4_{\pm1.1}$
eus_Latn	$45.4_{\pm6.4}$	$31.0_{\pm1.2}$	$63.1_{\pm0.6}$	$66.7_{\pm1.1}$	$72.6_{\pm0.6}$	$75.9_{\pm0.7}$
fin_Latn	$55.1_{\pm9.7}$	$31.1_{\pm1.1}$	$69.0_{\pm0.7}$	$73.0_{\pm0.7}$	$67.3_{\pm1.5}$	$77.7_{\pm0.4}$
fra_Latn	$88.0_{\pm1.5}$	$31.6_{\pm0.5}$	$79.1_{\pm0.5}$	$82.4_{\pm1.0}$	$80.0_{\pm1.3}$	$82.6_{\pm0.6}$
fuv_Latn	$28.5_{\pm0.4}$	$28.0_{\pm1.2}$	$29.5_{\pm0.8}$	$28.1_{\pm0.6}$	$27.9_{\pm0.5}$	$26.9_{\pm1.5}$
gaz_Latn	$31.4_{\pm0.6}$	$29.4_{\pm0.3}$	$41.6_{\pm1.7}$	$42.8_{\pm0.7}$	$45.0_{\pm0.3}$	$48.6_{\pm0.6}$
grn_Latn	$37.2_{\pm0.9}$	$31.0_{\pm1.2}$	$52.1_{\pm0.4}$	$52.5_{\pm0.9}$	$47.3_{\pm1.4}$	$54.1_{\pm0.8}$
guj_Gujr	$27.9_{\pm0.4}$	$30.5_{\pm1.3}$	$52.9_{\pm0.7}$	$55.9_{\pm1.7}$	$62.6_{\pm1.0}$	$64.7_{\pm0.6}$
hat_Latn	$38.2_{\pm2.5}$	$29.1_{\pm1.1}$	$63.3_{\pm1.3}$	$67.4_{\pm0.9}$	$65.9_{\pm0.9}$	$71.5_{\pm0.7}$
hau_Latn	$32.0_{\pm0.8}$	$28.4_{\pm0.6}$	$58.3_{\pm1.5}$	$62.1_{\pm0.4}$	$59.4_{\pm2.2}$	$59.9_{\pm1.1}$
heb_Hebr	$39.8_{\pm6.8}$	$32.8_{\pm0.4}$	$64.0_{\pm1.7}$	$66.6_{\pm0.3}$	$68.8_{\pm1.6}$	$71.7_{\pm0.7}$
hin_Deva	$55.1_{\pm5.7}$	$28.6_{\pm1.0}$	$62.4_{\pm1.6}$	$65.7_{\pm1.0}$	$70.6_{\pm0.6}$	$73.0_{\pm1.5}$
hrv_Latn	$63.4_{\pm9.5}$	$31.5_{\pm1.1}$	$73.1_{\pm1.0}$	$77.4_{\pm0.5}$	$69.7_{\pm1.2}$	$73.3_{\pm0.5}$
hun_Latn	$62.4_{\pm2.6}$	$30.9_{\pm1.0}$	$67.8_{\pm0.7}$	$71.6_{\pm0.5}$	$66.9_{\pm0.5}$	$72.7_{\pm1.5}$
hye_Armn	$27.8_{\pm0.6}$	$28.4_{\pm0.8}$	$56.6_{\pm1.5}$	$58.6_{\pm0.6}$	$52.0_{\pm1.3}$	$61.5_{\pm1.7}$
ibo_Latn	$31.0_{\pm0.9}$	$30.2_{\pm1.2}$	$48.8_{\pm1.1}$	$49.2_{\pm0.3}$	$47.0_{\pm1.6}$	$51.7_{\pm1.8}$
ilo_Latn	$38.6_{\pm1.2}$	$29.6_{\pm1.9}$	$62.6_{\pm0.9}$	$66.7_{\pm1.3}$	$61.9_{\pm0.8}$	$67.7_{\pm0.7}$
ind_Latn	$73.3_{\pm6.7}$	$30.6_{\pm0.6}$	$79.4_{\pm1.3}$	$82.5_{\pm0.6}$	$74.7_{\pm0.7}$	$76.5_{\pm0.4}$
isl_Latn	$44.2_{\pm5.4}$	$28.3_{\pm1.5}$	$61.6_{\pm0.3}$	$65.0_{\pm1.2}$	$56.3_{\pm0.8}$	$57.9_{\pm1.1}$
ita_Latn	$85.9_{\pm1.5}$	$31.9_{\pm1.7}$	$79.8_{\pm0.0}$	$82.4_{\pm0.5}$	$74.0_{\pm1.6}$	$78.2_{\pm1.4}$
jav_Latn	$45.2_{\pm5.3}$	$29.4_{\pm0.9}$	$71.4_{\pm1.0}$	$74.2_{\pm0.5}$	$56.6_{\pm1.0}$	$58.1_{\pm0.8}$
jpn_Jpan	$77.3_{\pm2.4}$	$30.4_{\pm1.1}$	$65.5_{\pm2.2}$	$67.1_{\pm0.9}$	$60.7_{\pm0.8}$	$65.1_{\pm0.7}$
kac_Latn	$32.7_{\pm0.5}$	$29.6_{\pm0.6}$	$39.1_{\pm0.4}$	$40.9_{\pm1.2}$	$37.0_{\pm0.3}$	$39.7_{\pm1.3}$
kan_Knda	$28.6_{\pm0.4}$	$30.3_{\pm1.0}$	$55.7_{\pm1.4}$	$56.9_{\pm1.3}$	$62.4_{\pm1.3}$	$65.6_{\pm1.0}$
kat_Geor	$27.1_{\pm1.6}$	$27.4_{\pm0.4}$	$50.4_{\pm1.9}$	$51.1_{\pm0.5}$	$50.3_{\pm2.0}$	$56.7_{\pm2.0}$
kaz_Cyrl	$40.7_{\pm4.7}$	$29.6_{\pm0.6}$	$55.9_{\pm1.3}$	$59.3_{\pm0.7}$	$65.3_{\pm0.5}$	$69.1_{\pm0.4}$
kea_Latn	$43.8_{\pm1.3}$	$31.0_{\pm0.3}$	$61.2_{\pm1.3}$	$65.6_{\pm1.5}$	$59.7_{\pm0.8}$	$62.8_{\pm0.5}$
khk_Cyrl	$33.9_{\pm2.0}$	$28.4_{\pm0.5}$	$44.4_{\pm1.5}$	$44.8_{\pm1.2}$	$48.1_{\pm0.5}$	$52.5_{\pm1.7}$
khm_Khmr	$29.6_{\pm2.2}$	$29.8_{\pm0.2}$	$47.3_{\pm0.8}$	$51.6_{\pm0.8}$	$44.4_{\pm0.8}$	$47.7_{\pm1.6}$
kin_Latn	$36.2_{\pm1.0}$	$29.3_{\pm0.2}$	$55.3_{\pm0.7}$	$57.1_{\pm0.8}$	$55.0_{\pm1.0}$	$58.3_{\pm0.8}$
kir_Cyrl	$40.7_{\pm3.0}$	$30.6_{\pm0.9}$	$56.6_{\pm1.5}$	$58.7_{\pm0.8}$	$62.9_{\pm1.3}$	$66.5_{\pm1.7}$
kor_Hang	$77.5_{\pm3.1}$	$32.3_{\pm0.8}$	$61.6_{\pm2.1}$	$62.3_{\pm0.7}$	$67.9_{\pm2.0}$	$69.0_{\pm0.7}$
lao_Laoo	$28.4_{\pm2.2}$	$30.2_{\pm0.8}$	$54.8_{\pm2.0}$	$58.4_{\pm1.5}$	$51.0_{\pm0.8}$	$51.6_{\pm1.0}$
lin_Latn	$33.2_{\pm1.3}$	$28.4_{\pm0.9}$	$53.8_{\pm1.6}$	$57.0_{\pm0.8}$	$52.1_{\pm0.6}$	$57.3_{\pm1.1}$
lit_Latn	$49.3_{\pm5.3}$	$31.2_{\pm1.6}$	$68.7_{\pm1.3}$	$72.3_{\pm0.5}$	$62.1_{\pm0.8}$	$68.1_{\pm0.9}$
lug_Latn	$31.1_{\pm1.6}$	$28.3_{\pm0.6}$	$44.6_{\pm1.3}$	$47.2_{\pm0.3}$	$42.6_{\pm1.0}$	$45.6_{\pm0.9}$
luo_Latn	$31.6_{\pm1.8}$	$28.6_{\pm0.6}$	$45.2_{\pm0.6}$	$45.4_{\pm0.9}$	$37.6_{\pm0.8}$	$42.6_{\pm0.2}$
lvs_Latn	$46.7_{\pm3.9}$	$29.6_{\pm1.6}$	$68.3_{\pm0.4}$	$70.6_{\pm1.3}$	$59.0_{\pm1.2}$	$68.6_{\pm2.1}$

Table 9: **Belebele (1/2).** We benchmark models on test portions of Belebele (cf. §4). S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on source-language (S-DEV) validation splits. Subscripts denote std. deviation. Metric: accuracy.

	Zero-Shot Cross-Lingual Transfer				Translate-Test	
	LLM2Vec	NLLB-Encoder	NLLB-LLM2Vec S1+FT	NLLB-LLM2Vec S1+S2	LLM2Vec NLLB-600M	LLM2Vec NLLB-3B
eng_Latn	$92.5_{\pm0.3}$	$80.4_{\pm0.2}$	$90.0_{\pm0.7}$	$91.4_{\pm0.2}$	$92.5_{\pm0.3}$	$92.5_{\pm0.3}$
AVG	$48.2_{\pm3.2}$	$30.4_{\pm0.4}$	$60.2_{\pm0.5}$	$62.6_{\pm0.5}$	$60.7_{\pm0.8}$	$64.2_{\pm0.7}$
mal_Mlym	$28.7_{\pm0.4}$	$30.0_{\pm0.9}$	$49.5_{\pm2.4}$	$49.9_{\pm0.4}$	$66.9_{\pm1.2}$	$65.1_{\pm0.8}$
mar_Deva	$42.8_{\pm5.7}$	$32.4_{\pm1.5}$	$59.1_{\pm1.7}$	$60.8_{\pm0.5}$	$64.3_{\pm0.8}$	$63.7_{\pm0.7}$
mkd_Cyrl	$64.4_{\pm6.7}$	$29.3_{\pm1.1}$	$72.6_{\pm0.3}$	$74.3_{\pm0.6}$	$68.4_{\pm0.9}$	$72.6_{\pm0.6}$
mlt_Latn	$41.1_{\pm5.8}$	$30.0_{\pm0.6}$	$62.8_{\pm0.6}$	$67.7_{\pm1.7}$	$67.9_{\pm0.4}$	$67.9_{\pm1.2}$
mri_Latn	$31.7_{\pm1.6}$	$26.8_{\pm1.0}$	$46.2_{\pm0.7}$	$47.2_{\pm1.4}$	$49.3_{\pm1.2}$	$52.1_{\pm0.3}$
mya_Mymr	$28.1_{\pm0.9}$	$28.9_{\pm0.4}$	$44.8_{\pm0.8}$	$47.4_{\pm0.5}$	$41.8_{\pm0.8}$	$46.1_{\pm0.8}$
nld_Latn	$79.1_{\pm5.2}$	$31.1_{\pm0.8}$	$78.1_{\pm1.4}$	$81.0_{\pm0.6}$	$74.8_{\pm0.3}$	$78.6_{\pm0.9}$
nob_Latn	$73.7_{\pm7.5}$	$32.5_{\pm0.3}$	$81.7_{\pm1.0}$	$84.6_{\pm0.2}$	$75.8_{\pm0.5}$	$79.5_{\pm1.1}$
npi_Deva	$42.1_{\pm2.8}$	$28.1_{\pm0.6}$	$59.4_{\pm1.7}$	$60.8_{\pm1.6}$	$55.8_{\pm0.4}$	$56.8_{\pm1.8}$
nso_Latn	$32.3_{\pm0.7}$	$29.3_{\pm2.2}$	$57.6_{\pm0.1}$	$60.0_{\pm1.5}$	$60.0_{\pm1.4}$	$63.6_{\pm1.6}$
nya_Latn	$30.6_{\pm1.2}$	$27.0_{\pm0.6}$	$52.2_{\pm1.0}$	$54.5_{\pm1.4}$	$48.4_{\pm1.4}$	$53.2_{\pm0.4}$
ory_Orya	$27.7_{\pm1.7}$	$30.4_{\pm0.9}$	$56.0_{\pm1.2}$	$57.1_{\pm1.2}$	$66.5_{\pm1.3}$	$71.3_{\pm0.8}$
pan_Guru	$28.6_{\pm0.7}$	$29.9_{\pm0.6}$	$55.4_{\pm2.5}$	$56.5_{\pm2.1}$	$64.8_{\pm1.1}$	$66.0_{\pm1.1}$
pbt_Arab	$35.7_{\pm4.3}$	$30.3_{\pm0.8}$	$48.9_{\pm1.1}$	$49.9_{\pm1.4}$	$60.3_{\pm0.6}$	$61.9_{\pm1.5}$
pes_Arab	$71.3_{\pm4.7}$	$31.6_{\pm1.1}$	$69.9_{\pm1.0}$	$71.4_{\pm0.3}$	$67.4_{\pm0.4}$	$70.7_{\pm0.8}$
plt_Latn	$34.4_{\pm1.0}$	$29.0_{\pm0.5}$	$61.9_{\pm1.2}$	$64.9_{\pm0.7}$	$63.9_{\pm0.8}$	$66.9_{\pm1.5}$
pol_Latn	$69.4_{\pm7.0}$	$30.8_{\pm1.0}$	$67.6_{\pm0.9}$	$71.0_{\pm0.8}$	$69.3_{\pm1.5}$	$75.4_{\pm1.4}$
por_Latn	$87.0_{\pm1.5}$	$32.9_{\pm1.0}$	$82.1_{\pm1.7}$	$84.1_{\pm0.7}$	$77.7_{\pm1.1}$	$79.7_{\pm1.5}$
ron_Latn	$74.7_{\pm7.5}$	$31.4_{\pm1.2}$	$76.2_{\pm1.1}$	$79.1_{\pm1.0}$	$72.1_{\pm1.7}$	$76.1_{\pm0.7}$
rus_Cyrl	$85.9_{\pm1.9}$	$32.6_{\pm0.3}$	$75.4_{\pm0.3}$	$79.1_{\pm1.2}$	$71.8_{\pm1.2}$	$80.7_{\pm0.6}$
shn_Mymr	$26.7_{\pm0.7}$	$25.9_{\pm1.2}$	$34.9_{\pm2.7}$	$37.3_{\pm0.5}$	$34.8_{\pm1.4}$	$36.1_{\pm0.6}$
sin_Sinh	$29.6_{\pm0.3}$	$28.1_{\pm1.1}$	$42.0_{\pm1.8}$	$43.8_{\pm1.6}$	$55.9_{\pm0.5}$	$58.7_{\pm1.4}$
slk_Latn	$62.5_{\pm7.8}$	$31.7_{\pm0.5}$	$72.4_{\pm1.1}$	$76.0_{\pm0.8}$	$69.2_{\pm1.2}$	$75.1_{\pm0.6}$
slv_Latn	$54.4_{\pm7.4}$	$31.4_{\pm0.6}$	$72.7_{\pm1.3}$	$76.0_{\pm0.6}$	$68.3_{\pm1.4}$	$75.0_{\pm1.7}$
sna_Latn	$34.4_{\pm1.6}$	$29.5_{\pm1.1}$	$59.6_{\pm1.1}$	$61.1_{\pm0.9}$	$55.0_{\pm0.3}$	$61.3_{\pm0.4}$
snd_Arab	$36.4_{\pm1.1}$	$30.1_{\pm1.7}$	$54.5_{\pm0.7}$	$56.6_{\pm0.3}$	$63.8_{\pm0.6}$	$60.3_{\pm1.5}$
som_Latn	$31.2_{\pm0.8}$	$26.8_{\pm0.9}$	$48.4_{\pm0.9}$	$51.2_{\pm1.1}$	$54.3_{\pm0.7}$	$55.4_{\pm0.9}$
sot_Latn	$31.4_{\pm1.6}$	$27.4_{\pm1.4}$	$54.8_{\pm1.3}$	$57.9_{\pm1.4}$	$61.0_{\pm0.4}$	$64.4_{\pm1.2}$
spa_Latn	$87.4_{\pm1.7}$	$30.8_{\pm0.9}$	$82.3_{\pm1.8}$	$85.6_{\pm0.4}$	$74.0_{\pm0.7}$	$79.5_{\pm1.1}$
srp_Cyrl	$65.4_{\pm0.3}$	$30.9_{\pm1.0}$	$72.9_{\pm1.2}$	$75.5_{\pm1.0}$	$66.7_{\pm1.4}$	$69.4_{\pm1.4}$
ssw_Latn	$31.5_{\pm1.4}$	$29.3_{\pm1.0}$	$50.2_{\pm0.8}$	$52.4_{\pm2.1}$	$50.9_{\pm1.4}$	$57.9_{\pm1.2}$
sun_Latn	$39.6_{\pm2.7}$	$30.2_{\pm1.5}$	$66.3_{\pm0.8}$	$70.1_{\pm0.6}$	$58.7_{\pm1.4}$	$61.1_{\pm1.3}$
swe_Latn	$76.9_{\pm6.2}$	$31.4_{\pm1.3}$	$79.7_{\pm1.1}$	$82.8_{\pm0.4}$	$71.7_{\pm1.3}$	$78.1_{\pm0.3}$
swh_Latn	$42.5_{\pm2.5}$	$30.4_{\pm1.1}$	$73.4_{\pm0.7}$	$75.5_{\pm0.9}$	$69.0_{\pm1.6}$	$75.7_{\pm1.1}$
tam_Taml	$28.6_{\pm0.7}$	$32.2_{\pm0.5}$	$51.7_{\pm2.0}$	$54.8_{\pm0.9}$	$62.0_{\pm1.4}$	$60.5_{\pm0.2}$
tel_Telu	$28.2_{\pm1.3}$	$30.6_{\pm1.1}$	$52.0_{\pm0.9}$	$54.6_{\pm1.0}$	$59.6_{\pm1.0}$	$61.6_{\pm1.0}$
tgk_Cyrl	$40.6_{\pm3.6}$	$29.2_{\pm1.0}$	$53.5_{\pm0.7}$	$54.7_{\pm0.6}$	$58.0_{\pm1.5}$	$63.9_{\pm1.1}$
tgl_Latn	$52.7_{\pm8.5}$	$29.1_{\pm0.4}$	$68.9_{\pm1.3}$	$72.2_{\pm1.3}$	$71.4_{\pm1.7}$	$71.1_{\pm1.4}$
tha_Thai	$69.4_{\pm5.0}$	$31.5_{\pm1.1}$	$56.3_{\pm1.3}$	$58.8_{\pm0.6}$	$58.2_{\pm1.5}$	$61.7_{\pm0.4}$
tir_Ethi	$27.1_{\pm1.7}$	$28.7_{\pm0.7}$	$42.4_{\pm1.0}$	$44.6_{\pm1.8}$	$42.1_{\pm2.7}$	$47.7_{\pm0.9}$
tsn_Latn	$33.0_{\pm1.2}$	$27.7_{\pm1.7}$	$54.3_{\pm0.3}$	$56.6_{\pm0.6}$	$54.6_{\pm1.8}$	$61.1_{\pm1.7}$
tso_Latn	$36.0_{\pm0.5}$	$29.1_{\pm0.3}$	$62.6_{\pm0.7}$	$64.5_{\pm1.7}$	$55.8_{\pm2.6}$	$64.3_{\pm0.8}$
tur_Latn	$66.7_{\pm8.6}$	$30.9_{\pm1.0}$	$61.3_{\pm1.0}$	$64.0_{\pm1.1}$	$68.6_{\pm1.5}$	$75.7_{\pm1.2}$
ukr_Cyrl	$77.9_{\pm3.1}$	$30.9_{\pm0.9}$	$71.5_{\pm0.6}$	$74.1_{\pm0.8}$	$70.6_{\pm0.8}$	$73.4_{\pm0.8}$
urd_Arab	$44.4_{\pm6.3}$	$31.4_{\pm0.8}$	$62.1_{\pm1.2}$	$63.6_{\pm1.6}$	$66.0_{\pm0.5}$	$67.4_{\pm1.1}$
uzn_Latn	$44.9_{\pm5.6}$	$30.0_{\pm0.9}$	$58.6_{\pm0.7}$	$61.0_{\pm0.8}$	$70.0_{\pm1.1}$	$75.0_{\pm1.7}$
vie_Latn	$82.1_{\pm1.1}$	$28.2_{\pm0.5}$	$72.4_{\pm1.9}$	$76.4_{\pm0.3}$	$71.7_{\pm0.1}$	$73.4_{\pm0.6}$
war_Latn	$48.2_{\pm3.8}$	$29.8_{\pm0.4}$	$66.6_{\pm0.4}$	$68.6_{\pm1.1}$	$69.4_{\pm1.2}$	$73.0_{\pm0.2}$
wol_Latn	$31.1_{\pm0.6}$	$27.7_{\pm0.6}$	$36.1_{\pm0.9}$	$36.9_{\pm0.7}$	$35.2_{\pm1.2}$	$33.7_{\pm1.3}$
xho_Latn	$32.4_{\pm0.7}$	$28.3_{\pm0.5}$	$59.6_{\pm0.5}$	$61.2_{\pm0.8}$	$59.1_{\pm0.9}$	$64.9_{\pm1.4}$
yor_Latn	$29.4_{\pm0.6}$	$29.1_{\pm1.6}$	$40.6_{\pm1.3}$	$41.4_{\pm0.7}$	$38.8_{\pm1.2}$	$42.0_{\pm0.8}$
zho_Hans	$86.4_{\pm2.3}$	$30.9_{\pm1.6}$	$78.2_{\pm0.6}$	$79.6_{\pm0.9}$	$70.0_{\pm1.1}$	$69.6_{\pm0.9}$
zho_Hant	$85.0_{\pm2.2}$	$33.2_{\pm1.3}$	$75.5_{\pm1.1}$	$76.8_{\pm0.5}$	$51.5_{\pm0.3}$	$57.9_{\pm0.5}$
zsm_Latn	$68.6_{\pm9.2}$	$29.0_{\pm0.5}$	$79.7_{\pm1.4}$	$81.5_{\pm0.8}$	$73.4_{\pm0.4}$	$73.9_{\pm1.1}$
zul_Latn	$31.1_{\pm0.4}$	$30.0_{\pm1.5}$	$54.4_{\pm0.6}$	$58.1_{\pm1.6}$	$56.6_{\pm1.0}$	$62.4_{\pm0.2}$

Table 10: **Belebele (2/2).** We benchmark models on test portions of Belebele (cf. §4). S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on source-language (S-DEV) validation splits. Subscripts denote std. deviation. Metric: accuracy.

Results by No. of Adaptation Steps.

	0	**3000**	**6000**	**10000**
XNLI-EN	$82.6_{\pm 0.6}$	$90.6_{\pm 0.3}$	$91.2_{\pm 0.3}$	$91.4_{\pm 0.2}$
AMNLI-AYM	$60.0_{\pm 0.7}$	$62.8_{\pm 2.4}$	$62k0_{\pm 1.2}$	$61.3_{\pm 0.8}$
AMNLI-GN	$65.7_{\pm 0.4}$	$68.5_{\pm 0.5}$	$69.5_{\pm 0.8}$	$69.7_{\pm 1.3}$
AMNLI-QUY	$60.7_{\pm 1.5}$	$61.8_{\pm 0.8}$	$61.4_{\pm 2.4}$	$61.5_{\pm 1.9}$
KARDEŞ-NLU-AZ	$79.3_{\pm 0.9}$	$84.1_{\pm 0.1}$	$83.7_{\pm 1.1}$	$83.1_{\pm 0.7}$
KARDEŞ-NLU-KK	$77.8_{\pm 0.9}$	$82.1_{\pm 0.4}$	$82.2_{\pm 0.9}$	$81.8_{\pm 0.2}$
KARDEŞ-NLU-KY	$77.9_{\pm 0.6}$	$81.9_{\pm 0.1}$	$81.6_{\pm 0.3}$	$81.4_{\pm 0.6}$
KARDEŞ-NLU-UZ	$78.9_{\pm 0.4}$	$83.3_{\pm 0.3}$	$82.9_{\pm 0.8}$	$82.3_{\pm 0.4}$
XNLI-AR	$75.6_{\pm 0.5}$	$81.1_{\pm 0.5}$	$82.1_{\pm 0.8}$	$82.0_{\pm 0.2}$
XNLI-BG	$79.5_{\pm 0.5}$	$86.0_{\pm 0.2}$	$86.2_{\pm 0.6}$	$86.6_{\pm 0.4}$
XNLI-DE	$79.1_{\pm 0.2}$	$85.6_{\pm 0.4}$	$85.8_{\pm 0.4}$	$85.8_{\pm 0.2}$
XNLI-EL	$78.1_{\pm 0.4}$	$79.1_{\pm 6.1}$	$75.2_{\pm 3.8}$	$79.6_{\pm 6.2}$
XNLI-ES	$79.8_{\pm 0.4}$	$86.9_{\pm 0.3}$	$87.3_{\pm 0.3}$	$87.5_{\pm 0.3}$
XNLI-FR	$78.8_{\pm 0.5}$	$86.2_{\pm 0.4}$	$86.6_{\pm 0.3}$	$86.8_{\pm 0.3}$
XNLI-HI	$72.7_{\pm 0.3}$	$76.4_{\pm 0.5}$	$76.6_{\pm 1.0}$	$76.9_{\pm 0.6}$
XNLI-RU	$76.9_{\pm 0.2}$	$83.4_{\pm 0.7}$	$83.7_{\pm 0.6}$	$83.9_{\pm 0.1}$
XNLI-SW	$73.5_{\pm 0.2}$	$79.5_{\pm 0.4}$	$79.6_{\pm 0.4}$	$79.6_{\pm 0.4}$
XNLI-TH	$71.6_{\pm 0.5}$	$77.0_{\pm 1.0}$	$77.5_{\pm 1.3}$	$77.5_{\pm 0.5}$
XNLI-TR	$76.4_{\pm 0.6}$	$79.6_{\pm 0.3}$	$79.6_{\pm 0.4}$	$80.0_{\pm 0.3}$
XNLI-UR	$68.5_{\pm 0.2}$	$71.8_{\pm 0.5}$	$71.9_{\pm 0.7}$	$72.1_{\pm 0.4}$
XNLI-VI	$77.7_{\pm 0.3}$	$83.0_{\pm 0.4}$	$83.5_{\pm 0.3}$	$83.3_{\pm 0.3}$
XNLI-ZH	$73.6_{\pm 0.2}$	$79.8_{\pm 0.7}$	$80.2_{\pm 0.7}$	$80.2_{\pm 0.7}$
AVG	$74.4_{\pm 0.3}$	$79.1_{\pm 0.2}$	$79.0_{\pm 0.4}$	$79.2_{\pm 0.2}$

Table 11: **NLI by No. of Adaptation Steps.** We benchmark NLLB-LLM2Vec S1+S2 on test portions of NLI benchmarks (cf. §4) by number of adaptation steps in S1. S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on source-language (S-DEV) validation splits. Subscripts denote std. deviation. Metric: accuracy.

	0	**3000**	**6000**	**10000**
ENG	$86.4_{\pm 0.8}$	$91.5_{\pm 1.1}$	$91.7_{\pm 0.5}$	$92.2_{\pm 0.5}$
ACE	$78.3_{\pm 2.3}$	$82.2_{\pm 1.2}$	$81.0_{\pm 2.8}$	$80.6_{\pm 3.8}$
BAN	$75.2_{\pm 1.9}$	$75.0_{\pm 2.2}$	$72.4_{\pm 3.1}$	$72.9_{\pm 3.7}$
BJN	$82.3_{\pm 1.7}$	$82.7_{\pm 0.7}$	$82.7_{\pm 1.3}$	$82.6_{\pm 2.0}$
BUG	$72.7_{\pm 2.3}$	$66.5_{\pm 5.3}$	$62.7_{\pm 5.9}$	$61.1_{\pm 6.9}$
IND	$85.5_{\pm 1.6}$	$88.6_{\pm 0.8}$	$89.3_{\pm 1.1}$	$89.3_{\pm 0.7}$
JAV	$81.1_{\pm 0.7}$	$85.6_{\pm 2.5}$	$84.1_{\pm 2.1}$	$85.3_{\pm 2.4}$
MIN	$76.2_{\pm 3.1}$	$80.5_{\pm 3.3}$	$78.3_{\pm 3.0}$	$80.4_{\pm 3.3}$
SUN	$80.8_{\pm 1.9}$	$83.9_{\pm 2.0}$	$82.6_{\pm 3.1}$	$83.2_{\pm 3.1}$
AVG	$79.0_{\pm 4.2}$	$80.8_{\pm 6.9}$	$79.2_{\pm 8.2}$	$79.4_{\pm 8.8}$

Table 12: **NusaX by No. of Adaptation Steps.** We benchmark NLLB-LLM2Vec S1+S2 on test portions of NusaX (cf. §4) by number of adaptation steps in S1. S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on source-language (S-DEV) validation splits. Subscripts denote std. deviation. Metric: accuracy.

	0	3000	6000	10000
eng_Latn	$74.9_{\pm 5.5}$	$90.0_{\pm 0.8}$	$91.7_{\pm 0.4}$	$99.4_{\pm 0.7}$
AVG	$51.3_{\pm 2.8}$	$61.8_{\pm 0.2}$	$62.2_{\pm 0.2}$	$62.6_{\pm 0.5}$
acm_Arab	$45.4_{\pm 1.6}$	$54.0_{\pm 0.2}$	$55.8_{\pm 1.3}$	$56.7_{\pm 0.6}$
afr_Latn	$62.2_{\pm 4.7}$	$77.3_{\pm 1.1}$	$79.7_{\pm 1.2}$	$80.1_{\pm 1.1}$
als_Latn	$54.3_{\pm 5.3}$	$71.0_{\pm 0.9}$	$71.7_{\pm 0.4}$	$72.9_{\pm 1.0}$
amh_Ethi	$43.8_{\pm 3.3}$	$51.1_{\pm 2.1}$	$49.9_{\pm 0.5}$	$50.4_{\pm 1.5}$
apc_Arab	$47.5_{\pm 3.5}$	$59.9_{\pm 0.9}$	$60.3_{\pm 0.4}$	$60.7_{\pm 0.6}$
arb_Arab	$53.3_{\pm 3.4}$	$65.3_{\pm 1.2}$	$67.9_{\pm 1.1}$	$68.1_{\pm 1.5}$
ars_Arab	$48.0_{\pm 1.9}$	$59.1_{\pm 0.2}$	$59.9_{\pm 1.5}$	$59.6_{\pm 1.1}$
ary_Arab	$40.2_{\pm 1.6}$	$49.1_{\pm 1.1}$	$48.1_{\pm 0.6}$	$48.3_{\pm 0.8}$
arz_Arab	$46.4_{\pm 1.9}$	$58.1_{\pm 1.3}$	$60.2_{\pm 1.6}$	$59.7_{\pm 0.8}$
asm_Beng	$45.6_{\pm 1.2}$	$52.6_{\pm 0.5}$	$53.0_{\pm 0.2}$	$54.0_{\pm 1.1}$
azj_Latn	$46.6_{\pm 1.5}$	$55.1_{\pm 0.1}$	$55.0_{\pm 0.4}$	$56.7_{\pm 0.8}$
bam_Latn	$38.6_{\pm 0.4}$	$40.3_{\pm 0.6}$	$39.3_{\pm 0.8}$	$39.5_{\pm 1.2}$
ben_Beng	$52.0_{\pm 2.3}$	$62.5_{\pm 1.3}$	$63.0_{\pm 1.0}$	$62.5_{\pm 0.8}$
ben_Latn	$30.6_{\pm 0.8}$	$31.7_{\pm 1.1}$	$27.6_{\pm 1.7}$	$28.2_{\pm 1.2}$
bod_Tibt	$30.5_{\pm 0.1}$	$34.5_{\pm 0.3}$	$33.5_{\pm 0.7}$	$33.9_{\pm 1.1}$
bul_Cyrl	$61.0_{\pm 3.8}$	$76.5_{\pm 0.3}$	$76.3_{\pm 0.4}$	$77.9_{\pm 0.1}$
cat_Latn	$63.7_{\pm 3.4}$	$78.9_{\pm 0.6}$	$80.9_{\pm 0.5}$	$82.3_{\pm 0.9}$
ceb_Latn	$55.4_{\pm 2.8}$	$67.9_{\pm 0.9}$	$69.3_{\pm 1.4}$	$70.7_{\pm 0.7}$
ces_Latn	$61.1_{\pm 3.7}$	$74.1_{\pm 0.6}$	$75.5_{\pm 0.4}$	$75.9_{\pm 1.5}$
ckb_Arab	$48.9_{\pm 0.8}$	$58.4_{\pm 1.0}$	$58.7_{\pm 1.0}$	$59.8_{\pm 1.2}$
dan_Latn	$66.2_{\pm 4.0}$	$80.8_{\pm 1.4}$	$82.8_{\pm 0.4}$	$83.5_{\pm 0.7}$
deu_Latn	$64.4_{\pm 4.7}$	$78.4_{\pm 0.7}$	$79.1_{\pm 0.0}$	$78.1_{\pm 0.9}$
ell_Grek	$51.7_{\pm 3.2}$	$64.6_{\pm 1.2}$	$67.3_{\pm 0.8}$	$67.5_{\pm 0.2}$
est_Latn	$54.4_{\pm 4.5}$	$67.5_{\pm 0.4}$	$68.9_{\pm 0.2}$	$70.1_{\pm 1.0}$
eus_Latn	$57.0_{\pm 2.0}$	$66.7_{\pm 0.9}$	$67.3_{\pm 0.6}$	$66.7_{\pm 1.1}$
fin_Latn	$58.5_{\pm 4.3}$	$72.1_{\pm 0.5}$	$72.0_{\pm 0.6}$	$73.0_{\pm 0.7}$
fra_Latn	$66.0_{\pm 4.3}$	$81.7_{\pm 0.7}$	$82.4_{\pm 0.7}$	$82.4_{\pm 1.0}$
fuv_Latn	$29.3_{\pm 0.4}$	$29.7_{\pm 0.2}$	$28.6_{\pm 0.9}$	$28.1_{\pm 0.6}$
gaz_Latn	$39.0_{\pm 1.1}$	$43.6_{\pm 1.5}$	$41.2_{\pm 1.5}$	$42.8_{\pm 0.7}$
grn_Latn	$46.3_{\pm 1.1}$	$54.7_{\pm 0.3}$	$54.2_{\pm 1.6}$	$52.5_{\pm 0.9}$
guj_Gujr	$47.0_{\pm 2.1}$	$55.3_{\pm 0.6}$	$55.9_{\pm 0.8}$	$55.9_{\pm 1.7}$
hat_Latn	$53.5_{\pm 2.5}$	$67.1_{\pm 1.0}$	$67.8_{\pm 0.9}$	$67.4_{\pm 0.9}$
hau_Latn	$48.6_{\pm 2.9}$	$60.8_{\pm 1.5}$	$61.4_{\pm 0.4}$	$62.1_{\pm 0.4}$
heb_Hebr	$54.3_{\pm 2.9}$	$65.6_{\pm 1.3}$	$65.4_{\pm 0.6}$	$66.6_{\pm 0.3}$
hin_Deva	$52.7_{\pm 2.2}$	$63.7_{\pm 0.8}$	$63.9_{\pm 1.6}$	$65.7_{\pm 1.0}$
hrv_Latn	$59.4_{\pm 4.9}$	$75.0_{\pm 1.1}$	$76.6_{\pm 0.3}$	$77.4_{\pm 0.5}$
hun_Latn	$57.2_{\pm 4.4}$	$71.5_{\pm 0.3}$	$71.6_{\pm 0.4}$	$71.6_{\pm 0.5}$
hye_Armn	$48.6_{\pm 2.8}$	$58.8_{\pm 0.7}$	$57.8_{\pm 1.0}$	$58.6_{\pm 0.6}$
ibo_Latn	$40.7_{\pm 2.7}$	$50.6_{\pm 0.8}$	$49.0_{\pm 0.3}$	$49.2_{\pm 0.3}$
ilo_Latn	$53.4_{\pm 2.2}$	$64.5_{\pm 1.1}$	$65.1_{\pm 0.8}$	$66.7_{\pm 1.3}$
ind_Latn	$66.4_{\pm 4.4}$	$81.2_{\pm 0.3}$	$81.7_{\pm 1.1}$	$82.5_{\pm 0.6}$
isl_Latn	$50.7_{\pm 3.1}$	$63.4_{\pm 0.6}$	$65.1_{\pm 0.8}$	$65.0_{\pm 1.2}$
ita_Latn	$64.9_{\pm 4.7}$	$80.3_{\pm 1.1}$	$81.4_{\pm 0.9}$	$82.4_{\pm 0.5}$
jav_Latn	$60.7_{\pm 3.7}$	$73.0_{\pm 0.1}$	$73.4_{\pm 0.4}$	$74.2_{\pm 0.5}$
jpn_Jpan	$53.5_{\pm 3.8}$	$68.0_{\pm 0.2}$	$67.6_{\pm 1.2}$	$67.1_{\pm 0.9}$
kac_Latn	$37.4_{\pm 0.6}$	$39.4_{\pm 1.2}$	$40.2_{\pm 1.1}$	$40.9_{\pm 1.2}$
kan_Knda	$49.9_{\pm 3.6}$	$56.4_{\pm 0.9}$	$56.5_{\pm 1.1}$	$56.9_{\pm 1.3}$
kat_Geor	$44.4_{\pm 2.4}$	$51.2_{\pm 0.7}$	$50.3_{\pm 0.9}$	$51.1_{\pm 0.5}$
kaz_Cyrl	$50.4_{\pm 3.2}$	$59.4_{\pm 0.4}$	$59.6_{\pm 0.9}$	$59.3_{\pm 0.7}$
kea_Latn	$52.1_{\pm 3.8}$	$63.3_{\pm 0.6}$	$64.5_{\pm 0.6}$	$65.6_{\pm 1.5}$
khk_Cyrl	$39.8_{\pm 2.2}$	$45.5_{\pm 0.8}$	$44.8_{\pm 1.3}$	$44.8_{\pm 1.2}$
khm_Khmr	$40.1_{\pm 0.5}$	$51.3_{\pm 0.8}$	$48.8_{\pm 1.5}$	$51.6_{\pm 0.8}$
kin_Latn	$48.0_{\pm 2.6}$	$55.9_{\pm 0.9}$	$57.0_{\pm 0.3}$	$57.1_{\pm 0.8}$
kir_Cyrl	$51.1_{\pm 2.8}$	$57.4_{\pm 1.1}$	$58.9_{\pm 1.2}$	$58.7_{\pm 0.8}$
kor_Hang	$52.3_{\pm 3.8}$	$63.7_{\pm 0.6}$	$63.0_{\pm 0.9}$	$62.3_{\pm 0.7}$
lao_Laoo	$46.6_{\pm 2.9}$	$58.5_{\pm 0.5}$	$58.6_{\pm 0.5}$	$58.4_{\pm 1.5}$
lin_Latn	$47.4_{\pm 3.6}$	$56.4_{\pm 0.6}$	$56.6_{\pm 0.4}$	$57.0_{\pm 0.8}$
lit_Latn	$58.0_{\pm 4.8}$	$69.7_{\pm 1.1}$	$70.1_{\pm 0.4}$	$72.3_{\pm 0.5}$
lug_Latn	$40.4_{\pm 2.1}$	$46.1_{\pm 0.1}$	$46.0_{\pm 1.2}$	$47.2_{\pm 0.3}$
luo_Latn	$39.4_{\pm 2.5}$	$46.5_{\pm 0.6}$	$46.9_{\pm 0.1}$	$45.4_{\pm 0.9}$
lvs_Latn	$57.3_{\pm 3.7}$	$69.3_{\pm 0.5}$	$69.9_{\pm 0.2}$	$70.6_{\pm 1.3}$

Table 13: **Belebele by No. of Adaptation Steps (1/2).** We benchmark NLLB-LLM2Vec S1+S2 on test portions of Belebele (cf. §4) by number of adaptation steps in S1. S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on source-language (S-DEV) validation splits. Subscripts denote std. deviation. Metric: accuracy.

	0	**3000**	**6000**	**10000**
eng_Latn	$74.9_{\pm 5.5}$	$90.0_{\pm 0.8}$	$91.7_{\pm 0.4}$	$99.4_{\pm 0.7}$
AVG	$51.3_{\pm 2.8}$	$61.8_{\pm 0.2}$	$62.2_{\pm 0.2}$	$62.6_{\pm 0.5}$
mal_Mlym	$44.0_{\pm 2.9}$	$52.1_{\pm 1.1}$	$50.6_{\pm 0.9}$	$49.9_{\pm 0.4}$
mar_Deva	$51.7_{\pm 3.7}$	$60.6_{\pm 0.9}$	$60.3_{\pm 1.4}$	$60.8_{\pm 0.5}$
mkd_Cyrl	$59.4_{\pm 3.1}$	$72.2_{\pm 1.2}$	$73.4_{\pm 0.5}$	$74.3_{\pm 0.6}$
mlt_Latn	$52.1_{\pm 1.5}$	$65.7_{\pm 0.7}$	$66.5_{\pm 0.5}$	$67.7_{\pm 1.7}$
mri_Latn	$41.2_{\pm 2.5}$	$45.8_{\pm 0.3}$	$46.7_{\pm 0.6}$	$47.2_{\pm 1.4}$
mya_Mymr	$40.6_{\pm 1.8}$	$47.3_{\pm 0.6}$	$46.6_{\pm 0.7}$	$47.4_{\pm 0.5}$
nld_Latn	$64.1_{\pm 3.9}$	$79.1_{\pm 0.4}$	$80.7_{\pm 0.4}$	$81.0_{\pm 0.6}$
nob_Latn	$65.2_{\pm 4.1}$	$81.8_{\pm 1.2}$	$83.9_{\pm 0.7}$	$84.6_{\pm 0.2}$
npi_Deva	$51.5_{\pm 2.8}$	$61.1_{\pm 1.1}$	$61.2_{\pm 0.4}$	$60.8_{\pm 1.6}$
nso_Latn	$49.6_{\pm 4.3}$	$58.0_{\pm 0.8}$	$60.4_{\pm 0.9}$	$60.0_{\pm 1.5}$
nya_Latn	$44.4_{\pm 1.5}$	$52.9_{\pm 1.2}$	$54.5_{\pm 1.4}$	$54.5_{\pm 1.4}$
ory_Orya	$48.2_{\pm 1.4}$	$57.1_{\pm 0.8}$	$57.0_{\pm 1.1}$	$57.1_{\pm 1.2}$
pan_Guru	$46.6_{\pm 2.0}$	$55.2_{\pm 1.4}$	$55.0_{\pm 1.1}$	$56.5_{\pm 2.1}$
pbt_Arab	$44.2_{\pm 2.3}$	$50.6_{\pm 1.3}$	$48.7_{\pm 0.8}$	$49.9_{\pm 1.4}$
pes_Arab	$60.1_{\pm 5.5}$	$72.4_{\pm 0.5}$	$71.6_{\pm 0.4}$	$71.4_{\pm 0.3}$
plt_Latn	$52.3_{\pm 2.8}$	$64.9_{\pm 0.7}$	$64.1_{\pm 0.5}$	$64.9_{\pm 0.7}$
pol_Latn	$55.7_{\pm 3.9}$	$69.0_{\pm 0.4}$	$70.7_{\pm 0.8}$	$71.0_{\pm 0.8}$
por_Latn	$68.2_{\pm 4.8}$	$83.2_{\pm 0.5}$	$83.8_{\pm 0.7}$	$84.1_{\pm 0.7}$
ron_Latn	$62.1_{\pm 4.3}$	$77.3_{\pm 0.9}$	$78.0_{\pm 1.0}$	$79.1_{\pm 1.0}$
rus_Cyrl	$61.6_{\pm 3.6}$	$76.2_{\pm 0.6}$	$78.1_{\pm 1.0}$	$79.1_{\pm 1.2}$
shn_Mymr	$33.3_{\pm 1.4}$	$36.4_{\pm 1.6}$	$36.4_{\pm 0.8}$	$37.3_{\pm 0.5}$
sin_Sinh	$38.4_{\pm 2.3}$	$43.9_{\pm 1.3}$	$44.1_{\pm 1.7}$	$43.8_{\pm 1.6}$
slk_Latn	$60.5_{\pm 4.6}$	$73.3_{\pm 0.2}$	$74.9_{\pm 0.3}$	$76.0_{\pm 0.8}$
slv_Latn	$59.9_{\pm 4.2}$	$74.9_{\pm 0.7}$	$76.2_{\pm 0.3}$	$76.0_{\pm 0.6}$
sna_Latn	$48.8_{\pm 2.2}$	$61.1_{\pm 1.0}$	$61.4_{\pm 0.4}$	$61.1_{\pm 0.9}$
snd_Arab	$48.6_{\pm 2.4}$	$57.6_{\pm 1.4}$	$57.1_{\pm 1.2}$	$56.6_{\pm 0.3}$
som_Latn	$43.3_{\pm 2.1}$	$52.0_{\pm 1.5}$	$51.4_{\pm 0.9}$	$51.2_{\pm 1.1}$
sot_Latn	$45.6_{\pm 3.7}$	$56.6_{\pm 0.1}$	$57.4_{\pm 1.4}$	$57.9_{\pm 1.4}$
spa_Latn	$67.0_{\pm 3.6}$	$84.6_{\pm 0.8}$	$85.3_{\pm 0.2}$	$85.6_{\pm 0.4}$
srp_Cyrl	$60.0_{\pm 3.9}$	$72.8_{\pm 1.0}$	$74.4_{\pm 0.8}$	$75.5_{\pm 1.0}$
ssw_Latn	$44.2_{\pm 3.9}$	$51.4_{\pm 0.2}$	$51.7_{\pm 0.2}$	$52.4_{\pm 2.1}$
sun_Latn	$54.8_{\pm 3.2}$	$68.8_{\pm 0.9}$	$68.9_{\pm 0.4}$	$70.1_{\pm 0.6}$
swe_Latn	$66.6_{\pm 4.6}$	$81.3_{\pm 0.7}$	$81.9_{\pm 0.1}$	$82.8_{\pm 0.4}$
swh_Latn	$60.3_{\pm 4.2}$	$74.9_{\pm 0.4}$	$74.6_{\pm 0.5}$	$75.5_{\pm 0.9}$
tam_Taml	$47.2_{\pm 3.9}$	$55.7_{\pm 1.7}$	$55.6_{\pm 0.9}$	$54.8_{\pm 0.9}$
tel_Telu	$47.0_{\pm 2.6}$	$56.0_{\pm 0.6}$	$54.3_{\pm 1.2}$	$54.6_{\pm 1.0}$
tgk_Cyrl	$45.6_{\pm 3.1}$	$55.6_{\pm 0.2}$	$55.4_{\pm 0.5}$	$54.7_{\pm 0.6}$
tgl_Latn	$58.0_{\pm 3.1}$	$71.8_{\pm 0.3}$	$72.1_{\pm 0.7}$	$72.2_{\pm 1.3}$
tha_Thai	$46.5_{\pm 1.9}$	$58.0_{\pm 0.6}$	$58.1_{\pm 0.8}$	$58.8_{\pm 0.6}$
tir_Ethi	$40.1_{\pm 2.0}$	$44.0_{\pm 0.7}$	$44.9_{\pm 1.5}$	$44.6_{\pm 1.8}$
tsn_Latn	$48.3_{\pm 3.8}$	$55.9_{\pm 1.0}$	$55.4_{\pm 0.9}$	$56.6_{\pm 0.6}$
tso_Latn	$54.9_{\pm 1.8}$	$64.3_{\pm 0.6}$	$63.7_{\pm 0.2}$	$64.5_{\pm 1.7}$
tur_Latn	$55.3_{\pm 3.2}$	$62.4_{\pm 0.7}$	$63.7_{\pm 0.8}$	$64.0_{\pm 1.1}$
ukr_Cyrl	$57.0_{\pm 3.1}$	$72.3_{\pm 0.2}$	$72.9_{\pm 1.1}$	$74.1_{\pm 0.8}$
urd_Arab	$53.8_{\pm 2.1}$	$63.2_{\pm 1.2}$	$63.5_{\pm 0.7}$	$63.6_{\pm 1.6}$
uzn_Latn	$49.8_{\pm 2.3}$	$60.5_{\pm 0.9}$	$61.3_{\pm 0.3}$	$61.0_{\pm 0.8}$
vie_Latn	$60.3_{\pm 3.4}$	$75.5_{\pm 0.2}$	$75.2_{\pm 1.0}$	$76.4_{\pm 0.3}$
war_Latn	$55.9_{\pm 4.0}$	$69.6_{\pm 0.4}$	$69.0_{\pm 1.0}$	$68.6_{\pm 1.1}$
wol_Latn	$34.7_{\pm 1.5}$	$38.6_{\pm 1.5}$	$37.0_{\pm 0.8}$	$36.9_{\pm 0.7}$
xho_Latn	$49.7_{\pm 2.5}$	$59.4_{\pm 1.1}$	$60.6_{\pm 0.1}$	$61.2_{\pm 0.8}$
yor_Latn	$36.1_{\pm 1.7}$	$41.6_{\pm 0.7}$	$41.1_{\pm 0.9}$	$41.4_{\pm 0.7}$
zho_Hans	$65.3_{\pm 3.7}$	$78.4_{\pm 0.7}$	$79.9_{\pm 0.5}$	$79.6_{\pm 0.9}$
zho_Hant	$63.9_{\pm 3.6}$	$77.4_{\pm 0.4}$	$77.6_{\pm 0.7}$	$76.8_{\pm 0.5}$
zsm_Latn	$66.1_{\pm 3.8}$	$80.1_{\pm 1.0}$	$80.7_{\pm 1.5}$	$81.5_{\pm 0.8}$
zul_Latn	$47.1_{\pm 2.3}$	$56.2_{\pm 0.9}$	$56.7_{\pm 0.9}$	$58.1_{\pm 1.6}$

Table 14: **Belebele by No. of Adaptation Steps (2/2).** We benchmark NLLB-LLM2Vec S1+S2 on test portions of Belebele (cf. §4) by number of adaptation steps in S1. S1 and S2 refer to self-supervised and task-specific stages of aligning NLLB with LLM2Vec (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on source-language (S-DEV) validation splits. Subscripts denote std. deviation. Metric: accuracy.

NLLB-GPT-2.

	EN	AR	BG	DE	EL	ES	FR	HI	RU	SW	TH	TR	UR	VI	ZH	AVG
Zero-Shot Cross-Lingual Transfer: Fine-tune multilingual model on English training set																
NLLB-GPT-2 FT	$82.2_{\pm0.1}$	$75.0_{\pm0.4}$	$78.3_{\pm0.6}$	$77.1_{\pm1.0}$	$75.6_{\pm0.5}$	$78.8_{\pm0.4}$	$77.9_{\pm0.8}$	$71.2_{\pm0.5}$	$75.6_{\pm0.4}$	$73.1_{\pm0.6}$	$71.1_{\pm0.8}$	$74.0_{\pm0.8}$	$68.4_{\pm0.9}$	$76.7_{\pm0.4}$	$73.4_{\pm0.6}$	$74.7_{\pm0.5}$
Translate-Test: Translate test data to English																
GPT-2 NLLB-600M	$00.0_{\pm0.0}$	$73.9_{\pm0.1}$	$76.3_{\pm0.3}$	$77.4_{\pm0.1}$	$77.5_{\pm0.6}$	$78.7_{\pm0.2}$	$78.4_{\pm0.5}$	$71.3_{\pm0.4}$	$73.8_{\pm0.2}$	$69.1_{\pm0.4}$	$69.0_{\pm0.3}$	$74.1_{\pm0.5}$	$65.6_{\pm0.1}$	$74.9_{\pm0.3}$	$71.0_{\pm0.5}$	$73.6_{\pm0.1}$
GPT-2 NLLB-3B	$00.0_{\pm0.3}$	$74.9_{\pm0.5}$	$78.1_{\pm0.1}$	$78.9_{\pm0.2}$	$77.9_{\pm0.3}$	$79.8_{\pm0.2}$	$79.4_{\pm0.1}$	$72.6_{\pm0.3}$	$76.0_{\pm0.4}$	$69.9_{\pm0.3}$	$71.0_{\pm0.1}$	$76.6_{\pm0.6}$	$66.0_{\pm0.4}$	$75.9_{\pm0.4}$	$73.7_{\pm1.0}$	$75.1_{\pm0.2}$

Table 15: **NLLB-GPT-2 XNLI (1/2).** We benchmark models on test portions of XNLI (cf. §4). S1 refers to the self-supervised stage of aligning NLLB with GPT-2 (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on **source-language (S-DEV) validation splits**. Subscripts denote std. deviation. Metric: accuracy.

	EN	AR	BG	DE	EL	ES	FR	HI	RU	SW	TH	TR	UR	VI	ZH	AVG
Zero-Shot Cross-Lingual Transfer: Fine-tune multilingual model on English training set																
NLLB-GPT-2 S1+FT	$82.2_{\pm0.1}$	$75.0_{\pm0.4}$	$78.4_{\pm0.5}$	$77.2_{\pm0.9}$	$75.8_{\pm0.6}$	$78.8_{\pm0.4}$	$78.0_{\pm0.8}$	$71.3_{\pm0.5}$	$75.7_{\pm0.4}$	$73.1_{\pm0.6}$	$71.1_{\pm0.8}$	$74.2_{\pm0.7}$	$68.4_{\pm0.9}$	$76.7_{\pm0.3}$	$73.5_{\pm0.4}$	$74.8_{\pm0.5}$
Translate-Test: Translate test data to English																
GPT-2 NLLB-600M	$85.0_{\pm0.1}$	$74.1_{\pm0.2}$	$76.8_{\pm0.6}$	$77.7_{\pm0.1}$	$77.8_{\pm0.3}$	$79.1_{\pm0.3}$	$78.6_{\pm0.5}$	$71.7_{\pm0.2}$	$74.0_{\pm0.1}$	$69.6_{\pm0.2}$	$69.4_{\pm0.2}$	$74.4_{\pm0.4}$	$65.9_{\pm0.4}$	$75.1_{\pm0.3}$	$71.7_{\pm0.6}$	$74.0_{\pm0.1}$
GPT-2 NLLB-3B	$85.0_{\pm0.1}$	$75.1_{\pm0.4}$	$78.6_{\pm0.3}$	$79.1_{\pm0.3}$	$78.3_{\pm0.3}$	$80.1_{\pm0.3}$	$79.6_{\pm0.3}$	$72.9_{\pm0.1}$	$76.4_{\pm0.1}$	$70.5_{\pm0.3}$	$71.4_{\pm0.3}$	$76.7_{\pm0.4}$	$66.4_{\pm0.1}$	$76.1_{\pm0.2}$	$74.4_{\pm0.4}$	$75.4_{\pm0.0}$

Table 16: **NLLB-GPT-2 XNLI (2/2).** We benchmark models on test portions of XNLI (cf. §4). S1 refers to the self-supervised stage of aligning NLLB with GPT-2 (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on **per target-language (T-DEV) validation splits**. Subscripts denote std. deviation. Metric: accuracy.

	AYM		GN		QUY		AVG		AZ		KK		KY		UZ		AVG	
	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV	S-DEV	T-DEV
Zero-Shot Cross-Lingual Transfer: Fine-tune multilingual model on English training set																		
NLLB-GPT-2 S1+FT	$59.7_{\pm0.5}$	$60.4_{\pm0.4}$	$66.8_{\pm0.9}$	$67.8_{\pm0.7}$	$60.0_{\pm0.8}$	$61.2_{\pm1.1}$	$62.2_{\pm0.6}$	$63.2_{\pm0.6}$	$77.4_{\pm0.4}$	$77.7_{\pm0.7}$	$75.3_{\pm0.3}$	$75.5_{\pm0.1}$	$75.6_{\pm0.3}$	$75.7_{\pm0.4}$	$76.6_{\pm0.6}$	$77.1_{\pm0.6}$	76.2±0.2	$76.5_{\pm0.3}$
Translate-Test: Translate test data to English																		
GPT-2 NLLB-600M	$50.3_{\pm1.1}$	$52.1_{\pm1.1}$	$58.0_{\pm0.9}$	$59.0_{\pm0.6}$	$54.0_{\pm0.5}$	$54.4_{\pm0.1}$	$54.1_{\pm0.5}$	$55.2_{\pm0.4}$	$77.2_{\pm0.2}$	$77.4_{\pm0.5}$	$73.2_{\pm0.1}$	$73.7_{\pm0.4}$	$73.5_{\pm0.4}$	$74.3_{\pm0.0}$	$75.0_{\pm0.5}$	$75.7_{\pm0.7}$	$74.7_{\pm0.0}$	$75.3_{\pm0.2}$
GPT-2 NLLB-3B	$46.2_{\pm1.0}$	$47.9_{\pm0.6}$	$59.7_{\pm1.5}$	$61.6_{\pm0.3}$	$51.9_{\pm1.0}$	$54.0_{\pm0.3}$	$52.6_{\pm0.5}$	$54.5_{\pm0.2}$	$78.9_{\pm0.3}$	$79.3_{\pm0.4}$	$75.0_{\pm0.4}$	$75.5_{\pm0.4}$	$71.7_{\pm0.4}$	$72.6_{\pm0.4}$	$75.9_{\pm0.4}$	$76.3_{\pm0.3}$	$75.4_{\pm0.4}$	$75.9_{\pm0.3}$

Table 17: **NLLB-GPT-2 on AmNLI & Kardeş-NLU.** We benchmark models on test portions of AmNLI and Kardeş-NLU (cf. §4). S1 refers to the self-supervised stage of aligning NLLB with GPT-2 (cf. §3). FT denotes supervised fine-tuning. Reported performance is averaged over three seeds on model checkpoints that maximize performance on source-language (S-DEV) and per target-language (T-DEV) validation splits. Subscripts denote std. deviation. Metric: accuracy.