LangSAMP: Language-Script Aware Multilingual Pretraining

Anonymous ACL submission

Abstract

Recent multilingual pretrained language models (mPLMs) often avoid using language embeddings – learnable vectors assigned to individual languages. However, this places a significant burden on token representations to encode all language-specific information, which may hinder language neutrality. To address this limitation, we propose **Lang**uage-**S**cript **A**ware **M**ultilingual **P**retraining (**LangSAMP**), a method that incorporates both **language** and **script** embeddings to enhance representation learning. Specifically, we integrate these embeddings into the output of the Transformer blocks before passing the final representations to the language modeling head for prediction. We apply LangSAMP to the continual pretraining of XLM-R (Conneau et al., 2020) on a highly multilingual corpus covering more than 500 languages. The resulting model consistently outperforms the baseline in zero-shot crosslingual transfer across diverse downstream tasks. Extensive analysis reveals that language and script embeddings capture language- and script-specific nuances, which benefits more language-neutral representations, proved by improved pairwise cosine similarity. In our case study, we also show language and script embeddings can be used to select better source languages for crosslingual transfer. We make our code and models publicly available.[1]

1 Introduction

Encoder-only mPLMs are often regarded as universal text encoders (Cer et al., 2018; Huang et al., 2019; Yang et al., 2020), where the sentence-level or token-level representations are applied to various downstream tasks across different languages (Wei et al., 2021). One of the most attractive aspects of these representations is their utility in crosslingual transfer (Zoph et al., 2016; Wu and Dredze, 2019; Artetxe et al., 2020a). That is, representations from

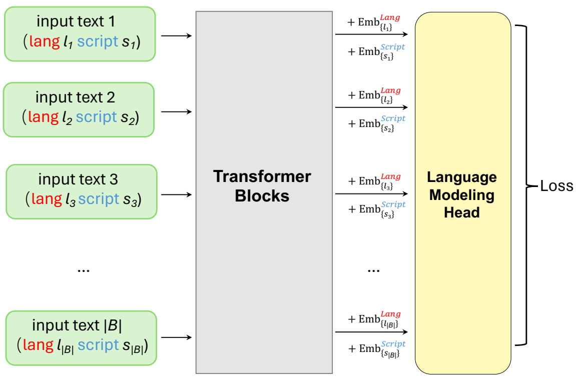

Figure 1: An illustration of LangSAMP for a single batch. Each text may come from different languages and different scripts. Language and script embeddings are added to the transformer output before feeding into the language modeling head. This setup improves the language neutrality of the representations as the auxiliary embeddings share the burden by encoding some language- and script-specific information useful for decoding specific tokens in masked language modeling.

a single source language can be used to fine-tune a multilingual task-specific model (e.g., an mPLM + a task-specific classifier). The fine-tuned model can be applied directly to other languages, without further training. Such a pipeline is particularly useful for low-resource languages, where training data is often scarce (Artetxe et al., 2020b).

The effectiveness of this pipeline depends on the transferability of crosslingual representations. However, previous studies have shown that the representations from recent mPLMs encode a lot of language- and script-specific information (Datta et al., 2020; Chang et al., 2022; Wen-Yi and Mimno, 2023). This is generally not advantageous, as language neutrality, i.e., representations from different languages share a unified subspace, is important for effective crosslingual transfer (Libovický et al., 2020; Chang et al., 2022; Hua et al., 2024). While some approaches attempt to post-align these representations (Cao et al., 2020; Pan et al., 2021; Liu et al., 2024b; Xhelili et al., 2024), limited ef-

[1] URL hidden for anonymity

forts have focused on enhancing language neutrality from the architectural perspective of mPLMs during pretraining.

Early mPLMs, such as XLM (Conneau and Lample, 2019) leverage language embeddings – learnable vectors assigned to different languages. These embeddings are added to the token embeddings before being fed into the transformer (Vaswani et al., 2017) blocks, aiming to alleviate the burden of encoding language-specific information within the token embeddings. Language embeddings can also guide generation toward the correct target language in machine translation (Conneau and Lample, 2019; Song et al., 2019; Liu et al., 2022). However, more recent mPLMs, such as XLM-R (Conneau et al., 2020) and mBERT (Devlin et al., 2019), have discarded these embeddings. The two primary reasons are that (1) mPLMs are expected to have a single, unified parameter set for all languages, and (2) they need to function seamlessly as universal text encoders without requiring language IDs as input. However, the removal inevitably reduces the language neutrality of token embeddings and representations (contextual token embeddings), which may negatively impact crosslingual transfer.

To address this limitation, this work proposes **Lang**uage-**S**cript **A**ware **M**ultilingual **P**retraining (**LANGSAMP**), a method that incorporates both **language** and **script** embeddings to facilitate better representation learning. Instead of adding these embeddings to the token embeddings before feeding them into the transformer blocks, we add them to the output of the transformer blocks (final contextual token embeddings) **before feeding them into the language modeling head**, as shown in Figure 1. In the pretraining phase, language and script IDs are required to obtain language and script embeddings, offloading the burden and helping decode specific tokens in masked language modeling. After pretraining, the backbone (token embeddings and transformer blocks) can function seamlessly as a universal text encoder, which can be fine-tuned together with a task-specific classifier for downstream tasks, without any language or script IDs as input, which are the same as most recent mPLMs.

To validate our approach, we continually pretrain XLM-R (Conneau et al., 2020) using LANGSAMP on Glot500-c (ImaniGooghari et al., 2023), a multilingual dataset containing over 500 languages. We evaluate the resulting model across a diverse set of downstream tasks, including sentence retrieval, text classification, and sequence labeling, consistently achieving superior performance compared to the baseline. We show better language neutrality is achieved – LANGSAMP improves the pairwise cosine similarity across languages. Additionally, we observe that language and script embeddings encapsulate typological features, making their similarities a useful resource for selecting optimal source languages in crosslingual transfer.

Our main contributions are as follows: (i) We propose LANGSAMP, an effective multilingual pretraining method to improve the language neutrality of representations. (ii) We conduct extensive experiments across a spectrum of downstream tasks, demonstrating that our method consistently improves crosslingual transfer performance. (iii) Our case study shows that language embeddings, as a byproduct, can effectively assist in selecting the optimal source language for crosslingual transfer.

2 Related Work

2.1 Multilingual Pretrained Language Models

Multilingual pretrained language models (mPLMs) are models that are trained on many languages, with one or multiple self-learning objectives, such as masked language modeling (MLM) (Devlin et al., 2019) or causal language modeling (Radford et al., 2019). These models can be generally classified as encoder-only (Devlin et al., 2019; Conneau et al., 2020; Liang et al., 2023), encoder-decoder (Liu et al., 2020; Fan et al., 2021; Xue et al., 2021), and decoder-only models (Lin et al., 2022; Shliazhko et al., 2022; Scao et al., 2022). Decoder-only models that have considerably many parameters and are pretrained on a lot of data are also referred to as large language models (LLMs) (Achiam et al., 2023; Touvron et al., 2023; Üstün et al., 2024), which are good at natural language generation tasks, typically for high- and medium-resource languages. In parallel, some recent encoder-only models attempt to scale *horizontally*, i.e., cover more languages, especially low-resource ones (Ogueji et al., 2021; Alabi et al., 2022; ImaniGooghari et al., 2023; Liu et al., 2024a). These highly multilingual encoder-only models are particularly good at understanding tasks in a zero-shot crosslingual fashion.

2.2 Language Embeddings

Language embeddings are vectors that explicitly or implicitly capture the linguistic characteristics of languages. Early works construct such embeddings using prior knowledge of the languages, resulting in vectors where each dimension encodes a specific linguistic feature (Östling, 2015; Ammar et al., 2016; Littell et al., 2017). However, such features have to be manually defined and may be unavailable for less-studied languages (Yu et al., 2021). Therefore, researchers also explore learning language embeddings directly from parallel corpora (Malaviya et al., 2017; Östling and Tiedemann, 2017; Bjerva and Augenstein, 2018; Tan et al., 2019; Liu et al., 2023; Chen et al., 2023) or monolingual corpora (Conneau and Lample, 2019; Yu et al., 2021). This is usually done by assigning an ID to each language, initializing a fixed-length learnable vector, and integrating the vector into the input from that language. The embeddings can capture linguistic features and help crosslingual tasks, e.g., guiding language-specific generation in machine translation in XLM (Conneau and Lample, 2019). This line of approaches requires language IDs as input for both pretraining and downstream fine-tuning. In contrast, language embeddings are only leveraged in our pretraining. The backbone can be used as a universal text encoder without language IDs for fine-tuning on downstream tasks.

3 Methodology

We present **LANGSAMP**, an approach that incorporates both **language** and **script** embeddings to facilitate learning more language-neutral representations in multilingual pretraining. LANGSAMP preserves the same architecture as the most recent multilingual encoder-only models, except for requiring auxiliary language and script IDs/embeddings in pretraining. In the fine-tuning stage, these auxiliary IDs and embeddings are not required. We introduce the key components in the following.

3.1 Language and Script Embeddings

Language and script embeddings are introduced to share the token representations' burden of encoding language- and script-specific information. Let $\boldsymbol{E}^{Lang} \in \mathbb{R}^{L\times D}$ and $\boldsymbol{E}^{Script} \in \mathbb{R}^{S\times D}$ be the language and script embeddings respectively, where L is the number of languages, S is the number of scripts, and D is the embedding dimension of the model. We use $\boldsymbol{E}_l^{Lang}$ (resp. $\boldsymbol{E}_s^{Script}$) to denote the embedding of a specific language l (resp. script s). Similar to token embeddings (which represent relations between tokens in vector space), the language/script embeddings are also expected to capture structural and typological similarities of languages (§5.2) and be useful for selecting good source language for crosslingual transfer (§5.4).

3.2 Language-Script Aware Modeling

In the standard MLM pretraining, Transformer blocks generate the final representation at a masked position. Subsequently, this representation is fed to the language modeling head to reconstruct the original token. Since the original token is used by a specific language and written in a specific script, language- or script-specific information is particularly necessary to decode this token. From this perspective, the Transformer output used for decoding is not language-neutral by nature. Our intuition is that we can ease the decoding by giving hints (e.g., the token should be generated in a specific language or script) to the language modeling head. In this way, the output of Transformer blocks does not need to encode much language- and script-specific information, and can thus be more language-neutral. Inspired by this, we add language and script embeddings to the output of Transformer blocks and feed the resulting representations to the language modeling head for decoding, as shown in Figure 2.

Formally, let a training instance (an input sentence) be $X = [x_1, x_2, \cdots, x_n]$ that comes from language l and is written in script s. We feed X into Transformer blocks and obtain the final contextualized embeddings from the last layer: $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n]$. We then add the language and script embedding to these outputs to form the final representations: $\boldsymbol{o}_i = \boldsymbol{h_i} + \boldsymbol{E}_l^{Lang} + \boldsymbol{E}_s^{Script}$. The final representations at the masked positions are used to decode the original tokens in MLM:

$$\mathcal{L}_{MLM} = -\sum_{i\in\mathcal{M}} \log P_{MLM}(x_i|\boldsymbol{o_i})$$

where $\mathcal{M}$ is the set of masked positions in X and $P_{MLM}(x_i|\boldsymbol{o_i})$ is the probability of decoding the original token x_i given the final representation $\boldsymbol{o}_i$, which is computed by the language modeling head. Since $\boldsymbol{E}_l^{Lang}$ and $\boldsymbol{E}_s^{Script}$ provide language and script-specific information, we expect that $\boldsymbol{h}_i$ will be more language-neutral (§5.3), which is beneficial to zero-shot crosslingual transfer (§4.3).

3.3 Fine-tuning on Downstream tasks

Since we only leverage language and script embedding in the pretraining for MLM, the core architecture (token embeddings + Transformer blocks) remains the same as most mainstream mPLMs, such

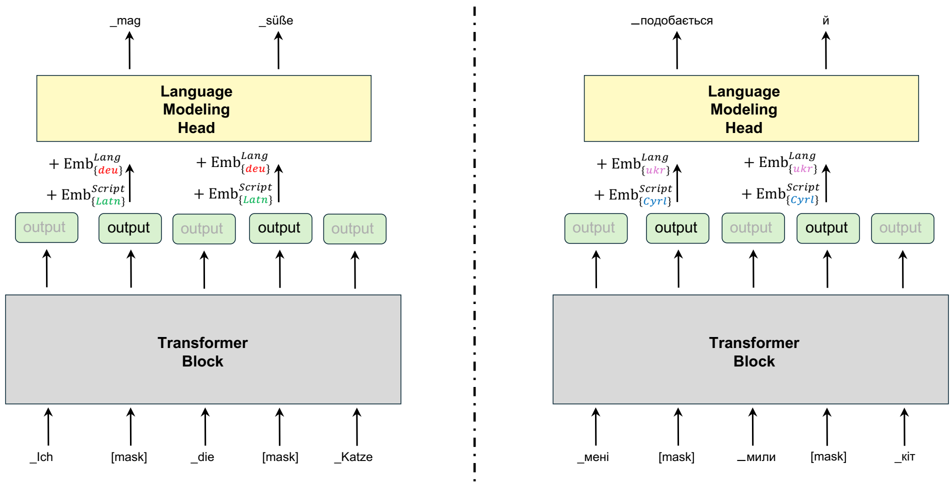

Figure 2: Illustration of LANGSAMP applied to a German sentence (left) and a Ukrainian sentence (right), both meaning "I like the cute cat". Language and script embeddings are added to the outputs from the transformer block. The resulting representation is used to predict the original tokens at the [mask] positions in MLM training.

as XLM-R. In this way, we **do not** need any language or script IDs as input to obtain the Transformer output, i.e., the final contextualized embeddings $\boldsymbol{H}$. This means our pretrained model can be fine-tuned in the standard way in the NLP pipeline. Specifically, for any downstream tasks that require a task-specific classifier (either token-level or sequence-level tasks), we can feed the final contextualized embeddings $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n]$ to the classifier and update the model parameters according to the fine-tuning objective, where language or script embeddings are not participating at all. In addition, as $\boldsymbol{H}$ is more language-neutral thanks to LANGSAMP, we expect the representations to boost zero-shot crosslingual transfer (§4.3).

4 Experiments

4.1 Setups

Training Corpora and Tokenizer We use Glot500-c (ImaniGooghari et al., 2023), a corpus that has monolingual data from more than 500 languages written in 30 different scripts. We treat each language-script as a separate entity and refer to those covered by XLM-R (Conneau et al., 2020) as *head languages* whereas the remaining are *tail languages* (also low-resource languages). We use the tokenizer of Glot500-m (ImaniGooghari et al., 2023) which is a SentencePiece Unigram tokenizer (Kudo and Richardson, 2018; Kudo, 2018) whose vocabulary is merged from the subwords in XLM-R and new subwords learned from Glot500-c.

Continued pretraining We use the weights from XLM-R to initialize our LANGSAMP model for MLM pretraining. **Language and script embeddings are randomly initialized with dimensions $\mathbb{R}^{610\times768}$ and $\mathbb{R}^{30\times768}$ respectively**. We continually train our model on Glot500-c, where we sample data from a multinomial distribution with a temperature of 0.3, to increase the amount of training instances of low- and medium-resource language. We use AdamW optimizer (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) with $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = $ 1e-6. The initial learning rate is set to 5e-5. The effective batch size is 1,024 in each training step where the gradient accumulation is 8 and per-GPU batch size is 32. We train the model on 4 NVIDIA RTX6000 GPUs. Each training instance in a batch contains sentences from **the same language-script** which are concatenated to a chunk of 512 tokens. Each batch contains instances from **different language-scripts**. We store checkpoints every 5K steps and apply early stopping with the best average performance on downstream tasks. We set the maximum steps to 150K. The training takes about 4 weeks.

Baseline To validate LANGSAMP, we create a baseline where language and script embeddings are not used. This baseline can be regarded as a reproduction of Glot500-m (ImaniGooghari et al., 2023). For a fair comparison, the training hyperparameters and training data (100% data of Glot500-c) are the same as LANGSAMP. However, in our ablation study §5.1, due to a constrained computing budget, we cannot continually pretrain model variants on full Glot500-c for validating each component individually (with/without language or script em-

	tail		head		Latn		non-Latn		all	
	Baseline	LANGSAMP	Baseline	LANGSAMP	Baseline	LANGSAMP	Baseline	LANGSAMP	Baseline	LANGSAMP
SR-B	36.9	**39.5**	60.6	**61.3**	40.7	**42.8**	51.2	**53.5**	42.9	**45.1**
SR-T	56.9	**58.6**	74.8	**76.1**	67.5	**68.7**	73.7	**75.6**	69.7	**71.1**
Taxi1500	46.1	**50.9**	59.3	**61.5**	47.3	**51.9**	58.1	**60.3**	49.4	**53.6**
SIB200	69.0	**70.2**	82.2	**82.6**	72.1	**73.1**	81.1	**81.7**	75.0	**75.9**
NER	59.7	**60.5**	**64.2**	**64.2**	66.8	**67.7**	**54.0**	53.6	62.1	**62.5**
POS	**61.9**	61.7	**76.2**	**76.2**	**74.8**	74.4	66.7	**67.2**	**71.8**	71.7

Table 1: Performance of LANGSAMP and baseline on six downstream tasks across five seeds. We report the performance by grouping languages according to two characteristics: (1) whether it is a head or a tail language and (2) whether it is written in Latin script or non-Latin script. LANGSAMP consistently achieves on-par performance or outperforms the baseline across all groups and downstream tasks. **Bold**: best result for each group in each task.

beddings). Instead, we create such variants and pretrain them using a small portion (5%) of Glot500-c. As a result, the baseline model in Table 1 is different from the vanilla model in Table 2.

4.2 Downstream Tasks

We consider the following three evaluation types, with two datasets for each type. The evaluation is done in an English-centric zero-shot crosslingual transfer style for evaluation types that requires fine-tuning. That is, we first fine-tune the pretrained model on the English train set, then select the best checkpoint on the English development set, and finally evaluate the best checkpoint on the test sets of all other languages. For Sentence Retrieval, which does not involve any fine-tuning, we simply use English as the retrieval query language. For all tasks, only a subset of languages (head and tail languages) supported by Glot500-c are considered. We show the detailed information of the used dataset and hyperparameter settings in §A. We introduce the evaluation types and datasets in the following.

Sentence Retrieval. We use Bible (SR-B) and Tatoeba (Artetxe and Schwenk, 2019) (SR-T). The pairwise similarity for retrieving the target sentences is calculated using the mean pooling of contextualized word embeddings at the 8th layer.

Text Classification. We use Taxi1500 (Ma et al., 2023) and SIB200 (Adelani et al., 2024). The former is a Bible dataset with 6 categories whereas the latter is based on FLORES-200 (Costa-jussà et al., 2022) with more modern genres like technology.

Sequence Labeling. We use WikiANN for named entity recognition (NER) (Pan et al., 2017) and Universal Dependencies (de Marneffe et al., 2021) for Part-Of-Speech (POS) tagging.

4.3 Results and Discussion

We evaluate the LANGSAMP model and baseline to understand how the integration of language and script embeddings influences crosslingual transfer. We group the transfer target languages based on two characteristics: (1) whether it is a head or tail language and (2) whether it is written in Latin or a non-Latin script. This grouping aims to directly identify the effectiveness of LANGSAMP on low-resource languages and languages written in a less common script. The results are shown in Table 1.

Both tail and head languages benefit. We observe consistent improvements in tail and head languages across tasks. The enhancement is more obvious in tail languages. For example, LANGSAMP improves the performance by 7% for tail languages vs 1% for head languages in SR-B. A similar phenomenon can also be seen for other tasks. This pattern indicates that LANGSAMP can be more helpful for those tail languages, for which the training data is scarce. With the help of language embeddings sharing the burden, the LANGSAMP model can have more language-neutral representations for these languages, resulting in better performance.

Both non-Latin and Latin languages benefit. We observe similar consistent improvements when grouping languages into Latin or non-Latin languages. Different from the trend seen in tail/head groups, we see that no group shows an obvious larger enhancement compared to the other group. This can be explained by the fact that head and tail languages are distributed more equally in Latn and non-Latn groups. In addition, the improvements indicate the incorporation of script embeddings is helpful. By decoupling some script-specific information from the representations, the output generated by the backbone is more script-neutral, leading to better crosslingual transfer across scripts.

	SR-B			SR-T			Taxi1500			SIB200			NER			POS		
	tail	head	all	tail	head	all	tail	head	all	tail	head	all	tail	head	all	tail	head	all
vanilla model	11.9	56.4	23.2	46.0	77.7	68.6	18.1	<u>58.6</u>	28.4	56.1	**83.0**	68.3	<u>55.1</u>	<u>62.8</u>	<u>59.3</u>	<u>49.9</u>	75.7	<u>67.8</u>
w/ $\boldsymbol{E}^{Lang}$	<u>13.1</u>	<u>57.9</u>	<u>24.5</u>	**49.1**	<u>79.0</u>	<u>70.5</u>	18.3	58.5	<u>28.5</u>	<u>57.2</u>	<u>82.7</u>	<u>68.8</u>	**55.2**	**63.0**	**59.5**	<u>49.9</u>	<u>75.8</u>	<u>67.8</u>
w/ $\boldsymbol{E}^{Script}$	12.5	57.4	23.9	<u>48.3</u>	78.4	69.8	<u>18.5</u>	57.0	28.2	56.6	82.1	68.2	<u>55.1</u>	62.4	59.0	**50.8**	**76.2**	**68.4**
w/ $\boldsymbol{E}^{Lang}$ and $\boldsymbol{E}^{Script}$	**13.4**	**58.7**	**24.9**	**49.1**	**79.5**	**70.8**	**20.6**	**58.8**	**30.3**	**57.9**	**83.0**	**69.3**	54.9	61.6	58.6	49.7	75.6	67.6

Table 2: Ablation study. We investigate the effectiveness of language and script embeddings on downstream performance. Note that the vanilla model and w/$\boldsymbol{E}^{Lang}$ and $\boldsymbol{E}^{Script}$ are different from Baseline and LANGSAMP in Table 1 because of smaller pretraining data size. By including both types of embeddings, the model achieves the overall best performance among all model variants. **Bold** (<u>underlined</u>): best (second-best) result for each column.

Improvements can vary slightly across tasks. We observe more consistent large improvements for sequence-level tasks – retrieval and classification – where LANGSAMP outperforms the baseline in all groups. However, on sequence labeling tasks, LANGSAMP achieves very close performance to the baseline. For example, LANGSAMP scores are 0.1 less compared to the baseline on NER. This could be related to the difficulty of the tasks: both NER and POS are relatively easy tasks and models can transfer well in prevalent classes, e.g., *nouns*, through shared vocabulary (ImaniGooghari et al., 2023; Liu et al., 2024a). Therefore, decoupling language- or script-script information from the Transformer output can be less helpful for these tasks. Nevertheless, the overall improvements across tasks indicate the superiority of LANGSAMP compared with the baseline.

5 Analysis

5.1 Ablation Study

In the ablation study, we want to explore the effectiveness of language embeddings and script embeddings individually. However, due to a limited computation budget, we cannot run experiments on the full corpora for each variant. Therefore, we select 5% data for each language from Glot500-c and continually pretrain XLM-R using the same hyperparameters used in the main experiments described in §4.1. Specifically, we consider four variants: **a**) model without language/script embeddings; **b**) model with only language embeddings; **c**) model with only script embeddings; and **d**) model with both language and script embeddings. The performance of each variant is shown in Table 2.

Either language or script embeddings help. The vanilla model achieves the overall worst performance among all model variants. As long as language or script embeddings are included, we generally observe a consistent improvement across all downstream tasks. This indicates that both language and script embeddings can share the burden of encoding too much language- and script-specific information in the token representations. As a result, the representations generated by the model variants with language or script embeddings are more language-neutral. The best overall performance is achieved when both language and script embeddings are used, suggesting decoupling both language- and script-specific information would be the best option for improving crosslingual transfer.

Improvement varies across task types. Similar to the findings in §4.3, we observe that including the auxiliary embeddings is very helpful for sequence-level tasks, especially sentence retrieval, where we observe the highest enhancement, while less helpful for token-level tasks. It is also noticeable that including language embeddings is the most effective for sentence retrieval (either best or the second best per column). On the other hand, the sequence labeling task does not enjoy large improvements: most model variants achieve on-par performance with each other. The reason has been discussed in §4.3: NER and POS are relatively simple tasks since models can transfer easily in prevalent classes. Nevertheless, the overall results show the effectiveness of the auxiliary embeddings.

5.2 Qualitative Exploration: Visualization

We visualize language and script embeddings in Figure 3. Only head language embeddings are chosen for better readability. We observe that similar or related languages are located close to each other. For example, **cmn** and **zho** (simplified and traditional Chinese, lower left) are closest to each other, as are **pes** (Iranian Persian) and **prs** (Dari). The languages that are mutually influenced by Chinese to a large extent, **jpn**, **kor**, and **vie**, are also close to each other. Most European languages, as well as Indian languages that belong to the Indo-European family, form a rather dense cluster in the middle.

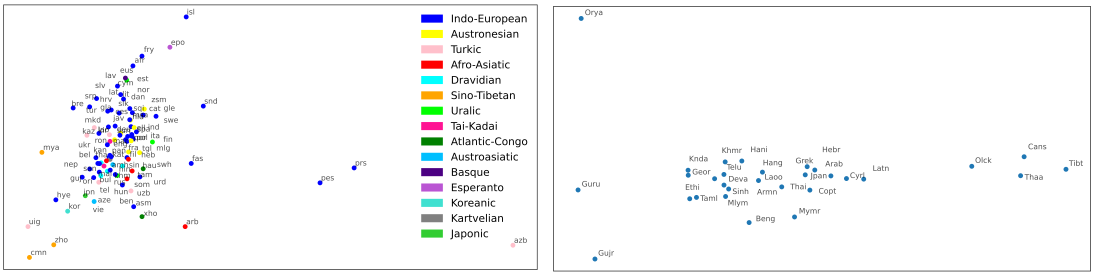

Figure 3: PCA visualizations of head language embeddings (left) and script embeddings (right). We see some related languages and scripts are close to each other, indicating that they implicitly encode language- and script-specific information. Data imbalance may have caused some languages/scripts with limited data to appear as outliers.

In the plot on the right, most scripts of the Indian subcontinent are found close to each other (**Deva**, **Telu**, **Mlym**, **Taml**, **Knda**, **Sinh**, **Beng**), despite some outliers (e.g., **Gujr** and **Guru**), probably due to small amount of data that are written in these scripts. **Hani** and scripts of languages that are mutually influenced (**Hang** and **Jpan**) are not far from each other. The same is true for two very related scripts, **Thai** and **Laoo**. In summary, the learnable language and script embeddings can capture language- and script-specific information in the training, which can be helpful for the language-neutrality of the output of transformer blocks.

5.3 Quantitative Exploration: Similarity

We expect that LANGSAMP can generate more language-neutral representations, meaning that representations of semantically equivalent sentences from different languages are similar. To evaluate this, we selected 10 high-resource languages that differ typologically and use a diverse set of scripts: **eng_Latn**, **rus_Cyrl**, **zho_Hani**, **arb_Arab**, **hin_Deva**, **jpn_Jpan**, **tur_Latn**, **spa_Latn**, **ind_Latn**, and **swa_Latn**. We calculated the pairwise cosine similarity of sentence representations using 100 randomly sampled parallel sentences from SR-B. Sentence representations are obtained by mean-pooling the token representations at the 8th layer, followed by subtracting the language centroid (the average of all 100 sentence representations for that language). We report the pairwise cosine similarity in Figure 5 in §B and show the improvement (by percentage) in Figure 4.

We can observe that the similarity between any two languages is improved in LANGSAMP. The enhancement is especially noticeable for typologically distinct languages using different scripts. For example, arb_Arab is in a different language family and written in a different script compared to the other 9 languages, the similarity involving arb_Arab is greatly improved: 4.7% for eng_Latn and 4.1% rus_Cyrl. Importantly, since LANGSAMP does not incorporate additional parallel data, this improvement is solely attributed to the inclusion of language and script embeddings during pretraining. This indicates that LANGSAMP effectively generates more language-neutral representations by decoupling language- and script-specific features into auxiliary embeddings.

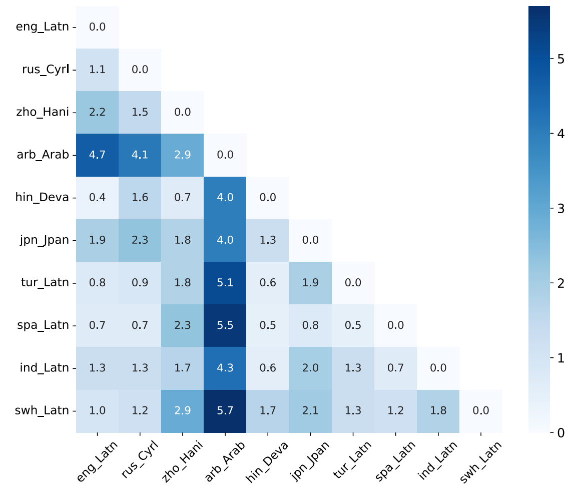

Figure 4: Similarity improvement (by percentage) from baseline to LANGSAMP in terms of the pairwise cosine similarity. Similarity is increased for each pair, indicating better language neutrality of the representations.

5.4 Case Study: Source Language Selection

Previous studies show language similarities have been useful for selecting good source languages for crosslingual transfer (Lin et al., 2019; Lauscher et al., 2020; Nie et al., 2023; Wang et al., 2023b,a; Lin et al., 2024). We expect this to also apply to the similarities induced by our language embeddings. Therefore, we conduct a case study and use the languages mentioned in §5.3 as the donor languages. When performing the downstream task for a specific target language, instead of always using English as the source language, we select the

	tail		head		Latn		non-Latn		all	
	English	Donor	English	Donor	English	Donor	English	Donor	English	Donor
Taxi1500	47.3	**48.3**	59.1	**60.3**	48.4	**49.0**	58.1	**60.5**	50.2	**51.2**
SIB200	**67.9**	**67.9**	81.2	**81.6**	71.0	**71.1**	80.3	**80.6**	74.0	**74.2**
NER	61.2	**61.7**	64.1	**65.6**	**67.5**	66.9	54.6	**58.5**	62.8	**63.8**
POS	**63.2**	53.8	**77.0**	72.3	**75.5**	68.4	**68.1**	63.6	**72.8**	66.6

Table 3: Performance of LANGSAMP, using English vs the closest donor language (based on cosine similarity induced from language embeddings) as the source language for zero-shot crosslingual transfer. Each number is the average over all target languages in a class. **Bold**: the result that is better for an English/Donor comparison.

	Taxi1500		SIB200		NER		POS	
tha	eng	jpn	eng	jpn	eng	jpn	eng	jpn
	63.8	**63.8**	85.4	**85.7**	2.1	**10.2**	**58.3**	27.5
yue	eng	zho	eng	zho	eng	zho	eng	zho
	55.4	**67.7**	-	-	25.7	**73.5**	42.6	**80.9**
san	eng	hin	eng	hin	eng	hin	eng	hin
	-	-	72.9	**76.6**	38.4	**53.4**	25.5	**32.7**
urd	eng	hin	eng	hin	eng	hin	eng	hin
	-	-	79.1	**80.6**	65.1	**76.8**	69.7	**89.7**
lin	eng	swh	eng	swh	eng	swh	eng	swh
	47.1	**54.7**	68.2	**73.3**	47.6	**55.9**	-	-
run	eng	swh	eng	swh	eng	swh	eng	swh
	48.0	**55.2**	65.2	**72.7**	-	-	-	-

Table 4: Languages with large improvements when using the closest donor language. In each task, the first/second column indicates results using English/the donor language as the source language. "-" indicates the language is not covered by the task.

donor language that is the most cosine-similar to the target language. We evaluate the LANGSAMP model on Taxi1500, SIB200, NER, and POS in a zero-shot crosslingual transfer style. The aggregated results are reported in Table 3 and we select representative target languages that benefit from choosing a good donor language in Table 4.

Effects of donor varies across tasks. Our results suggest that the performance gain from using a donor language varies across tasks. The gain in the text classification task is more consistent than the sequence labeling task. We assume the primary reason is that the training data for NER and POS are not parallel and the amount is highly variable across languages. For example, English has much more data than some of the other donor languages for these two tasks.

Non-Latin languages benefit more. For the text classification task, greater improvements can be observed in non-Latin script languages than in Latin script languages. This reflects previous findings that non-Latin script languages are less represented in mPLMs (Muller et al., 2021) and indicates the effectiveness of leveraging language embeddings in selecting better donor languages for them.

Donor is frequently from the same family. We find language embeddings frequently identify a donor language of the same family as the target language, leading to a large performance improvement over English as the source. For example, as shown in Table 4, **zho_Hani** as a donor language for **yue_Hani** leads to large performance gains on all three tasks. Similar gains are seen using **hin_Deva** for **san_Deva**. Positive effects can also be found across scripts, as in the case of using **hin_Deva** for **urd_Arab**, two very similar languages written in different scripts.

Interesting cases of unrelated donors. We also notice some interesting cases where the closest donor language is not or only partially related to the target language but nevertheless aids transfer performance as shown in Table 4. For example, **jpn_Jpan** has a positive effect for **tha_Thai**. Similarly, for **tuk_Latn**, using **rus_Cyrl** as the source achieves better transfer performance than English.

6 Conclusion

We propose LANGSAMP, a multilingual pretraining approach that leverages auxiliary language and script embeddings to facilitate more language-neutral representations by offloading the burden of encoding language- and script-specific information within the Transformer outputs. Through extensive experiments, we show LANGSAMP consistently outperforms the baseline on various downstream tasks. Our ablation study confirms the effectiveness of both language and script embeddings. LANGSAMP exhibits improved language neutrality, as reflected by increased pairwise similarity across all donor languages. Furthermore, our case study demonstrates that the auxiliary embeddings encode language- and script-specific information, facilitating the selection of optimal source languages for more effective crosslingual transfer.

Limitations

Due to the constraints of computing resources, we are not able to continue pretraining the model using the full Glot500-c data in **our ablation study**. However, as all variants are trained in a strictly controlled environment, their results can be compared in a fair way, and the consistent improvement suggests the effectiveness of the language embeddings.

In addition, we do not consider the possibility of introducing language and script embeddings before the Transformer blocks. Although this is also a possible architecture, it does not fulfill our aim and therefore is not relevant to us. Our primary prerequisite is that the resulting model can work as a universal text encoder without any language or script IDs as input, just like most highly multilingual models (e.g., XLM-R (Conneau et al., 2020) and mBERT (Devlin et al., 2019)). LANGSAMP only requires language or script IDs in the pretraining stage. After that, the backbone (token embeddings + the Transformer blocks) acts exactly as a universal text encoder. But we leave the exploration of whether such an architecture can bring about large-scale better language embeddings (not our motivation) for future research in the community.

Another potential limitation is the coverage of languages and scripts. Our model uses 610 languages and 30 scripts from Glot500-c. For low-resource languages not supported by our model, we can still generate representations since language IDs are not required as input. However, without a corresponding language embedding, it becomes challenging to select the optimal donor language for crosslingual transfer. Nonetheless, when adapting to these languages, the language embeddings can be expanded, similar to the approach commonly used for vocabulary extension.

References

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

David Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba Alabi, Yanke Mao, Haonan Gao, and En-Shiun Lee. 2024. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245, St. Julian's, Malta. Association for Computational Linguistics.

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

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

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

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020b. A call for more rigor in unsupervised cross-lingual learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7375–7388, Online. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Johannes Bjerva and Isabelle Augenstein. 2018. From phonology to syntax: Unsupervised linguistic typology at different levels with language embeddings. 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 907–916, New Orleans, Louisiana. Association for Computational Linguistics.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Tyler Chang, Zhuowen Tu, and Benjamin Bergen. 2022. The geometry of multilingual language model representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 119–136, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yiyi Chen, Russa Biswas, and Johannes Bjerva. 2023. Colex2Lang: Language embeddings from semantic typology. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 673–684, Tórshavn, Faroe Islands. University of Tartu Library.

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 and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Arindrima Datta, Bhuvana Ramabhadran, Jesse Emond, Anjuli Kannan, and Brian Roark. 2020. Language-agnostic multilingual modeling. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8239–8243.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

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

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Tianze Hua, Tian Yun, and Ellie Pavlick. 2024. mOthello: When do cross-lingual representation alignment and cross-lingual transfer emerge in multilingual models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1585–1598, Mexico City, Mexico. Association for Computational Linguistics.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2485–2494, Hong Kong, China. Association for Computational Linguistics.

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

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

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

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023*

Conference on Empirical Methods in Natural Language Processing, pages 13142–13152, Singapore. Association for Computational Linguistics.

Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1663–1674, Online. Association for Computational Linguistics.

Peiqin Lin, Chengzhi Hu, Zheyu Zhang, Andre Martins, and Hinrich Schuetze. 2024. mPLM-sim: Better cross-lingual similarity and transfer in multilingual pretrained language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 276–310, St. Julian's, Malta. Association for Computational Linguistics.

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

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Yihong Liu, Haris Jabbar, and Hinrich Schuetze. 2022. Flow-adapter architecture for unsupervised machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1253–1266, Dublin, Ireland. Association for Computational Linguistics.

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

Yihong Liu, Chunlan Ma, Haotian Ye, and Hinrich Schuetze. 2024b. TransliCo: A contrastive learning framework to address the script barrier in multilingual pretrained language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2476–2499, Bangkok, Thailand. Association for Computational Linguistics.

Yihong Liu, Haotian Ye, Leonie Weissweiler, Philipp Wicke, Renhao Pei, Robert Zangenfeind, and Hinrich Schütze. 2023. A crosslingual investigation of conceptualization in 1335 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12969–13000, Toronto, Canada. Association for Computational Linguistics.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Chunlan Ma, Ayyoob ImaniGooghari, Haotian Ye, Ehsaneddin Asgari, and Hinrich Schütze. 2023. Taxi1500: A multilingual dataset for text classification in 1500 languages. *arXiv preprint arXiv:2305.08487*.

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

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

Ercong Nie, Sheng Liang, Helmut Schmid, and Hinrich Schütze. 2023. Cross-lingual retrieval augmented prompt for low-resource languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8320–8340, Toronto, Canada. Association for Computational Linguistics.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages

116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robert Östling. 2015. Word order typology through multilingual word alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211, Beijing, China. Association for Computational Linguistics.

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 644–649, Valencia, Spain. Association for Computational Linguistics.

Lin Pan, Chung-Wei Hang, Haode Qi, Abhishek Shah, Saloni Potdar, and Mo Yu. 2021. Multilingual BERT post-pretraining alignment. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 210–219, Online. Association for Computational Linguistics.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. *arXiv preprint arXiv:2204.07580*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973, Hong Kong, China. Association for Computational Linguistics.

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

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, et al. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Mingyang Wang, Heike Adel, Lukas Lange, Jannik Strötgen, and Hinrich Schuetze. 2023a. GradSim: Gradient-based language grouping for effective multilingual training. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4631–4646, Singapore. Association for Computational Linguistics.

Mingyang Wang, Heike Adel, Lukas Lange, Jannik Strötgen, and Hinrich Schütze. 2023b. NLNDE at SemEval-2023 task 12: Adaptive pretraining and source language selection for low-resource multilingual sentiment analysis. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 488–497, Toronto, Canada. Association for Computational Linguistics.

Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. 2021. On learning universal representations across languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Andrea W Wen-Yi and David Mimno. 2023. Hyperpolyglot LLMs: Cross-lingual interpretability in token embeddings. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1124–1131, Singapore. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Orgest Xhelili, Yihong Liu, and Hinrich Schuetze. 2024. Breaking the script barrier in multilingual pre-trained language models with transliteration-based post-training alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11283–11296, Miami, Florida, USA. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

Dian Yu, Taiqi He, and Kenji Sagae. 2021. Language embeddings for typology and cross-lingual transfer learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7210–7225, Online. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Settings and Hyperparameters

We show the information of the evaluation datasets and used measures in Table 5 and introduce the detailed settings and hyperparameters as follows.

Sentence Retrieval We use English-aligned sentences (up to 500 and 1000 for SR-B and SR-T respectively) from languages covered by Glot500-c (ImaniGooghari et al., 2023). No fine-tuning is needed for this evaluation type: we directly use each model as a text encoder and generate the sentence-level representation by averaging the contextual token embeddings at the **8th** layer, similar to previous work (Jalili Sabet et al., 2020; ImaniGooghari et al., 2023; Liu et al., 2024a). We perform retrieval by sorting the pairwise similarities.

	|head|	|tail|	|Latn|	|non-Latn|	#class	measure (%)
SR-B	94	275	290	79	-	top-10 Acc.
SR-T	70	28	64	34	-	top-10 Acc.
Taxi1500	89	262	281	70	6	F1 score
SIB200	78	94	117	55	7	F1 score
NER	89	75	104	60	7	F1 score
POS	63	28	57	34	18	F1 score

Table 5: Information of the evaluation datasets and used measures. |head| (resp. |tail|): number of head (resp. tail) language-scripts. |Latn| (resp. |non-Latn|): number of languages written in Latin script (resp. non-Latn scripts). #class: the number of the categories if it belongs to a text classification or sequence labeling task.

Text Classification We add a 6-class or 7-class (for Taxi1500 and SIB200 respectively) sequence-level classification head onto the backbone model (no language or script IDs are required as input since the language modeling head is not needed in this sequence-level classification model). By default, we train the model on the English train set and store the best checkpoint on the English validation set. We train all models using AdamW optimizer (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) for a maximum of 40 epochs, with a learning rate of 1e-5 and an effective batch size of 16 (batch size of 8, gradient accumulation of 2). We use a single GTX 1080 Ti GPU for training. The evaluation is done in zero-shot transfer: we directly apply the best checkpoint to the test sets of all other languages.

Sequence Labeling We add a 7-class or 18-class (for NER and POS respectively) token-level classification head onto the backbone model (no language or script IDs are required as input since the language modeling head is not needed in this token-level classification model). Similarly, we train the model on the English train set and store the best checkpoint on the English validation set by default. We train all models using AdamW optimizer (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) for a maximum of 10 epochs. The learning rate is set to 2e-5 and the effective batch size is set to 32 (batch size of 8, gradient accumulation of 4). The training is done on a single GTX 1080 Ti GPU. The evaluation is done in zero-shot transfer: we directly apply the best checkpoint to the test sets of all other languages.

B Pairwise Cosine Similarity

As introduced in §5.3, we select 10 topologically different languages that are written in diverse

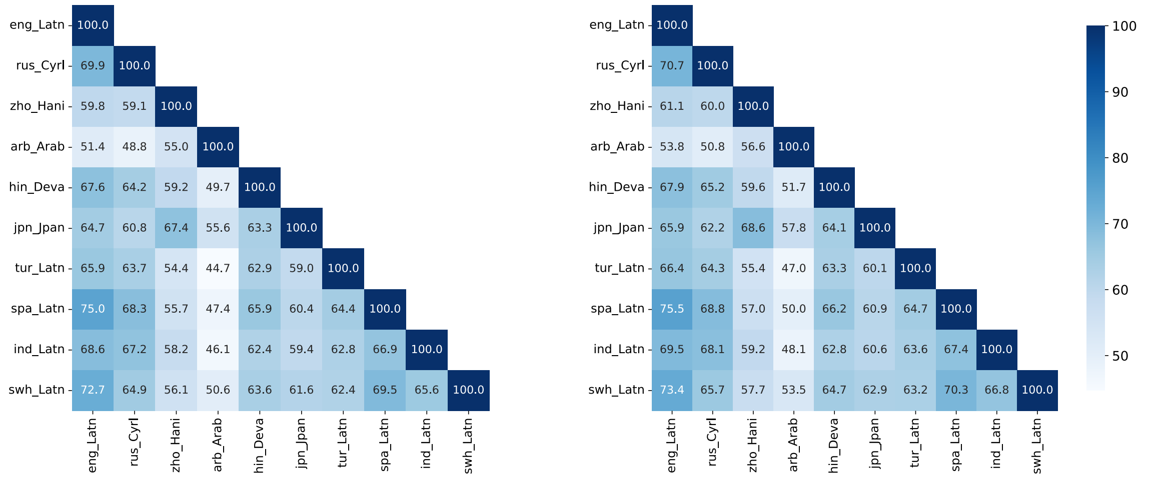

Figure 5: Comparison between baseline (left) and LANGSAMP (right) in terms of the pairwise cosine similarity. LANGSAMP achieves better similarity for each pair, indicating improved language neutrality of the representations.

scripts to assess the language neutrality: **eng_Latn**, **rus_Cyrl**, **zho_Hani**, **arb_Arab**, **hin_Deva**, **jpn_Jpan**, **tur_Latn**, **spa_Latn**, **ind_Latn**, and **swa_Latn**. We report the pairwise cosine similarity for the baseline and LANGSAMP in Figure 5.

It can be observed that the similarity between any two languages in LANGSAMP is consistently higher than in the baseline. The absolute increase is small in general, due to the fact that (1) without the introduction of the auxiliary language and script embeddings, the baseline already assigns good similarity to translations and (2) LANGSAMP does not introduce any additional parallel data in the pretraining, which is usually regarded as important to improve the similarity. Nevertheless, the consistent improvement indicates that LANGSAMP effectively improves the language neutrality by decoupling language- and script-specific features into auxiliary embeddings.

C Results for Each Language Family

We report the aggregated results for each language family for each task in Table 6. We see consistent improvement for all language families in sentence retrieval and text classification tasks. For sequence tagging tasks, LANGSAMP achieves similar performance compared with the baseline. This trend is similar to the main results we report in §4.3.

D Complete Crosslingual Transfer Results

We report the complete results of English-centric zero-shot crosslingual performance of baseline and LANGSAMP for all tasks and languages in Table 7, 8 (**SR-B**), Table 9 (**SR-T**), Table 10, 11(**Taxi1500**), 12 (**SIB200**), Table 13 (**NER**), and Table 14 (**POS**). Each result is the average over fine-tuning the baseline or LANGSAMP under five random seeds.

E Transfer Results Using English and Closest Donor Language

We report the complete results of the zero-shot crosslingual performance of LANGSAMP when using English and the closest donor language as the source language in Table 15, 16 (**Taxi1500**), 17 (**SIB200**), Table 18 (**NER**), and Table 19 (**POS**). Each result is directly obtained from a single run. We fine-tune the LANGSAMP using different donor languages under the same random seed.

SR-B

	(indo1319, 93)	(atla1278, 69)	(aust1307, 55)	(turk1311, 23)	(sino1245, 23)	(maya1287, 15)	(afro1255, 12)	(other, 79)	(all, 369)
Baseline	61.4	37.3	42.9	60.9	31.6	15.5	29.5	31.3	42.9
LANGSAMP	**62.0**	**40.2**	**45.1**	**63.3**	**34.8**	**15.7**	**32.0**	**34.6**	**45.1**

SR-T

	(indo1319, 54)	(atla1278, 2)	(aust1307, 7)	(turk1311, 7)	(sino1245, 3)	(maya1287, 0)	(afro1255, 5)	(other, 20)	(all, 98)
Baseline	74.2	50.0	48.7	71.3	81.7	-	52.1	68.7	69.7
LANGSAMP	**75.2**	**50.6**	**50.2**	**74.6**	**83.0**	-	**54.2**	**70.5**	**71.1**

Taxi1500

	(indo1319, 87)	(atla1278, 68)	(aust1307, 51)	(turk1311, 18)	(sino1245, 22)	(maya1287, 15)	(afro1255, 11)	(other, 79)	(all, 351)
Baseline	60.2	41.6	50.7	59.1	48.7	41.2	34.5	45.1	49.4
LANGSAMP	**62.9**	**47.0**	**55.3**	**62.9**	**53.8**	**45.8**	**39.0**	**49.3**	**53.6**

SIB200

	(indo1319, 71)	(atla1278, 33)	(aust1307, 17)	(turk1311, 10)	(sino1245, 5)	(maya1287, 0)	(afro1255, 13)	(other, 23)	(all, 172)
Baseline	82.1	59.0	76.4	80.5	67.4	-	73.0	75.1	75.0
LANGSAMP	**82.7**	**60.5**	**78.0**	**81.8**	**68.7**	-	**73.1**	**75.7**	**75.9**

NER

	(indo1319, 94)	(atla1278, 5)	(aust1307, 12)	(turk1311, 12)	(sino1245, 7)	(maya1287, 0)	(afro1255, 6)	(other, 28)	(all, 164)
Baseline	66.5	**62.0**	**58.9**	**62.5**	**37.9**	-	54.7	56.0	62.1
LANGSAMP	**67.4**	61.3	58.5	60.9	34.7	-	**56.1**	**57.3**	**62.5**

POS

	(indo1319, 54)	(atla1278, 2)	(aust1307, 4)	(turk1311, 5)	(sino1245, 3)	(maya1287, 1)	(afro1255, 6)	(other, 16)	(all, 91)
Baseline	**78.2**	**61.3**	**74.7**	**72.5**	34.1	**63.8**	**65.4**	60.6	**71.8**
LANGSAMP	78.1	61.2	73.3	71.1	**40.3**	59.7	64.8	**60.7**	71.7

Table 6: Aggregated performance of the baseline and LANGSAMP for 7 major language families on all tasks. We report the average performance for **indo1319** (Indo-European), **atla1278** (Atlantic-Congo), **aust1307** (Austronesian), **turk1311** (Turkic), **sino1245** (Sino-Tibetan), **maya1287** (Mayan), and **afro1255** (Afro-Asiatic). We classify the remaining languages into the group "**other**". In addition, we report the average over all languages (group "**all**"). The number of languages in that family is shown in parentheses. **Bold**: best result for each task.

Language	Baseline	LangSamp	Language	Baseline	LangSamp	Language	Baseline	LangSamp	Language	Baseline	LangSamp
ace_Latn	43.8	**49.4**	ach_Latn	37.6	**40.6**	acr_Latn	17.6	**18.6**	afr_Latn	**74.2**	72.4
agw_Latn	31.0	**38.2**	ahk_Latn	3.4	**3.8**	aka_Latn	41.8	**48.4**	aln_Latn	**70.0**	**70.0**
als_Latn	**54.4**	**54.4**	alt_Cyrl	53.8	**57.0**	alz_Latn	36.2	**37.4**	amh_Ethi	44.4	**51.2**
aoj_Latn	15.6	**18.6**	arb_Arab	9.6	**11.6**	arn_Latn	18.2	**23.0**	ary_Arab	11.2	**13.0**
arz_Arab	**15.2**	**15.2**	asm_Beng	**59.2**	59.0	ayr_Latn	37.6	**46.0**	azb_Arab	55.6	**59.0**
aze_Latn	73.4	**75.4**	bak_Cyrl	58.8	**62.2**	bam_Latn	38.4	**44.8**	ban_Latn	33.0	**33.2**
bar_Latn	32.2	**34.0**	bba_Latn	26.2	**31.0**	bbc_Latn	**60.8**	58.8	bci_Latn	**12.0**	11.8
bcl_Latn	75.4	**79.0**	bel_Cyrl	**70.6**	69.6	bem_Latn	51.0	**54.4**	ben_Beng	53.4	**55.4**
bhw_Latn	28.4	**30.6**	bim_Latn	31.4	**42.8**	bis_Latn	45.2	**50.8**	bod_Tibt	29.6	**33.6**
bqc_Latn	27.4	**29.2**	bre_Latn	**31.8**	30.0	bts_Latn	**62.4**	62.0	btx_Latn	**57.2**	55.8
bul_Cyrl	79.8	**80.0**	bum_Latn	32.8	**35.2**	bzj_Latn	69.8	**70.2**	cab_Latn	11.6	**11.8**
cac_Latn	10.8	**11.8**	cak_Latn	**17.8**	16.6	caq_Latn	26.0	**29.8**	cat_Latn	**85.4**	83.2
cbk_Latn	54.8	**56.2**	cce_Latn	41.8	**45.4**	ceb_Latn	70.4	**70.6**	ces_Latn	**68.2**	67.0
cfm_Latn	34.4	**38.8**	che_Cyrl	10.2	**11.2**	chk_Latn	35.2	**43.0**	chv_Cyrl	45.0	**54.4**
ckb_Arab	31.2	**32.8**	cmn_Hani	**41.4**	40.8	cnh_Latn	38.2	**43.2**	crh_Cyrl	67.2	**70.0**
crs_Latn	**85.6**	84.4	csy_Latn	40.2	**49.6**	ctd_Latn	44.4	**50.6**	ctu_Latn	**16.6**	16.0
cuk_Latn	**17.0**	**17.0**	cym_Latn	**45.6**	43.8	dan_Latn	**72.4**	71.8	deu_Latn	73.8	**74.0**
djk_Latn	**38.0**	**38.0**	dln_Latn	46.6	**51.4**	dtp_Latn	17.0	**17.8**	dyu_Latn	33.0	**40.2**
dzo_Tibt	28.4	**33.0**	efi_Latn	41.6	**53.6**	ell_Grek	48.2	**49.2**	enm_Latn	**69.4**	**69.4**
epo_Latn	**67.4**	65.8	est_Latn	**66.4**	66.0	eus_Latn	23.8	**24.2**	ewe_Latn	33.2	**34.8**
fao_Latn	**79.8**	78.4	fas_Arab	80.2	**84.2**	fij_Latn	30.0	**31.0**	fil_Latn	**77.6**	77.2
fin_Latn	65.4	**66.0**	fon_Latn	20.2	**25.2**	fra_Latn	**87.4**	87.2	fry_Latn	**47.0**	44.0
gaa_Latn	34.4	**40.6**	gil_Latn	30.0	**31.6**	giz_Latn	32.4	**36.4**	gkn_Latn	20.4	**24.2**
gkp_Latn	13.2	**14.6**	gla_Latn	**39.0**	38.0	gle_Latn	**41.2**	38.4	glv_Latn	37.2	**38.6**
gom_Latn	33.2	**36.0**	gor_Latn	21.8	**23.0**	grc_Grek	44.4	**47.0**	guc_Latn	**9.8**	8.2
gug_Latn	28.2	**31.2**	guj_Gujr	**69.8**	67.6	gur_Latn	17.6	**18.2**	guw_Latn	36.8	**45.4**
gya_Latn	27.6	**32.6**	gym_Latn	**13.6**	13.0	hat_Latn	**76.4**	74.6	hau_Latn	57.6	**59.6**
haw_Latn	28.0	**30.4**	heb_Hebr	21.6	**23.0**	hif_Latn	33.2	**34.6**	hil_Latn	74.0	**79.8**
hin_Deva	**75.6**	74.6	hin_Latn	34.2	**36.2**	hmo_Latn	44.2	**57.0**	hne_Deva	71.6	**73.6**
hnj_Latn	39.6	**46.6**	hra_Latn	43.4	**46.4**	hrv_Latn	**80.4**	79.8	hui_Latn	19.8	**22.0**
hun_Latn	65.6	**69.0**	hus_Latn	14.8	**16.2**	hye_Armn	62.8	**65.6**	iba_Latn	70.2	**71.6**
ibo_Latn	**32.4**	31.6	ifa_Latn	26.2	**29.0**	ifb_Latn	**28.6**	**28.6**	ikk_Latn	30.2	**46.4**
ilo_Latn	53.4	**54.4**	ind_Latn	78.4	**78.6**	isl_Latn	71.0	**71.8**	ita_Latn	76.2	**76.8**
ium_Latn	20.0	**23.2**	ixl_Latn	13.8	**14.4**	izz_Latn	19.6	**22.6**	jam_Latn	**61.0**	59.2
jav_Latn	**55.4**	52.0	jpn_Jpan	65.8	**67.6**	kaa_Cyrl	71.2	**75.0**	kaa_Latn	32.0	**37.6**
kab_Latn	12.2	**13.4**	kac_Latn	22.2	**27.0**	kal_Latn	12.6	**16.8**	kan_Knda	50.0	**52.8**
kat_Geor	49.6	**52.4**	kaz_Cyrl	69.4	**70.4**	kbp_Latn	21.8	**26.8**	kek_Latn	16.6	**18.6**
khm_Khmr	39.4	**43.0**	kia_Latn	24.6	**28.8**	kik_Latn	44.4	**48.4**	kin_Latn	56.6	**60.2**
kir_Cyrl	69.8	**70.2**	kjb_Latn	23.4	**26.0**	kjh_Cyrl	45.6	**50.6**	kmm_Latn	33.8	**38.0**
kmr_Cyrl	**42.0**	40.2	kmr_Latn	60.2	**60.4**	knv_Latn	7.0	**8.4**	kor_Hang	60.8	**64.0**
kpg_Latn	42.6	**48.8**	krc_Cyrl	59.8	**62.2**	kri_Latn	61.4	**62.6**	ksd_Latn	31.4	**41.0**
kss_Latn	5.2	**6.0**	ksw_Mymr	26.2	**28.0**	kua_Latn	43.0	**43.8**	lam_Latn	20.4	**22.8**
lao_Laoo	41.6	**47.2**	lat_Latn	56.6	**58.0**	lav_Latn	69.8	**71.2**	ldi_Latn	**22.4**	22.0
leh_Latn	**46.8**	45.8	lhu_Latn	**4.4**	4.2	lin_Latn	64.6	**71.0**	lit_Latn	**67.0**	66.6
loz_Latn	**46.8**	45.6	ltz_Latn	**63.8**	63.2	lug_Latn	37.2	**40.8**	luo_Latn	**42.8**	42.6
lus_Latn	46.6	**53.2**	lzh_Hani	59.8	**62.4**	mad_Latn	42.6	**44.6**	mah_Latn	30.4	**33.8**
mai_Deva	52.6	**56.0**	mal_Mlym	51.6	**57.4**	mam_Latn	**10.2**	**10.2**	mar_Deva	68.4	**71.4**
mau_Latn	2.8	**3.4**	mbb_Latn	22.0	**29.8**	mck_Latn	**55.6**	53.4	mcn_Latn	34.2	**40.8**
mco_Latn	**6.6**	6.4	mdy_Ethi	21.4	**30.6**	meu_Latn	48.8	**52.0**	mfe_Latn	**77.4**	**77.4**

Table 7: Top-10 accuracy of models on **SR-B** (Part I).

Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP
mgh_Latn	17.4	**20.8**	mgr_Latn	**48.6**	47.2	mhr_Cyrl	37.4	**43.2**	min_Latn	**32.4**	29.6
miq_Latn	28.8	**36.8**	mkd_Cyrl	78.4	**78.8**	mlg_Latn	60.2	**61.2**	mlt_Latn	48.0	**50.4**
mos_Latn	32.2	**32.8**	mps_Latn	16.4	**20.6**	mri_Latn	45.6	**55.0**	mrw_Latn	34.0	**40.6**
msa_Latn	43.6	**44.2**	mwm_Latn	24.0	**25.6**	mxv_Latn	**7.0**	**7.0**	mya_Mymr	25.8	**28.0**
myv_Cyrl	26.6	**30.6**	mzh_Latn	24.6	**25.4**	nan_Latn	13.2	**13.6**	naq_Latn	16.8	**26.8**
nav_Latn	**8.6**	**8.6**	nbl_Latn	**49.4**	48.4	nch_Latn	**21.6**	**21.6**	ncj_Latn	18.8	**19.4**
ndc_Latn	32.4	**36.2**	nde_Latn	51.0	**54.8**	ndo_Latn	41.0	**44.0**	nds_Latn	**38.4**	**38.4**
nep_Deva	56.4	**59.0**	ngu_Latn	**26.2**	26.0	nia_Latn	25.6	**28.0**	nld_Latn	**78.4**	78.0
nmf_Latn	25.6	**28.2**	nnb_Latn	33.2	**38.8**	nno_Latn	**76.8**	75.8	nob_Latn	**85.4**	85.0
nor_Latn	**85.8**	83.4	npi_Deva	77.4	**80.8**	nse_Latn	48.4	**51.8**	nso_Latn	46.2	**50.2**
nya_Latn	**57.6**	**57.6**	nyn_Latn	**48.8**	47.4	nyy_Latn	23.4	**24.6**	nzi_Latn	29.2	**34.4**
ori_Orya	51.2	**53.4**	ory_Orya	46.4	**49.8**	oss_Cyrl	41.4	**56.4**	ote_Latn	12.0	**13.2**
pag_Latn	**55.2**	52.2	pam_Latn	37.4	**41.2**	pan_Guru	**46.2**	45.4	pap_Latn	72.8	**75.0**
pau_Latn	17.0	**23.4**	pcm_Latn	**69.8**	69.4	pdt_Latn	**69.4**	66.0	pes_Arab	74.2	**75.2**
pis_Latn	51.4	**54.8**	pls_Latn	27.0	**31.8**	plt_Latn	60.2	**60.8**	poh_Latn	10.6	**11.4**
pol_Latn	73.8	**75.6**	pon_Latn	21.4	**24.0**	por_Latn	**81.8**	81.0	prk_Latn	42.0	**47.4**
prs_Arab	84.6	**87.0**	pxm_Latn	18.2	**19.8**	qub_Latn	30.6	**35.6**	quc_Latn	**18.6**	17.4
qug_Latn	53.6	**59.2**	quh_Latn	40.2	**43.8**	quw_Latn	46.2	**50.4**	quy_Latn	47.4	**54.4**
quz_Latn	59.4	**63.6**	qvi_Latn	49.2	**57.6**	rap_Latn	17.0	**17.8**	rar_Latn	**20.4**	19.8
rmy_Latn	30.4	**32.2**	ron_Latn	**69.4**	69.0	rop_Latn	35.8	**41.4**	rug_Latn	37.8	**38.4**
run_Latn	48.2	**52.4**	rus_Cyrl	74.6	**76.4**	sag_Latn	39.6	**45.4**	sah_Cyrl	43.4	**45.8**
san_Deva	**24.2**	23.6	san_Latn	**7.8**	7.4	sba_Latn	28.0	**29.2**	seh_Latn	67.4	**69.4**
sin_Sinh	45.6	**49.0**	slk_Latn	**69.8**	69.2	slv_Latn	**61.2**	60.8	sme_Latn	35.0	**37.6**
smo_Latn	27.6	**28.8**	sna_Latn	38.4	**41.2**	snd_Arab	**67.2**	65.0	som_Latn	**35.0**	34.8
sop_Latn	**32.4**	28.8	sot_Latn	48.4	**52.4**	spa_Latn	80.8	**81.4**	sqi_Latn	62.2	**64.8**
srm_Latn	**28.2**	26.6	srn_Latn	75.4	**75.6**	srp_Cyrl	**87.2**	85.8	srp_Latn	**85.8**	85.4
ssw_Latn	42.8	**47.0**	sun_Latn	52.0	**54.0**	suz_Deva	21.0	**22.6**	swe_Latn	**78.6**	77.0
swh_Latn	**71.6**	71.4	sxn_Latn	20.6	**20.8**	tam_Taml	47.0	**50.6**	tat_Cyrl	68.2	**70.4**
tbz_Latn	13.2	**18.2**	tca_Latn	10.0	**13.8**	tdt_Latn	50.0	**53.6**	tel_Telu	48.0	**50.2**
teo_Latn	19.4	**19.6**	tgk_Cyrl	69.2	**69.4**	tgl_Latn	**79.6**	78.0	tha_Thai	33.8	**38.0**
tih_Latn	42.2	**46.4**	tir_Ethi	32.2	**34.8**	tlh_Latn	62.0	**66.4**	tob_Latn	**11.6**	11.4
toh_Latn	36.8	**41.8**	toi_Latn	**39.4**	**39.4**	toj_Latn	**14.8**	12.6	ton_Latn	16.0	**16.6**
top_Latn	**6.6**	6.0	tpi_Latn	58.0	**62.2**	tpm_Latn	**27.4**	23.0	tsn_Latn	32.6	**34.6**
tso_Latn	50.0	**51.0**	tsz_Latn	21.2	**25.8**	tuc_Latn	25.6	**32.4**	tui_Latn	29.8	**31.0**
tuk_Cyrl	67.4	**69.4**	tuk_Latn	67.6	**70.0**	tum_Latn	**58.4**	57.0	tur_Latn	70.2	**70.4**
twi_Latn	35.0	**42.0**	tyv_Cyrl	**44.2**	43.4	tzh_Latn	19.0	**19.8**	tzo_Latn	**14.2**	13.6
udm_Cyrl	41.6	**45.2**	uig_Arab	47.4	**50.8**	uig_Latn	57.2	**58.8**	ukr_Cyrl	67.0	**68.0**
urd_Arab	60.4	**61.4**	uzb_Cyrl	80.6	**81.2**	uzb_Latn	**70.0**	68.2	uzn_Cyrl	82.4	**83.0**
ven_Latn	37.2	**42.0**	vie_Latn	68.0	**69.4**	wal_Latn	35.0	**43.4**	war_Latn	42.6	**44.0**
wbm_Latn	37.6	**46.2**	wol_Latn	31.8	**33.2**	xav_Latn	3.8	**4.0**	xho_Latn	42.6	**44.2**
yan_Latn	16.4	**27.2**	yao_Latn	37.4	**37.6**	yap_Latn	15.8	**19.6**	yom_Latn	37.6	**40.0**
yor_Latn	27.4	**28.8**	yua_Latn	**13.2**	12.8	yue_Hani	**17.2**	**17.2**	zai_Latn	29.0	**30.6**
zho_Hani	41.6	**41.8**	zlm_Latn	**84.8**	**84.8**	zom_Latn	39.6	**45.0**	zsm_Latn	90.0	**91.0**

Table 8: Top-10 accuracy of models on **SR-B** (Part II).

Language	Baseline	LangSAMP	Language	Baseline	LangSAMP	Language	Baseline	LangSAMP	Language	Baseline	LangSAMP
afr_Latn	77.9	**80.4**	amh_Ethi	47.0	**52.4**	ara_Arab	**69.4**	68.7	arz_Arab	61.8	**63.9**
ast_Latn	80.3	**84.3**	aze_Latn	82.6	**84.1**	bel_Cyrl	**83.6**	83.0	ben_Beng	72.1	**74.9**
bos_Latn	90.1	**90.4**	bre_Latn	17.4	**18.2**	bul_Cyrl	87.5	**89.2**	cat_Latn	78.2	**78.6**
cbk_Latn	**49.4**	48.0	ceb_Latn	39.0	**42.5**	ces_Latn	**75.7**	73.5	cmn_Hani	87.1	**87.4**
csb_Latn	38.3	**38.7**	cym_Latn	52.2	**55.0**	dan_Latn	91.7	**92.9**	deu_Latn	95.5	**95.7**
dtp_Latn	17.0	**19.3**	ell_Grek	79.3	**82.7**	epo_Latn	71.8	**74.8**	est_Latn	68.2	**69.9**
eus_Latn	52.2	**55.4**	fao_Latn	**77.1**	75.6	fin_Latn	72.3	**74.2**	fra_Latn	**85.3**	85.2
fry_Latn	75.1	**79.2**	gla_Latn	38.4	**38.6**	gle_Latn	44.8	**48.3**	glg_Latn	**77.1**	76.4
gsw_Latn	58.1	**63.2**	heb_Hebr	71.4	**74.9**	hin_Deva	**88.1**	87.3	hrv_Latn	**87.9**	87.5
hsb_Latn	**49.7**	**49.7**	hun_Latn	71.5	**73.2**	hye_Armn	79.1	**81.3**	ido_Latn	54.6	**55.8**
ile_Latn	71.2	**71.5**	ina_Latn	89.2	**90.7**	ind_Latn	88.1	**88.9**	isl_Latn	84.0	**84.5**
ita_Latn	84.1	**85.7**	jpn_Jpan	**77.2**	77.1	kab_Latn	10.8	**11.0**	kat_Geor	71.2	**72.4**
kaz_Cyrl	74.6	**77.7**	khm_Khmr	57.5	**63.0**	kor_Hang	80.8	**81.1**	kur_Latn	49.8	**52.4**
lat_Latn	39.2	**42.1**	lfn_Latn	55.8	**56.8**	lit_Latn	70.4	**72.9**	lvs_Latn	76.2	**78.1**
mal_Mlym	87.5	**91.6**	mar_Deva	79.8	**81.6**	mhr_Cyrl	27.7	**33.4**	mkd_Cyrl	**79.6**	79.4
mon_Cyrl	78.2	**80.5**	nds_Latn	71.3	**72.5**	nld_Latn	92.4	**93.4**	nno_Latn	85.5	**87.4**
nob_Latn	94.5	**95.3**	oci_Latn	**46.6**	44.9	pam_Latn	**10.2**	**10.2**	pes_Arab	86.7	**86.9**
pms_Latn	49.5	**50.9**	pol_Latn	**84.3**	83.4	por_Latn	90.2	**90.7**	ron_Latn	86.0	**86.9**
rus_Cyrl	91.6	**92.1**	slk_Latn	77.9	**78.2**	slv_Latn	**76.2**	75.9	spa_Latn	**88.6**	88.3
sqi_Latn	84.1	**85.2**	srp_Latn	**89.7**	89.6	swe_Latn	89.4	**89.6**	swh_Latn	**45.1**	44.9
tam_Taml	**50.2**	45.0	tat_Cyrl	71.2	**74.6**	tel_Telu	72.6	**74.8**	tgl_Latn	73.9	**74.2**
tha_Thai	75.4	**79.2**	tuk_Latn	62.1	**68.0**	tur_Latn	79.1	**82.0**	uig_Arab	64.7	**68.4**
ukr_Cyrl	84.9	**86.5**	urd_Arab	78.5	**81.7**	uzb_Cyrl	65.0	**67.3**	vie_Latn	**88.9**	88.8
war_Latn	22.7	**25.2**	wuu_Hani	79.0	**82.4**	xho_Latn	54.9	**56.3**	yid_Hebr	65.8	**67.6**

Table 9: Top-10 accuracy of models on **SR-T**.

Language	Baseline	LangSAMP	Language	Baseline	LangSAMP	Language	Baseline	LangSAMP	Language	Baseline	LangSAMP
ace_Latn	**66.3**	64.2	ach_Latn	35.8	**40.3**	acr_Latn	44.2	**51.0**	afr_Latn	**60.0**	58.8
agw_Latn	51.0	**56.3**	ahk_Latn	**8.0**	6.3	aka_Latn	42.5	**49.0**	aln_Latn	55.3	**58.1**
als_Latn	56.2	**58.1**	alt_Cyrl	47.2	**49.7**	alz_Latn	31.1	**38.5**	amh_Ethi	**8.8**	7.7
aoj_Latn	34.1	**42.6**	arn_Latn	40.9	**44.5**	ary_Arab	32.9	**33.8**	arz_Arab	35.4	**40.8**
asm_Beng	62.5	**64.5**	ayr_Latn	52.7	**57.3**	azb_Arab	**63.5**	62.3	aze_Latn	66.0	**70.2**
bak_Cyrl	59.7	**59.9**	bam_Latn	43.3	**49.1**	ban_Latn	42.5	**48.0**	bar_Latn	44.1	**49.2**
bba_Latn	39.4	**43.4**	bci_Latn	29.6	**33.5**	bcl_Latn	54.0	**63.2**	bel_Cyrl	59.7	**61.4**
bem_Latn	45.7	**50.5**	ben_Beng	61.8	**66.6**	bhw_Latn	44.4	**54.4**	bim_Latn	49.4	**50.2**
bis_Latn	65.8	**71.7**	bqc_Latn	31.6	**37.7**	bre_Latn	35.7	**42.9**	btx_Latn	52.9	**63.9**
bul_Cyrl	64.9	**65.5**	bum_Latn	38.6	**46.9**	bzj_Latn	66.3	**68.1**	cab_Latn	22.9	**31.1**
cac_Latn	42.7	**47.0**	cak_Latn	51.2	**55.2**	caq_Latn	39.7	**45.5**	cat_Latn	**63.4**	62.2
cbk_Latn	62.0	**68.8**	cce_Latn	41.3	**47.8**	ceb_Latn	52.9	**55.5**	ces_Latn	59.7	**66.8**
cfm_Latn	54.5	**65.6**	che_Cyrl	17.3	**23.2**	chv_Cyrl	54.8	**62.2**	cmn_Hani	67.4	**70.2**
cnh_Latn	61.4	**64.6**	crh_Cyrl	60.4	**64.1**	crs_Latn	**65.3**	64.6	csy_Latn	52.4	**64.2**
ctd_Latn	52.5	**59.3**	ctu_Latn	50.3	**51.3**	cuk_Latn	39.1	**43.7**	cym_Latn	**50.0**	49.1
dan_Latn	62.0	**64.2**	deu_Latn	53.0	**56.0**	djk_Latn	46.8	**55.5**	dln_Latn	47.7	**61.7**
dtp_Latn	50.0	**51.3**	dyu_Latn	46.4	**57.7**	dzo_Tibt	55.9	**57.4**	efi_Latn	52.1	**56.9**
ell_Grek	59.6	**62.2**	eng_Latn	74.2	**76.1**	enm_Latn	**72.1**	71.9	epo_Latn	56.0	**58.9**
est_Latn	**56.9**	56.2	eus_Latn	23.2	**25.9**	ewe_Latn	42.7	**52.2**	fao_Latn	56.6	**60.2**
fas_Arab	**72.0**	70.1	fij_Latn	43.7	**48.9**	fil_Latn	56.9	**58.8**	fin_Latn	57.7	**59.5**
fon_Latn	43.0	**44.2**	fra_Latn	64.7	**70.4**	fry_Latn	39.1	**43.2**	gaa_Latn	39.4	**42.4**
gil_Latn	40.9	**44.9**	giz_Latn	41.6	**50.2**	gkn_Latn	37.2	**42.8**	gkp_Latn	31.9	**38.6**
gla_Latn	47.8	**48.8**	gle_Latn	41.6	**42.5**	glv_Latn	37.4	**44.7**	gom_Latn	34.9	**37.9**
gor_Latn	42.6	**50.4**	guc_Latn	32.8	**39.4**	gug_Latn	33.7	**40.9**	guj_Gujr	68.1	**69.5**
gur_Latn	33.7	**43.3**	guw_Latn	48.7	**53.6**	gya_Latn	**40.6**	39.8	gym_Latn	40.4	**47.2**
hat_Latn	62.5	**65.2**	hau_Latn	53.8	**59.1**	haw_Latn	29.2	**39.2**	heb_Hebr	17.9	**20.8**
hif_Latn	44.5	**47.6**	hil_Latn	64.7	**67.7**	hin_Deva	66.0	**69.7**	hmo_Latn	58.4	**65.5**
hne_Deva	65.7	**66.7**	hnj_Latn	63.7	**67.1**	hra_Latn	50.4	**56.1**	hrv_Latn	62.8	**68.0**
hui_Latn	46.0	**51.1**	hun_Latn	63.7	**68.4**	hus_Latn	35.6	**42.2**	hye_Armn	69.7	**71.4**
iba_Latn	57.1	**61.6**	ibo_Latn	56.2	**58.3**	ifa_Latn	46.5	**55.2**	ifb_Latn	48.7	**50.6**
ikk_Latn	46.8	**52.3**	ilo_Latn	49.8	**60.7**	ind_Latn	76.1	**78.3**	isl_Latn	51.2	**58.0**
ita_Latn	63.5	**66.3**	ium_Latn	56.2	**59.4**	ixl_Latn	31.7	**39.6**	izz_Latn	39.4	**48.9**
jam_Latn	63.6	**68.5**	jav_Latn	46.2	**51.6**	jpn_Jpan	63.6	**63.7**	kaa_Cyrl	57.7	**66.8**
kab_Latn	23.3	**30.4**	kac_Latn	**49.2**	45.7	kal_Latn	30.0	**37.2**	kan_Knda	65.6	**65.8**
kat_Geor	**59.6**	57.6	kaz_Cyrl	**64.3**	62.4	kbp_Latn	34.5	**37.4**	kek_Latn	44.5	**46.6**
khm_Khmr	**69.5**	66.2	kia_Latn	40.9	**52.2**	kik_Latn	40.4	**46.7**	kin_Latn	43.9	**56.8**
kir_Cyrl	66.5	**67.7**	kjb_Latn	45.4	**48.5**	kjh_Cyrl	49.9	**55.1**	kmm_Latn	46.3	**57.2**
kmr_Cyrl	50.1	**51.6**	knv_Latn	43.1	**45.1**	kor_Hang	70.3	**72.4**	kpg_Latn	63.9	**65.6**
krc_Cyrl	55.7	**63.0**	kri_Latn	58.8	**64.1**	ksd_Latn	53.3	**53.5**	kss_Latn	**21.8**	17.9
ksw_Mymr	47.7	**50.0**	kua_Latn	41.0	**45.9**	lam_Latn	31.9	**38.0**	lao_Laoo	**71.9**	70.5
lat_Latn	57.0	**64.0**	lav_Latn	62.5	**64.8**	ldi_Latn	26.7	**34.8**	leh_Latn	44.4	**48.3**
lhu_Latn	22.7	**27.3**	lin_Latn	47.3	**55.5**	lit_Latn	61.1	**61.8**	loz_Latn	49.2	**49.8**
ltz_Latn	**53.3**	52.1	lug_Latn	41.9	**52.6**	luo_Latn	36.8	**44.8**	lus_Latn	47.5	**54.8**
lzh_Hani	61.1	**68.5**	mad_Latn	59.4	**63.0**	mah_Latn	33.8	**45.2**	mai_Deva	**64.1**	63.4
mal_Mlym	**7.1**	6.1	mam_Latn	27.6	**34.8**	mar_Deva	60.8	**61.9**	mau_Latn	**6.9**	5.9
mbb_Latn	52.2	**55.2**	mck_Latn	40.7	**46.2**	mcn_Latn	35.1	**44.2**	mco_Latn	21.9	**26.2**
mdy_Ethi	48.5	**54.5**	meu_Latn	46.9	**57.9**	mfe_Latn	68.4	**69.9**	mgh_Latn	31.2	**33.6**
mgr_Latn	45.9	**48.4**	mhr_Cyrl	40.9	**41.0**	min_Latn	50.3	**53.7**	miq_Latn	51.0	**54.2**
mkd_Cyrl	68.7	**72.9**	mlg_Latn	47.0	**51.7**	mlt_Latn	49.0	**53.5**	mos_Latn	35.8	**44.6**

Table 10: F1 scores of models on **Taxi1500** (Part I).

Language	Baseline	LangSAMP	Language	Baseline	LangSAMP	Language	Baseline	LangSAMP	Language	Baseline	LangSAMP
mps_Latn	51.3	**56.3**	mri_Latn	41.5	**49.2**	mrw_Latn	47.8	**48.5**	msa_Latn	46.0	**49.0**
mwm_Latn	51.3	**57.8**	mxv_Latn	14.3	**27.9**	mya_Mymr	56.6	**57.8**	myv_Cyrl	41.3	**47.8**
mzh_Latn	39.1	**42.7**	nan_Latn	25.5	**32.3**	naq_Latn	39.0	**45.6**	nav_Latn	21.6	**25.8**
nbl_Latn	46.0	**52.3**	nch_Latn	40.9	**46.1**	ncj_Latn	34.5	**41.7**	ndc_Latn	38.2	**43.7**
nde_Latn	46.0	**52.3**	ndo_Latn	45.4	**50.7**	nds_Latn	39.5	**47.2**	nep_Deva	70.3	**72.8**
ngu_Latn	42.1	**44.0**	nld_Latn	61.7	**61.9**	nmf_Latn	40.8	**47.5**	nnb_Latn	36.7	**45.9**
nno_Latn	62.3	**66.4**	nob_Latn	59.3	**60.6**	nor_Latn	61.2	**61.4**	npi_Deva	70.3	**70.6**
nse_Latn	42.7	**45.6**	nso_Latn	**53.2**	52.4	nya_Latn	54.2	**61.6**	nyn_Latn	41.6	**47.3**
nyy_Latn	30.7	**38.1**	nzi_Latn	34.6	**37.6**	ori_Orya	**69.8**	69.5	ory_Orya	**70.8**	69.0
oss_Cyrl	46.7	**57.3**	ote_Latn	**35.5**	35.4	pag_Latn	50.1	**54.7**	pam_Latn	38.8	**46.0**
pan_Guru	**66.8**	65.4	pap_Latn	65.7	**66.5**	pau_Latn	41.4	**43.9**	pcm_Latn	63.3	**67.7**
pdt_Latn	58.1	**58.7**	pes_Arab	**70.3**	69.9	pis_Latn	66.5	**67.9**	pls_Latn	45.5	**50.3**
plt_Latn	**52.3**	50.7	poh_Latn	47.5	**49.4**	pol_Latn	64.4	**68.6**	pon_Latn	52.8	**53.2**
por_Latn	67.3	**72.5**	prk_Latn	55.6	**56.8**	prs_Arab	68.1	**69.9**	pxm_Latn	40.5	**41.3**
qub_Latn	56.7	**59.1**	quc_Latn	50.0	**54.0**	qug_Latn	62.1	**68.0**	quh_Latn	61.4	**68.9**
quw_Latn	52.0	**56.1**	quy_Latn	70.7	**71.1**	quz_Latn	63.8	**67.2**	qvi_Latn	61.3	**64.0**
rap_Latn	47.2	**48.4**	rar_Latn	45.6	**53.8**	rmy_Latn	44.6	**48.1**	ron_Latn	60.2	**67.6**
rop_Latn	56.0	**57.6**	rug_Latn	50.0	**55.4**	run_Latn	49.5	**54.1**	rus_Cyrl	69.3	**72.9**
sag_Latn	43.9	**46.5**	sah_Cyrl	58.5	**62.8**	sba_Latn	36.7	**41.6**	seh_Latn	46.8	**49.4**
sin_Sinh	66.2	**66.5**	slk_Latn	59.2	**60.9**	slv_Latn	61.5	**63.2**	sme_Latn	34.8	**48.0**
smo_Latn	53.5	**61.2**	sna_Latn	39.5	**45.4**	snd_Arab	67.3	**68.8**	som_Latn	31.9	**36.5**
sop_Latn	32.2	**40.2**	sot_Latn	43.9	**48.1**	spa_Latn	64.3	**68.2**	sqi_Latn	71.3	**72.1**
srm_Latn	47.6	**53.4**	srn_Latn	63.1	**65.7**	srp_Latn	64.3	**70.7**	ssw_Latn	36.6	**47.4**
sun_Latn	53.7	**56.3**	suz_Deva	57.6	**61.0**	swe_Latn	67.5	**69.9**	swh_Latn	61.0	**64.6**
sxn_Latn	46.7	**51.8**	tam_Taml	72.2	**74.3**	tat_Cyrl	64.2	**67.5**	tbz_Latn	35.1	**44.2**
tca_Latn	41.0	**49.2**	tdt_Latn	58.6	**66.6**	tel_Telu	69.8	**72.1**	teo_Latn	23.1	**26.5**
tgk_Cyrl	63.9	**66.3**	tgl_Latn	56.9	**58.8**	tha_Thai	65.2	**66.8**	tih_Latn	56.6	**60.5**
tir_Ethi	49.3	**52.2**	tlh_Latn	62.2	**66.2**	tob_Latn	40.6	**44.6**	toh_Latn	37.3	**41.7**
toi_Latn	39.4	**49.2**	toj_Latn	35.7	**40.2**	ton_Latn	46.9	**49.8**	top_Latn	21.2	**26.0**
tpi_Latn	68.4	**69.5**	tpm_Latn	43.2	**52.8**	tsn_Latn	44.2	**45.0**	tsz_Latn	35.9	**42.9**
tuc_Latn	55.5	**61.4**	tui_Latn	44.8	**47.5**	tuk_Latn	55.9	**63.0**	tum_Latn	47.9	**50.5**
tur_Latn	61.3	**67.2**	twi_Latn	40.4	**49.2**	tyv_Cyrl	56.8	**62.7**	tzh_Latn	37.9	**44.5**
tzo_Latn	37.4	**42.9**	udm_Cyrl	53.1	**54.0**	ukr_Cyrl	63.9	**69.2**	urd_Arab	**60.6**	59.7
uzb_Latn	57.6	**58.7**	uzn_Cyrl	64.3	**66.7**	ven_Latn	42.6	**46.1**	vie_Latn	69.6	**70.0**
wal_Latn	41.1	**50.4**	war_Latn	43.3	**51.1**	wbm_Latn	56.1	**56.6**	wol_Latn	32.3	**40.6**
xav_Latn	28.0	**33.6**	xho_Latn	44.5	**50.1**	yan_Latn	50.1	**52.1**	yao_Latn	38.9	**46.8**
yap_Latn	37.5	**40.5**	yom_Latn	35.4	**39.5**	yor_Latn	46.0	**48.4**	yua_Latn	35.7	**39.9**
yue_Hani	57.7	**60.2**	zai_Latn	38.5	**44.5**	zho_Hani	64.2	**67.7**	zlm_Latn	**69.4**	69.2

Table 11: F1 scores of models on **Taxi1500** (Part II).

Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP
ace_Latn	71.5	**73.6**	acm_Arab	82.2	**83.0**	afr_Latn	82.3	**82.7**	ajp_Arab	**83.4**	81.8
aka_Latn	62.2	**67.2**	als_Latn	82.4	**84.4**	amh_Ethi	**74.2**	73.6	apc_Arab	**83.9**	82.9
arb_Arab	**83.8**	82.9	ary_Arab	**81.5**	80.2	arz_Arab	**84.5**	84.1	asm_Beng	83.6	**84.2**
ast_Latn	**88.4**	88.0	ayr_Latn	51.1	**53.8**	azb_Arab	71.5	**74.7**	azj_Latn	87.0	**88.0**
bak_Cyrl	84.6	**86.6**	bam_Latn	**47.9**	47.6	ban_Latn	80.3	**83.0**	bel_Cyrl	**83.7**	83.4
bem_Latn	63.0	**63.9**	ben_Beng	83.3	**84.3**	bjn_Latn	77.1	**78.5**	bod_Tibt	**73.5**	69.2
bos_Latn	86.5	**88.2**	bul_Cyrl	86.1	**87.5**	cat_Latn	84.8	**86.4**	ceb_Latn	81.8	**84.6**
ces_Latn	**89.1**	86.9	cjk_Latn	46.6	**48.1**	ckb_Arab	**83.9**	80.2	crh_Latn	74.0	**76.2**
cym_Latn	**75.9**	75.4	dan_Latn	86.8	**87.4**	deu_Latn	86.5	**87.8**	dyu_Latn	42.6	**44.5**
dzo_Tibt	68.7	**72.6**	ell_Grek	79.5	**80.0**	eng_Latn	**90.8**	90.0	epo_Latn	**83.8**	82.2
est_Latn	80.6	**81.6**	eus_Latn	82.1	**82.2**	ewe_Latn	49.3	**51.5**	fao_Latn	83.7	**84.9**
fij_Latn	56.1	**58.0**	fin_Latn	82.1	**82.9**	fon_Latn	41.7	**44.6**	fra_Latn	87.9	**89.6**
fur_Latn	77.6	**80.2**	gla_Latn	**57.6**	54.3	gle_Latn	62.2	**64.1**	glg_Latn	87.8	**89.0**
grn_Latn	**75.0**	74.5	guj_Gujr	83.9	**84.7**	hat_Latn	77.4	**79.1**	hau_Latn	**62.7**	62.1
heb_Hebr	77.9	**79.2**	hin_Deva	84.1	**84.4**	hne_Deva	77.9	**80.1**	hrv_Latn	87.3	**89.0**
hun_Latn	86.8	**87.6**	hye_Armn	**83.0**	82.5	ibo_Latn	72.3	**74.1**	ilo_Latn	75.8	**79.6**
ind_Latn	88.7	**89.1**	isl_Latn	78.5	**79.1**	ita_Latn	87.7	**89.2**	jav_Latn	80.2	**80.3**
jpn_Jpan	87.1	**87.9**	kab_Latn	31.1	**36.9**	kac_Latn	49.3	**52.3**	kam_Latn	49.1	**49.5**
kan_Knda	**83.2**	82.0	kat_Geor	81.8	**83.7**	kaz_Cyrl	84.2	**84.9**	kbp_Latn	**45.1**	44.2
kea_Latn	75.4	**77.0**	khm_Khmr	84.3	**84.4**	kik_Latn	57.1	**59.9**	kin_Latn	69.5	**70.5**
kir_Cyrl	**80.7**	80.3	kmb_Latn	48.2	**49.5**	kmr_Latn	**70.7**	70.0	kon_Latn	65.3	**69.2**
kor_Hang	**85.2**	83.9	lao_Laoo	**85.1**	84.2	lij_Latn	77.7	**79.6**	lim_Latn	74.7	**75.2**
lin_Latn	69.3	**71.4**	lit_Latn	**86.5**	84.7	lmo_Latn	77.7	**79.1**	ltz_Latn	76.6	**79.1**
lua_Latn	**59.1**	56.4	lug_Latn	55.5	**59.1**	luo_Latn	52.6	**53.0**	lus_Latn	65.3	**67.9**
lvs_Latn	**84.4**	83.6	mai_Deva	83.4	**84.0**	mal_Mlym	**80.6**	79.9	mar_Deva	**84.1**	82.5
min_Latn	77.7	**79.6**	mkd_Cyrl	83.3	**84.6**	mlt_Latn	82.9	**83.0**	mos_Latn	44.9	**46.6**
mri_Latn	54.4	**59.3**	mya_Mymr	80.1	**81.6**	nld_Latn	**86.5**	85.8	nno_Latn	**86.6**	86.4
nob_Latn	85.8	**86.1**	npi_Deva	**86.8**	86.0	nso_Latn	61.3	**61.9**	nya_Latn	71.1	**72.7**
oci_Latn	83.1	**84.9**	ory_Orya	79.7	**80.3**	pag_Latn	78.7	**79.7**	pan_Guru	77.4	**79.0**
pap_Latn	77.2	**79.0**	pes_Arab	87.6	**89.2**	plt_Latn	68.4	**68.5**	pol_Latn	86.4	**86.7**
por_Latn	87.3	**88.6**	prs_Arab	85.8	**88.4**	quy_Latn	63.7	**64.0**	ron_Latn	**86.4**	84.5
run_Latn	**68.3**	67.2	rus_Cyrl	87.6	**87.9**	sag_Latn	52.4	**55.1**	san_Deva	**77.9**	77.8
sat_Olck	53.0	**57.4**	scn_Latn	77.6	**78.2**	sin_Sinh	**84.5**	84.1	slk_Latn	86.1	**87.0**
slv_Latn	**86.4**	85.5	smo_Latn	73.4	**74.1**	sna_Latn	**59.3**	58.0	snd_Arab	72.1	**76.9**
som_Latn	**61.8**	59.8	sot_Latn	65.3	**67.6**	spa_Latn	**86.4**	86.2	srd_Latn	74.0	**75.8**
srp_Cyrl	**85.8**	85.2	ssw_Latn	67.5	**68.1**	sun_Latn	84.0	**85.2**	swe_Latn	86.6	**87.3**
swh_Latn	76.0	**78.6**	szl_Latn	74.3	**75.5**	tam_Taml	80.6	**84.3**	tat_Cyrl	84.0	**85.2**
tel_Telu	85.3	**85.7**	tgk_Cyrl	**81.6**	80.9	tgl_Latn	81.9	**83.0**	tha_Thai	87.4	**88.9**
tir_Ethi	59.9	**61.4**	tpi_Latn	80.6	**82.3**	tsn_Latn	**59.1**	55.2	tso_Latn	59.3	**61.2**
tuk_Latn	**78.3**	78.2	tum_Latn	70.3	**70.8**	tur_Latn	82.9	**83.6**	twi_Latn	61.4	**68.0**
uig_Arab	77.7	**80.0**	ukr_Cyrl	**84.7**	84.5	umb_Latn	**45.9**	45.8	urd_Arab	81.3	**81.9**
vec_Latn	**82.0**	81.1	vie_Latn	84.9	**85.8**	war_Latn	81.7	**83.4**	wol_Latn	49.2	**52.1**
xho_Latn	62.4	**64.0**	yor_Latn	46.6	**51.8**	zsm_Latn	**87.2**	86.6	zul_Latn	**73.8**	73.6

Table 12: F1 scores of models on **SIB200**.

Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP
ace_Latn	41.8	**42.2**	afr_Latn	76.5	**77.4**	als_Latn	**82.4**	**82.4**	amh_Ethi	**48.9**	41.0
ara_Arab	**57.1**	54.4	arg_Latn	78.0	**82.2**	arz_Arab	55.6	**57.5**	asm_Beng	65.8	**68.1**
ast_Latn	83.0	**84.9**	aym_Latn	**45.9**	44.3	aze_Latn	63.3	**66.0**	bak_Cyrl	60.4	**62.3**
bar_Latn	68.6	**70.1**	bel_Cyrl	**74.6**	**74.6**	ben_Beng	**72.7**	71.7	bih_Deva	**56.2**	55.3
bod_Tibt	18.1	**38.6**	bos_Latn	72.1	**73.8**	bre_Latn	63.3	**64.3**	bul_Cyrl	**75.0**	74.5
cat_Latn	83.3	**84.1**	cbk_Latn	**53.8**	52.5	ceb_Latn	53.8	**56.7**	ces_Latn	78.6	**78.7**
che_Cyrl	25.3	**56.5**	chv_Cyrl	**80.0**	73.5	ckb_Arab	72.9	**74.4**	cos_Latn	55.6	**57.0**
crh_Latn	**51.0**	49.0	csb_Latn	58.5	**60.6**	cym_Latn	**63.7**	59.6	dan_Latn	81.1	**81.6**
deu_Latn	76.5	**76.8**	diq_Latn	**55.2**	54.1	div_Thaa	43.0	**53.2**	ell_Grek	73.2	**74.1**
eml_Latn	42.3	**42.9**	eng_Latn	**83.7**	83.3	epo_Latn	67.5	**71.4**	est_Latn	72.3	**74.8**
eus_Latn	56.4	**57.0**	ext_Latn	45.1	**49.8**	fao_Latn	**71.1**	69.0	fas_Arab	**51.8**	50.0
fin_Latn	75.0	**75.2**	fra_Latn	76.4	**77.6**	frr_Latn	**55.9**	54.8	fry_Latn	**77.4**	77.2
fur_Latn	55.3	**55.7**	gla_Latn	59.8	**64.7**	gle_Latn	72.8	**72.9**	glg_Latn	80.1	**81.5**
grn_Latn	**56.0**	55.7	guj_Gujr	54.3	**58.9**	hbs_Latn	62.6	**63.8**	heb_Hebr	49.3	**50.7**
hin_Deva	69.3	**69.5**	hrv_Latn	77.3	**77.8**	hsb_Latn	73.6	**73.8**	hun_Latn	76.0	**77.4**
hye_Armn	**55.9**	55.4	ibo_Latn	**59.1**	55.2	ido_Latn	**81.9**	79.7	ilo_Latn	72.7	**74.7**
ina_Latn	58.0	**58.4**	ind_Latn	**64.7**	62.1	isl_Latn	**72.4**	71.6	ita_Latn	77.9	**79.2**
jav_Latn	**56.1**	54.9	jbo_Latn	**22.9**	**22.9**	jpn_Jpan	**21.3**	15.3	kan_Knda	58.2	**63.4**
kat_Geor	67.4	**67.8**	kaz_Cyrl	50.8	**50.9**	khm_Khmr	43.2	**46.9**	kin_Latn	**67.6**	66.7
kir_Cyrl	**48.4**	42.3	kor_Hang	**53.6**	51.9	ksh_Latn	56.7	**60.9**	kur_Latn	62.5	**65.2**
lat_Latn	**74.2**	73.5	lav_Latn	73.2	**75.2**	lij_Latn	41.4	**47.1**	lim_Latn	66.7	**67.8**
lin_Latn	49.5	**49.8**	lit_Latn	**75.3**	75.0	lmo_Latn	**76.3**	72.5	ltz_Latn	68.5	**68.9**
lzh_Hani	**14.0**	7.3	mal_Mlym	**65.1**	63.2	mar_Deva	**65.2**	61.7	mhr_Cyrl	59.8	**61.6**
min_Latn	**44.2**	43.4	mkd_Cyrl	76.3	**76.9**	mlg_Latn	**59.4**	57.8	mlt_Latn	64.6	**74.0**
mon_Cyrl	**67.5**	66.1	mri_Latn	**50.4**	46.3	msa_Latn	68.8	**69.0**	mwl_Latn	48.5	**51.5**
mya_Mymr	**57.9**	54.5	mzn_Arab	46.2	**46.9**	nan_Latn	86.5	**86.7**	nap_Latn	62.5	**62.6**
nds_Latn	**80.9**	75.8	nep_Deva	56.5	**61.0**	nld_Latn	81.4	**81.5**	nno_Latn	**76.9**	76.4
nor_Latn	75.9	**77.9**	oci_Latn	68.2	**72.6**	ori_Orya	**28.6**	**28.6**	oss_Cyrl	**58.8**	50.6
pan_Guru	45.3	**46.5**	pms_Latn	75.0	**80.9**	pnb_Arab	**68.1**	67.8	pol_Latn	**77.9**	77.8
por_Latn	76.8	**79.8**	pus_Arab	**44.2**	40.0	que_Latn	62.4	**66.4**	roh_Latn	**61.7**	56.9
ron_Latn	78.7	**78.9**	rus_Cyrl	**70.3**	69.5	sah_Cyrl	**71.8**	71.4	san_Deva	34.6	**36.6**
scn_Latn	65.2	**69.1**	sco_Latn	82.0	**91.5**	sgs_Latn	61.8	**67.2**	sin_Sinh	**58.3**	54.5
slk_Latn	77.0	**77.7**	slv_Latn	79.2	**80.3**	snd_Arab	**43.6**	41.0	som_Latn	52.8	**58.9**
spa_Latn	73.0	**78.6**	sqi_Latn	75.6	**77.1**	srp_Cyrl	**64.8**	63.6	sun_Latn	**56.3**	55.6
swa_Latn	68.3	**68.9**	swe_Latn	**70.2**	68.7	szl_Latn	67.0	**70.9**	tam_Taml	55.4	**59.3**
tat_Cyrl	**68.2**	60.5	tel_Telu	**52.3**	50.5	tgk_Cyrl	60.8	**61.4**	tgl_Latn	75.8	**76.4**
tha_Thai	**5.0**	0.9	tuk_Latn	55.5	**57.1**	tur_Latn	76.1	**77.2**	uig_Arab	**50.2**	47.6
ukr_Cyrl	**77.2**	76.4	urd_Arab	**69.8**	63.5	uzb_Latn	**74.0**	72.9	vec_Latn	**69.6**	65.9
vep_Latn	**70.2**	68.0	vie_Latn	72.3	**73.2**	vls_Latn	73.7	**77.6**	vol_Latn	56.7	**61.0**
war_Latn	**62.8**	**62.8**	wuu_Hani	**40.8**	19.4	xmf_Geor	**65.3**	60.8	yid_Hebr	47.5	**58.2**
yor_Latn	65.5	**65.8**	yue_Hani	**23.5**	18.4	zea_Latn	63.0	**65.8**	zho_Hani	**24.7**	18.1

Table 13: F1 scores of models on **NER**.

Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP	Language	Baseline	LANGSAMP
afr_Latn	88.0	**88.2**	ajp_Arab	**71.1**	68.8	aln_Latn	**51.9**	50.6	amh_Ethi	**67.8**	65.4
ara_Arab	66.9	**67.4**	bam_Latn	41.4	**43.5**	bel_Cyrl	**85.9**	85.1	ben_Beng	**83.7**	82.1
bre_Latn	**60.7**	59.4	bul_Cyrl	**88.6**	87.9	cat_Latn	**86.5**	86.1	ceb_Latn	**65.9**	63.1
ces_Latn	**85.1**	84.6	cym_Latn	**66.4**	64.7	dan_Latn	90.3	**90.7**	deu_Latn	**87.9**	87.4
ell_Grek	**86.6**	84.6	eng_Latn	**96.0**	95.9	est_Latn	83.7	**83.9**	eus_Latn	**65.3**	62.1
fao_Latn	**89.3**	88.1	fas_Arab	71.5	**72.6**	fin_Latn	**82.2**	81.7	fra_Latn	**86.7**	**86.7**
gla_Latn	57.0	**57.3**	gle_Latn	64.1	**64.9**	glg_Latn	**83.0**	82.1	glv_Latn	**50.7**	50.2
grc_Grek	**72.6**	71.9	grn_Latn	**20.9**	20.0	gsw_Latn	79.2	**80.3**	hbo_Hebr	37.1	**38.4**
heb_Hebr	**69.8**	68.7	hin_Deva	69.6	**72.7**	hrv_Latn	**85.8**	85.3	hsb_Latn	**82.7**	82.4
hun_Latn	81.3	**83.1**	hye_Armn	84.2	**84.9**	hyw_Armn	**81.6**	81.5	ind_Latn	**84.0**	83.1
isl_Latn	**82.8**	**82.8**	ita_Latn	88.3	**88.8**	jav_Latn	**73.6**	72.7	jpn_Jpan	25.0	**35.3**
kaz_Cyrl	**76.9**	75.2	kmr_Latn	**74.0**	73.8	kor_Hang	**52.7**	51.8	lat_Latn	**72.6**	72.2
lav_Latn	**84.0**	83.6	lij_Latn	**77.4**	76.3	lit_Latn	**81.5**	80.9	lzh_Hani	22.7	**24.3**
mal_Mlym	**86.3**	84.2	mar_Deva	**81.7**	77.9	mlt_Latn	79.4	**79.8**	myv_Cyrl	**64.2**	63.5
nap_Latn	**82.4**	**82.4**	nds_Latn	77.0	**77.9**	nld_Latn	88.3	**88.4**	nor_Latn	**88.1**	87.8
pcm_Latn	56.9	**57.3**	pol_Latn	**84.2**	82.7	por_Latn	**88.2**	87.8	quc_Latn	**63.8**	59.7
ron_Latn	81.4	**82.0**	rus_Cyrl	**89.0**	88.4	sah_Cyrl	**75.7**	71.5	san_Deva	**25.6**	24.8
sin_Sinh	**56.0**	55.7	slk_Latn	**84.8**	**84.8**	slv_Latn	**77.2**	76.7	sme_Latn	**73.2**	72.3
spa_Latn	**87.5**	87.1	sqi_Latn	76.0	**77.4**	srp_Latn	**85.4**	85.0	swe_Latn	**92.6**	92.4
tam_Taml	73.8	**73.9**	tat_Cyrl	70.4	**70.8**	tel_Telu	**81.7**	80.9	tgl_Latn	**75.2**	74.1
tha_Thai	58.3	**58.9**	tur_Latn	**71.3**	70.7	uig_Arab	**68.4**	67.3	ukr_Cyrl	**85.1**	85.0
urd_Arab	59.0	**67.0**	vie_Latn	**68.2**	67.5	wol_Latn	**60.9**	59.9	xav_Latn	**11.1**	9.2

Table 14: F1 scores of models on **POS**.

Language	English	Closest donor	Language	English	Closest donor	Language	English	Closest donor	Language	English	Closest donor
ace_Latn	**63.3**	60.1	ach_Latn	35.6	**48.1**	acr_Latn	**48.8**	46.7	afr_Latn	**58.6**	58.5
ahk_Latn	5.4	**8.3**	aka_Latn	**44.9**	41.2	aln_Latn	**56.2**	54.7	als_Latn	**57.1**	57.1
alz_Latn	34.1	**43.0**	aoj_Latn	40.9	**46.2**	arb_Arab	**55.4**	55.4	arn_Latn	43.1	**44.4**
arz_Arab	33.7	**40.3**	asm_Beng	53.4	**61.5**	ayr_Latn	52.7	**62.2**	azb_Arab	**61.0**	61.0
bak_Cyrl	54.7	**59.7**	bam_Latn	48.9	**55.6**	ban_Latn	**43.0**	42.5	bar_Latn	**47.8**	43.3
bci_Latn	34.6	**37.1**	bcl_Latn	54.2	**60.5**	bel_Cyrl	59.1	**61.5**	bem_Latn	44.2	**49.4**
bhw_Latn	**50.2**	46.9	bim_Latn	47.3	**55.1**	bis_Latn	**68.4**	68.1	bqc_Latn	33.2	**41.6**
btx_Latn	**56.7**	53.8	bul_Cyrl	62.5	**62.6**	bum_Latn	39.6	**42.2**	bzj_Latn	**65.7**	60.3
cac_Latn	43.8	**46.0**	cak_Latn	51.0	**57.9**	caq_Latn	42.7	**51.0**	cat_Latn	61.2	**62.3**
cce_Latn	**43.8**	38.0	ceb_Latn	**49.8**	49.1	ces_Latn	63.3	**63.7**	cfm_Latn	**58.3**	57.1
chk_Latn	**42.8**	38.9	chv_Cyrl	60.3	**64.3**	ckb_Arab	58.3	**67.0**	cmn_Hani	60.8	**73.0**
crh_Cyrl	61.4	**67.7**	crs_Latn	62.3	**63.5**	csy_Latn	**58.3**	56.7	ctd_Latn	**56.6**	55.8
cuk_Latn	39.1	**40.8**	cym_Latn	**51.9**	46.0	dan_Latn	**58.1**	54.0	deu_Latn	**51.5**	51.5
dln_Latn	**54.4**	54.4	dtp_Latn	51.5	**51.6**	dyu_Latn	**55.6**	48.2	dzo_Tibt	50.6	**58.1**
ell_Grek	**56.9**	53.9	eng_Latn	**78.0**	78.0	enm_Latn	**70.8**	67.0	epo_Latn	**58.3**	58.3
eus_Latn	**25.2**	21.4	ewe_Latn	46.4	**52.1**	fao_Latn	56.5	**64.8**	fas_Arab	69.6	**70.2**
fil_Latn	56.7	**58.7**	fin_Latn	**56.4**	55.7	fon_Latn	**36.8**	35.4	fra_Latn	**66.8**	66.8
gaa_Latn	36.9	**47.7**	gil_Latn	40.4	**47.2**	giz_Latn	48.4	**48.5**	gkn_Latn	**40.0**	34.1
gla_Latn	**45.6**	45.6	gle_Latn	41.8	**45.1**	glv_Latn	37.3	**48.7**	gom_Latn	34.8	**41.6**
guc_Latn	**39.6**	37.6	gug_Latn	39.0	**46.0**	guj_Gujr	67.1	**70.4**	gur_Latn	37.0	**44.2**
gya_Latn	39.6	**41.8**	gym_Latn	45.4	**52.9**	hat_Latn	**63.0**	60.0	hau_Latn	54.0	**59.6**
heb_Hebr	**16.7**	15.2	hif_Latn	42.4	**53.6**	hil_Latn	**63.7**	61.6	hin_Deva	**64.8**	64.8
hne_Deva	64.1	**67.5**	hnj_Latn	61.5	**63.2**	hra_Latn	48.2	**53.1**	hrv_Latn	**62.7**	60.7
hun_Latn	65.2	**65.9**	hus_Latn	37.6	**40.7**	hye_Armn	67.2	**69.3**	iba_Latn	57.9	**59.2**
ifa_Latn	49.7	**51.5**	ifb_Latn	**48.3**	48.1	ikk_Latn	46.6	**52.5**	ilo_Latn	**58.8**	55.7
isl_Latn	53.5	**61.2**	ita_Latn	62.8	**67.1**	ium_Latn	51.4	**58.0**	ixl_Latn	36.6	**38.2**
jam_Latn	**66.1**	61.0	jav_Latn	43.9	**47.6**	jpn_Jpan	**58.6**	58.6	kaa_Latn	57.7	**62.6**
kac_Latn	44.5	**47.3**	kal_Latn	31.5	**34.5**	kan_Knda	60.6	**67.5**	kat_Geor	55.2	**62.2**
kbp_Latn	34.9	**39.5**	kek_Latn	**41.5**	40.3	khm_Khmr	**64.7**	64.7	kia_Latn	48.0	**51.7**
kin_Latn	47.2	**52.5**	kir_Cyrl	61.1	**64.7**	kjb_Latn	44.7	**48.1**	kjh_Cyrl	**52.3**	51.1
kmr_Cyrl	45.5	**53.1**	knv_Latn	**42.6**	40.5	kor_Hang	69.8	**71.3**	kpg_Latn	**64.1**	57.4
kri_Latn	**63.2**	56.0	ksd_Latn	54.2	**54.4**	kss_Latn	16.2	**21.6**	ksw_Mymr	**50.4**	50.3
lam_Latn	34.7	**35.6**	lao_Laoo	69.1	**72.7**	lat_Latn	57.2	**62.9**	lav_Latn	**60.4**	57.7
leh_Latn	**43.5**	37.2	lhu_Latn	22.3	**29.0**	lin_Latn	47.1	**54.7**	lit_Latn	58.3	**59.7**
ltz_Latn	**48.2**	48.2	lug_Latn	**46.1**	39.0	luo_Latn	40.6	**41.2**	lus_Latn	**51.6**	51.6
mad_Latn	55.3	**63.0**	mah_Latn	**41.6**	38.3	mai_Deva	**62.7**	60.5	mam_Latn	**33.9**	33.2
mau_Latn	5.5	**8.4**	mbb_Latn	52.6	**53.1**	mck_Latn	**41.9**	41.2	mcn_Latn	37.7	**39.3**
mdy_Ethi	51.6	**57.6**	meu_Latn	54.9	**55.8**	mfe_Latn	66.0	**66.2**	mgh_Latn	30.3	**33.1**
mhr_Cyrl	36.0	**38.5**	min_Latn	**49.9**	40.7	miq_Latn	**52.2**	52.2	mkd_Cyrl	**71.2**	70.3
mlt_Latn	**50.7**	50.7	mos_Latn	40.3	**41.2**	mps_Latn	**57.1**	53.1	mri_Latn	50.9	**52.6**

Table 15: F1 scores of LANGSAMP on **Taxi1500** using English and the closest donor language as source (Part I).

Language	English	Closest donor	Language	English	Closest donor	Language	English	Closest donor	Language	English	Closest donor
msa_Latn	41.7	**42.0**	mwm_Latn	**55.1**	55.0	mxv_Latn	**29.6**	27.4	mya_Mymr	**54.4**	53.4
mzh_Latn	39.7	**45.1**	nan_Latn	31.5	**31.8**	naq_Latn	41.7	**43.7**	nav_Latn	21.1	**29.5**
nch_Latn	**44.0**	36.6	ncj_Latn	38.6	**39.1**	ndc_Latn	34.7	**36.6**	nde_Latn	45.7	**49.8**
nds_Latn	**49.6**	44.0	nep_Deva	68.0	**72.1**	ngu_Latn	43.4	**48.2**	nld_Latn	**61.1**	53.7
nnb_Latn	40.7	**46.1**	nno_Latn	**63.1**	63.1	nob_Latn	57.2	**58.2**	nor_Latn	56.4	**57.8**
nse_Latn	45.9	**48.5**	nso_Latn	**48.6**	48.6	nya_Latn	**56.0**	47.4	nyn_Latn	43.0	**44.1**
nzi_Latn	33.0	**33.8**	ori_Orya	**67.3**	67.3	ory_Orya	66.9	**70.7**	oss_Cyrl	55.5	**57.5**
pag_Latn	**55.5**	52.5	pam_Latn	**42.0**	37.8	pan_Guru	**64.1**	64.1	pap_Latn	**65.6**	59.8
pcm_Latn	**66.1**	65.9	pdt_Latn	**60.0**	56.5	pes_Arab	**69.0**	69.0	pis_Latn	64.3	**65.0**
plt_Latn	46.8	**52.9**	poh_Latn	44.3	**45.5**	pol_Latn	64.8	**65.1**	pon_Latn	50.5	**52.2**
prk_Latn	52.9	**53.0**	prs_Arab	69.2	**70.0**	pxm_Latn	34.5	**41.5**	qub_Latn	51.5	**56.3**
qug_Latn	**65.0**	61.3	quh_Latn	**66.7**	58.8	quw_Latn	55.9	**56.0**	quy_Latn	65.5	**67.7**
qvi_Latn	**62.0**	58.5	rap_Latn	48.9	**49.3**	rar_Latn	48.9	**51.9**	rmy_Latn	45.4	**49.1**
rop_Latn	**56.6**	54.7	rug_Latn	53.8	**55.1**	run_Latn	48.0	**55.2**	rus_Cyrl	**68.1**	68.1
sah_Cyrl	55.1	**57.6**	sba_Latn	39.1	**41.4**	seh_Latn	45.0	**46.7**	sin_Sinh	64.1	**66.9**
slv_Latn	**63.8**	60.7	sme_Latn	**42.8**	37.6	smo_Latn	**60.8**	54.2	sna_Latn	42.6	**44.9**
som_Latn	33.9	**35.5**	sop_Latn	**36.4**	36.0	sot_Latn	43.5	**45.5**	spa_Latn	**64.2**	64.2
srm_Latn	48.1	**48.4**	srn_Latn	**63.7**	62.8	srp_Latn	64.9	**65.2**	ssw_Latn	**43.7**	37.7
suz_Deva	**58.0**	57.8	swe_Latn	**66.8**	65.3	swh_Latn	**59.8**	59.8	sxn_Latn	**46.6**	40.2
tat_Cyrl	62.2	**68.2**	tbz_Latn	36.4	**39.5**	tca_Latn	43.3	**50.3**	tdt_Latn	**60.3**	55.1
teo_Latn	**23.7**	23.1	tgk_Cyrl	**60.9**	60.9	tgl_Latn	56.7	**58.7**	tha_Thai	**63.8**	63.8
tir_Ethi	**50.1**	50.1	tlh_Latn	**65.0**	65.0	tob_Latn	43.3	**50.4**	toh_Latn	37.1	**39.0**
toj_Latn	**36.6**	34.1	ton_Latn	47.3	**51.5**	top_Latn	**21.9**	21.3	tpi_Latn	63.8	**67.6**
tsn_Latn	39.8	**44.1**	tsz_Latn	40.4	**41.0**	tuc_Latn	**57.4**	56.9	tui_Latn	**43.7**	43.7
tum_Latn	**47.6**	43.2	tur_Latn	**62.1**	62.1	twi_Latn	**41.4**	38.9	tyv_Cyrl	59.8	**60.3**
tzo_Latn	**39.5**	39.5	udm_Cyrl	49.6	**49.9**	ukr_Cyrl	**62.4**	62.2	uzb_Latn	53.5	**57.7**
ven_Latn	41.9	**48.6**	vie_Latn	62.4	**65.4**	wal_Latn	**48.9**	42.7	war_Latn	47.7	**54.5**
wol_Latn	**37.2**	33.9	xav_Latn	**25.5**	23.7	xho_Latn	**44.9**	44.4	yan_Latn	50.3	**53.5**
yap_Latn	42.8	**42.9**	yom_Latn	**37.6**	34.1	yor_Latn	**41.8**	35.4	yua_Latn	40.1	**43.2**
zai_Latn	**42.6**	41.4	zho_Hani	**60.7**	60.7	zlm_Latn	**68.4**	65.5	zom_Latn	**44.6**	44.4
zul_Latn	51.9	**52.2**									

Table 16: F1 scores of LANGSAMP on **Taxi1500** using English and the closest donor language as source (Part II).

Language	English	Closest donor	Language	English	Closest donor	Language	English	Closest donor	Language	English	Closest donor
ace_Latn	69.9	**72.4**	acm_Arab	80.6	**81.4**	afr_Latn	81.4	**81.8**	ajp_Arab	81.4	**83.0**
als_Latn	**82.3**	82.3	amh_Ethi	**72.6**	72.6	apc_Arab	81.7	**83.2**	arb_Arab	**81.5**	81.5
arz_Arab	82.1	**84.4**	asm_Beng	**83.0**	83.0	ast_Latn	87.1	**87.6**	ayr_Latn	48.6	**51.1**
azj_Latn	**86.5**	84.0	bak_Cyrl	84.3	**86.5**	bam_Latn	**46.5**	42.2	ban_Latn	79.5	**81.3**
bem_Latn	**61.1**	51.4	ben_Beng	83.7	**84.0**	bjn_Latn	75.9	**77.9**	bod_Tibt	65.7	**71.0**
bul_Cyrl	86.3	**86.6**	cat_Latn	**85.7**	85.2	ceb_Latn	81.2	**83.2**	ces_Latn	**86.3**	85.6
ckb_Arab	**80.0**	76.8	crh_Latn	**76.8**	75.7	cym_Latn	73.6	**76.6**	dan_Latn	85.0	**86.0**
dyu_Latn	**43.6**	42.4	dzo_Tibt	**68.2**	59.8	ell_Grek	**79.5**	78.8	eng_Latn	**88.9**	88.9
est_Latn	**78.9**	78.1	eus_Latn	78.8	**80.7**	ewe_Latn	**49.9**	46.7	fao_Latn	**84.4**	83.6
fin_Latn	80.9	**81.5**	fon_Latn	**40.8**	38.1	fra_Latn	**87.8**	87.8	fur_Latn	77.4	**77.9**
gle_Latn	61.5	**64.4**	glg_Latn	**87.6**	87.6	grn_Latn	71.6	**73.2**	guj_Gujr	82.1	**83.4**
hau_Latn	59.3	**64.2**	heb_Hebr	76.8	**80.2**	hin_Deva	**82.8**	82.8	hne_Deva	77.9	**79.5**
hun_Latn	86.6	**87.5**	hye_Armn	**81.3**	80.3	ibo_Latn	**71.4**	71.3	ilo_Latn	76.1	**76.7**
isl_Latn	78.0	**78.3**	ita_Latn	86.4	**87.5**	jav_Latn	**79.9**	79.7	jpn_Jpan	**86.8**	86.8
kac_Latn	**48.9**	46.6	kam_Latn	45.8	**48.3**	kan_Knda	82.9	**83.0**	kat_Geor	**83.7**	81.0
kbp_Latn	**42.8**	42.2	kea_Latn	**73.1**	73.1	khm_Khmr	**82.7**	82.7	kik_Latn	55.1	**56.7**
kir_Cyrl	79.3	**80.1**	kmb_Latn	**46.2**	42.6	kmr_Latn	**69.8**	68.9	kon_Latn	**65.2**	63.4
lao_Laoo	**83.4**	82.9	lij_Latn	**76.4**	74.9	lim_Latn	**74.1**	73.0	lin_Latn	68.2	**73.3**
lmo_Latn	77.0	**78.3**	ltz_Latn	**76.4**	76.4	lua_Latn	**54.4**	54.3	lug_Latn	**58.2**	55.8
lus_Latn	**64.8**	64.8	lvs_Latn	**83.2**	83.0	mai_Deva	**82.9**	82.1	mal_Mlym	**79.8**	79.3
min_Latn	76.7	**79.8**	mkd_Cyrl	**83.6**	82.8	mlt_Latn	**81.3**	81.3	mos_Latn	**44.7**	40.9
mya_Mymr	**80.5**	78.8	nld_Latn	85.1	**86.4**	nno_Latn	**86.0**	86.0	nob_Latn	**84.8**	84.4
nso_Latn	**57.6**	57.6	nya_Latn	69.2	**70.9**	oci_Latn	**85.0**	84.1	ory_Orya	78.6	**79.0**
pan_Guru	**76.4**	76.4	pap_Latn	76.9	**78.1**	pes_Arab	**87.5**	87.3	plt_Latn	67.5	**69.3**
por_Latn	85.3	**86.8**	prs_Arab	85.0	**85.5**	quy_Latn	**62.6**	59.7	ron_Latn	84.0	**84.4**
rus_Cyrl	**86.8**	86.8	sag_Latn	**51.3**	50.2	san_Deva	72.9	**76.6**	sat_Olck	**56.4**	53.5
sin_Sinh	**82.7**	82.7	slk_Latn	**85.4**	85.1	slv_Latn	84.2	**87.4**	smo_Latn	74.2	**75.3**
snd_Arab	**70.4**	70.4	som_Latn	58.9	**61.1**	sot_Latn	**64.1**	63.2	spa_Latn	**84.4**	84.4
srp_Cyrl	84.8	**85.0**	ssw_Latn	64.1	**65.2**	sun_Latn	82.6	**85.2**	swe_Latn	84.2	**86.2**
szl_Latn	**72.4**	72.4	tam_Taml	**81.2**	81.2	tat_Cyrl	**83.6**	83.6	tel_Telu	84.0	**85.4**
tgl_Latn	**82.1**	81.7	tha_Thai	85.4	**85.7**	tir_Ethi	**60.3**	60.3	tpi_Latn	**80.3**	75.7
tso_Latn	57.3	**60.3**	tuk_Latn	78.1	**78.5**	tum_Latn	65.4	**68.5**	tur_Latn	**80.4**	80.4
uig_Arab	**75.5**	75.5	ukr_Cyrl	**84.3**	83.8	umb_Latn	41.0	**46.5**	urd_Arab	79.1	**80.6**
vie_Latn	**86.2**	83.9	war_Latn	80.7	**81.3**	wol_Latn	**50.5**	46.4	xho_Latn	**60.1**	59.8
zho_Hans	**89.6**	89.2	zho_Hant	**88.8**	88.8	zsm_Latn	**86.4**	86.0	zul_Latn	68.1	**69.8**

Table 17: F1 scores of LANGSAMP on **SIB200**. using English and the closest donor language as source.

Language	English	Closest donor	Language	English	Closest donor	Language	English	Closest donor	Language	English	Closest donor
ace_Latn	41.5	**56.9**	afr_Latn	75.8	**80.3**	als_Latn	**80.9**	80.9	amh_Ethi	**39.7**	39.7
arg_Latn	82.2	**88.8**	arz_Arab	55.1	**82.6**	asm_Beng	**69.0**	45.9	ast_Latn	84.6	**85.8**
aze_Latn	65.0	**74.0**	bak_Cyrl	62.5	**72.2**	bar_Latn	**68.2**	62.8	bel_Cyrl	74.9	**79.7**
bih_Deva	56.2	**67.6**	bod_Tibt	35.2	**35.7**	bos_Latn	70.1	**75.2**	bre_Latn	63.3	**66.0**
cat_Latn	83.8	**85.1**	cbk_Latn	**53.7**	48.9	ceb_Latn	**56.0**	26.8	ces_Latn	**77.9**	69.6
chv_Cyrl	73.6	**84.3**	ckb_Arab	**76.0**	60.6	cos_Latn	**63.0**	61.9	crh_Latn	52.7	**59.4**
cym_Latn	61.7	**62.1**	dan_Latn	**81.4**	81.3	deu_Latn	**74.6**	74.6	diq_Latn	54.0	**72.2**
ell_Grek	71.9	**72.0**	eml_Latn	**41.3**	41.3	eng_Latn	**83.5**	83.5	epo_Latn	**68.3**	68.3
eus_Latn	60.9	**65.1**	ext_Latn	44.2	**48.6**	fao_Latn	68.7	**79.2**	fas_Arab	**55.0**	53.6
fra_Latn	**76.5**	76.5	frr_Latn	**52.0**	52.0	fry_Latn	**74.6**	73.9	fur_Latn	**58.2**	54.0
gle_Latn	**72.6**	69.6	glg_Latn	80.7	**86.1**	grn_Latn	55.1	**59.8**	guj_Gujr	**61.2**	61.0
heb_Hebr	52.0	**52.9**	hin_Deva	**69.4**	69.4	hrv_Latn	77.2	**79.8**	hsb_Latn	**74.3**	69.7
hye_Armn	53.0	**62.2**	ibo_Latn	58.1	**58.4**	ido_Latn	**82.6**	81.5	ilo_Latn	**80.0**	74.9
ind_Latn	**67.6**	67.6	isl_Latn	70.1	**75.4**	ita_Latn	78.2	**79.5**	jav_Latn	56.0	**86.4**
jpn_Jpan	**22.0**	22.0	kan_Knda	57.5	**61.8**	kat_Geor	**68.7**	60.1	kaz_Cyrl	50.5	**57.1**
kin_Latn	**69.6**	67.3	kir_Cyrl	44.3	**60.9**	kor_Hang	50.4	**51.2**	ksh_Latn	**59.7**	51.4
lat_Latn	71.9	**81.4**	lav_Latn	**74.4**	69.0	lij_Latn	45.2	**54.2**	lim_Latn	**69.3**	61.2
lit_Latn	74.2	**76.1**	lmo_Latn	**73.6**	65.5	ltz_Latn	**67.9**	67.9	lzh_Hani	**14.8**	14.8
mar_Deva	62.5	**76.6**	mhr_Cyrl	60.6	**72.3**	min_Latn	42.6	**57.5**	mkd_Cyrl	72.2	**73.1**
mlt_Latn	**75.9**	75.9	mon_Cyrl	**68.7**	60.9	mri_Latn	**50.0**	47.0	msa_Latn	67.6	**73.0**
mya_Mymr	55.3	**56.3**	mzn_Arab	43.3	**47.2**	nan_Latn	**88.1**	36.6	nap_Latn	**63.0**	55.3
nep_Deva	56.9	**60.4**	nld_Latn	**80.8**	80.0	nno_Latn	**77.6**	77.6	nor_Latn	77.9	**80.4**
ori_Orya	**34.2**	34.2	oss_Cyrl	50.6	**59.1**	pan_Guru	**51.5**	51.5	pms_Latn	**80.9**	78.4
pol_Latn	**77.7**	71.1	por_Latn	78.9	**84.9**	pus_Arab	42.6	**45.3**	que_Latn	**70.4**	55.5
ron_Latn	**77.8**	75.5	rus_Cyrl	**67.5**	67.5	sah_Cyrl	71.9	**77.9**	san_Deva	38.4	**53.4**
sco_Latn	**86.4**	84.5	sgs_Latn	66.4	**69.8**	sin_Sinh	**53.0**	51.2	slk_Latn	**76.4**	55.9
snd_Arab	**41.8**	41.8	som_Latn	**57.5**	56.2	spa_Latn	**77.6**	77.6	sqi_Latn	76.8	**78.7**
sun_Latn	50.8	**75.1**	swa_Latn	**71.8**	71.8	swe_Latn	**70.9**	65.8	szl_Latn	**70.9**	70.9
tat_Cyrl	63.8	**76.5**	tel_Telu	48.1	**49.0**	tgk_Cyrl	**68.4**	68.4	tgl_Latn	71.9	**73.7**
tuk_Latn	54.4	**57.3**	tur_Latn	**77.1**	77.1	uig_Arab	47.7	**62.3**	ukr_Cyrl	76.6	**85.3**
uzb_Latn	73.2	**76.0**	vec_Latn	68.0	**75.1**	vep_Latn	**72.0**	63.0	vie_Latn	**72.3**	49.7
vol_Latn	**61.0**	36.5	war_Latn	**64.9**	56.1	wuu_Hani	35.7	**66.7**	xmf_Geor	**69.3**	55.7
yor_Latn	**69.3**	41.7	yue_Hani	25.7	**73.5**	zea_Latn	62.9	**75.4**	zho_Hani	**25.2**	25.2

Table 18: F1 scores of LANGSAMP on **NER** using English and the closest donor language as source.

Language	English	Closest donor	Language	English	Closest donor	Language	English	Closest donor	Language	English	Closest donor
afr_Latn	**88.5**	79.5	ajp_Arab	**71.1**	41.9	aln_Latn	**53.4**	45.1	amh_Ethi	**66.8**	66.8
bam_Latn	**43.0**	31.2	bel_Cyrl	86.4	**93.8**	ben_Beng	**87.5**	80.2	bre_Latn	61.1	**62.3**
cat_Latn	86.8	**95.8**	ceb_Latn	**66.7**	32.5	ces_Latn	**85.4**	73.3	cym_Latn	**65.5**	60.4
deu_Latn	**88.2**	88.2	ell_Grek	**84.9**	75.5	eng_Latn	**96.0**	96.0	est_Latn	**84.7**	77.4
fao_Latn	**88.7**	67.5	fas_Arab	**72.2**	69.1	fin_Latn	**82.2**	75.8	fra_Latn	**85.8**	85.8
gle_Latn	64.6	**65.5**	glg_Latn	83.6	**87.8**	glv_Latn	51.9	**57.8**	grc_Grek	**71.6**	71.6
gsw_Latn	**82.7**	82.7	hbo_Hebr	**38.9**	37.4	heb_Hebr	67.9	**69.3**	hin_Deva	**77.2**	77.2
hsb_Latn	**83.7**	73.4	hun_Latn	**82.2**	42.0	hye_Armn	**85.1**	84.9	hyw_Armn	**83.0**	56.8
isl_Latn	**82.7**	81.2	ita_Latn	88.9	**92.4**	jav_Latn	75.4	**78.8**	jpn_Jpan	**33.1**	33.1
kmr_Latn	**76.6**	61.6	kor_Hang	**52.7**	45.3	lat_Latn	72.8	**74.2**	lav_Latn	**83.7**	78.4
lit_Latn	**82.1**	80.7	lzh_Hani	**24.5**	24.5	mal_Mlym	**86.0**	52.1	mar_Deva	**84.1**	81.7
myv_Cyrl	**65.9**	58.4	nap_Latn	**82.4**	70.6	nds_Latn	**79.1**	34.0	nld_Latn	**88.2**	82.2
pcm_Latn	**58.2**	48.1	pol_Latn	84.2	**89.1**	por_Latn	87.9	**92.0**	quc_Latn	**63.3**	52.6
rus_Cyrl	**88.7**	88.7	sah_Cyrl	74.2	**74.5**	san_Deva	25.5	**32.7**	sin_Sinh	**56.2**	34.4
slv_Latn	77.6	**79.0**	sme_Latn	**74.8**	60.6	spa_Latn	**87.8**	87.8	sqi_Latn	**77.5**	72.7
swe_Latn	**92.7**	83.2	tam_Taml	**74.6**	74.6	tat_Cyrl	**72.4**	70.9	tel_Telu	**80.9**	55.9
tha_Thai	**58.3**	27.5	tur_Latn	**71.2**	71.2	uig_Arab	**68.2**	48.3	ukr_Cyrl	85.6	**91.7**
vie_Latn	**68.4**	32.4	wol_Latn	**61.6**	57.4	xav_Latn	**16.7**	11.2	yor_Latn	**62.7**	46.5
zho_Hani	**47.4**	47.4									

Table 19: F1 scores of LANGSAMP on **POS** using English and the closest donor language as source.