OFA: A Framework of Initializing Unseen Subword Embeddings for Efficient Large-scale Multilingual Continued Pretraining

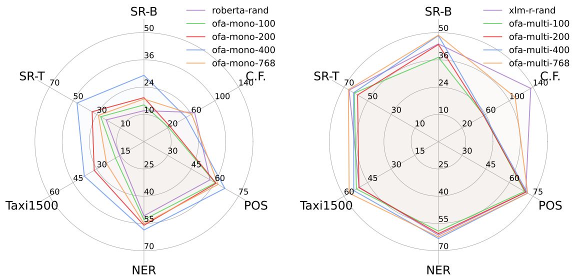
Anonymous ACL submission

Abstract

Instead of pretraining multilingual language models from scratch, a more efficient method is to adapt existing pretrained language models (PLMs) to new languages via vocabulary extension and continued pretraining. However, this method usually randomly initializes the embeddings of new subwords and introduces substantially more embedding parameters to the model, thus weakening the efficiency. To address these issues, we propose a novel framework: **O**ne **F**or **A**ll (**OFA**), which wisely initializes the embeddings of unseen subwords and thus can adapt a PLM to multiple languages efficiently and effectively. OFA takes advantage of external well-aligned multilingual static word vectors and injects the alignment knowledge into the subword embeddings. In addition, OFA applies matrix factorization and replaces the cumbersome embeddings with two lower-dimensional matrices, which largely reduces the number of parameters. We show OFA accelerates the convergence of continued pretraining, which is environmentally friendly as much fewer carbon footprints are generated. Through extensive experiments, we demonstrate OFA can achieve competitive or better performance than default continued pretraining baselines on a wide range of crosslingual downstream tasks. We make our code and models publicly available.

1 Introduction

Multilingual PLMs, such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), have demonstrated remarkable zero-shot crosslingual capability (Huang et al., 2019; Artetxe et al., 2020). That is, with only finetuning in some (high-resource) languages to perform a task, the multilingual model can be directly applied to other (low-resource) languages. However, training such multilingual PLMs from scratch requires massive data of different languages, and most importantly, considerable computing resources and energy (Wang et al., 2019; Bender et al., 2021; Zhou et al., 2023). Therefore, continued pretraining from existing models has been a good alternative (Wang et al., 2022; Alabi et al., 2022; ImaniGooghari et al., 2023). However, two problems are generally overlooked in the context of multilingual continued pertaining with vocabulary extension: **(a)** the random initialization of embeddings for new subwords does not actively use any lexical knowledge encoded in the model; **(b)** the introduction of many new parameters may pose efficiency problem.



(a) RoBERTa-based models (b) XLM-R-based models

Figure 1: Qualitative comparisons between baselines and OFA. OFA consistently achieves competitive or better performance than the baselines using both (a) monolingual (RoBERTa) or (b) multilingual (XLM-R) PLMs as the source model, with fewer carbon footprints (C.F.) during the continued pretraining, indicating higher efficiency. The stride of each axis in the chart is different.

Regarding **(a)**, the default random initialization approach which samples from a given distribution, e.g., a Gaussian (Hewitt, 2021; de Vries and Nissim, 2021; Marchisio et al., 2023), does not actively use the lexical knowledge of the original embeddings. To better leverage existing knowledge, some recent works propose to initialize the embeddings for target-language subwords by exploiting both external crosslingual static word vectors and the original PLM embeddings (Tran, 2020; Minixhofer et al., 2022; Dobler and de Melo, 2023). Unfortunately, these methods either bilingualize a PLM or create a new monolingual LM for a single target language at a time, which is not ideal in the context of multilingual continued pretraining. Therefore, our goal is to adapt to many languages all at once and wisely initialize the new subword embeddings for large-scale multilingual continued pretraining.

Regarding **(b)**, adapting to more languages will unarguably introduce more parameters. According to Chung et al. (2021), the embedding matrix of multilingual models makes up around 50% of the model's entire parameters. This percentage can be further increased when adding more new subwords as a consequence of adapting to more languages. In the monolingual setting, the factorized embedding parameterization shows effectiveness without sacrificing much performance (Lan et al., 2020). A similar method is also expected to succeed in multilingual models, given that embeddings are inherently more redundant: *words from different languages that refer to the same concept often have similar representations*. Therefore, we aim to reduce the number of parameters in the embeddings through factorized parameterization.

To this end, we introduce **OFA**, a framework that wisely initializes the embeddings of new subwords with a factorized parameterization for efficient large-scale multilingual continued pretraining. OFA first factorizes the embeddings of the source PLM and uses two smaller matrices to replace it. In the lower-dimensional space, the embeddings of the non-shared new subwords are represented as combinations of the embeddings of some subwords from the source PLM, weighted by the similarity extracted from well-aligned external static multilingual vectors (Liu et al., 2023a) that cover 1,335 languages. The embeddings of the shared subwords are directly copied. Finally, OFA copies all non-embedding parameters of the source PLM model and exchanges the source tokenizer (the tokenizer of the source PLM) with the target tokenizer (the tokenizer after vocabulary extension).

We use a monolingual PLM, i.e., RoBERTa (Liu et al., 2019) and a multilingual PLM, i.e., XLM-R (Conneau et al., 2020) as our source models. We first apply OFA to these models and then continued pretrain the resulting models on the Glot500-c corpus (ImaniGooghari et al., 2023). The final models are evaluated on a diverse set of downstream tasks, including sentence retrieval, text classification, and sequence labeling. OFA not only accelerates the convergence of continued pretraining thus much fewer carbon footprints are generated, but also achieves competitive or better performance on all tasks compared with randomly initialized or full-dimensional baselines, as shown in Figure 1.

The contributions of this work are as follows: (i) We propose OFA, a framework that wisely initializes the embeddings of unseen subwords with factorized parametrization, targeted on efficient multilingual continued pretraining. (ii) We conduct extensive and strictly controlled experiments on a wide range of downstream tasks and show that OFA is effective and boosts crosslingual transfer. (iii) We show OFA is efficient and environmentally friendly: achieving better performance with less GPU consumption and fewer carbon footprints.

2 Related Work

There are generally two ways to obtain a multilingual PLM. The first way is to pretrain a model from scratch directly on a number of languages with a specific self-learning objective, e.g., masked language modeling (MLM) (Devlin et al., 2019). The typical models that adopt such a strategy are encoder-only models such as mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), IndicBERT (Kakwani et al., 2020), AfriBERTa (Ogueji et al., 2021) and XLM-V (Liang et al., 2023), decoder-only models such as XGLM (Lin et al., 2022), mGPT (Shliazhko et al., 2022) and BLOOM (Scao et al., 2022), and encoder-decoder models such as mBART (Liu et al., 2020) and mT5 (Xue et al., 2021). The alternative way is to use publicly available multilingual PLMs as the source models and continued pretrain them on a set of target languages (Wang et al., 2022; Alabi et al., 2022; ImaniGooghari et al., 2023). This continued pretraining approach is in favor because it consumes fewer resources than training from scratch, which is important when the computation budget is limited given the continually increasing model size (Tay et al., 2022; Gupta et al., 2023).

One key reason why this continued pretraining approach works is the crosslingual ability of the original multilingual PLMs (Pires et al., 2019; K et al., 2020; Chai et al., 2022). With this ability, during continued pretraining, the model could leverage the knowledge gained in the previous pretraining phase as a prior, and adapt to the new languages quickly. Some prior works attempt to actively capitalize latent knowledge encoded in the parameters (embeddings or the transformer body) of the source PLM (Artetxe et al., 2020; Pfeiffer et al., 2021) when transferring to new languages. However, embeddings of new subwords are randomly initialized. Most recently, Tran (2020), Minixhofer et al. (2022) and Dobler and de Melo (2023) explore the possibility of leveraging both the source PLM embeddings and well-aligned external crosslingual word vectors to initialize the embeddings of new subwords for a **single** target language at a time. However, how this type of method could be efficiently applied to multilingual scenarios is left unexplored. Our work, in contrast to former research, aims to establish a framework to adapt a PLM, regardless of monolingual or multilingual, to multiple languages. In addition, our framework is targeted towards parameter efficiency, which is friendly to a limited computation budget.

3 Preliminary: Embedding Factorization

We first introduce one key technique used by OFA: source embedding factorization. Although matrix factorization itself is not new and is widely leveraged, e.g., in ALBERT (Lan et al., 2020) (a monolingual model) to lower memory consumption. We instead look at this factorization from a **multilingual perspective** and provide the intuition as to why such low-rank parameterization is effective in large-scale **multilingual continued pretraining**.

Given the embeddings $\boldsymbol{E}^s \in \mathbb{R}^{|V^s| \times D}$ from a source PLM that is pretrained on some source languages S, where V^s is its subword vocabulary and D is the embedding dimension, we propose to factorize the matrix $\boldsymbol{E}^s$ into lower-dimensional embeddings $\boldsymbol{F}^s \in \mathbb{R}^{|V^s| \times D'}$ and an orthogonal up-projection matrix $\boldsymbol{P} \in \mathbb{R}^{D' \times D}$: $\boldsymbol{E}^s \approx \boldsymbol{F}^s \boldsymbol{P}$, where $D' < D$. $\boldsymbol{P}$ can be interpreted as the embeddings of a set of D'-dimensional latent semantic concepts that are language-agnostic, serving as the basis of a semantic space in $\mathbb{R}^D$ for all subwords. Thus we refer to $\boldsymbol{P}$ as the *primitive embeddings*. $\boldsymbol{F}^s$ can be regarded as *coordinates* of all subwords in V^s in the space spanned by $\boldsymbol{P}$. The final representation of a subword v will be the linear combination of the primitive embeddings: $\boldsymbol{P}^T \boldsymbol{F}^s_{\{v\}}$.

By factorizing the embeddings into the language-agnostic part $\boldsymbol{P}$ and language-specific part $\boldsymbol{F}^s$, we can reduce the number of trainable parameters from $|V^s| \times D$ to $|V^s| \times D' + D' \times D$. This reduction of parameters can be prominent when $D' \ll D$. In addition, as $\boldsymbol{P}$ is shared across languages, we only need to find the target coordinates $\boldsymbol{F}^t \in \mathbb{R}^{|V^t| \times D'}$ under the same basis $\boldsymbol{P}$ when we want to adapt the model to new languages whose vocabulary is V^t. This is much more efficient than finding $\boldsymbol{E}^t \in \mathbb{R}^{|V^t| \times D}$, considering $|V^t|$ can be considerably large in a multilingual setting. Lastly, any coordinates in $\boldsymbol{F}^t$ can be up-projected back to $\mathbb{R}^D$ through $\boldsymbol{P}$, corresponding to the hidden size of the transformer body of the source PLM.

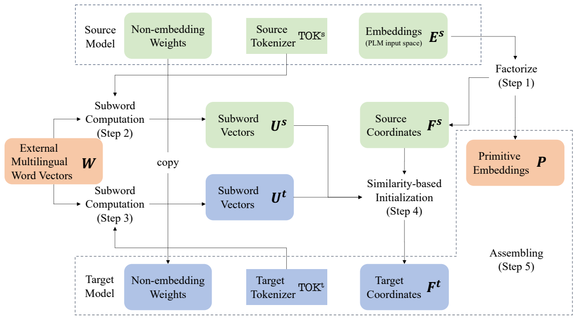


Figure 2: Summary of OFA. Different color indicates the block is specific to different languages. Green: source languages; blue: target languages; orange: both.

4 OFA Framework

OFA initializes the embeddings of new subwords in a factorized parametrization. The basic idea of OFA is as follows. We leverage an external multilingual word vector[1] space (which provides high-quality representations of both source and target languages) to induce a measure of semantic similarity on the joint set of subwords and words of both source and target languages. This similarity measure then allows us to initialize subwords of target languages with semantically meaningful representations in the source PLM embedding space. We show the summary of OFA framework in Figure 2 and describe the process step by step as follows.

Problem Setting. Given well-aligned external static multilingual word vectors $\boldsymbol{W}$ (vocabulary V), a source PLM (subword embeddings are $\boldsymbol{E}^s$) with its tokenizer $\texttt{TOK}^s$ (vocabulary V^s) and target tokenizer $\texttt{TOK}^t$ (vocabulary V^t), we want to find **a good initialization** of embeddings for all subwords in V^t, i.e., $\boldsymbol{F}^t$, which are **in lower dimensions**.

Step 1. We factorize $\boldsymbol{E}^s$ from the source PLM to primitive embeddings $\boldsymbol{P}$ and source coordinates $\boldsymbol{F}^s$. $\boldsymbol{P}$ will serve as the base of subword embeddings for all languages, and $\boldsymbol{F}^s$ will be used to initialize the desired target coordinates $\boldsymbol{F}^t$ in **Step 4**. We simply let $\boldsymbol{F}^s = \boldsymbol{E}^s$ for baseline models (no matrix factorization is applied to $\boldsymbol{E}^s$).

[1]To avoid confusion, we use the word "word vectors" to refer to any vector in the external static word vector space, and "embedding" to refer to the embeddings in the PLM space.

Step 2. We use the source tokenizer $\texttt{TOK}^s$ to tokenize all words in V. We then create a directed bipartite graph between words in V and subwords in V^s that can be tokenized from those words. We use $\overrightarrow{\text{ColexNet+}}$ (Liu et al., 2023a) as the word vectors, as they show very strong crosslinguality and reflect conceptual similarity (Liu et al., 2023b; Ye et al., 2023) in many languages (see §C for additional details of the word vectors). Next, we create the vector of a subword as the average of the vector of the words that are connected with the subword:

$$\vec{c} = \frac{1}{|N(c)|} \sum_{v \in N(c)} \boldsymbol{W}_{\{v\}}$$

where c is a subword in the graph and $N(c)$ is the set of neighbors of c in the graph (these neighbors are $\in V$). The intuition behind this calculation is that any words that include the same subword are related to the concept that the subword represents, and therefore those words should contribute to the representation of the subword. If a subword in V^s is not in the graph, we create its vector as zero. In this way, we create vectors for all subwords in V^s. We refer to the created subword vectors as $\boldsymbol{U}^s$.

Step 3. We create subword vectors for all subwords in V^t in the same way as described in Step 2, using target decoder $\texttt{TOK}^t$, all words in V, and the multilingual word vectors $\boldsymbol{W}$. The created subword vectors are denoted as $\boldsymbol{U}^t$. Note that $\boldsymbol{U}^t$ and $\boldsymbol{U}^s$ are in the same vector space as $\boldsymbol{W}$, because both of them are created based on $\boldsymbol{W}$.

Step 4. We then leverage the source coordinates $\boldsymbol{F}^s$, source-language subword vectors $\boldsymbol{U}^s$ and target-language subword vectors $\boldsymbol{U}^t$ to initialize target coordinates $\boldsymbol{F}^t$. To begin with, we deal with the subwords shared by V^s and V^t. For these subwords, we simply copy their coordinates from $\boldsymbol{F}^s$ to $\boldsymbol{F}^t$, which is also done by Dobler and de Melo (2023). For the remaining subwords, which are probably from new languages and not covered by V^s, we follow WECHSEL (Minixhofer et al., 2022) to find a good initialization based on similarity. Specifically, for each subword $x \in V^s$ and each subword $y \in V^t$, we calculate the cosine similarity between x and y in the subword vector space:

$$s_{(x,y)} = \text{cos-sim}(\boldsymbol{U}^s_{\{x\}},\ \boldsymbol{U}^t_{\{y\}})$$

The coordinate of each non-shared subword in V^t is finally initialized as a convex combination of source-language coordinates in $\boldsymbol{F}^s$:

$$\boldsymbol{F}^t_{\{y\}} = \frac{\sum_{x \in \mathbb{N}(y)} \exp(s_{(x,y)}/\tau) \cdot \boldsymbol{F}^s_{\{x\}}}{\sum_{x' \in \mathbb{N}(y)} \exp(s_{(x',y)}/\tau)}$$

where $\mathbb{N}(y)$ is the set of k nearest source-language subwords of the target-language subword y and τ is the temperature (we set $k = 10$ and $\tau = 0.1$ by default, following Minixhofer et al. (2022) who report the optimal choices in their experiments). In case the vector of a subword y in $\boldsymbol{U}^t$ is zero, we randomly initialize its coordinate $\boldsymbol{F}^t_{\{y\}}$ from a Gaussian distribution $\mathcal{N}(\mathbb{E}[\boldsymbol{F}^s], \text{Var}[\boldsymbol{F}^s])$. Note that $\boldsymbol{F}^t$ is roughly in the embedding space of $\boldsymbol{F}^s$, instead of in the vector space of $\boldsymbol{U}^s$ and $\boldsymbol{U}^t$.

Step 5. We finally assemble a target model by using the transformer body of the source PLM (all parameters except for its subword embeddings), the primitive embeddings $\boldsymbol{P}$, and the initialized target coordinates $\boldsymbol{F}^t$. The dimension of $\boldsymbol{F}^t$ is the same as the transformer body if no matrix factorization is applied, otherwise, we need to up-project the coordinates with $\boldsymbol{P}$ to suit the hidden dimension of the transformer body. In this way, we transform a source PLM into a multilingual model that has fewer parameters, which serves as a good start for efficient multilingual continued pretraining.

5 Experiments

5.1 Setups

We use a SentencePiece (Kudo and Richardson, 2018) tokenizer that has a vocabulary size of 401K as the target tokenizer. The vocabulary is merged from the subwords in XLM-R (Conneau et al., 2020) and new subwords learned from the Glot500-c corpus (ImaniGooghari et al., 2023) (See §A for details of the Glot500-c corpus.). The target tokenizer is the same as the tokenizer used in Glot500-m (ImaniGooghari et al., 2023). We then created 8 models using OFA framework as follows:

OFA-mono-xxx: we construct target models by OFA using English RoBERTA (Liu et al., 2019) as the source model. xxx denotes the latent dimension used in the factorization, where singular value decomposition (SVD) is used and top-k eigenvalues / eigenvectors are selected. We use four different dimensions: 100, 200, 400 and 768. When the dimension is 768, no matrix factorization is applied. The vocabulary and the tokenizer are the same as Glot500-m. Then we continued pretrain these assembled models on the Glot500-c corpus.

OFA-multi-xxx: we use the same setting as OFA-mono-xxx to construct target models (latent dimension: 100, 200, 400, 768), where XLM-R is used as the source model. Then we continued pretrain these models on the Glot500-c corpus.

The model architecture of OFA-mono-768 and OFA-multi-768 is the same as Glot500-m, where the embeddings are tied with the parameters of the language modeling head. For lower-dimensional models, two matrices are used to map the representation back to vocabulary space for masked language modeling. The parameters of the two matrices are tied to the primitive embeddings and target coordinates. We continued pretrain all models using MLM objective and follow the training hyperparameters used by ImaniGooghari et al. (2023). Each training step contains an effective batch of 384 samples randomly picked from all language-scripts[2]. We refer to the languages that are covered by XLM-R as **head** languages and the rest of languages as **tail** languages. We store checkpoints for each model every 10K steps and apply early stopping with the best average performance on downstream tasks. We train all models on **four** NVIDIA RTX A6000 GPUs for a maximum of four weeks. See §B for a detailed description of hyperparameter settings of continued pretraining and evaluation.

5.2 Baselines

We consider the following baselines for comparison with OFA (see Table 1 for the number of parameters under different latent embedding dimensions):

RoBERTa A monolingual PLM trained on English corpus (Liu et al., 2019). Its embeddings and tokenizer do not cover most of the new subwords of our models. The vocabulary size is 50K.

RoBERTa-rand We replace the embeddings of RoBERTa with new embeddings (the vocabulary size is 401K, the same as OFA-mono-768), which are constructed by copying the shared subwords and **randomly** initializing the embeddings of remaining subwords not covered by RoBERTa from a Gaussian distribution with a mean and variance of the original RoBERTa embeddings, similar to Minixhofer et al. (2022). Glot500-m tokenizer is used for tokenization. We then continued pretrain it on Glot500-c with the same hyperparameters.

XLM-R A strong multilingual PLM trained on 100 languages (Conneau et al., 2020). We use the

	D'=100	D'=200	D'=400	D=768
Model Params.	126M	167M	247M	395M
Embedding Params.	40M	80M	161M	309M

Table 1: Model parameters under different latent dimensions. When D'=100, 200, or 400, each corresponds to two OFA-initialized models (based on RoBERTa or XLM-R). D=768 not only corresponds to OFA-768, but also baselines RoBERTa-rand and XLM-R-rand, as they have the same architecture. By decreasing latent dimensions, the model parameters decrease drastically.

base version, where the embedding dimension is 768. The vocabulary size is 250K.

XLM-R-rand Similar to RoBERTa-rand, this model extends the vocabulary from XLM-R and the embeddings of subwords not covered by XLM-R are randomly initialized from a Gaussian distribution with a mean and variance of the original XLM-R embeddings.[3] Glot500-m tokenizer is used for tokenization. The model is then continued pretrained on Glot500-c with the same hyperparameters.

5.3 Downstream Tasks

Sentence Retrieval. We consider two datasets: Tatoeba (Artetxe and Schwenk, 2019) (SR-T) and Bible (SR-B). We select up to 1,000 English-aligned sentences for SR-T, following the same setting used by Hu et al. (2020). For SR-B, we select up to 500 English-aligned sentences. We report the top-10 accuracy by finding the nearest neighbors of the representation of each English sentence. Following Jalili Sabet et al. (2020), the representations are calculated by taking the average of the contextualized word embedding at the 8th layer.

Sequence Labeling. We consider two types of tasks: named entity recognition (NER) and Part-Of-Speech (POS) tagging. We use WikiANN dataset (Pan et al., 2017) for NER and Universal Dependencies (de Marneffe et al., 2021) of version v2.11 for POS. We finetune the models only on the English train set, select the best model on the English dev set, and then report the zero-shot performance on the test sets of other languages. F1 scores are reported for both NER and POS.

Text Classification. We use Taxi1500 (Ma et al., 2023), a text classification dataset that provides train/valid/test sets with 6 classes in more than

[2]A language-script is a combination of ISO 639-3 and script, which is used by the Glot500-c corpus.

[3]The model is named Glot500-m in ImaniGooghari et al. (2023). To be consistent with other names used in this paper, we call it XLM-R-rand. All models are trained on the same infrastructure for a strictly controlled experimental setting.

	SR-B			SR-T			Taxi1500			NER			POS		
	tail	head	all	tail	head	all	tail	head	all	tail	head	all	tail	head	all
RoBERTa	3.2	3.9	3.4	8.1	4.9	5.8	5.5	6.9	5.8	30.4	26.4	28.2	21.1	28.6	26.3
RoBERTa-rand	11.0	14.7	11.9	24.9	20.9	22.0	14.2	19.1	15.5	52.1	49.8	50.8	47.1	61.4	57.0
OFA-mono-100	13.1	20.3	14.9	26.8	26.5	26.6	15.8	24.8	18.1	53.3	52.6	52.9	50.6	64.8	60.4
OFA-mono-200	<u>16.1</u>	<u>25.9</u>	<u>18.6</u>	<u>33.2</u>	<u>34.3</u>	<u>33.9</u>	<u>29.8</u>	<u>37.0</u>	<u>31.6</u>	<u>55.8</u>	<u>56.1</u>	<u>56.0</u>	49.0	66.1	60.8
OFA-mono-400	**25.4**	**40.4**	**29.2**	**41.6**	**48.7**	**46.7**	**35.1**	**46.4**	**37.9**	**58.2**	**59.0**	**58.6**	**57.0**	**70.6**	**66.4**
OFA-mono-768	16.0	23.6	17.9	28.6	28.5	28.6	22.1	28.9	23.8	54.8	55.3	55.1	<u>51.7</u>	<u>66.7</u>	<u>62.1</u>
XLM-R	7.4	54.2	19.3	32.6	66.2	56.6	15.5	59.8	26.7	47.6	61.8	55.3	42.1	**76.1**	65.6
XLM-R-rand	38.6	<u>60.4</u>	44.2	<u>55.6</u>	<u>69.7</u>	<u>65.7</u>	47.0	<u>59.9</u>	50.3	60.3	62.3	61.4	60.6	74.9	70.5
OFA-multi-100	33.0	49.7	37.3	54.9	63.8	61.3	50.5	56.7	52.1	58.6	59.8	59.2	60.4	73.9	69.7
OFA-multi-200	39.4	57.0	43.9	51.8	61.1	58.5	49.0	54.9	50.5	59.5	61.4	60.6	60.5	74.9	70.5
OFA-multi-400	**44.5**	60.0	<u>48.5</u>	54.8	64.7	61.8	<u>51.9</u>	59.3	<u>53.8</u>	**62.5**	**64.0**	**63.3**	**63.2**	75.4	<u>71.6</u>
OFA-multi-768	<u>43.8</u>	**62.7**	**48.7**	**56.1**	**70.4**	**66.3**	**54.3**	**63.8**	**56.7**	<u>60.6</u>	<u>63.9</u>	<u>62.4</u>	<u>62.4</u>	<u>75.8</u>	**71.7**

Table 2: Performance of the models initialized with OFA and baselines on five multilingual tasks across 5 seeds. We report the performance as an average over head, tail, and all language-scripts for each model. Models initialized with OFA constantly perform better than baselines. **Bold** (<u>underlined</u>): best (second-best) result per controlled group.

1,500 languages. Following ImaniGooghari et al. (2023), we select a subset of languages (354) supported by the models for evaluation. Same as in NER and POS, we report the zero-shot performance (in F1 scores) using English as the source.

5.4 Results and Discussions

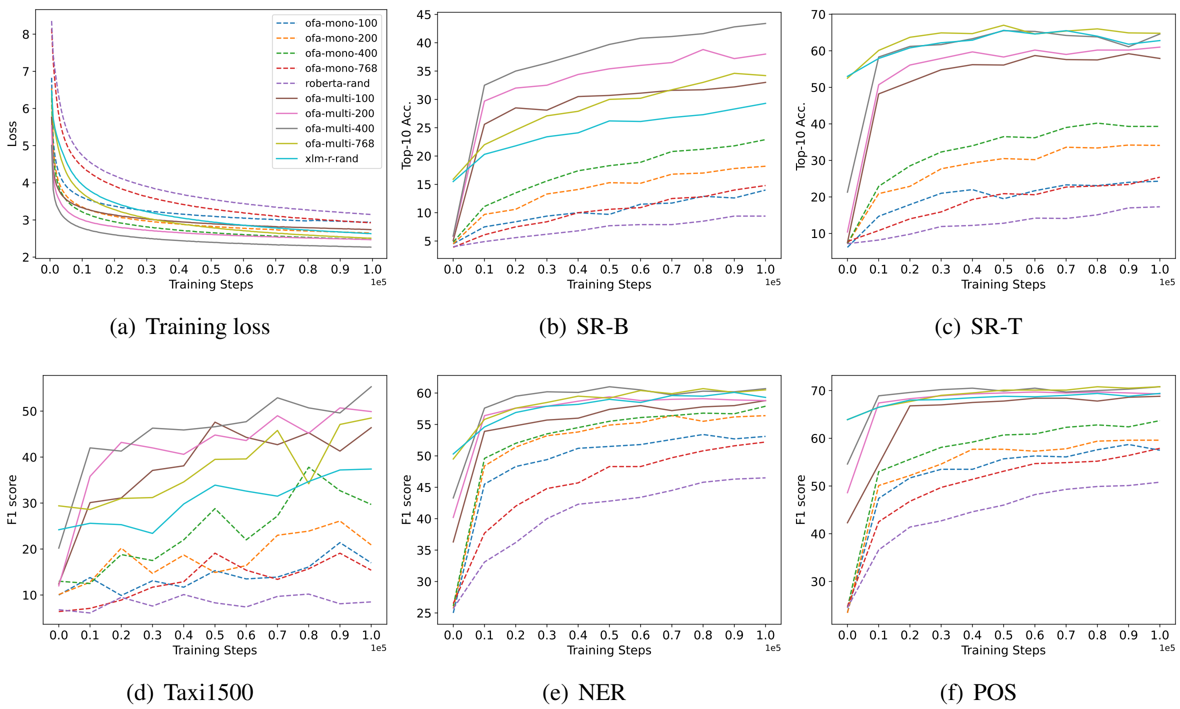
Table 2 shows the performance of the models initialized with OFA and baselines on downstream tasks (see complete results for each language-script in §E). Models initialized with OFA demonstrate a consistent improvement compared with the baselines. When the source model is monolingual, with random initialization of unseen subwords, RoBERTa-rand just obtains 11.9, 22.0, and 15.5 on SR-B, SR-T, and Taxi1500 respectively (averaged overall), which are 6.0, 6.6, 8.3 lower than its counterpart OFA-mono-768. In the sequence labeling task we also see similar improvement: OFA-mono-768 achieves 4.3 and 5.1 better than RoBERTa-rand on NER and POS respectively. Such an increase is even higher when compared with RoBERTa, as RoBERTa is a monolingual model. When the source model is multilingual, models initialized with OFA also achieve remarkable performance. OFA-multi-768 achieves better performance than XLM-R on every task. Compared with XLM-R-rand, it also achieves better performance, which indicates the effectiveness of the initialization with the help of external multilingual embeddings.

The embedding dimension also plays a crucial role in the performance. Typically, we see an improvement in performance as we increase the latent dimension, particularly from 100 to 400 for both OFA-mono and OFA-multi models. This is expected as a larger dimension often induces better expressiveness. Nevertheless, the improvement from dimension 400 to 768, is not consistently large, and in some cases, it even leads to performance declines. For example, OFA-mono-400 outperforms OFA-mono-768 on all downstream tasks. We assume this is because a monolingual model with many parameters might not be easy to adapt to diverse languages. A smaller embedding dimension can ease the burden and facilitate the pretraining, thus achieving better performance. Similarly, OFA-multi-400 is very competitive to OFA-multi-768 (OFA-multi-400 is even better on NER and POS). We attribute this to the "redundancy" of the embeddings in multilingual PLMs (see §D for an analysis). By using factorization, we keep the most important information that is shared across languages. Thus there is a trade-off. When the dimension is very small, e.g., 100, there is a risk of information loss. However, with a moderate size, e.g., 400, the model is less redundant and equipped with enough expressiveness to achieve good performance.

6 Analysis

6.1 Continued training Progression

To analyze how different embedding dimensions and initialization methods can influence the continued training, we visualize the training loss of models that are initialized with OFA and two baseline models, i.e., RoBERTa-rand and XLM-R-rand. In addition, we evaluate all these models on five downstream tasks at 10K-step intervals until 100K steps. The results are shown in Figure 3. From Fig. 3 (a), when the embedding dimension is 768, the models initialized with OFA converge faster com-



(a) Training loss (b) SR-B (c) SR-T

(d) Taxi1500 (e) NER (f) POS

Figure 3: The training loss as well as the performance on five downstream tasks from step 0 (without continued pretraining) to step 100K (10th checkpoints). We see that models initialized by OFA converge faster than baseline models (RoBERTa-rand and XLM-R-rand) whose new subwords are randomly initialized during continued pretraining. For most of the downstream tasks, models with lower embedding dimensions can achieve better performance after only 10K steps compared with their full-dimensional counterparts (OFA-mono-768 and OFA-multi-768).

pared with the models being randomly initialized, regardless of whether the source model is monolingual or multilingual. The faster convergence is also related to the performance, as OFA-mono-768 (resp. OFA-multi-768) constantly performs better than RoBERTa-rand (resp. XLM-R-rand) throughout steps for all tasks. This indicates that OFA, which explicitly leverages information encoded in source PLM embeddings and external multilingual word vectors, is superior to random initialization.

We also observe models with smaller dimensions tend to learn information faster in the initial steps, indicated by the speed of MLM loss drop. As explained earlier, smaller dimensions mean fewer parameters which eases the burden in continued pretraining, especially when the source model is monolingual. On the other hand, faster learning speed explains why models with smaller dimensions generally perform better than their full-dimensional counterparts (OFA-mono-768 or OFA-multi-768) in the early training phase. For example, with only 167M parameters, OFA-multi-200 achieves better or very close performance on each task compared with OFA-multi-768, which is two times larger. We also observe that all models, especially OFA-multi models, quickly reach a performance plateau on NER and POS tasks. This aligns with the finding that syntactic knowledge is acquired rapidly in the training progression (Blevins et al., 2022; Müller-Eberstein et al., 2023). This also suggests that sequence labeling might be a straightforward task where the model can transfer prevalent classes such as *verb* and *noun*, possibly through shared vocabulary (ImaniGooghari et al., 2023).

Combined with the analysis above, better initialization and smaller embedding dimensions enable an efficient multilingual continued pretraining and better performance in downstream tasks with fewer training steps. Lightweight models also reduce GPU consumption and allow for larger batch sizes. Therefore, the proposed OFA framework can be very useful where a limited computation budget is presented, e.g., in most laboratories or institutions.

In addition, as there are recent concerns regarding the environmental impact of training or operating LMs (Bender et al., 2021; Rae et al., 2021; Weidinger et al., 2022), we also report some related statistics when continued pretraining our models in Table 3. There are two benefits of using OFA with factorized embedding parameterization: (1) the av-

Models	best-checkpoint	avg. T	C.F.
OFA-mono-100	110K	3.8h	21.7
OFA-mono-200	120K	3.9h	24.3
OFA-mono-400	230K	4.3h	51.3
OFA-mono-768	250K	4.7h	60.9
RoBERTa-rand	270K	4.7h	65.8
OFA-multi-100	290K	3.8h	57.1
OFA-multi-200	280K	3.9h	56.6
OFA-multi-400	260K	4.3h	58.0
OFA-multi-768	450K	4.7h	110.0
XLM-R-rand	560K	4.7h	136.4

Table 3: Additional information: best checkpoint, average training time (avg. T) spent per 10K steps until the best checkpoint, and carbon footprint (C.F.: in kg of CO_2 eq.) of different models in continued pretraining.

Models	Settings	SR-B	SR-T	Taxi1500	NER	POS
OFA-mono-100	w/o	4.5	6.2	10.0	25.0	23.5
	w/	**14.9**	**26.6**	**18.1**	**52.9**	**60.4**
OFA-mono-200	w/o	4.5	7.2	10.1	25.7	23.4
	w/	**18.6**	**33.9**	**31.6**	**56.0**	**60.8**
OFA-mono-400	w/o	4.8	7.2	13.0	26.1	24.5
	w/	**29.2**	**46.7**	**37.9**	**58.6**	**66.4**
OFA-mono-768	w/o	3.9	7.8	8.2	26.5	24.7
	w/	**17.9**	**28.6**	**23.8**	**55.1**	**62.1**
OFA-multi-100	w/o	5.1	7.5	12.4	36.3	42.3
	w/	**37.3**	**61.3**	**52.1**	**59.2**	**69.7**
OFA-multi-200	w/o	5.7	10.4	12.0	40.2	48.6
	w/	**43.9**	**58.5**	**50.5**	**60.6**	**70.5**
OFA-multi-400	w/o	5.9	21.3	20.2	43.3	54.6
	w/	**48.5**	**61.8**	**53.8**	**63.3**	**71.6**
OFA-multi-768	w/o	15.9	52.5	29.4	49.5	63.9
	w/	**48.7**	**66.3**	**56.7**	**62.4**	**71.7**

Table 4: Performance of models initialized with OFA under settings of w/o and w/ continued pretraining. Continued pretraining largely improves the performance.

erage training time per 10K steps is shortened and (2) overall less training time is required to reach the best checkpoints compared to the random baseline. Considering that there is no huge difference in terms of the performance in downstream tasks, initializing by OFA with lower embedding dimensions can largely reduce the carbon emissions[4] and therefore is more environmentally friendly.

6.2 Influence of Continued Pretraining

Continued pretraining has a different impact on models with different embedding dimensions for different downstream tasks. Therefore, we compare how the model performance varies with or without continued pretraining, as shown in Table 4.

Although most models without continued pretraining perform generally badly, we see some exceptions. For example, OFA-multi-768 achieves more than 52.5 accuracy in SR-T, while only 15.9 in SR-B. The major reason is that SR-B contains many tail language-scripts that are not covered by XLM-R. On the contrary, SR-T covers many head languages. The continued pretraining also has less impact on sequence labeling, i.e., NER and POS, where the model can use the knowledge already encoded in its parameters to perform well in English, and then transfer to other languages through shared vocabulary, or the already existing crosslinguality when the source model is multilingual.

When the source model is monolingual, the performance without continued pretraining is bad no matter what embedding dimension is used. However, the higher-dimension model achieves constantly better performance than lower-dimension ones when the source model is multilingual. This can be explained by the fact that the source multilingual model already has strong crosslinguality and a higher dimension can better restore the original information encoded in XLM-R's embedding matrix. Nevertheless, the benefits of higher dimensions diminish after continued pretraining. Combined with Figure 3, we see that even the smallest model, i.e., OFA-multi-100, quickly surpasses OFA-multi-768 in SR-B and Taxi500 tasks after 10K training steps. We therefore could conclude that the models initialized with OFA could quickly adapt to new languages in the continued pretraining, especially when the source model is already multilingual.

7 Conclusion

In this work, we present OFA, a framework that wisely initializes unseen subword embeddings with factorized embedding parameterization for efficient large-scale multilingual continued pretraining. We conduct extensive and strictly controlled experiments by continued pretraining models that are initialized from monolingual or multilingual PLMs. We evaluate these models on a wide range of downstream tasks. We show that models initialized with OFA enjoy faster convergence during training and achieve competitive or better performance on downstream tasks, compared with the baselines where embeddings of new subwords are randomly initialized. We also show that with smaller embedding dimensions, the continued pretraining is further facilitated: training time is shortened and models achieve better performance in the early training phase. Therefore, this work contributes to efficient large-scale multilingual continued pretraining.

[4]Estimations were conducted using the MachineLearning Impact calculator presented in (Lacoste et al., 2019).

Limitations

In this work, we apply OFA to two models, RoBERTa, a monolingual PLM, and XLM-R, a multilingual PLM, and show the superiority of the proposed initialization method compared to the random initialization. However, both are encoder-only models and they are pretrained / continued pretrained only using the MLM objective. Theoretically, this approach should be able to extend to other types of models, e.g., decoder-only and encoder-decoder models, or other types of training objectives, e.g., next-word prediction or translation objectives, since our approach is **only related to the initialization stage** of continued pretraining and not restricted to any model architectures or training objectives. We do not try all possibilities in terms of architectures / objectives as that is not the major focus of this work, and we have a limited computation budget. We would leave such exploration using OFA in different architectures / objectives for future research in the community.

Another possible limitation is that, while we inject external knowledge into the subword embeddings before continued pretraining, such knowledge may diminish due to catastrophic forgetting (Kirkpatrick et al., 2017). That is, due to continued pretraining, the model gradually loses the initial knowledge. This is not wanted and we would expect methods such as active forgetting (Chen et al., 2023) could alleviate the problem by restoring the constructed embeddings from OFA every certain step in the continued pretraining. However, this again is not the major focus of this paper and we would call for exploration in this direction.

References

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

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

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.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2022. Analyzing the mono- and cross-lingual pretraining dynamics of multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3575–3590, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuan Chai, Yaobo Liang, and Nan Duan. 2022. Cross-lingual ability of multilingual masked language models: A study of language structure. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4702–4712, Dublin, Ireland. Association for Computational Linguistics.

Yihong Chen, Kelly Marchisio, Roberta Raileanu, David Ifeoluwa Adelani, Pontus Stenetor, Sebastian Riedel, and Mikel Artetx. 2023. Improving language plasticity via pretraining with active forgetting. *arXiv preprint arXiv:2307.01163*.

Hyung Won Chung, Thibault Févry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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.

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

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

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

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

Alexandre François. 2008. Semantic maps and the typology of colexification. *From polysemy to semantic change: Towards a typology of lexical semantic associations*, 106:163.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. 2023. Continual pre-training of large language models: How to (re) warm your model? *arXiv preprint arXiv:2308.04014*.

John Hewitt. 2021. Initializing new word embeddings for pretrained language models.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

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.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: an empirical study. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, 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*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

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.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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. *arXiv preprint arXiv:2301.10472*.

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

Yihong Liu, Haotian Ye, Leonie Weissweiler, Renhao Pei, and Hinrich Schuetze. 2023a. Crosslingual transfer learning for low-resource languages based on multilingual colexification graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8376–8401, Singapore. Association for Computational Linguistics.

Yihong Liu, Haotian Ye, Leonie Weissweiler, Philipp Wicke, Renhao Pei, Robert Zangenfeind, and Hinrich Schütze. 2023b. 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 pretraining for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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*.

Kelly Marchisio, Patrick Lewis, Yihong Chen, and Mikel Artetxe. 2023. Mini-model adaptation: Efficiently extending pretrained models to new languages via aligned shallow training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5474–5490, Toronto, Canada. Association for Computational Linguistics.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3158–3163, Reykjavik, Iceland. European Language Resources Association (ELRA).

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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

Max Müller-Eberstein, Rob van der Goot, Barbara Plank, and Ivan Titov. 2023. Subspace chronicles: How linguistic information emerges, shifts and interacts during language model training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13190–13208, Singapore. 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.

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.

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

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. 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.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models:

Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

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*.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2022. Scale efficiently: Insights from pretraining and finetuning transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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

Hai Wang, Dian Yu, Kai Sun, Jianshu Chen, and Dong Yu. 2019. Improving pre-trained multilingual model with vocabulary expansion. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 316–327, Hong Kong, China. Association for Computational Linguistics.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, Julia Haas, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 214–229, New York, NY, USA. Association for Computing Machinery.

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.

Haotian Ye, Yihong Liu, and Hinrich Schütze. 2023. A study of conceptual language similarity: comparison and evaluation. *arXiv preprint arXiv:2305.13401*.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.

A Glot500-c

The Glot500-c corpus (ImaniGooghari et al., 2023)[5] contains 511 languages in 30 different scripts. The total number of sentences is 1.5B and the median number of sentences per language-script is 120K. Because some languages can be written in multiple scripts, the corpus treats each **language-script** as a separate entity. For example, Tajik-Cyrillic and Tajik-Arabic will be considered as different entities as there are two different scripts used for Tajik in the corpus. The corpus is divided into train/dev/test sets for each language. Dev and test sets have 1000 sentences. Same as (ImaniGooghari et al., 2023), we only use the training data to continued pretrain all of our models.

B Detailed Hyperparameters

B.1 Continued Pretraining

We continued pretrain both the baseline models (RoBERTa-rand and XLM-R-rand) and models initialized with OFA using basically the same hyperparameters as used in ImaniGooghari et al. (2023). Specifically, we use MLM objective with the standard mask rate of 15%. We use Adam optimizer (Kingma and Ba, 2015) 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 set to 384. Each batch contains training samples concatenated up to the maximum sequence length of 512 and randomly picked from all language-scripts in the Glot500-c corpus. The only difference from ours to ImaniGooghari et al. (2023) is that we use **four** RTX A6000 GPUs while they use **eight** RTX A6000 GPUs. Therefore, we set the per-GPU batch to 12, and the gradient accumulation to 8, fulfilling $4 \times 12 \times 8 = 384$. The gradient accumulation in ImaniGooghari et al. (2023) is set to 4, as they use four more GPUs. We use FP16 training (mixed precision (Micikevicius et al., 2018)). The different gradient accumulation and usage of

[5] https://github.com/cisnlp/Glot500

	|head|	|tail|	#class	measure (%)
SR-B	94	275	-	top-10 Acc.
SR-T	70	28	-	top-10 Acc.
Taxi1500	90	264	6	F1 score
NER	89	75	7	F1 score
POS	63	28	18	F1 score

Table 5: Downstream tasks and measures. |head| (resp. |tail|): head (resp. tail) language-scripts according to ImaniGooghari et al. (2023) (a language-script is head if it is covered by XLM-R, otherwise it is tail); #class: the number of the categories if it is a (sequence-level or token-level) classification task.

mixed-precision might be the reason why the performance of our baseline XLM-R-rand is slightly different from the performance reported in ImaniGooghari et al. (2023). The continue-pretraining is done using scripts adapted from HuggingFace[6].

B.2 Downstream Tasks

The outline of the evaluation is shown in Table 5. We introduce the detailed hyperparameters used for each downstream task in the following.

SR-B. We use up to 500 English-aligned sentences from languages that are supported by the model, where most of the languages are tail languages (275). The retrieval task is performed without any training: we directly use the model after continued pretraining to encode all sentences. Each sentence is represented by taking the average of the contextual embedding at the **8th** layer. We then compute the top-10 accuracy for each pair (English and another language) by finding the nearest neighbors (in the other language) of the representation of each English sentence.

SR-T. We use up to 1000 English-aligned sentences from Tatoeba, which mainly contains head languages (70). The evaluation setting is the same as SR-B and top-10 accuracy is reported.

Taxi1500. We finetune the continued pretrained model (a sequence-level classification model in 6 classes) on the English train set and select the best checkpoint using the English dev set. We train each model for a maximum of 40 epochs with early stopping on a single GTX 1080 Ti GPU. Adam optimizer is used, the learning rate is set to 1e-5 and the effective batch size is set to 16 (batch size of 8 and gradient accumulation of 2). We then evaluate the zero-shot performance by evaluating the finetuned model on the test sets of all other language-scripts. F1 score is reported for each language-script.

[6] https://huggingface.co/

NER. We finetune the continued pretrained model (a token-level classification model in 7 classes) on the English train set and select the best checkpoint using the English dev set. We train each model for a maximum of 10 epochs with early stopping on a single GTX 1080 Ti GPU. Adam optimizer is used, the learning rate is set to 2e-5 and the effective batch size is set to 32 (batch size of 8 and gradient accumulation of 4). We then evaluate the zero-shot performance by evaluating the finetuned model on the test sets of all other language-scripts. F1 score is reported for each language-script.

POS. We finetune the continued pretrained model (a token-level classification model in 18 classes) on the English train set and select the best checkpoint using the English dev set. We train each model for a maximum of 10 epochs with early stopping on a single GTX 1080 Ti GPU. Adam optimizer is used, the learning rate is set to 2e-5 and the effective batch size is set to 32 (batch size of 8 and gradient accumulation of 4). We then evaluate the zero-shot performance by evaluating the finetuned model on the test sets of all other language-scripts. F1 score is reported for each language-script.

C Multilingual Word Vectors and Coverage

Two important factors that influence the effectiveness of OFA initialization are (1) the quality of the external multilingual word vectors and (2) the coverage of the multilingual word vectors in terms of languages and new subwords in the target model.

In this work, we use $\overrightarrow{\text{ColexNet+}}$ (Liu et al., 2023a), multilingual word vectors learned from colexification[7] (François, 2008) graphs built from 1,335 translations (one for a specific language identified by its ISO-639-3 code) of Parallel Bible Corpus (Mayer and Cysouw, 2014). The patterns of colexifications are extracted by Conceptualizer (Liu et al., 2023b), a statistic concept-grams alignment method. The tokens in the word vectors are ngrams (mostly word types as the algorithm prefers longer ngrams) within whitespace tokenized words. According to Liu et al. (2023a), $\overrightarrow{\text{ColexNet+}}$ outperforms a bunch of strong multilingual word vector

[7] Colexifications are a linguistic phenomenon where different meanings are expressed by the same word.

Source models	Copy	Similarity	Random	Coverage
RoBERTa	27K	179K	195K	51.5%
XLM-R	255K	84K	62K	84.6%

Table 6: The number of subwords being initialized by copying from the original embeddings (**Copy**); through the similarity-based method introduced in OFA (**Similarity**); and randomly from a Gaussian distribution (**Random**) when using $\overrightarrow{\text{ColexNet+}}$ as the external multilingual word vectors. Coverage shows the percentage of the subword being wisely initialized: (Copy + Similarity) / (Copy + Similarity + Random). The coverage is high for both of the source models. As the new vocabulary is extended from XLM-R, many subword embeddings are directly copied when using XLM-R as the source model.

baselines on crosslingual transfer tasks, especially for low-resource languages. we therefore choose to use $\overrightarrow{\text{ColexNet+}}$ as our multilingual word vectors.

We want as many as possible subwords to be initialized wisely (either directly copied for shared subwords or initialized by the similarity-based method in OFA), instead of being randomly initialized from a Gaussian distribution. This requires that the chosen external multilingual word vectors cover many subwords. Therefore we report the number of subwords being initialized (1) **by copying**, (2) **through the similarity-based method**, and (3) **randomly** when using $\overrightarrow{\text{ColexNet+}}$ as our external multilingual word vectors in Table 6. We see that for either the monolingual model as the source model (RoBERTa) or the multilingual model as the source model (XLM-R), the coverage (subwords being wisely initialized over all subwords) is more than 50%, indicating that the words included in $\overrightarrow{\text{ColexNet+}}$ cover a large number of subwords even though it is trained from a genre-specific corpus.

D Redundancy in Multilingual PLMs

To figure out how "redundant" the embeddings are in monolingual or multilingual PLMs, we use principle component analysis (PCA) to perform dimension reduction to the embeddings of various PLMs. We select monolingual PLMs: BERT (Devlin et al., 2019) of English and GPT-2 (Radford et al., 2019), and multilingual PLMs: mBERT (Devlin et al., 2019), base and large versions of XLM-R (Conneau et al., 2020), Glot500-m (ImaniGooghari et al., 2023) and XLM-V (Liang et al., 2023). The embedding dimension and vocabulary size of each PLM are shown in Table 7. We report how much variance is explained (information preserved) when

PLM	emb dim.	|V|
BERT-eng	768	31K
GPT-2	768	50K
mBERT	768	120K
XLM-R-base	768	250K
XLM-R-large	1024	250K
Glot500-m	768	401K
XLM-V	768	901K

Table 7: Embedding dimensions and vocabulary size of several monolingual and multilingual PLMs.

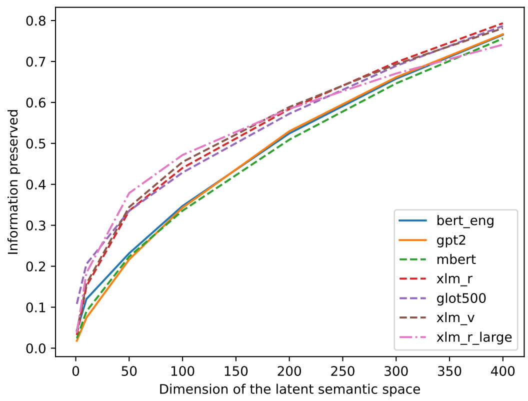


Figure 4: Information preserved (percentage of variance explained by the selected components) under different dimensions of the semantic space (number of principal components). Generally trend: multilingual models generally preserve more information than monolingual ones when embeddings are reduced to the same dimension.

keeping different numbers of principle components in the sorted order by their eigenvalues (until the first 400 components) in Figure 4. The general trend is that multilingual PLMs tend to be more "redundant" than monolingual ones: only keeping the first 100 components, about 50% variance can be explained in Glot500-m and XLM-R-large embeddings. Similarly, the information preserved is more than 40% in XLM-R-base and XLM-V, which is higher than the percentage in monolingual models GPT-2 and English BERT (about 30% is preserved), when the first 100 components are kept.

We also assume this "redundancy" is related to the crosslinguality of the PLMs. If the embedding matrix is more redundant, this indicates the many tokens referring to the same concept from different languages share similar representation space, therefore better crosslinguality is expected. For example, both base and large versions of XLM-R are more redundant than mBERT according to Figure 4, indicating better crosslinguality, which aligns with the finding that XLM-R constantly outperforms mBERT in many NLP downstream tasks (Conneau et al., 2020). However, the high redundancy, in turn, suggests an unnecessary over-parameterization. Thus we could use matrix factorization to remove some redundancy to reduce the number of parameters while not sacrificing much performance, which is exactly what we propose in the OFA framework: replacing the cumbersome embedding matrix with two smaller matrices.

E Complete Results for Each Task and Language

We report the complete results for all tasks and languages in Table 8, 9, 10 11 (SR-B), Table 12 (SR-T), Table 13, 14, 15, 16 (Taxi1500), Table 17, 18 (NER), and Table 19 (POS).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
ace_Latn	2.6	10.4	18.8	18.0	**24.4**	16.4	4.4	51.8	39.4	43.2	48.4	**54.6**
ach_Latn	4.0	10.6	8.4	9.8	**18.0**	13.0	4.4	46.6	30.4	37.2	**54.4**	43.2
acr_Latn	2.0	8.2	9.8	10.0	**15.6**	9.4	2.6	18.4	13.2	19.8	**27.0**	22.8
afr_Latn	4.4	14.8	22.8	31.8	**33.8**	23.8	**76.8**	71.4	65.0	71.6	71.0	74.4
agw_Latn	3.8	10.4	14.8	15.4	**25.4**	20.8	5.8	31.4	25.8	31.0	**41.6**	37.8
ahk_Latn	2.4	3.2	3.4	2.8	**4.0**	3.8	3.0	3.2	2.6	3.6	**4.4**	3.2
aka_Latn	5.0	10.8	14.0	18.8	**32.6**	18.6	5.0	46.4	46.0	48.2	**53.0**	51.6
aln_Latn	9.6	24.6	23.6	39.6	**61.4**	41.8	67.8	70.0	68.2	69.2	**72.0**	71.2
als_Latn	7.8	25.4	27.8	31.2	**49.2**	37.2	51.4	55.8	51.0	53.6	**56.8**	54.4
alt_Cyrl	2.4	8.2	9.8	12.8	**21.4**	16.2	12.6	52.8	42.6	54.4	57.0	**58.8**
alz_Latn	3.2	12.0	9.6	10.0	**18.2**	11.2	4.6	37.6	30.0	35.0	**40.2**	36.6
amh_Ethi	2.0	5.2	10.6	12.0	**30.2**	12.6	35.4	**52.4**	28.0	44.0	48.6	50.8
aoj_Latn	2.4	5.8	7.8	7.6	**14.0**	9.6	5.0	15.0	13.4	14.4	**23.6**	17.0
arb_Arab	1.8	5.0	7.0	8.6	**11.4**	8.0	7.0	**15.2**	11.6	14.6	14.4	14.6
arn_Latn	4.0	9.6	10.2	11.0	**14.8**	12.6	4.8	**30.8**	16.8	22.4	28.6	29.6
ary_Arab	2.2	4.0	5.4	4.6	**8.8**	6.0	2.8	9.6	7.4	12.8	**18.8**	12.2
arz_Arab	2.4	6.2	7.2	6.6	**14.4**	7.8	5.4	20.0	14.4	26.8	**29.4**	19.2
asm_Beng	2.4	6.8	13.0	19.0	**36.4**	12.4	26.2	59.6	46.6	61.2	**63.0**	61.2
ayr_Latn	3.0	8.4	14.6	13.4	**21.8**	15.0	4.8	32.4	30.0	40.6	**53.8**	45.2
azb_Arab	2.2	8.6	11.6	15.2	**29.0**	14.0	7.4	55.4	51.0	63.6	**72.0**	60.6
aze_Latn	2.6	18.4	18.2	32.4	**60.8**	30.4	71.0	74.0	67.4	69.2	73.8	**77.0**
bak_Cyrl	2.2	9.4	13.4	18.6	**32.2**	17.8	5.4	66.6	55.8	65.4	65.8	**71.0**
bam_Latn	3.0	11.0	14.0	11.6	**19.6**	14.2	3.4	38.0	34.0	47.4	48.0	**53.6**
ban_Latn	4.0	7.8	11.6	11.0	**16.0**	11.0	9.0	36.2	28.0	31.8	**41.0**	39.8
bar_Latn	7.0	8.6	13.0	13.4	**17.8**	13.0	13.4	29.4	24.8	39.4	41.6	**46.6**
bba_Latn	2.4	8.8	12.6	10.2	**18.8**	12.0	3.8	23.8	22.8	27.2	34.4	**36.6**
bbc_Latn	3.2	15.0	20.2	23.8	**40.2**	24.6	7.8	59.6	48.4	52.8	**63.2**	**63.2**
bci_Latn	2.6	7.2	**8.0**	5.8	7.6	7.6	4.4	**13.4**	9.6	10.4	13.2	11.6
bcl_Latn	4.0	32.4	33.4	33.4	**65.6**	42.6	10.2	77.4	75.0	77.6	80.6	**82.8**
bel_Cyrl	2.6	13.0	20.8	26.8	**44.8**	18.2	67.2	65.2	53.6	66.6	64.6	**70.4**
bem_Latn	2.8	12.8	18.8	25.6	**36.4**	21.0	6.6	53.2	52.6	59.0	64.8	**66.8**
ben_Beng	2.2	6.2	15.4	17.0	**31.4**	11.4	46.4	**58.0**	44.0	50.6	56.4	57.4
bhw_Latn	5.0	9.8	12.8	13.2	**20.6**	12.4	4.4	38.2	28.6	**40.8**	40.2	**40.8**
bim_Latn	3.4	10.8	10.6	9.4	**19.2**	14.8	4.2	42.4	28.2	32.0	42.8	**59.0**
bis_Latn	3.8	24.2	18.0	24.0	**43.4**	26.4	7.0	49.6	36.6	36.8	47.4	**50.8**
bod_Tibt	2.2	6.4	7.8	12.4	**28.0**	11.8	2.0	21.8	27.0	40.6	**46.8**	37.4
bqc_Latn	2.8	6.6	8.4	9.6	**15.8**	8.2	3.4	35.4	21.8	32.0	37.6	**40.6**
bre_Latn	6.4	9.0	8.8	10.0	**10.8**	9.6	17.6	33.4	24.6	28.8	34.6	**34.8**
bts_Latn	3.2	18.8	22.6	22.6	**41.6**	25.2	6.0	65.8	53.0	58.4	**70.4**	68.2
btx_Latn	3.8	16.4	16.6	17.4	**35.0**	25.6	11.0	54.0	41.4	53.2	61.8	**62.6**
bul_Cyrl	2.2	16.8	31.8	40.0	**62.8**	38.8	81.2	79.4	67.8	78.8	77.8	**81.6**
bum_Latn	2.6	7.8	6.4	7.4	**11.8**	7.0	4.8	27.2	30.8	30.8	**44.4**	36.2
bzj_Latn	6.2	21.4	22.6	27.8	**45.4**	27.2	7.8	68.4	61.0	68.2	**76.0**	71.0
cab_Latn	2.2	5.6	5.6	7.2	**10.4**	7.6	5.8	13.4	11.8	15.8	**18.0**	15.2
cac_Latn	2.4	5.6	6.4	7.6	**9.6**	6.2	3.6	9.4	9.4	12.2	**14.4**	11.6
cak_Latn	2.4	8.4	8.8	13.6	**16.0**	10.8	3.4	16.8	11.6	17.0	**20.6**	19.0
caq_Latn	2.6	8.4	12.0	10.6	**19.4**	8.4	3.2	28.0	25.4	29.8	**42.8**	36.0
cat_Latn	12.6	30.6	38.4	42.0	**65.2**	37.4	**86.6**	81.0	74.2	80.4	81.2	83.4
cbk_Latn	10.0	20.4	23.2	35.4	**54.0**	31.8	31.8	57.8	57.8	57.0	**69.6**	60.6
cce_Latn	3.8	10.0	14.0	17.2	**22.2**	14.8	5.2	42.4	35.2	42.4	51.4	**53.2**
ceb_Latn	3.6	31.0	32.8	44.8	**51.6**	36.4	14.2	73.2	67.0	72.0	**73.4**	72.4
ces_Latn	4.0	10.8	21.6	21.2	**34.6**	21.2	**75.2**	63.0	53.4	60.8	64.0	66.2
cfm_Latn	3.8	13.0	10.4	14.8	**25.4**	15.8	4.6	41.4	36.6	38.6	45.4	**47.2**
che_Cyrl	2.0	3.8	4.8	5.2	**6.4**	4.8	3.4	9.4	9.4	11.8	**14.4**	10.2
chk_Latn	3.6	9.8	15.2	15.2	**22.4**	13.6	5.4	44.4	31.6	44.6	49.4	**52.8**
chv_Cyrl	2.2	9.2	10.2	18.4	**26.6**	16.6	4.6	51.8	44.8	58.2	**61.0**	59.6
ckb_Arab	2.2	8.2	12.4	16.0	**24.8**	12.2	4.0	32.2	31.2	31.2	34.0	**34.2**
cmn_Hani	2.4	14.0	21.0	29.2	**41.0**	28.8	39.2	42.4	38.6	42.6	42.8	**43.2**
cnh_Latn	3.8	11.0	10.6	15.8	**25.0**	14.4	4.8	46.2	36.2	44.0	48.4	**58.6**
crh_Cyrl	2.6	10.0	11.6	22.8	**37.8**	25.0	8.8	68.2	62.0	72.2	74.4	**75.8**
crs_Latn	4.6	33.6	41.2	44.4	**62.4**	39.8	7.4	84.0	81.2	87.0	**88.6**	85.8
csy_Latn	3.0	13.8	9.8	15.2	**22.4**	21.0	3.8	50.0	37.0	44.2	55.4	**57.4**
ctd_Latn	3.8	13.6	8.6	12.8	**25.8**	20.4	4.2	52.6	37.0	48.0	55.2	**61.2**
ctu_Latn	2.8	6.2	8.2	6.4	**9.4**	6.6	2.8	20.0	13.2	16.6	20.8	**21.6**
cuk_Latn	3.8	4.6	6.8	7.2	**9.2**	7.2	5.0	14.0	12.8	15.4	**22.4**	18.8
cym_Latn	3.6	6.8	9.4	10.2	**17.8**	9.2	38.8	**47.0**	33.0	44.0	46.2	46.8
dan_Latn	5.4	25.4	35.8	36.6	**52.4**	36.4	**71.6**	67.2	59.4	67.4	63.2	69.0
deu_Latn	10.2	24.4	33.6	39.2	**58.8**	33.8	**78.8**	74.6	65.4	73.6	75.0	76.6
djk_Latn	3.0	10.8	12.6	16.2	**21.4**	16.0	4.6	38.4	32.0	38.0	**47.0**	40.4
dln_Latn	3.6	12.6	12.2	14.6	**24.4**	20.6	5.2	53.2	44.2	56.4	**66.2**	60.0
dtp_Latn	3.6	6.6	6.2	12.4	**13.4**	8.6	5.4	18.0	14.4	18.2	**24.2**	23.4
dyu_Latn	2.6	7.8	9.8	11.0	**18.2**	13.4	4.2	35.0	29.6	42.2	42.2	**46.2**
dzo_Tibt	2.0	5.0	5.6	11.6	**23.6**	8.4	2.2	18.0	20.2	31.0	**45.4**	34.8
efi_Latn	3.6	11.4	18.4	20.4	**31.0**	23.2	4.4	46.6	43.2	45.2	54.8	**59.4**
ell_Grek	2.2	8.2	14.8	21.8	**33.2**	14.8	**52.6**	48.6	40.4	47.0	48.0	49.4
enm_Latn	29.4	52.4	38.8	46.6	54.8	**58.8**	39.8	68.8	68.8	**74.4**	**74.4**	70.8
epo_Latn	7.0	17.6	27.0	36.6	**45.2**	30.6	64.6	63.0	51.2	60.2	59.2	**67.6**
est_Latn	2.8	10.4	16.0	16.2	**31.6**	21.6	**72.0**	62.8	53.4	60.0	65.4	68.0
eus_Latn	3.8	5.6	7.4	7.6	**9.6**	6.8	**26.2**	24.0	14.8	19.2	20.0	23.4
ewe_Latn	2.0	9.6	11.4	16.0	**23.0**	15.4	4.6	37.0	31.8	30.8	41.0	**43.4**
fao_Latn	4.2	22.4	30.0	37.2	**53.2**	31.2	24.0	77.6	73.6	78.6	81.0	**82.6**
fas_Arab	2.6	18.8	26.8	44.0	**72.2**	41.8	78.2	86.6	78.8	85.4	87.4	**89.4**
fij_Latn	3.2	9.8	14.8	12.2	18.4	**19.4**	3.8	34.6	27.2	33.6	36.0	**36.8**
fil_Latn	3.8	34.0	35.2	52.8	**67.2**	52.4	60.4	80.6	71.2	78.0	82.0	**82.4**
fin_Latn	3.6	7.6	12.0	12.0	**24.4**	11.8	**75.6**	58.0	36.0	46.6	49.4	62.6
fon_Latn	2.0	7.2	8.8	8.2	**13.4**	9.4	2.6	19.8	17.8	19.0	31.6	**33.8**
fra_Latn	9.0	36.6	34.4	35.8	**65.6**	47.2	**88.6**	82.8	76.4	81.4	82.6	86.4
fry_Latn	6.4	16.4	20.4	20.2	**29.6**	18.4	27.8	47.4	41.6	46.8	49.2	**51.6**
gaa_Latn	2.4	11.6	13.2	18.2	**26.8**	18.4	3.8	41.4	35.4	31.4	49.4	**53.6**
gil_Latn	3.8	9.0	11.2	9.8	**15.0**	11.6	5.6	26.6	28.0	**36.6**	31.0	33.2
giz_Latn	2.4	8.6	8.8	11.8	**17.6**	14.8	6.2	38.4	26.6	33.8	**44.2**	40.8
gkn_Latn	2.4	7.6	6.2	8.8	**11.2**	8.6	4.0	23.4	13.4	22.6	**30.6**	30.0
gkp_Latn	2.2	4.8	7.4	5.6	**7.8**	6.2	3.0	13.8	9.4	13.4	**19.2**	18.0

Table 8: Top-10 accuracy of baselines and models initialized with OFA on **SR-B** (Part I).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
gla_Latn	3.6	7.4	8.2	9.4	**14.8**	9.2	25.2	38.2	27.4	34.8	**41.4**	39.8
gle_Latn	3.0	6.6	9.6	11.2	**16.0**	10.4	35.0	**41.6**	29.0	35.4	37.0	40.8
glv_Latn	3.0	8.4	9.8	11.8	**13.8**	10.4	5.8	35.2	31.4	39.8	**46.6**	44.8
gom_Latn	3.6	5.4	5.6	5.6	**13.0**	8.0	6.0	37.2	27.0	37.6	**47.6**	42.0
gor_Latn	3.0	8.4	8.8	10.2	**12.8**	9.8	3.8	20.0	16.6	21.2	**27.6**	25.0
grc_Grek	2.2	3.6	9.8	11.2	**21.8**	11.0	17.4	49.6	33.2	41.2	46.8	**50.4**
guc_Latn	2.4	6.0	5.4	5.2	**8.8**	6.4	3.4	9.0	9.4	9.6	10.8	**11.2**
gug_Latn	3.4	8.0	9.8	12.0	**16.0**	9.8	4.6	33.8	29.8	35.0	**40.0**	38.8
guj_Gujr	2.0	8.4	18.0	24.2	**47.8**	12.0	53.8	69.6	55.0	60.4	67.4	**74.0**
gur_Latn	3.0	8.4	**11.0**	8.0	**11.0**	7.4	3.8	18.6	16.6	19.8	**25.0**	21.2
guw_Latn	2.6	7.6	12.8	16.6	**26.2**	15.0	4.0	38.4	38.4	43.6	48.6	**50.0**
gya_Latn	2.6	12.4	10.4	13.8	**21.0**	13.4	3.6	32.8	27.8	30.8	**47.4**	40.4
gym_Latn	3.0	5.2	8.8	7.8	**9.4**	6.8	3.6	13.6	10.0	13.0	**16.2**	15.6
hat_Latn	2.8	14.4	20.8	29.8	**54.6**	26.0	6.0	78.2	68.8	75.6	**79.8**	79.2
hau_Latn	4.4	8.8	9.2	13.2	14.0	**16.4**	28.8	54.0	48.6	53.8	59.0	**63.4**
haw_Latn	2.8	8.8	14.4	13.0	**19.8**	12.2	4.2	34.8	30.6	30.2	35.6	**36.2**
heb_Hebr	2.0	3.2	6.4	10.8	**12.6**	4.6	**25.0**	23.0	18.6	21.4	21.8	22.2
hif_Latn	4.6	11.0	13.0	12.0	**20.2**	12.2	12.2	25.8	28.2	**41.2**	38.2	27.4
hil_Latn	3.0	24.8	29.6	39.4	**58.0**	33.4	11.0	79.8	72.4	74.2	79.2	**80.6**
hin_Deva	2.6	14.4	25.0	35.0	**64.0**	24.8	67.0	74.8	70.4	73.8	78.4	**78.8**
hin_Latn	2.8	7.6	9.2	12.6	**18.2**	9.6	13.6	32.6	32.4	41.6	**43.0**	34.2
hmo_Latn	3.0	16.2	24.4	28.0	**40.4**	24.8	6.4	**62.8**	44.6	45.8	52.2	61.6
hne_Deva	1.8	8.8	18.8	24.0	**42.8**	24.0	13.4	76.6	56.0	77.4	**86.2**	83.0
hnj_Latn	2.6	10.2	16.0	28.2	**47.2**	23.6	2.8	53.4	38.8	47.8	53.2	**57.6**
hra_Latn	4.0	8.8	11.8	14.6	**18.6**	14.8	5.2	47.8	37.6	50.6	54.0	**57.0**
hrv_Latn	5.8	33.0	44.8	56.6	**72.2**	47.2	79.8	78.4	74.4	78.2	**81.2**	80.6
hui_Latn	2.6	5.8	7.6	9.0	**13.0**	10.6	3.8	19.4	14.2	18.6	**27.8**	24.8
hun_Latn	3.0	9.0	10.8	12.8	**23.6**	15.6	**76.4**	59.2	38.6	49.0	55.2	64.4
hus_Latn	2.6	7.6	5.6	7.8	**9.8**	7.2	3.6	15.8	11.4	13.0	17.8	**19.0**
hye_Armn	1.6	9.0	15.4	23.6	**42.0**	13.6	30.8	67.6	49.0	64.0	**68.8**	65.8
iba_Latn	3.8	17.4	17.6	26.8	**44.4**	26.4	14.4	**76.4**	57.0	66.0	72.0	69.6
ibo_Latn	2.6	8.8	14.2	17.8	**27.4**	14.0	5.0	28.4	23.2	25.4	**35.0**	32.8
ifa_Latn	2.8	9.8	9.4	11.6	**19.8**	14.2	4.4	28.4	17.8	24.4	29.2	**33.4**
ifb_Latn	2.6	9.4	12.0	14.8	**21.2**	11.2	4.8	27.8	17.8	25.6	29.0	**32.2**
ikk_Latn	2.6	10.6	11.6	16.6	**26.0**	17.6	3.0	40.2	29.6	38.8	49.4	**51.2**
ilo_Latn	4.0	15.4	16.8	22.2	**40.0**	27.4	6.2	55.2	46.4	54.6	61.2	**62.6**
ind_Latn	3.4	31.2	37.0	50.0	**72.6**	51.0	**82.6**	78.0	71.0	72.4	78.0	78.8
isl_Latn	3.8	15.4	22.2	26.2	**42.8**	20.6	62.6	70.8	55.6	62.8	67.6	**73.4**
ita_Latn	10.4	34.6	42.8	56.0	**69.6**	46.0	75.4	75.8	70.8	73.2	74.6	**78.4**
ium_Latn	2.8	7.2	10.2	7.0	**14.8**	8.4	3.2	24.4	18.4	21.0	25.2	**26.4**
ixl_Latn	2.2	6.4	5.4	6.8	**8.4**	6.4	4.0	10.4	9.0	12.2	**17.4**	13.2
izz_Latn	2.8	6.8	8.0	11.6	**13.6**	11.8	2.8	16.8	14.0	19.4	**28.6**	23.0
jam_Latn	4.0	22.0	18.6	24.2	**38.6**	30.2	6.6	63.4	55.8	61.4	**67.8**	66.4
jav_Latn	3.0	11.8	16.2	11.4	**22.4**	15.8	25.4	56.8	41.6	48.2	55.0	**58.8**
jpn_Jpan	3.6	12.2	13.8	23.2	**38.8**	20.6	65.0	63.6	40.0	51.4	58.6	**71.2**
kaa_Cyrl	2.0	9.8	12.8	21.0	**32.0**	18.2	17.6	72.8	61.2	72.0	73.8	**76.0**
kaa_Latn	2.8	7.6	9.8	9.8	**19.0**	11.2	9.2	41.6	31.4	35.4	**44.2**	43.8
kab_Latn	2.8	5.4	5.6	4.6	6.0	**8.4**	3.4	14.2	11.8	18.6	**22.4**	20.0
kac_Latn	3.0	6.8	8.2	9.4	**17.8**	9.4	3.6	27.0	13.4	19.2	29.2	**33.0**
kal_Latn	3.2	4.2	6.2	6.2	**8.2**	6.4	3.4	14.2	10.8	15.8	**20.6**	18.0
kan_Knda	1.8	5.2	9.2	11.8	**21.4**	9.8	**51.2**	47.8	29.2	41.0	41.6	46.0
kat_Geor	2.0	7.2	12.6	21.0	**37.0**	15.4	54.2	52.0	39.4	45.8	49.2	**54.6**
kaz_Cyrl	2.0	8.2	12.8	15.6	**27.2**	14.4	61.4	67.6	48.2	62.2	65.2	**71.2**
kbp_Latn	2.4	8.0	9.0	11.0	**16.2**	11.6	2.6	29.0	16.0	23.4	28.0	**33.4**
kek_Latn	2.6	9.6	6.0	8.0	**12.0**	8.0	5.0	16.4	11.4	16.8	**22.4**	20.2
khm_Khmr	2.0	7.6	12.6	15.8	**30.6**	12.2	28.4	43.6	28.6	41.6	39.8	**47.2**
kia_Latn	3.8	9.6	10.0	11.6	**16.8**	14.4	4.0	29.0	19.8	28.0	30.0	**34.8**
kik_Latn	2.6	12.8	15.6	14.4	**32.2**	15.4	3.2	47.4	39.8	48.8	55.0	**56.4**
kin_Latn	4.4	15.6	19.0	24.2	**40.0**	19.2	5.0	56.4	60.4	63.6	**66.4**	63.8
kir_Cyrl	2.0	11.0	13.8	24.0	**36.0**	20.6	54.8	68.6	56.4	63.8	67.0	**71.4**
kjb_Latn	2.4	11.0	11.2	11.8	**19.2**	11.4	4.0	25.0	15.4	20.0	**28.4**	27.6
kjh_Cyrl	2.2	7.8	10.6	11.8	**19.6**	12.4	11.0	44.2	41.6	51.4	56.4	**59.0**
kmm_Latn	4.0	8.6	9.0	9.8	**19.4**	15.4	4.8	39.2	23.4	34.0	39.0	**47.2**
kmr_Cyrl	2.0	6.8	7.6	11.6	**24.8**	8.0	4.0	32.0	30.8	39.2	**46.0**	37.6
kmr_Latn	2.2	14.2	18.6	26.0	**37.4**	21.0	35.8	62.2	56.6	61.8	**67.0**	64.0
knv_Latn	1.8	3.6	4.4	4.8	**7.2**	5.0	2.8	6.4	4.6	7.2	9.0	**10.2**
kor_Hang	2.2	5.8	11.0	17.0	**32.8**	14.0	**64.0**	63.8	42.2	53.2	59.8	62.8
kpg_Latn	3.4	15.8	17.8	20.6	**38.2**	24.2	5.2	45.0	34.4	45.0	**55.0**	54.0
krc_Cyrl	2.0	9.2	11.6	14.8	**28.4**	20.2	9.2	60.6	52.8	58.4	**67.4**	64.6
kri_Latn	3.2	19.8	20.4	29.4	**46.0**	25.2	2.8	56.4	49.0	51.4	62.4	**68.6**
ksd_Latn	4.0	12.2	15.6	14.6	21.2	**21.6**	7.0	40.2	31.4	35.6	33.2	**45.4**
kss_Latn	2.0	2.4	3.2	4.0	**4.4**	3.0	2.2	4.4	3.2	4.6	**5.2**	4.2
ksw_Mymr	2.0	4.4	7.6	10.2	**15.2**	8.4	1.6	19.0	16.2	23.4	**28.2**	25.4
kua_Latn	2.8	10.2	13.0	15.2	**27.4**	14.0	4.8	39.8	40.6	**54.6**	**54.6**	45.2
lam_Latn	2.4	5.4	10.4	9.4	**11.6**	7.2	4.6	22.2	20.4	**27.0**	26.6	25.0
lao_Laoo	2.0	5.6	11.0	15.2	**29.2**	9.0	31.4	**46.8**	30.4	39.4	40.2	43.2
lat_Latn	10.8	19.6	24.0	26.4	**34.8**	31.0	52.2	55.2	45.0	52.8	52.6	**58.0**
lav_Latn	4.8	15.4	19.8	19.4	**36.2**	25.6	**74.2**	67.4	56.8	62.4	64.6	71.0
ldi_Latn	3.0	8.0	**10.4**	10.2	10.0	9.0	5.4	21.4	20.0	25.0	**29.0**	28.6
leh_Latn	2.8	11.0	13.2	16.8	**32.2**	21.2	5.6	54.4	44.4	53.6	55.8	**60.0**
lhu_Latn	2.2	3.6	2.6	3.8	**5.2**	2.6	2.0	4.0	3.4	4.0	**6.8**	3.0
lin_Latn	3.4	13.6	21.0	23.0	**42.0**	26.6	6.6	70.4	61.2	69.2	**76.8**	73.8
lit_Latn	3.8	9.6	13.6	16.2	**23.4**	18.6	**74.4**	60.4	43.8	52.2	55.6	66.8
loz_Latn	3.2	12.6	12.6	17.2	**23.2**	21.0	6.8	43.6	50.4	**57.2**	56.0	55.0
ltz_Latn	8.6	22.2	19.8	24.6	**44.8**	32.6	9.8	71.8	63.0	65.6	**74.2**	72.8
lug_Latn	3.0	7.8	11.8	19.2	**26.2**	16.8	4.6	35.8	37.4	48.0	**53.0**	**53.0**
luo_Latn	4.0	10.6	12.6	11.0	**21.0**	12.4	6.4	42.0	33.0	42.8	**53.6**	47.4
lus_Latn	4.0	6.6	11.4	10.6	**18.0**	15.2	3.8	51.6	43.0	50.8	58.2	**62.2**
lzh_Hani	3.4	15.6	15.8	34.6	**51.4**	31.4	25.0	63.0	56.2	61.2	**66.4**	64.0
mad_Latn	3.6	9.8	10.4	13.0	**19.2**	14.0	7.6	37.8	33.6	43.2	**47.8**	46.0
mah_Latn	3.8	10.0	16.8	11.4	**19.6**	12.0	4.8	30.6	21.0	32.6	**34.6**	30.0
mai_Deva	2.2	7.0	16.4	20.4	**33.4**	16.2	6.4	54.6	43.6	55.4	58.6	**59.8**
mal_Mlym	2.0	4.6	10.6	14.2	**26.2**	7.6	49.4	49.8	34.2	42.6	**51.4**	48.4

Table 9: Top-10 accuracy of baselines and models initialized with OFA on **SR-B** (Part II).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
mam_Latn	2.6	5.8	6.0	6.4	**7.4**	6.0	3.8	11.0	8.4	11.6	**13.2**	10.4
mar_Deva	2.2	9.4	20.0	24.6	**43.2**	17.8	66.2	65.2	54.4	60.6	66.2	**72.8**
mau_Latn	2.0	3.4	2.8	3.2	**4.6**	4.2	2.4	3.2	3.2	3.6	3.2	**3.8**
mbb_Latn	2.4	11.8	10.0	16.0	**26.8**	17.0	3.0	25.0	19.4	27.6	**34.6**	30.8
mck_Latn	2.6	13.0	19.6	20.0	**35.2**	22.2	5.2	65.6	53.0	59.0	64.8	**67.2**
mcn_Latn	3.4	10.0	9.6	12.0	**23.4**	12.0	6.0	45.4	25.0	36.2	43.0	**46.0**
mco_Latn	2.2	5.0	4.6	4.4	**6.0**	4.8	2.6	**8.0**	4.8	6.8	7.8	6.6
mdy_Ethi	2.0	4.6	8.6	8.0	**16.4**	7.0	2.8	30.0	20.4	32.6	**47.2**	35.0
meu_Latn	3.2	11.8	18.4	20.2	**28.4**	23.0	5.6	51.0	43.8	48.4	49.8	**55.2**
mfe_Latn	5.2	30.6	37.4	39.6	**61.6**	33.4	9.0	77.2	72.6	79.6	**83.6**	83.4
mgh_Latn	3.2	6.4	8.4	8.8	**10.6**	9.4	5.2	15.8	16.6	19.0	**25.6**	24.0
mgr_Latn	3.0	14.0	16.0	22.6	**31.6**	18.6	4.0	52.4	50.0	53.8	**58.2**	55.6
mhr_Cyrl	2.2	8.0	10.0	14.2	**23.8**	11.6	6.6	30.4	33.2	41.2	**52.0**	41.8
min_Latn	3.4	9.6	13.0	14.4	**19.4**	12.4	9.4	31.8	21.6	32.0	32.0	**34.6**
miq_Latn	3.8	7.4	9.4	9.6	**19.6**	13.6	4.4	40.0	23.2	30.6	37.8	**47.2**
mkd_Cyrl	2.6	22.6	32.4	47.4	**67.4**	38.2	76.6	**78.8**	69.2	77.2	78.4	77.6
mlg_Latn	4.0	10.6	9.6	11.8	**18.0**	11.4	29.0	58.4	42.6	57.0	60.2	**61.4**
mlt_Latn	4.0	15.4	14.8	25.2	**37.2**	24.4	5.8	46.6	44.6	51.0	**53.4**	53.0
mos_Latn	3.4	4.4	11.0	8.8	**17.0**	9.2	4.2	33.4	28.4	32.4	39.8	**46.4**
mps_Latn	2.2	7.4	6.6	9.2	**16.2**	12.4	3.2	14.8	11.2	15.8	20.6	**23.0**
mri_Latn	3.6	15.4	15.4	16.8	**32.2**	19.0	4.2	44.8	46.0	50.8	**52.0**	51.4
mrw_Latn	2.4	11.8	15.8	15.2	**25.2**	14.8	6.0	33.0	23.0	32.0	39.8	**45.8**
msa_Latn	3.0	22.8	27.6	34.0	**42.2**	32.2	40.0	43.4	41.0	41.2	**44.8**	44.2
mwm_Latn	2.0	6.8	11.2	12.2	**18.8**	10.2	2.6	25.4	13.6	20.8	28.4	**33.0**
mxv_Latn	2.6	3.8	4.8	5.4	**6.8**	4.8	3.0	6.8	4.6	6.4	**8.8**	6.6
mya_Mymr	1.8	4.0	6.6	11.2	**15.4**	7.8	20.2	26.2	19.2	26.8	27.8	**28.2**
myv_Cyrl	2.2	5.8	8.2	9.4	**16.4**	8.0	4.6	32.4	27.0	38.4	**43.0**	35.8
mzh_Latn	3.0	10.0	8.2	10.6	**16.6**	11.6	4.6	25.0	16.8	23.4	**33.8**	33.2
nan_Latn	2.4	6.6	6.8	5.6	**7.8**	5.4	3.2	13.6	11.8	12.0	13.8	**14.6**
naq_Latn	2.2	4.0	6.4	7.0	**11.8**	7.6	3.0	18.0	15.8	22.2	25.2	**30.4**
nav_Latn	2.2	5.0	6.2	5.4	**6.8**	5.0	2.4	10.0	8.2	10.0	11.6	**12.0**
nbl_Latn	3.2	12.0	15.8	22.0	**34.2**	18.0	9.2	47.6	53.4	59.2	**64.4**	57.6
nch_Latn	3.2	5.4	10.8	10.4	**11.2**	9.8	4.4	17.4	11.6	14.8	**20.8**	17.6
ncj_Latn	2.8	6.0	8.2	8.0	**12.2**	8.4	4.6	19.0	10.2	18.4	20.6	**21.0**
ndc_Latn	3.4	12.0	14.8	22.2	**31.4**	16.2	5.2	37.6	35.8	**42.0**	41.0	41.4
nde_Latn	3.0	13.8	17.0	24.0	**36.6**	22.6	13.0	54.2	53.0	57.8	57.4	**63.0**
ndo_Latn	3.6	8.0	12.8	12.2	**19.4**	11.6	5.2	36.6	39.4	49.6	**59.6**	46.4
nds_Latn	5.2	12.2	14.8	18.4	**28.0**	16.4	9.6	37.0	36.4	43.0	**43.8**	41.2
nep_Deva	2.4	10.6	16.0	24.4	**42.2**	26.8	35.6	59.6	49.4	55.8	61.6	**63.8**
ngu_Latn	2.8	10.2	11.4	13.2	**18.0**	8.6	4.6	21.8	22.4	27.6	**28.4**	21.6
nia_Latn	2.6	7.6	8.8	9.4	**13.0**	8.8	4.6	25.0	20.0	28.4	**35.6**	27.4
nld_Latn	6.0	28.8	34.4	39.2	**61.6**	37.8	78.0	78.6	71.0	75.8	79.6	**83.2**
nmf_Latn	3.8	7.2	8.2	7.4	**13.8**	10.2	4.6	26.4	18.4	28.2	31.6	**35.2**
nnb_Latn	2.6	9.8	11.0	13.0	**22.8**	12.6	3.6	33.0	32.0	42.0	**44.8**	43.2
nno_Latn	5.0	33.0	32.0	47.4	**65.4**	40.4	58.4	74.6	75.2	77.8	76.8	**79.0**
nob_Latn	3.8	38.8	45.4	63.0	**78.6**	48.8	82.6	83.8	78.4	83.8	84.8	**85.8**
nor_Latn	5.6	34.6	50.8	57.6	**76.2**	47.8	81.2	85.4	83.2	82.6	83.4	**87.2**
npi_Deva	2.0	14.2	23.4	34.4	**63.4**	33.4	50.6	80.4	70.6	80.0	81.8	**84.2**
nse_Latn	3.4	13.2	20.0	19.6	**31.2**	18.6	5.2	51.8	52.4	55.6	**57.8**	56.0
nso_Latn	3.8	15.0	14.0	21.8	**42.6**	24.0	6.0	44.8	51.2	52.2	**57.8**	54.0
nya_Latn	2.8	10.8	16.6	18.6	**39.2**	22.6	4.0	61.6	58.8	66.2	65.8	**69.2**
nyn_Latn	2.4	9.8	13.8	20.0	**32.6**	16.6	4.4	45.0	45.8	55.6	**56.2**	55.4
nyy_Latn	2.4	5.2	5.8	8.6	**14.4**	6.6	3.0	20.0	14.0	18.8	24.0	**25.8**
nzi_Latn	3.0	7.0	11.2	8.0	**20.8**	11.6	3.2	31.8	32.0	28.8	**44.8**	44.4
ori_Orya	2.0	5.8	17.4	23.4	**36.4**	13.8	42.6	63.6	43.0	58.0	**68.0**	66.2
ory_Orya	1.8	6.8	14.8	16.6	**27.6**	12.4	31.4	56.0	37.2	51.0	57.4	**57.8**
oss_Cyrl	1.6	7.8	14.8	14.2	**29.2**	13.6	4.2	54.6	45.2	60.8	**68.0**	59.2
ote_Latn	2.6	4.4	4.4	6.8	**9.0**	6.0	3.6	11.0	7.2	10.4	**18.4**	17.6
pag_Latn	4.2	18.6	17.4	18.8	**39.6**	24.6	8.0	55.8	46.6	58.6	**59.8**	59.0
pam_Latn	3.2	11.6	14.8	19.4	**30.0**	19.2	8.2	44.4	35.8	44.2	**50.4**	42.6
pan_Guru	2.0	6.4	12.8	18.0	**29.2**	11.8	43.2	52.8	36.8	44.6	51.2	**56.4**
pap_Latn	7.6	27.2	27.0	38.8	**61.8**	38.2	12.4	72.4	69.8	76.8	77.0	**78.4**
pau_Latn	3.4	7.6	7.4	6.6	**15.0**	11.4	4.4	23.2	12.0	18.0	**27.6**	24.6
pcm_Latn	9.0	28.4	34.4	43.0	**57.0**	39.0	13.6	69.2	65.4	70.6	69.2	**72.6**
pdt_Latn	3.6	19.0	20.4	27.0	**43.8**	22.8	9.2	65.2	56.0	71.4	78.0	**78.6**
pes_Arab	1.8	16.0	25.6	43.0	**66.6**	39.8	69.4	77.4	70.2	76.4	77.0	**79.4**
pis_Latn	3.8	23.2	19.8	22.2	**33.4**	22.0	6.4	50.2	45.4	44.8	52.8	**56.0**
pls_Latn	3.4	7.6	8.4	11.8	**15.2**	10.2	5.0	28.0	21.4	28.6	**32.6**	32.0
plt_Latn	3.2	10.8	10.6	11.4	**18.8**	11.6	26.6	60.2	42.2	57.8	62.0	**62.6**
poh_Latn	3.0	5.0	6.0	4.0	**6.6**	5.6	3.4	11.0	9.2	12.2	**15.4**	12.4
pol_Latn	3.0	12.8	17.0	17.2	**37.4**	20.4	**79.2**	67.8	53.2	66.2	68.4	74.2
pon_Latn	3.4	8.6	9.2	9.2	**16.0**	13.6	5.6	24.2	21.4	23.8	24.4	**26.0**
por_Latn	12.4	37.2	43.8	53.4	**72.4**	52.6	**81.6**	80.0	74.6	80.4	80.0	81.2
prk_Latn	2.6	15.4	23.4	23.6	**45.8**	29.4	3.6	56.4	37.0	52.6	**60.4**	59.6
prs_Arab	2.8	18.8	25.6	45.6	**76.0**	43.6	79.4	**87.2**	78.6	85.4	86.0	**87.2**
pxm_Latn	3.2	7.4	6.8	9.8	**14.2**	8.0	3.2	15.8	14.6	18.6	**24.0**	15.8
qub_Latn	3.2	7.4	10.6	14.0	**22.2**	10.6	4.6	37.0	29.4	37.2	**44.0**	41.8
quc_Latn	2.2	7.6	8.2	10.0	**11.4**	8.8	3.6	18.6	10.8	17.4	23.6	**24.0**
qug_Latn	2.8	8.6	15.2	19.8	**37.0**	25.4	4.8	58.8	49.6	55.4	**64.4**	64.0
quh_Latn	3.2	9.4	14.4	17.8	**24.0**	17.4	4.6	37.8	39.8	49.0	49.0	**51.4**
quw_Latn	3.2	8.8	12.8	12.8	**24.4**	14.4	6.2	44.6	39.0	49.0	**58.6**	58.0
quy_Latn	2.8	12.0	17.0	22.2	**37.6**	23.2	4.6	54.6	46.6	53.0	55.6	**64.8**
quz_Latn	2.2	12.4	19.6	23.0	**44.8**	25.6	4.8	66.4	52.2	65.4	65.4	**69.6**
qvi_Latn	3.4	13.0	16.4	22.0	**35.6**	21.8	4.4	51.6	39.4	48.6	58.6	**64.6**
rap_Latn	2.4	7.4	7.0	8.2	**11.0**	7.4	3.2	**20.2**	14.2	14.4	19.6	19.6
rar_Latn	2.6	5.6	10.0	10.2	**18.6**	10.8	3.2	21.2	19.6	22.4	**24.4**	23.0
rmy_Latn	5.0	11.0	11.2	16.0	**17.4**	12.6	6.8	25.8	38.0	39.4	**42.2**	38.0
ron_Latn	6.6	22.2	24.0	30.8	**46.6**	27.8	**72.2**	67.0	55.0	60.4	63.6	69.4
rop_Latn	3.0	15.8	18.2	27.6	**42.4**	19.8	4.6	39.8	35.4	41.6	**50.0**	**50.0**
rug_Latn	4.2	7.4	10.4	15.6	**24.2**	11.8	3.6	42.6	21.8	32.4	41.0	**46.4**
run_Latn	3.4	16.6	17.2	29.4	**34.0**	20.4	5.4	53.0	55.0	60.0	59.6	**63.6**
rus_Cyrl	2.0	21.4	29.8	35.6	**61.4**	33.8	**75.8**	72.8	70.6	72.2	73.0	75.2
sag_Latn	3.4	15.2	11.2	18.8	24.8	**25.6**	6.0	43.2	29.8	43.2	**53.6**	47.4

Table 10: Top-10 accuracy of baselines and models initialized with OFA on **SR-B** (Part III).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
sah_Cyrl	1.6	8.6	13.2	16.6	**29.0**	16.2	6.2	49.8	37.4	47.0	**52.6**	52.0
san_Deva	1.6	5.4	6.0	8.0	**14.2**	7.2	13.8	23.2	16.4	19.6	**27.2**	24.8
san_Latn	2.4	3.2	3.0	3.0	**5.2**	3.0	4.6	9.4	4.4	6.4	**10.0**	9.8
sba_Latn	2.4	8.4	11.6	14.4	**15.8**	10.2	2.8	26.0	23.0	21.8	30.2	**34.0**
seh_Latn	3.0	15.4	22.4	27.2	**40.0**	29.2	6.4	63.6	62.4	75.2	**75.6**	74.2
sin_Sinh	1.8	4.6	7.4	10.6	**19.0**	8.2	44.8	48.0	28.4	37.8	**48.4**	44.4
slk_Latn	3.8	11.4	21.0	20.0	**37.2**	21.8	**75.2**	66.4	54.8	63.4	64.0	68.8
slv_Latn	6.4	14.8	19.4	21.6	**37.2**	22.0	**63.6**	57.8	49.0	52.2	52.8	59.0
sme_Latn	2.8	8.2	12.6	10.4	**22.0**	15.2	6.8	37.8	35.6	45.8	**50.6**	45.2
smo_Latn	2.6	11.8	10.8	15.0	**23.6**	16.4	4.4	30.8	22.8	29.2	**35.0**	32.0
sna_Latn	2.8	11.8	14.6	22.4	**31.8**	18.6	7.0	43.8	42.6	47.6	45.0	**48.0**
snd_Arab	2.2	6.4	10.8	15.2	**29.6**	11.4	52.2	57.0	40.0	59.8	65.2	**68.8**
som_Latn	2.6	6.2	6.0	5.6	**8.4**	7.0	22.2	**40.6**	21.4	25.8	29.0	38.4
sop_Latn	2.8	6.6	12.8	16.6	**18.8**	9.6	5.2	26.6	27.4	30.8	**33.2**	30.8
sot_Latn	3.8	16.6	17.0	28.2	**45.8**	26.4	6.0	51.0	52.6	56.0	59.8	**61.0**
spa_Latn	20.6	46.2	49.6	64.0	**76.0**	59.4	**81.2**	81.0	76.6	80.0	80.4	78.2
sqi_Latn	8.8	28.0	24.2	37.4	**57.0**	42.4	58.2	63.0	61.0	63.8	**66.0**	64.2
srm_Latn	3.0	8.4	8.6	13.4	**21.2**	11.0	4.0	26.8	17.2	27.2	**34.4**	30.8
srn_Latn	5.6	32.0	24.4	34.6	**61.6**	31.6	6.8	73.4	69.6	72.0	**79.8**	77.2
srp_Cyrl	2.6	29.6	46.4	63.0	**79.6**	55.4	83.0	85.4	84.0	**88.8**	88.0	87.6
srp_Latn	7.4	35.2	51.8	63.8	**79.8**	56.0	85.0	85.0	82.4	86.6	**87.2**	86.8
ssw_Latn	2.4	10.6	13.6	16.8	**33.4**	14.2	4.8	44.0	41.8	51.2	53.8	**54.8**
sun_Latn	4.2	10.8	14.6	15.8	**27.6**	19.2	22.4	50.2	45.4	50.0	54.0	**56.6**
suz_Deva	2.2	4.0	4.8	6.8	**13.6**	8.4	3.6	25.2	13.8	26.4	**32.8**	22.8
swe_Latn	4.8	25.0	33.8	30.8	**52.0**	34.6	**79.8**	77.2	65.0	71.0	73.4	77.4
swh_Latn	3.4	12.8	18.8	23.2	**49.4**	32.2	47.8	72.0	62.8	72.0	71.8	**76.6**
sxn_Latn	3.2	6.4	10.0	9.8	**13.4**	8.2	4.8	22.6	19.4	22.0	**26.4**	24.0
tam_Taml	2.2	4.2	8.6	11.6	**25.8**	4.8	42.8	**51.2**	31.8	39.4	47.4	47.8
tat_Cyrl	1.8	12.2	17.2	23.4	**41.8**	20.8	8.2	65.0	61.0	68.6	**74.4**	71.8
tbz_Latn	1.6	4.4	8.6	7.0	**12.2**	9.6	2.6	15.0	12.4	21.6	**27.2**	22.0
tca_Latn	2.6	5.8	6.8	7.2	**10.2**	7.0	2.4	11.8	8.4	10.0	**17.8**	16.0
tdt_Latn	3.6	17.6	18.0	22.4	**38.4**	17.6	6.2	50.6	44.2	50.2	**62.0**	59.4
tel_Telu	1.8	4.4	11.4	13.0	**23.8**	8.6	44.4	42.2	30.4	34.2	42.6	**48.6**
teo_Latn	3.6	6.4	8.4	8.6	**10.0**	7.8	5.8	16.0	16.6	22.2	**26.2**	21.0
tgk_Cyrl	1.8	14.8	19.2	27.2	**49.2**	23.4	4.6	67.4	62.8	61.8	**75.0**	72.4
tgl_Latn	3.4	37.0	36.2	53.4	**66.6**	52.2	61.0	79.2	70.8	77.4	**81.8**	80.6
tha_Thai	2.0	5.4	9.0	15.2	**28.6**	9.6	30.0	34.8	27.8	38.0	37.2	**39.6**
tih_Latn	2.2	15.4	15.2	16.2	**30.8**	15.6	5.2	46.6	30.4	37.8	47.8	**54.8**
tir_Ethi	1.8	6.2	9.0	14.0	**24.8**	10.4	7.4	37.2	31.8	39.2	**48.4**	43.8
tlh_Latn	6.0	28.4	27.8	37.6	**48.6**	29.4	7.8	61.8	60.8	64.8	**73.4**	71.4
tob_Latn	2.4	4.0	5.4	8.4	**9.4**	6.8	2.2	13.8	8.6	11.6	**16.6**	16.0
toh_Latn	2.6	9.6	12.8	14.0	**25.2**	16.0	4.0	41.0	32.8	40.2	46.4	**47.4**
toi_Latn	3.4	9.8	14.0	16.6	**29.0**	14.0	4.2	41.0	36.8	45.4	**45.8**	42.4
toj_Latn	3.0	7.6	7.2	8.2	**8.8**	7.4	4.2	13.4	10.6	11.8	**15.8**	14.6
ton_Latn	2.4	7.0	7.0	10.0	**13.6**	5.8	4.2	15.0	13.2	17.0	**22.0**	16.0
top_Latn	2.6	4.2	3.4	4.8	**5.4**	4.2	3.4	5.4	4.6	6.0	**8.2**	5.8
tpi_Latn	4.4	29.6	20.6	36.2	**52.6**	43.6	5.8	59.6	50.6	50.6	55.0	**62.6**
tpm_Latn	2.4	10.6	11.6	7.2	16.0	**16.8**	3.6	34.2	25.4	30.0	27.4	**36.2**
tsn_Latn	3.0	8.4	10.6	14.2	**21.8**	12.4	5.4	23.0	34.8	35.6	**38.8**	36.8
tso_Latn	3.6	13.6	14.6	22.0	**32.4**	20.0	5.6	49.2	51.6	56.6	59.4	**60.4**
tsz_Latn	2.2	6.4	8.0	8.8	**15.2**	10.0	5.6	25.6	23.2	25.0	28.4	**30.4**
tuc_Latn	3.0	9.4	7.2	14.0	**15.2**	12.6	2.6	24.8	20.4	24.6	**31.2**	27.8
tui_Latn	3.0	7.8	10.4	12.2	**14.4**	10.2	3.6	26.2	19.4	27.8	**41.0**	35.4
tuk_Cyrl	2.0	10.2	15.6	16.2	**27.6**	18.8	13.6	64.8	55.0	67.0	**71.6**	65.8
tuk_Latn	3.4	8.8	12.2	18.6	**40.0**	18.6	9.6	68.0	59.6	69.2	**74.4**	71.2
tum_Latn	3.2	12.6	19.2	27.0	**36.0**	23.0	5.2	54.8	53.0	**67.0**	61.8	61.2
tur_Latn	2.6	13.8	15.4	17.8	**39.4**	25.8	**74.4**	66.4	54.0	63.4	65.6	69.6
twi_Latn	2.4	8.6	12.6	16.4	**26.8**	15.4	3.8	42.8	36.8	40.4	47.2	**47.4**
tyv_Cyrl	2.0	6.6	9.8	10.4	**19.0**	11.0	6.8	43.0	32.2	46.8	**52.4**	50.8
tzh_Latn	3.0	7.4	7.2	7.2	**11.8**	8.2	6.0	15.8	15.6	20.0	**25.6**	20.6
tzo_Latn	2.2	5.8	6.6	7.2	**7.8**	7.4	3.8	13.6	9.4	11.0	13.6	**14.0**
udm_Cyrl	2.0	9.4	11.8	13.6	**23.6**	12.0	6.0	45.8	37.2	47.4	**56.8**	47.4
uig_Arab	2.0	4.6	6.8	10.4	**22.4**	7.0	45.8	56.0	32.0	43.6	52.8	**58.2**
uig_Latn	2.8	6.8	7.6	10.8	**18.2**	11.0	9.8	57.4	51.0	57.4	**63.2**	63.0
ukr_Cyrl	2.2	12.8	21.8	29.4	**47.4**	20.2	66.0	64.8	54.2	65.8	65.4	**66.4**
urd_Arab	2.2	13.4	27.6	30.8	**50.6**	22.2	47.6	62.2	56.2	63.4	64.6	**65.4**
uzb_Cyrl	2.6	14.8	25.4	43.8	**70.2**	33.0	6.2	81.0	76.2	78.8	82.2	**82.8**
uzb_Latn	3.4	9.6	14.6	19.8	**38.6**	17.0	54.8	73.6	56.0	64.4	67.2	**74.6**
uzn_Cyrl	1.8	19.8	22.6	42.8	**65.8**	34.6	5.4	82.4	78.4	80.6	82.4	**85.0**
ven_Latn	2.6	8.8	11.2	17.0	**30.2**	13.6	4.8	37.0	36.6	47.6	44.8	**54.4**
vie_Latn	2.4	7.6	17.0	18.2	**29.2**	15.2	**72.8**	67.0	47.8	60.0	60.8	66.2
wal_Latn	3.0	5.8	7.4	9.8	**15.0**	9.0	4.2	37.8	30.4	48.6	**57.8**	48.6
war_Latn	3.6	20.8	26.0	31.8	**37.4**	25.0	9.8	50.4	45.6	52.6	47.4	**53.8**
wbm_Latn	2.8	15.6	19.4	21.4	**40.8**	23.6	3.8	53.8	30.0	44.6	55.8	**57.4**
wol_Latn	3.6	8.8	9.0	6.0	**12.8**	7.8	4.6	35.0	29.0	41.0	**47.0**	36.0
xav_Latn	2.4	3.0	3.2	3.4	**4.0**	**4.0**	2.2	3.8	3.2	4.4	5.0	**5.2**
xho_Latn	2.6	10.8	16.8	18.6	**30.2**	16.2	10.4	45.8	38.4	48.6	49.6	**53.2**
yan_Latn	2.6	7.4	9.6	9.4	**17.2**	9.4	4.2	29.4	16.2	26.0	27.0	**34.0**
yao_Latn	3.2	8.6	11.2	10.4	**22.4**	10.8	4.4	40.6	39.4	47.2	**52.0**	45.8
yap_Latn	4.0	8.8	6.0	8.8	**12.2**	10.6	4.0	18.2	12.6	18.2	18.8	**20.0**
yom_Latn	2.8	8.8	11.6	12.4	**22.2**	14.8	4.8	37.4	33.6	41.4	**42.6**	40.2
yor_Latn	3.0	5.4	9.4	10.8	**18.0**	11.2	3.4	33.0	24.2	30.0	**37.2**	33.8
yua_Latn	2.8	7.6	7.8	7.8	**9.4**	8.6	3.8	9.6	10.8	14.8	**17.4**	14.2
yue_Hani	2.2	6.2	10.8	8.6	**12.0**	**12.0**	**17.2**	14.4	13.4	13.8	14.2	13.0
zai_Latn	4.0	8.8	11.2	13.6	**19.8**	13.0	6.2	22.6	24.0	26.6	**36.0**	30.0
zho_Hani	2.4	12.6	23.4	30.4	**41.4**	25.4	40.4	43.8	40.0	44.6	44.4	**45.0**
zlm_Latn	3.4	35.8	40.2	49.0	**72.0**	53.6	83.4	84.4	79.8	80.2	84.2	**85.6**
zom_Latn	3.6	14.2	8.4	13.0	**23.2**	18.2	3.6	49.0	36.2	45.2	49.6	**53.8**
zsm_Latn	2.6	40.2	42.8	58.4	**82.2**	62.4	90.2	88.8	84.8	86.0	**90.4**	88.6
zul_Latn	3.4	9.6	16.4	19.8	**37.4**	15.4	11.0	53.2	44.8	53.6	54.6	**59.6**

Table 11: Top-10 accuracy of baselines and models initialized with OFA on **SR-B** (Part IV).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
afr_Latn	4.1	19.9	31.5	33.7	**40.0**	22.9	71.9	74.2	73.7	65.0	69.1	**75.6**
amh_Ethi	6.5	14.3	16.1	16.7	**28.6**	16.7	35.1	42.9	40.5	39.3	**46.4**	44.0
ara_Arab	1.0	8.4	13.0	18.8	**31.4**	11.5	**59.2**	57.3	56.8	51.9	50.3	54.2
arz_Arab	2.1	13.6	18.2	23.1	**39.4**	17.0	32.5	**57.4**	56.2	47.8	52.0	49.3
ast_Latn	42.5	63.8	61.4	63.0	**70.1**	60.6	59.8	**83.5**	81.1	75.6	81.9	**83.5**
aze_Latn	2.2	22.2	24.6	37.4	**50.8**	31.7	62.6	**75.7**	68.3	66.0	69.8	73.9
bel_Cyrl	1.4	15.9	29.4	34.4	**54.6**	25.9	70.0	75.0	70.4	67.4	69.5	**75.3**
ben_Beng	1.3	6.1	13.5	17.7	**36.5**	13.1	54.1	**65.5**	52.8	46.9	56.9	64.2
bos_Latn	11.9	56.2	59.3	72.0	**79.9**	62.1	78.5	**89.5**	86.7	85.0	86.2	87.3
bre_Latn	4.2	6.6	4.4	7.7	**8.1**	7.0	10.3	18.1	14.1	13.8	14.9	**18.7**
bul_Cyrl	1.4	23.0	32.2	44.9	**65.8**	40.3	**84.4**	82.8	73.7	75.2	77.0	**84.4**
cat_Latn	13.3	39.0	34.3	46.7	**58.9**	43.8	72.8	73.0	71.5	65.4	70.8	**77.9**
cbk_Latn	10.5	25.7	19.8	25.4	**33.0**	26.4	33.2	46.9	47.5	42.9	42.1	**48.0**
ceb_Latn	4.7	22.8	19.5	24.8	**28.0**	24.8	15.2	36.8	**40.2**	39.5	39.0	39.8
ces_Latn	3.1	15.8	19.7	26.7	**38.0**	23.3	**71.1**	64.9	60.8	58.5	60.4	69.2
cmn_Hani	1.5	14.6	32.4	46.1	**69.7**	32.9	**79.5**	79.3	75.7	67.9	69.0	78.4
csb_Latn	7.1	16.2	16.2	17.4	**26.9**	19.0	21.3	35.2	31.2	35.6	**42.7**	40.7
cym_Latn	4.9	13.4	13.2	17.6	**21.7**	15.0	45.7	**53.0**	46.6	45.0	51.3	52.2
dan_Latn	6.3	46.7	62.3	71.0	**76.2**	55.0	**91.9**	89.6	87.2	82.8	86.6	90.0
deu_Latn	13.8	40.7	52.7	61.4	**78.1**	55.9	**95.9**	92.6	92.6	88.2	91.6	95.0
dtp_Latn	2.6	8.2	5.5	9.8	**13.8**	10.2	5.6	18.4	17.1	18.2	**23.0**	20.8
ell_Grek	1.0	7.6	18.3	26.3	**40.6**	17.8	**76.2**	69.2	57.6	62.1	61.9	71.9
epo_Latn	7.6	31.0	36.5	41.5	**56.1**	37.2	64.9	68.6	66.0	64.1	65.1	**72.0**
est_Latn	3.3	13.5	13.8	19.6	**28.8**	18.2	63.9	62.7	54.9	47.4	53.9	**65.3**
eus_Latn	4.7	8.2	10.2	10.3	**14.5**	11.1	45.9	**50.0**	35.6	38.0	37.5	49.8
fao_Latn	8.4	38.5	53.8	57.3	**65.3**	45.0	45.0	80.2	**84.0**	73.7	80.9	75.6
fin_Latn	2.3	11.5	12.8	14.4	**28.2**	15.0	**81.9**	61.5	48.5	46.4	50.0	65.8
fra_Latn	7.5	35.5	28.5	35.6	**54.5**	43.0	**85.7**	80.3	76.1	74.6	76.9	83.2
fry_Latn	22.5	48.6	52.6	52.0	**58.4**	50.3	60.1	72.8	**83.2**	74.0	76.3	72.3
gla_Latn	3.7	6.6	6.6	9.4	**10.4**	7.5	21.0	36.3	29.0	32.3	37.8	**38.5**
gle_Latn	3.0	7.8	8.4	9.6	**22.8**	10.5	32.0	44.3	34.2	35.2	37.8	**44.5**
glg_Latn	16.2	41.3	40.4	48.2	**60.1**	43.9	72.6	71.4	72.4	63.1	70.0	**76.4**
gsw_Latn	17.1	40.2	35.0	43.6	**45.3**	39.3	36.8	59.8	61.5	56.4	59.8	**65.8**
heb_Hebr	1.1	6.8	15.2	19.5	**34.4**	7.3	**76.3**	59.6	57.0	49.0	55.5	59.9
hin_Deva	1.4	15.0	24.9	35.3	**62.1**	27.2	73.8	**83.1**	74.2	70.4	74.6	83.0
hrv_Latn	4.9	45.9	55.6	66.4	**80.1**	58.6	79.6	86.7	83.4	82.5	84.4	**87.1**
hsb_Latn	3.1	14.3	17.6	21.1	**28.2**	19.3	21.5	47.0	47.2	44.1	**48.2**	45.5
hun_Latn	2.6	10.8	10.9	14.6	**27.5**	15.6	**76.1**	61.3	47.5	46.5	48.5	63.9
hye_Armn	1.2	7.8	26.7	30.9	**49.9**	18.9	64.6	71.8	65.2	59.6	66.3	**72.1**
ido_Latn	10.6	30.8	36.7	43.8	**48.5**	37.7	25.7	53.5	**61.0**	52.1	53.9	55.4
ile_Latn	16.3	42.3	40.2	50.5	**57.9**	44.5	34.6	71.3	**76.8**	66.4	66.1	69.8
ina_Latn	25.0	56.9	58.8	70.1	**78.8**	62.9	62.7	88.3	89.6	86.3	85.8	**90.1**
ind_Latn	2.7	33.6	42.7	59.8	**70.9**	52.2	84.3	**87.5**	79.7	78.0	80.6	86.7
isl_Latn	1.9	18.0	23.5	32.0	**56.9**	19.3	78.7	78.0	74.9	72.7	76.9	**81.5**
ita_Latn	13.1	43.1	43.3	56.5	**68.0**	50.7	81.3	82.8	78.4	73.9	75.7	**83.3**
jpn_Jpan	1.4	9.4	19.4	23.6	**43.1**	18.0	**74.4**	70.1	57.1	56.8	66.0	69.7
kab_Latn	2.3	6.0	4.0	3.4	6.0	**6.2**	3.7	13.1	12.1	14.4	**17.7**	14.2
kat_Geor	1.3	11.8	17.7	25.7	**40.6**	20.6	**61.1**	57.1	53.6	47.3	50.3	52.9
kaz_Cyrl	2.3	18.3	20.9	25.9	**39.8**	22.8	60.3	**64.7**	59.8	52.9	58.3	63.7
khm_Khmr	1.7	5.3	12.5	22.3	**34.1**	12.2	41.1	**55.5**	45.7	48.8	52.1	53.6
kor_Hang	1.3	5.3	11.7	16.5	**38.7**	9.6	**73.4**	69.5	50.9	55.6	59.2	69.6
kur_Latn	7.3	17.6	20.0	23.7	**30.2**	23.4	24.1	49.5	**52.0**	44.4	47.1	47.3
lat_Latn	11.8	21.5	19.1	23.6	**27.2**	23.6	33.6	39.6	**40.1**	35.2	36.5	37.7
lfn_Latn	15.4	33.3	35.9	40.4	**50.8**	38.2	32.5	58.8	**59.2**	52.0	56.8	57.5
lit_Latn	2.7	9.3	15.7	20.7	**30.9**	16.0	**73.4**	61.4	51.3	51.1	52.7	63.2
lvs_Latn	3.2	15.7	20.2	30.0	**39.3**	22.2	**73.4**	67.6	58.6	56.9	59.8	69.2
mal_Mlym	1.6	4.2	18.5	22.7	**46.0**	7.4	**80.1**	77.4	65.5	63.3	69.7	75.8
mar_Deva	1.0	9.3	13.8	23.2	**44.7**	14.5	63.5	**70.7**	60.5	58.4	61.4	69.5
mhr_Cyrl	1.5	5.4	6.4	9.6	**17.6**	8.5	6.5	25.8	30.6	27.1	**33.5**	30.0
mkd_Cyrl	1.1	20.0	28.6	45.4	**60.7**	30.7	70.5	75.2	69.2	67.6	69.5	**77.0**
mon_Cyrl	3.0	14.8	15.7	23.9	**43.2**	17.0	60.9	**75.9**	58.0	61.8	69.8	72.7
nds_Latn	7.0	29.1	32.6	38.1	**49.5**	30.1	28.8	70.3	67.6	68.6	70.9	**74.1**
nld_Latn	7.9	37.1	45.7	53.7	**69.7**	41.8	**90.3**	88.2	86.0	83.0	85.1	90.0
nno_Latn	6.1	42.1	53.2	62.9	**71.7**	49.1	70.7	85.3	**86.4**	82.5	84.1	85.1
nob_Latn	4.3	53.4	69.1	77.0	**85.2**	61.1	93.5	**94.3**	91.4	87.4	89.6	93.7
oci_Latn	7.7	20.6	16.4	23.8	**34.4**	22.6	22.9	41.7	41.4	41.7	42.5	**44.4**
pam_Latn	2.5	**6.8**	4.4	5.6	5.7	4.9	4.8	7.7	**12.6**	10.2	10.7	7.7
pes_Arab	1.0	16.8	24.4	45.6	**66.5**	34.4	83.3	83.2	74.0	75.6	78.9	**84.7**
pms_Latn	7.6	26.7	13.9	24.6	**32.2**	23.6	16.6	55.2	47.8	53.5	**56.4**	50.9
pol_Latn	2.7	17.7	26.3	29.0	**44.6**	24.3	**82.6**	75.8	68.4	63.8	67.6	77.5
por_Latn	12.7	44.2	47.6	57.7	**75.2**	57.6	**91.0**	85.9	84.8	82.4	85.4	89.6
ron_Latn	9.0	30.8	34.3	41.1	**58.3**	39.6	**86.0**	82.8	71.0	69.6	74.1	83.0
rus_Cyrl	1.3	21.0	37.3	47.0	**68.4**	40.4	**89.6**	85.2	80.0	76.3	77.6	86.9
slk_Latn	3.1	18.2	22.9	31.6	**43.5**	26.3	**73.2**	69.0	62.6	61.5	62.5	70.2
slv_Latn	6.4	28.2	29.6	39.6	**53.1**	34.0	**72.1**	70.8	67.4	63.9	67.9	71.9
spa_Latn	19.0	49.3	51.2	63.5	**73.9**	60.6	**85.5**	84.3	80.0	77.3	81.8	84.9
sqi_Latn	8.0	33.8	32.4	48.3	**70.0**	47.5	72.2	82.1	76.3	76.1	80.2	**84.0**
srp_Latn	3.2	32.8	47.5	59.4	**77.2**	52.2	78.1	86.2	82.7	82.6	84.3	**87.3**
swe_Latn	5.2	41.6	43.9	58.3	**68.6**	47.0	**90.4**	85.6	81.5	74.8	78.1	87.4
swh_Latn	9.7	20.5	19.0	31.0	**32.8**	26.9	30.3	**45.9**	41.8	40.8	39.7	43.3
tam_Taml	3.6	7.5	15.3	18.2	**35.5**	12.1	46.9	53.4	58.3	47.6	**59.3**	54.1
tat_Cyrl	1.3	14.1	17.6	25.6	**42.5**	23.1	10.3	63.6	60.0	57.9	61.3	**65.9**
tel_Telu	4.3	10.3	18.4	24.4	**40.2**	17.1	58.5	59.4	**62.4**	58.5	**62.4**	61.1
tgl_Latn	3.0	34.5	32.8	44.9	**63.0**	43.1	47.6	72.3	64.1	61.5	65.1	**74.0**
tha_Thai	2.0	7.3	26.8	36.5	**61.9**	14.2	56.8	68.6	72.6	64.8	68.6	**73.0**
tuk_Latn	6.9	19.7	14.8	24.1	**32.0**	19.7	16.3	58.1	55.2	51.2	**60.1**	58.1
tur_Latn	1.7	13.1	16.0	22.1	**34.2**	24.6	**77.9**	71.2	59.1	59.6	58.8	73.3
uig_Arab	1.1	5.6	8.6	11.6	**25.1**	7.7	38.8	**57.1**	47.2	44.4	50.5	51.8
ukr_Cyrl	1.6	17.7	26.0	35.6	**55.5**	27.4	77.1	77.1	68.7	63.9	66.6	**77.3**
urd_Arab	1.0	11.6	21.1	32.7	**57.5**	20.0	54.4	75.8	70.1	61.9	72.4	**78.9**
uzb_Cyrl	5.4	20.1	23.8	29.9	**36.0**	23.8	25.2	**61.9**	59.3	50.9	50.5	60.5
vie_Latn	1.1	6.7	13.5	21.4	**45.7**	17.4	**85.4**	83.0	60.5	67.6	72.5	83.2
war_Latn	3.7	11.9	11.3	15.6	**19.0**	15.5	8.0	25.7	**28.9**	28.8	26.3	26.9
wuu_Hani	1.2	9.3	23.4	33.5	**52.3**	23.8	56.1	76.7	67.6	65.2	68.5	**76.8**
xho_Latn	15.5	28.9	28.9	33.8	35.2	**40.8**	28.9	**52.8**	51.4	50.0	45.8	**52.8**
yid_Hebr	1.2	6.8	15.6	21.3	**36.2**	8.5	37.3	65.3	62.5	62.3	**68.5**	63.6
yue_Hani	1.2	10.9	21.8	34.7	**56.7**	22.4	50.3	**72.5**	63.7	63.3	69.3	71.0
zsm_Latn	3.5	36.4	46.5	58.2	**75.2**	54.1	81.4	90.2	81.8	78.3	82.0	**91.0**

Table 12: Top-10 accuracy of baselines and models initialized with OFA on **SR-T**.

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
ace_Latn	6.1	17.7	26.7	38.8	**50.4**	33.6	13.4	**66.6**	62.2	62.4	63.9	65.7
ach_Latn	4.9	9.5	8.7	18.1	**28.3**	18.6	10.9	33.4	**46.7**	43.0	38.9	41.9
acr_Latn	4.9	5.3	14.2	24.1	**36.2**	15.4	8.8	44.3	**53.0**	46.6	52.5	45.6
afr_Latn	8.6	41.7	41.6	43.3	**44.4**	37.4	**65.7**	61.8	59.3	62.8	65.5	58.7
agw_Latn	7.3	8.5	13.5	34.6	**38.3**	30.1	13.9	48.7	59.8	**60.3**	49.7	58.3
ahk_Latn	4.9	4.9	**6.7**	4.9	4.9	4.9	9.3	**14.2**	6.1	4.9	4.9	9.1
aka_Latn	4.9	18.9	20.2	27.2	32.8	**35.7**	9.1	39.2	**57.4**	53.9	51.3	43.4
aln_Latn	12.3	27.3	17.1	45.9	**52.2**	30.5	53.8	**57.0**	51.6	55.7	56.2	54.0
als_Latn	12.7	28.8	15.5	43.4	**51.4**	31.2	**57.8**	53.7	53.9	55.9	55.8	54.1
alt_Cyrl	4.9	14.6	13.6	32.0	**32.9**	15.6	25.4	46.5	46.0	46.4	**57.6**	49.9
alz_Latn	4.9	9.4	9.0	21.0	**23.4**	17.1	11.8	37.6	34.8	32.8	31.4	**41.4**
amh_Ethi	4.9	7.1	4.9	**10.5**	4.9	5.4	9.3	4.9	**11.6**	6.0	4.9	10.3
aoj_Latn	4.9	6.2	13.9	21.5	**32.1**	24.8	12.2	35.8	44.8	49.2	**53.1**	44.4
arn_Latn	4.9	15.6	23.1	22.9	24.9	**29.6**	9.1	36.5	41.7	39.2	41.3	**52.0**
ary_Arab	4.9	6.6	5.0	**19.5**	17.7	10.5	14.5	26.9	30.3	34.6	**36.5**	34.5
arz_Arab	4.9	10.6	4.9	25.5	**35.5**	15.4	21.9	38.3	36.3	41.0	43.4	**47.4**
asm_Beng	4.9	14.0	11.4	36.3	**44.6**	29.6	47.3	55.2	51.1	53.4	**64.8**	61.3
ayr_Latn	4.9	4.9	6.0	32.3	**48.8**	16.4	7.7	48.4	62.7	61.8	61.1	**67.3**
azb_Arab	4.9	29.3	26.3	31.9	**42.0**	29.6	16.1	65.3	**67.9**	56.8	67.7	61.3
aze_Latn	4.7	17.6	37.0	42.1	**54.4**	40.5	64.6	68.2	68.8	66.6	72.5	**73.6**
bak_Cyrl	4.9	6.1	9.2	29.2	**42.4**	20.3	22.6	61.3	57.7	61.8	**71.8**	68.3
bam_Latn	4.9	20.7	13.2	**31.2**	27.9	20.9	7.7	44.8	50.9	48.4	44.8	**58.7**
ban_Latn	4.9	11.3	11.4	25.5	**32.4**	13.1	18.9	**51.3**	38.1	49.8	43.2	49.9
bar_Latn	4.9	15.6	17.8	26.2	**27.8**	12.8	34.1	50.4	48.6	40.1	50.9	**57.6**
bba_Latn	4.9	13.0	5.0	**30.7**	26.7	24.0	8.6	49.1	46.8	38.5	**50.3**	44.7
bci_Latn	4.9	11.5	13.6	10.1	**19.9**	6.7	8.4	29.0	32.1	24.3	29.0	**36.6**
bcl_Latn	4.9	23.0	21.8	37.7	**50.6**	38.8	31.5	54.6	**67.8**	59.8	61.3	62.0
bel_Cyrl	4.9	25.5	20.6	39.0	**45.4**	25.5	62.0	59.5	55.2	53.0	60.8	**64.7**
bem_Latn	4.9	11.5	14.3	34.6	**43.1**	27.2	15.8	41.5	44.6	42.1	**57.0**	56.4
ben_Beng	4.9	8.3	11.6	29.3	**45.5**	17.8	63.4	59.5	61.0	55.7	62.0	**71.6**
bhw_Latn	7.3	11.7	19.6	26.2	**30.7**	18.9	14.9	36.4	**54.2**	53.4	51.5	45.3
bim_Latn	4.9	12.2	15.0	19.0	**21.3**	15.9	9.1	53.2	53.5	47.3	58.5	**65.6**
bis_Latn	7.2	19.7	19.6	53.8	**64.1**	36.4	14.8	70.3	**72.9**	65.6	71.2	71.6
bqc_Latn	4.9	11.4	4.9	**17.0**	12.4	11.7	9.1	42.3	29.7	30.7	36.7	**50.7**
bre_Latn	4.9	**12.1**	11.2	7.1	4.9	4.9	30.3	37.0	35.2	37.0	28.6	**39.5**
btx_Latn	4.9	21.0	32.6	33.7	**44.6**	24.8	24.6	60.0	55.4	55.1	57.5	**62.9**
bul_Cyrl	4.9	20.9	42.1	44.4	**58.2**	36.3	**69.2**	68.2	62.9	60.2	63.9	67.6
bum_Latn	4.9	12.6	18.4	19.6	**23.2**	15.8	14.0	39.5	**46.3**	40.8	38.3	42.1
bzj_Latn	4.9	32.8	35.9	44.3	**58.4**	30.3	13.3	65.0	64.5	59.5	66.7	**68.7**
cab_Latn	4.9	10.3	4.9	**21.6**	15.8	10.9	8.0	22.7	24.7	25.5	**28.4**	27.0
cac_Latn	4.9	8.6	15.9	34.4	**35.0**	15.3	10.5	43.6	48.8	58.4	**60.0**	55.6
cak_Latn	4.9	13.8	7.1	38.2	**39.6**	11.7	10.7	54.5	51.2	54.3	51.0	**61.1**
caq_Latn	4.9	8.5	21.9	32.4	**39.6**	17.0	8.3	43.2	49.1	40.6	**52.0**	51.7
cat_Latn	16.8	14.8	34.6	41.4	**55.3**	28.5	**65.6**	58.2	60.5	61.0	60.7	62.3
cbk_Latn	15.5	26.6	42.5	54.9	**64.6**	37.0	51.8	65.9	64.5	55.6	61.9	**69.2**
cce_Latn	4.9	22.8	14.5	27.6	**34.3**	22.2	9.7	51.1	49.1	44.9	**52.3**	49.3
ceb_Latn	4.9	23.7	26.7	35.9	**50.9**	31.7	26.2	57.9	53.1	51.6	51.3	**66.8**
ces_Latn	4.9	12.3	26.1	30.4	**38.4**	20.7	**67.7**	61.8	56.3	49.1	62.4	63.8
cfm_Latn	4.9	13.8	**21.4**	21.3	19.4	6.1	9.1	55.1	60.6	64.7	**67.1**	65.4
che_Cyrl	4.9	5.0	4.9	**14.8**	6.0	4.9	11.4	14.6	17.7	21.4	17.2	**25.2**
chv_Cyrl	4.9	13.0	14.6	28.9	**39.8**	25.5	13.4	51.6	65.2	51.5	62.3	**67.2**
cmn_Hani	4.9	32.2	23.5	54.9	**65.1**	35.1	**71.9**	65.4	68.3	64.2	68.6	68.9
cnh_Latn	4.9	10.0	16.8	16.6	**20.1**	6.9	9.7	59.7	58.7	60.4	**65.2**	62.9
crh_Cyrl	4.9	5.1	17.1	36.8	**45.9**	42.0	14.7	65.9	63.7	60.6	65.9	**71.1**
crs_Latn	4.9	33.2	30.6	53.1	**66.4**	43.9	16.5	67.3	**67.8**	65.5	65.1	67.7
csy_Latn	4.9	8.4	15.9	**24.9**	24.3	21.2	11.8	53.4	51.0	60.6	60.1	**61.7**
ctd_Latn	4.9	4.9	21.2	**26.6**	22.5	21.2	9.4	52.4	59.8	59.0	50.8	**65.7**
ctu_Latn	4.9	6.8	19.4	**26.6**	25.1	19.7	13.0	53.5	53.1	60.0	58.4	**63.3**
cuk_Latn	4.9	15.4	7.4	22.8	**24.9**	7.9	14.2	43.6	37.9	38.3	35.7	**54.3**
cym_Latn	4.9	11.1	13.6	22.4	**27.5**	19.6	**52.9**	44.5	37.0	44.2	39.0	51.0
dan_Latn	4.9	26.1	43.3	36.3	**51.0**	33.2	62.1	55.4	**62.9**	57.3	51.9	58.9
deu_Latn	4.9	22.3	29.4	28.8	**29.6**	25.5	**53.9**	48.7	50.3	42.7	49.4	50.3
djk_Latn	4.9	25.6	19.5	34.2	**53.1**	23.7	14.7	49.1	**57.7**	45.8	56.2	56.0
dln_Latn	4.9	11.1	**25.7**	18.7	20.8	6.2	11.0	38.5	**64.1**	60.2	45.7	57.4
dtp_Latn	4.9	12.3	18.3	27.0	21.8	**30.2**	10.8	54.3	55.9	49.4	**59.0**	56.2
dyu_Latn	4.9	20.0	6.1	19.9	**24.9**	19.5	5.1	52.1	**59.9**	59.0	55.5	56.0
dzo_Tibt	4.9	7.9	15.1	32.6	**38.3**	14.5	4.9	41.2	64.7	55.5	**69.2**	61.9
efi_Latn	4.9	11.2	16.3	34.6	**52.7**	38.0	13.7	41.3	47.6	56.8	52.8	**65.9**
ell_Grek	4.9	14.7	15.0	33.3	**34.9**	21.3	46.6	58.5	51.4	49.3	62.6	**66.1**
eng_Latn	72.8	**76.7**	74.7	72.7	76.1	73.4	74.6	74.8	73.5	74.4	75.9	**78.9**
enm_Latn	53.7	63.5	62.9	69.7	**74.3**	73.1	57.5	62.6	72.2	71.8	**75.3**	69.7
epo_Latn	4.9	21.6	20.6	34.4	**50.6**	19.4	**63.0**	60.4	51.9	59.6	55.0	59.8
est_Latn	4.9	10.7	10.2	12.3	**24.1**	12.1	**67.1**	58.9	54.3	51.7	56.8	64.5
eus_Latn	6.9	11.2	11.4	9.5	8.9	**13.3**	22.7	17.2	23.0	17.6	14.9	**25.1**
ewe_Latn	4.9	17.6	30.5	25.4	**32.6**	30.2	7.3	37.4	50.0	52.2	47.5	**53.2**
fao_Latn	4.9	20.0	19.6	28.9	**44.0**	27.2	33.6	61.1	**65.3**	56.4	57.7	63.8
fas_Arab	4.9	31.0	45.1	54.3	**60.4**	55.0	68.7	**75.4**	73.8	71.7	74.0	70.8
fij_Latn	5.0	21.0	6.5	35.3	**37.3**	29.1	13.0	45.4	44.5	50.6	**57.5**	49.5
fil_Latn	4.8	20.2	30.9	45.5	**47.1**	37.7	53.7	61.2	61.5	51.8	64.0	**67.3**
fin_Latn	4.9	15.7	19.2	20.3	**24.3**	11.6	**60.0**	54.9	46.1	38.8	41.8	**60.0**
fon_Latn	4.9	12.7	8.1	**29.4**	25.6	17.6	6.2	42.1	51.6	43.7	53.2	**57.2**
fra_Latn	19.6	30.9	47.9	53.7	**59.2**	35.2	**74.8**	70.4	64.0	66.8	68.4	74.3
fry_Latn	4.5	16.0	15.1	24.0	**32.0**	14.7	40.1	45.5	47.8	40.1	43.7	**50.1**
gaa_Latn	4.9	17.5	7.5	**28.4**	27.7	24.2	5.0	38.2	38.7	49.4	48.0	**53.6**
gil_Latn	4.9	9.9	7.9	**26.8**	25.0	19.1	8.4	42.6	37.5	45.2	**48.8**	46.8
giz_Latn	4.9	17.5	14.0	31.0	**41.5**	24.2	9.0	52.1	46.9	41.2	44.2	**52.6**
gkn_Latn	4.9	11.0	6.1	23.3	**25.5**	16.8	9.7	36.3	41.0	40.4	**49.8**	48.7
gkp_Latn	4.9	4.9	4.9	**25.9**	16.1	8.2	6.0	23.4	**45.4**	41.2	41.6	39.1
gla_Latn	4.9	8.8	10.6	19.7	**29.9**	18.3	36.2	53.3	37.7	40.7	39.5	**55.4**

Table 13: F1 scores of baselines and models initialized with OFA on **Taxi1500** (Part I).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
gle_Latn	4.9	12.2	5.2	18.0	20.6	**24.1**	40.1	38.6	37.8	34.7	37.5	**54.6**
glv_Latn	4.9	23.8	12.1	8.3	**32.9**	15.4	11.7	43.8	**45.0**	41.8	43.0	42.8
gom_Latn	4.9	11.4	**14.9**	8.0	10.3	7.6	13.0	31.7	**48.3**	37.3	47.2	44.8
gor_Latn	4.9	7.3	18.2	19.4	28.0	**29.0**	18.5	47.5	41.9	49.3	45.6	**54.0**
guc_Latn	4.9	20.3	19.2	25.0	**36.5**	19.8	8.7	35.2	26.9	32.9	33.9	**41.7**
gug_Latn	4.9	15.4	16.7	**29.3**	26.3	24.1	15.3	39.0	**51.6**	44.8	50.0	44.7
guj_Gujr	4.9	8.4	25.3	43.3	**51.9**	30.2	62.9	72.4	68.4	69.6	68.1	**73.7**
gur_Latn	4.9	10.5	11.9	21.1	**23.9**	8.4	7.4	44.9	49.5	41.5	45.8	**53.9**
guw_Latn	4.9	9.0	11.7	**33.1**	28.6	27.5	12.0	48.7	56.5	54.9	**64.9**	60.5
gya_Latn	4.9	7.6	4.9	31.1	**45.1**	32.8	5.0	42.0	46.5	41.4	46.4	**47.8**
gym_Latn	4.9	6.8	4.9	23.6	**29.1**	11.7	10.9	47.1	47.9	47.2	**57.6**	52.0
hat_Latn	4.9	25.0	34.1	47.3	**63.2**	30.7	14.5	64.4	68.4	58.4	71.9	**72.4**
hau_Latn	5.9	13.9	13.0	26.3	**45.9**	17.3	44.3	54.1	48.9	47.7	53.8	**65.3**
haw_Latn	4.9	10.6	14.7	24.9	**28.0**	9.1	9.0	38.4	41.0	42.8	43.1	**52.7**
heb_Hebr	7.0	10.7	9.7	8.1	13.0	**19.4**	17.9	18.1	24.5	**27.9**	21.0	22.0
hif_Latn	4.9	4.9	8.1	13.2	**18.6**	6.0	19.2	45.1	39.6	42.2	**53.7**	51.1
hil_Latn	6.9	26.8	28.8	45.5	**67.6**	38.4	33.8	66.6	66.7	66.9	65.8	**78.4**
hin_Deva	4.9	17.3	21.4	40.5	**66.5**	41.5	66.7	66.4	66.0	61.0	68.3	**68.9**
hmo_Latn	4.9	15.6	11.3	30.9	**46.2**	40.7	15.3	55.7	58.5	63.2	62.6	**64.0**
hne_Deva	4.9	23.3	18.6	42.1	**56.6**	39.1	41.0	66.7	67.6	67.6	69.0	**73.0**
hnj_Latn	4.9	6.1	19.5	42.4	**51.3**	38.1	15.2	58.3	65.7	**69.6**	65.3	65.5
hra_Latn	4.9	4.9	14.5	12.6	**22.2**	8.8	13.3	49.2	**59.7**	47.6	55.1	58.0
hrv_Latn	8.2	34.5	37.6	44.1	**60.8**	37.7	61.0	64.0	55.1	60.8	**71.1**	**71.1**
hui_Latn	4.9	12.9	5.0	30.3	**31.0**	22.8	9.3	39.5	45.0	**54.8**	51.5	45.5
hun_Latn	4.9	9.0	13.4	**21.9**	16.3	16.0	**75.5**	61.2	45.8	50.0	56.9	60.8
hus_Latn	4.9	5.2	4.9	**30.7**	14.9	12.7	10.7	36.6	38.5	41.8	36.0	**42.3**
hye_Armn	4.9	10.0	39.5	50.3	**68.3**	34.5	72.1	**72.2**	64.2	59.1	70.0	69.2
iba_Latn	4.9	17.6	36.5	43.4	**56.6**	28.5	40.7	55.4	63.5	62.2	64.1	**64.9**
ibo_Latn	4.9	10.4	14.0	34.0	**41.1**	28.5	8.0	42.8	54.6	53.9	**65.9**	63.6
ifa_Latn	4.9	20.9	18.5	19.4	**25.4**	20.4	12.5	48.4	57.2	50.7	54.9	**58.6**
ifb_Latn	4.9	**24.3**	19.4	18.4	19.4	23.9	8.9	36.4	48.8	50.8	54.2	**54.9**
ikk_Latn	4.9	6.7	7.3	23.8	**31.9**	22.6	9.5	52.9	47.9	58.3	**63.6**	52.3
ilo_Latn	4.9	15.6	23.6	39.0	**39.4**	22.9	20.0	57.0	61.8	58.5	58.4	**69.0**
ind_Latn	6.1	46.6	45.1	65.5	**66.6**	47.7	75.6	72.5	73.1	69.6	74.4	**75.9**
isl_Latn	4.9	23.9	18.1	22.7	**29.4**	24.2	60.3	58.3	53.5	48.9	55.8	**66.6**
ita_Latn	9.6	31.6	38.4	55.1	**56.8**	38.6	**71.2**	65.0	63.3	62.6	68.6	67.9
ium_Latn	4.9	13.8	22.4	41.8	**60.3**	17.5	7.4	59.2	62.1	**67.0**	62.9	61.7
ixl_Latn	4.9	16.7	4.9	**19.0**	15.1	4.9	12.6	25.2	42.2	**42.7**	39.4	35.6
izz_Latn	4.9	9.2	4.9	21.1	11.5	**22.4**	12.3	41.6	47.8	46.2	52.4	**61.1**
jam_Latn	4.9	23.0	31.9	49.1	**58.0**	36.8	18.0	68.2	57.7	59.7	66.2	**70.5**
jav_Latn	4.9	12.1	23.0	**30.1**	28.7	15.1	48.7	52.0	45.6	48.5	51.0	**57.6**
jpn_Jpan	4.9	9.1	11.3	47.5	**54.5**	24.7	**71.0**	60.6	69.9	63.0	64.1	66.4
kaa_Cyrl	4.9	5.0	4.9	18.7	**30.4**	13.9	16.7	54.8	58.9	46.9	64.1	**66.4**
kab_Latn	4.9	10.3	10.8	**13.8**	8.1	6.3	9.1	23.0	28.3	26.4	**30.0**	24.0
kac_Latn	4.9	16.9	7.1	16.9	**39.4**	8.3	11.3	47.8	43.4	50.7	45.1	**51.3**
kal_Latn	4.9	5.8	13.5	**15.1**	13.3	12.4	10.3	29.4	34.6	29.4	**40.8**	39.3
kan_Knda	4.9	5.3	14.7	29.8	**42.4**	32.3	69.9	64.2	60.8	50.7	66.1	**76.9**
kat_Geor	4.9	26.0	38.4	44.3	**55.7**	35.9	66.6	55.6	54.2	55.8	65.2	**68.1**
kaz_Cyrl	4.9	5.0	10.4	30.3	**38.5**	25.1	63.4	57.3	**66.1**	61.5	63.4	62.9
kbp_Latn	4.9	9.4	16.9	32.2	**34.1**	15.1	4.9	**43.8**	39.6	41.1	38.8	41.9
kek_Latn	4.9	4.9	15.6	**32.8**	28.5	14.7	7.7	37.4	43.0	36.7	43.2	**51.6**
khm_Khmr	4.9	4.9	23.0	45.1	**64.3**	25.0	63.6	**71.0**	65.3	64.9	68.4	68.7
kia_Latn	4.9	14.3	7.1	26.1	**28.4**	15.9	13.4	57.7	56.3	53.7	53.8	**60.1**
kik_Latn	4.9	8.8	14.7	25.8	**29.9**	21.1	6.4	36.3	49.2	48.7	44.2	**49.4**
kin_Latn	4.9	14.8	15.4	50.7	**61.1**	32.1	17.0	58.3	53.6	49.6	60.8	**62.1**
kir_Cyrl	4.9	5.3	15.9	37.4	**47.6**	33.4	61.4	67.1	63.7	65.5	63.6	**68.0**
kjb_Latn	4.9	7.0	8.3	34.1	**38.8**	19.8	8.8	48.1	54.2	56.6	**64.4**	63.9
kjh_Cyrl	4.9	9.9	15.4	26.6	**32.8**	23.7	21.6	50.2	51.1	46.7	**61.5**	55.8
kmm_Latn	4.9	8.5	13.2	17.9	**31.4**	7.7	9.1	42.5	50.2	51.0	56.7	**59.7**
kmr_Cyrl	4.9	11.4	14.7	**26.3**	19.2	18.8	9.5	41.9	38.3	43.5	**50.9**	46.6
knv_Latn	4.9	14.4	12.9	**25.0**	24.1	17.3	8.6	40.0	41.8	45.4	51.1	**55.3**
kor_Hang	4.9	13.5	15.3	39.9	**55.1**	33.8	**72.7**	66.9	65.4	54.6	62.7	71.4
kpg_Latn	6.1	10.7	33.0	36.9	**55.1**	35.0	10.6	62.0	60.3	70.6	66.8	**71.1**
krc_Cyrl	4.9	6.7	12.6	37.0	**44.6**	33.3	24.8	51.6	61.3	53.0	**66.3**	65.8
kri_Latn	6.1	19.7	25.4	53.5	**69.7**	34.4	10.8	57.5	58.7	57.3	61.5	**67.8**
ksd_Latn	4.9	12.8	12.2	**44.0**	21.6	10.4	12.7	**61.5**	53.4	50.0	54.6	56.9
kss_Latn	4.9	4.9	6.1	14.9	**17.8**	4.3	4.9	11.6	27.0	**29.5**	29.4	25.4
ksw_Mymr	4.9	7.2	6.0	28.4	**58.4**	18.1	4.9	**57.4**	56.3	54.7	56.4	55.6
kua_Latn	4.9	21.1	17.6	**32.4**	23.4	24.9	17.5	46.8	**51.2**	41.4	50.7	48.1
lam_Latn	4.9	7.3	11.1	**27.7**	25.3	18.8	12.8	36.8	43.1	35.8	45.1	**51.7**
lao_Laoo	4.9	6.3	22.6	50.4	**69.2**	41.2	73.5	76.8	72.7	66.4	74.8	**78.4**
lat_Latn	18.2	17.4	26.5	30.9	**50.7**	32.2	65.9	54.6	55.7	50.6	58.5	**67.8**
lav_Latn	4.9	21.0	7.3	30.2	**38.0**	20.6	**69.9**	62.6	49.2	52.7	55.7	68.9
ldi_Latn	4.9	11.3	7.3	14.6	**22.5**	6.5	13.7	26.2	26.2	22.4	30.2	**35.8**
leh_Latn	4.9	17.8	15.0	25.3	**40.8**	21.6	14.3	44.3	52.9	48.9	52.7	**59.0**
lhu_Latn	4.9	11.3	**14.0**	13.2	13.4	4.9	6.3	25.3	31.4	28.9	**36.6**	28.3
lin_Latn	4.9	9.3	17.2	29.0	**42.1**	30.8	12.7	43.5	59.2	**60.9**	54.4	55.1
lit_Latn	4.9	17.1	19.0	**31.6**	30.7	16.5	**65.1**	54.6	40.7	44.3	52.5	60.9
loz_Latn	4.9	13.3	11.5	21.5	**25.8**	18.1	13.8	47.4	56.1	52.8	53.2	**58.9**
ltz_Latn	4.9	15.4	27.8	25.5	**34.3**	25.0	27.2	50.7	53.2	54.3	52.8	**58.6**
lug_Latn	4.9	16.3	21.2	28.9	**40.9**	26.2	13.7	45.9	51.7	44.0	59.6	**61.8**
luo_Latn	5.1	16.0	11.7	34.0	33.5	**36.2**	10.6	37.0	44.4	**51.0**	46.2	44.7
lus_Latn	4.9	14.1	22.9	27.3	**31.5**	14.9	9.1	39.3	53.1	**57.8**	55.8	51.6
lzh_Hani	4.9	20.7	38.0	49.6	**58.0**	35.3	62.9	66.4	67.8	**68.5**	61.5	64.6
mad_Latn	6.1	16.4	10.7	30.3	**51.9**	19.3	24.6	61.4	60.8	60.5	63.3	**66.2**
mah_Latn	4.9	16.4	8.1	27.9	**30.4**	13.2	10.6	33.3	38.9	47.0	46.9	**50.6**
mai_Deva	4.9	13.9	18.0	42.8	**61.4**	41.1	30.5	64.9	67.3	62.4	69.4	**69.5**
mal_Mlym	4.9	5.7	**9.5**	7.0	6.2	4.9	10.5	4.8	**11.5**	7.7	5.0	4.8
mam_Latn	4.9	13.1	4.9	**26.7**	12.7	6.9	9.2	32.3	36.1	36.5	31.8	**37.6**

Table 14: F1 scores of baselines and models initialized with OFA on **Taxi1500** (Part II).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
mar_Deva	4.9	10.9	14.1	38.9	**41.4**	33.7	60.7	56.8	68.8	61.4	67.6	**74.4**
mau_Latn	**4.9**	**4.9**	**4.9**	**4.9**	**4.9**	**4.9**	6.5	7.1	**8.2**	4.9	4.9	7.1
mbb_Latn	4.9	19.3	19.1	33.3	34.8	**39.0**	8.7	57.4	**61.5**	55.8	58.7	58.8
mck_Latn	4.9	20.6	21.1	25.1	**36.8**	18.8	18.2	44.4	49.2	43.1	51.0	**51.1**
mcn_Latn	4.9	9.8	4.9	21.3	**22.1**	15.0	10.7	**49.1**	47.1	44.5	41.9	47.5
mco_Latn	4.9	4.9	4.9	7.3	**17.6**	10.5	8.2	14.4	**35.0**	21.5	23.3	18.8
mdy_Ethi	4.9	4.9	13.4	27.0	**37.3**	7.0	4.9	48.7	56.7	54.9	55.8	**58.8**
meu_Latn	4.9	18.8	8.2	**35.3**	35.0	29.2	15.6	53.0	**56.4**	51.3	53.2	54.9
mfe_Latn	9.1	30.0	31.5	60.6	**70.3**	39.8	15.6	67.9	65.8	61.7	**73.2**	71.9
mgh_Latn	4.9	9.9	11.7	13.2	**14.2**	13.9	9.4	35.3	36.4	**38.2**	36.6	37.0
mgr_Latn	4.9	18.8	16.6	23.9	**30.5**	27.0	15.8	41.7	39.3	42.3	49.7	**58.3**
mhr_Cyrl	4.9	16.3	9.8	23.4	**24.5**	21.7	10.5	41.7	48.2	43.5	**54.8**	52.4
min_Latn	6.1	13.9	11.8	19.8	**31.9**	15.2	23.9	**62.6**	59.0	57.1	49.3	55.6
miq_Latn	4.9	19.2	7.2	21.3	**33.5**	6.1	5.2	33.5	54.4	55.5	**57.9**	53.8
mkd_Cyrl	4.9	28.0	53.2	56.9	**69.5**	39.9	**74.4**	67.5	68.8	64.6	71.6	70.4
mlg_Latn	4.9	14.2	23.1	**31.9**	31.2	22.7	38.3	**56.3**	48.4	39.1	52.3	55.8
mlt_Latn	4.9	16.9	30.5	36.9	**39.3**	22.3	14.7	44.2	48.5	48.0	55.8	**59.7**
mos_Latn	4.9	8.6	4.9	25.1	**32.7**	12.2	10.7	38.1	45.3	**49.5**	46.6	47.3
mps_Latn	6.1	9.8	17.1	17.4	**33.1**	24.0	11.6	51.9	51.1	57.4	56.7	**57.9**
mri_Latn	4.9	20.2	20.3	**31.4**	28.5	22.1	8.5	44.4	47.8	46.0	**58.9**	53.3
mrw_Latn	6.4	7.8	6.2	**34.2**	31.1	20.7	16.7	**59.5**	51.9	57.2	49.7	55.1
msa_Latn	4.9	19.9	19.9	**35.0**	32.9	19.8	43.5	**54.4**	38.4	39.6	47.7	52.3
mwm_Latn	4.9	5.0	12.3	**27.6**	24.7	17.9	6.7	47.9	48.4	**60.1**	52.1	56.5
mxv_Latn	4.9	**9.3**	4.9	4.8	4.9	5.9	11.7	17.2	**30.1**	17.2	21.3	26.4
mya_Mymr	4.9	6.9	8.3	15.7	**42.8**	4.9	50.0	65.0	55.6	53.6	66.0	**70.7**
myv_Cyrl	4.9	8.2	12.2	30.7	**32.4**	24.0	14.2	49.1	40.1	43.6	41.2	**53.3**
mzh_Latn	4.9	7.1	6.2	30.5	**37.9**	27.3	12.6	43.4	**46.7**	42.1	42.1	42.0
nan_Latn	4.9	4.9	4.9	**18.1**	16.5	8.2	6.4	29.9	31.5	20.2	35.1	**42.6**
naq_Latn	4.9	6.7	4.9	17.3	**21.6**	11.2	7.7	35.7	39.7	40.5	37.0	**49.2**
nav_Latn	4.9	10.4	9.6	**14.2**	9.8	6.6	6.9	15.6	22.2	24.9	**29.5**	23.4
nbl_Latn	4.9	16.2	18.5	32.4	**38.6**	29.9	20.2	40.0	52.3	47.0	**56.7**	49.4
nch_Latn	4.9	9.0	12.9	27.4	**33.6**	17.0	6.4	40.1	39.7	41.2	43.4	**48.9**
ncj_Latn	4.9	7.6	22.0	25.7	**29.4**	11.0	7.4	46.5	47.3	37.7	42.7	**51.5**
ndc_Latn	4.9	21.0	18.8	29.3	**32.6**	23.4	18.5	44.2	**47.8**	45.4	47.6	47.0
nde_Latn	4.9	16.2	18.5	32.4	**38.6**	29.9	20.2	40.0	52.3	47.0	**56.7**	49.4
ndo_Latn	4.9	21.4	23.4	**31.6**	28.1	24.9	16.1	47.0	48.8	50.1	**51.7**	**51.7**
nds_Latn	4.9	26.4	13.6	24.9	**30.4**	18.1	15.4	34.6	**52.0**	41.8	41.0	45.0
nep_Deva	4.9	16.2	10.8	42.6	**63.6**	41.7	65.9	66.8	67.0	60.9	62.4	**77.5**
ngu_Latn	4.9	6.5	17.8	25.9	**27.0**	12.2	10.9	45.5	46.1	48.6	46.5	**49.6**
nld_Latn	5.9	30.9	35.0	39.8	**50.3**	38.7	66.4	**67.9**	62.8	63.1	63.9	62.5
nmf_Latn	4.9	4.9	7.9	16.7	**18.0**	7.2	11.9	34.5	38.7	45.9	**47.7**	45.8
nnb_Latn	4.9	7.6	21.8	28.5	**35.9**	21.4	10.9	36.7	51.8	47.7	49.6	**55.1**
nno_Latn	4.9	36.0	43.7	44.2	**63.5**	37.7	59.4	61.2	59.6	54.3	**65.6**	63.6
nob_Latn	4.9	35.5	43.9	56.7	**56.9**	36.5	67.9	64.2	63.7	53.8	62.2	**68.0**
nor_Latn	4.9	33.0	47.0	50.4	**53.8**	39.2	**67.1**	62.2	64.2	58.0	62.6	**67.1**
npi_Deva	4.9	20.9	19.6	48.9	**66.4**	45.9	65.2	**72.3**	66.6	66.9	71.2	68.7
nse_Latn	4.9	14.4	16.1	25.4	**37.3**	24.0	15.7	50.3	47.4	44.3	**54.3**	52.3
nso_Latn	4.9	7.3	5.0	31.1	**44.9**	21.5	15.8	53.8	**61.7**	55.1	58.3	61.3
nya_Latn	4.9	25.3	18.1	32.1	**54.7**	37.0	16.0	48.3	59.5	60.9	60.5	**64.1**
nyn_Latn	4.9	17.7	15.9	29.4	**43.7**	23.3	15.6	40.9	45.9	43.5	42.3	**53.2**
nyy_Latn	4.9	8.2	4.9	**25.4**	24.3	13.1	8.1	32.9	28.9	23.8	32.8	**37.8**
nzi_Latn	4.9	14.6	15.9	18.7	**20.5**	18.8	6.5	33.0	39.2	39.9	**41.7**	40.6
ori_Orya	4.9	10.7	8.3	45.5	**63.4**	39.8	63.0	**72.1**	66.9	64.9	69.5	71.6
ory_Orya	4.9	9.1	10.1	48.8	**64.1**	36.5	61.8	69.8	68.7	63.8	70.9	**72.0**
oss_Cyrl	4.9	11.9	14.4	41.0	**42.9**	33.6	9.4	53.0	61.0	**61.5**	59.3	61.3
ote_Latn	4.9	4.9	4.9	19.0	**21.2**	17.1	5.5	39.0	38.9	35.8	29.4	**42.6**
pag_Latn	4.9	14.9	25.1	**34.7**	30.2	24.4	22.0	51.1	56.4	58.3	55.2	**59.3**
pam_Latn	4.9	16.2	17.4	**25.4**	20.3	13.5	25.8	38.5	**46.6**	37.7	46.3	45.7
pan_Guru	4.9	13.1	22.3	43.7	**50.6**	26.8	64.8	66.4	64.0	65.3	64.8	**68.2**
pap_Latn	12.3	36.6	38.0	64.4	**69.8**	55.3	36.3	68.7	**73.4**	59.9	66.9	69.8
pau_Latn	4.9	12.7	19.6	17.3	**31.1**	16.8	15.6	38.0	**46.6**	39.5	40.6	36.6
pcm_Latn	26.2	53.0	38.8	62.7	**65.5**	56.4	31.8	64.5	64.3	57.4	63.5	**65.5**
pdt_Latn	4.9	35.2	23.7	33.3	**58.8**	27.8	18.1	58.1	59.9	58.1	**67.2**	59.5
pes_Arab	4.9	28.2	46.3	52.7	**60.6**	47.9	72.6	73.2	72.3	70.6	**73.4**	71.6
pis_Latn	8.0	28.1	26.1	55.6	**66.4**	44.4	12.5	67.7	66.2	61.2	64.1	**69.7**
pls_Latn	4.9	13.9	20.3	**41.6**	36.5	26.8	16.2	48.9	55.4	50.0	55.8	**61.2**
plt_Latn	4.9	11.9	21.3	**39.6**	33.3	16.9	32.3	54.0	53.4	46.9	54.3	**54.8**
poh_Latn	4.9	20.4	20.9	24.7	**26.3**	15.5	12.7	50.4	**56.9**	42.1	45.2	51.4
pol_Latn	4.9	21.7	22.1	24.9	**36.1**	24.2	**68.8**	68.1	51.5	64.1	67.1	68.5
pon_Latn	4.9	23.3	27.1	36.9	**44.9**	28.8	7.9	50.2	47.4	54.1	56.2	**57.4**
por_Latn	17.6	25.4	38.2	51.0	**59.8**	34.1	**73.4**	69.6	67.3	61.4	68.9	67.7
prk_Latn	4.9	7.3	14.4	34.5	**49.1**	29.8	11.2	58.4	51.0	62.4	59.6	**66.2**
prs_Arab	4.9	33.3	44.4	52.4	**58.4**	42.3	**74.4**	72.4	71.4	72.4	72.5	73.9
pxm_Latn	4.9	16.3	10.8	16.5	15.6	**17.4**	11.5	33.2	44.0	45.5	51.4	**52.3**
qub_Latn	4.9	4.9	13.6	27.8	**52.7**	21.8	10.1	63.3	64.0	56.3	61.3	**64.8**
quc_Latn	4.9	17.1	11.6	**30.0**	27.3	24.6	15.3	42.0	54.6	46.5	49.9	**56.3**
qug_Latn	4.9	7.5	10.5	28.6	**58.6**	27.3	12.0	66.6	68.5	65.9	61.5	**71.0**
quh_Latn	4.9	11.3	11.2	41.4	**64.3**	23.6	12.1	69.3	72.0	65.7	71.0	**72.1**
quw_Latn	4.9	9.0	16.8	22.4	**45.7**	16.5	11.2	46.6	**61.2**	56.0	56.1	60.5
quy_Latn	4.9	23.4	14.8	39.2	**64.3**	29.0	11.1	67.8	74.1	68.1	69.8	**75.7**
quz_Latn	4.9	21.3	10.6	37.3	**55.5**	27.1	12.5	63.7	68.6	**71.5**	70.8	68.9
qvi_Latn	4.9	6.6	9.3	22.5	**41.3**	24.9	7.6	55.0	65.0	66.2	68.2	**70.6**
rap_Latn	4.9	11.8	9.4	**34.7**	21.0	8.8	5.4	45.5	**56.0**	49.1	53.9	39.5
rar_Latn	4.9	15.2	13.0	**31.8**	30.7	17.2	9.0	40.2	**57.2**	49.8	56.9	50.2
rmy_Latn	6.1	14.6	15.4	29.4	**30.2**	18.8	16.0	51.8	49.0	39.2	46.9	**57.1**
ron_Latn	11.3	24.6	28.5	25.7	**47.4**	28.2	67.0	59.2	58.5	60.9	64.1	**69.5**
rop_Latn	7.3	14.8	31.4	46.7	**57.7**	38.8	13.6	59.8	59.2	59.0	**65.1**	59.6
rug_Latn	4.9	12.0	5.0	25.9	8.5	**35.0**	6.2	53.7	60.8	47.0	61.1	**61.9**
run_Latn	4.9	16.7	16.1	41.4	**55.4**	21.4	17.7	54.8	**59.8**	54.4	56.2	49.8

Table 15: F1 scores of baselines and models initialized with OFA on **Taxi1500** (Part III).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
rus_Cyrl	4.9	19.5	47.2	50.0	**64.3**	44.3	68.9	69.5	66.3	64.8	71.9	**75.0**
sag_Latn	4.7	17.1	4.9	21.6	**29.1**	7.2	11.9	38.4	41.1	**52.4**	42.9	48.4
sah_Cyrl	4.9	4.9	26.7	**28.5**	28.3	24.6	14.8	53.4	50.6	57.8	**64.6**	60.8
sba_Latn	4.9	13.8	5.0	**29.1**	25.3	23.8	7.9	43.0	40.2	**45.9**	40.7	38.9
seh_Latn	4.9	19.8	17.2	27.9	**36.0**	28.8	13.4	43.8	**62.7**	48.3	55.1	51.0
sin_Sinh	4.9	10.0	16.4	42.6	**47.3**	26.0	65.8	69.5	60.2	59.2	58.9	**72.2**
slk_Latn	4.9	20.8	29.5	33.3	**40.1**	28.5	**72.6**	57.7	49.6	45.3	54.0	63.5
slv_Latn	4.9	23.8	28.4	36.7	**52.2**	23.6	66.6	66.4	58.0	56.1	57.0	**67.1**
sme_Latn	4.9	15.5	23.5	**30.0**	22.5	19.2	12.3	46.8	40.3	43.0	39.6	**52.5**
smo_Latn	4.9	11.6	16.5	45.4	**55.2**	19.2	12.8	61.5	62.1	59.7	60.4	**66.1**
sna_Latn	4.9	20.8	19.0	35.0	**38.3**	27.7	14.4	37.4	**49.6**	42.7	45.7	48.7
snd_Arab	4.9	12.0	22.7	35.6	**52.1**	42.1	66.4	**71.2**	65.7	66.6	66.5	70.1
som_Latn	4.9	10.3	4.9	9.4	**16.1**	9.4	41.7	41.1	33.1	25.8	33.2	**43.9**
sop_Latn	4.9	13.6	15.2	19.7	**20.9**	15.4	12.7	29.1	43.9	35.8	38.8	**47.7**
sot_Latn	4.9	5.1	8.1	23.8	**33.6**	15.6	15.3	49.2	51.5	47.0	42.8	**62.1**
spa_Latn	17.9	38.4	44.6	**60.9**	60.5	41.5	**74.0**	68.6	61.8	67.0	67.4	66.9
sqi_Latn	22.2	33.9	17.8	54.0	**59.0**	33.7	74.4	72.8	68.3	**75.4**	74.7	70.8
srm_Latn	4.9	20.0	14.7	32.9	**34.8**	24.8	14.1	51.9	53.6	46.8	**58.7**	55.7
srn_Latn	4.9	35.7	29.8	50.5	**65.3**	34.2	15.9	64.3	63.6	64.6	**66.6**	62.8
srp_Latn	6.0	30.1	47.2	50.5	**62.0**	45.1	67.8	67.1	58.6	59.2	69.9	**72.6**
ssw_Latn	4.9	15.5	17.8	26.9	30.5	**31.1**	14.9	37.2	43.9	41.5	50.0	**55.9**
sun_Latn	6.1	16.1	20.0	34.9	**47.4**	28.0	52.9	**58.2**	53.3	52.1	52.1	57.0
suz_Deva	4.9	11.2	9.4	32.4	**45.4**	18.2	16.4	54.5	61.6	55.5	**69.8**	62.9
swe_Latn	4.9	30.7	42.9	41.1	**47.5**	27.3	**74.6**	70.0	65.6	63.7	71.0	70.5
swh_Latn	4.9	9.3	17.4	32.4	**57.0**	24.3	61.3	62.3	55.2	53.0	60.4	**64.7**
sxn_Latn	4.9	13.1	14.5	**38.6**	38.5	17.1	13.1	46.9	42.8	43.6	44.6	**47.3**
tam_Taml	4.9	4.9	22.1	39.9	**58.2**	16.3	62.9	63.5	62.7	59.6	68.0	**74.8**
tat_Cyrl	4.9	7.2	21.7	33.5	**44.9**	40.6	27.8	64.5	66.7	62.3	**73.3**	70.4
tbz_Latn	4.9	5.6	11.9	**28.5**	25.7	17.3	6.9	44.8	44.4	48.2	**56.5**	49.6
tca_Latn	4.9	13.2	8.8	**33.2**	16.9	18.0	9.4	36.8	44.2	46.0	**59.6**	55.2
tdt_Latn	4.9	12.7	23.4	47.6	**51.9**	33.8	15.9	55.3	63.6	60.9	59.3	**70.9**
tel_Telu	4.9	11.2	18.3	25.1	**50.4**	26.1	68.7	63.3	68.7	59.6	66.3	**75.4**
teo_Latn	4.9	11.3	4.9	**18.5**	10.1	12.6	14.2	25.2	**32.6**	30.1	26.2	29.2
tgk_Cyrl	5.1	18.5	39.2	48.5	**52.9**	32.9	9.8	**67.1**	66.8	57.5	63.7	65.9
tgl_Latn	4.8	20.2	30.9	45.5	**47.1**	37.7	53.7	61.2	61.5	51.8	64.0	**67.3**
tha_Thai	4.9	6.6	18.1	58.4	**68.8**	17.2	68.8	64.7	62.3	68.6	72.9	**74.7**
tih_Latn	4.9	18.4	6.2	36.1	**42.4**	37.1	12.8	56.9	62.7	58.7	63.7	**65.7**
tir_Ethi	4.9	4.9	15.1	30.2	**34.1**	11.6	19.5	64.8	55.3	53.2	59.9	**69.9**
tlh_Latn	25.9	48.0	55.8	**64.1**	63.8	47.9	35.0	64.4	67.9	60.5	63.9	**69.4**
tob_Latn	4.9	4.7	8.6	**33.9**	28.4	8.7	7.5	38.0	54.2	43.2	53.0	**54.5**
toh_Latn	4.9	19.3	18.8	29.4	**34.1**	22.3	15.4	44.2	45.2	40.3	37.6	**53.3**
toi_Latn	4.9	15.8	17.2	**28.4**	27.0	16.3	17.6	45.4	39.9	43.8	**51.2**	50.7
toj_Latn	4.9	4.9	11.3	**26.4**	19.9	12.4	14.5	35.9	35.9	39.8	41.9	**46.8**
ton_Latn	4.9	17.7	18.7	**22.6**	22.3	17.8	9.3	47.2	50.7	53.8	53.8	**54.0**
top_Latn	4.9	6.7	4.9	**16.7**	6.3	10.2	10.7	18.1	**33.9**	26.5	24.8	18.7
tpi_Latn	8.1	28.7	26.4	57.8	**67.8**	41.6	12.9	66.9	65.1	58.3	65.9	**69.2**
tpm_Latn	4.9	15.9	11.8	30.0	**33.2**	13.4	12.1	**56.6**	47.0	41.6	47.4	50.1
tsn_Latn	4.9	4.9	4.9	23.1	**25.0**	16.7	11.4	41.2	45.6	40.7	41.0	**52.4**
tsz_Latn	4.9	12.1	7.3	**27.6**	18.5	17.8	10.5	36.9	44.3	39.9	41.4	**51.0**
tuc_Latn	4.9	11.3	7.0	34.5	**41.7**	36.4	8.7	50.2	55.5	54.2	51.6	**66.5**
tui_Latn	4.9	4.9	6.0	15.6	19.9	**22.6**	8.6	**50.6**	43.3	47.6	41.9	46.9
tuk_Latn	4.9	8.6	25.7	32.5	**53.0**	32.6	21.1	66.4	60.7	63.6	63.6	**68.7**
tum_Latn	4.9	17.9	18.2	33.7	**40.1**	24.5	13.3	41.7	**53.2**	44.7	46.4	47.5
tur_Latn	4.9	10.4	24.8	32.0	**45.9**	38.3	66.1	64.2	55.9	56.2	62.4	**67.6**
twi_Latn	4.9	13.9	19.8	29.8	**33.7**	28.7	8.9	40.2	50.9	47.8	53.5	**54.1**
tyv_Cyrl	4.9	11.8	20.3	37.8	**46.7**	30.6	17.2	58.8	60.5	56.2	**66.6**	62.6
tzh_Latn	4.9	4.9	11.3	17.8	**30.0**	13.0	11.4	39.7	41.1	41.8	39.1	**49.3**
tzo_Latn	4.9	4.9	16.2	6.5	**20.4**	9.3	7.7	38.3	36.0	**43.9**	40.0	42.7
udm_Cyrl	4.9	8.0	15.9	23.5	**28.4**	24.3	12.6	52.8	52.0	53.0	59.9	**61.0**
ukr_Cyrl	4.9	29.7	30.7	39.4	**52.1**	32.8	67.8	57.6	58.7	47.5	60.1	**70.6**
urd_Arab	4.9	12.1	6.6	28.7	**44.2**	20.8	53.6	**60.1**	50.1	56.6	55.7	58.5
uzb_Latn	4.9	13.1	6.1	33.4	**39.2**	14.3	53.3	61.3	58.8	54.4	62.0	**64.4**
uzn_Cyrl	4.9	16.6	25.2	47.5	**52.0**	43.5	11.3	**69.8**	68.2	64.4	65.6	69.1
ven_Latn	4.9	19.0	12.8	22.4	**29.6**	24.7	10.9	44.2	45.8	46.4	**48.4**	44.4
vie_Latn	4.9	12.9	15.5	36.5	**48.5**	25.9	**68.8**	65.9	52.6	56.5	58.1	64.4
wal_Latn	4.9	5.2	14.1	**27.8**	23.1	12.0	17.4	49.7	43.9	40.5	44.2	**53.8**
war_Latn	4.9	14.8	25.4	32.8	**47.4**	29.6	21.9	50.0	51.3	48.2	53.7	**57.2**
wbm_Latn	4.9	7.3	16.6	35.7	**54.1**	28.2	10.8	57.2	50.9	**65.7**	59.5	65.4
wol_Latn	4.9	8.9	10.0	10.7	**11.6**	11.0	15.2	34.8	36.2	41.6	**45.2**	43.6
xav_Latn	4.9	4.9	4.9	**16.4**	8.0	7.7	10.3	28.4	27.9	28.0	32.2	**46.7**
xho_Latn	4.9	15.0	8.3	25.7	**36.1**	26.4	20.7	44.6	42.6	42.0	47.5	**51.7**
yan_Latn	4.9	12.1	6.1	20.5	**41.1**	11.1	11.1	46.4	48.0	51.8	57.6	**63.2**
yao_Latn	4.9	14.9	17.7	**25.1**	24.6	17.3	13.5	43.5	44.3	43.4	51.5	**52.6**
yap_Latn	4.9	14.6	11.1	**28.0**	18.8	23.2	10.6	42.7	43.8	48.0	**48.2**	46.4
yom_Latn	4.9	13.8	17.0	**26.6**	22.8	20.5	14.4	31.7	32.0	**41.1**	35.2	36.9
yor_Latn	4.9	4.2	4.9	**21.1**	**21.1**	4.2	14.6	44.8	39.1	49.9	**51.3**	50.2
yua_Latn	4.9	11.4	7.2	18.7	**26.2**	10.6	12.4	26.8	36.1	37.8	32.5	**40.1**
yue_Hani	4.9	20.0	12.2	44.4	**56.3**	15.8	60.1	59.3	59.8	55.7	62.5	**65.5**
zai_Latn	4.9	10.0	7.3	22.9	**30.6**	24.4	14.2	35.2	33.5	42.6	40.6	**51.4**
zho_Hani	4.9	31.1	38.8	48.4	**64.2**	37.8	**71.4**	68.3	69.1	65.8	65.4	70.4
zlm_Latn	4.9	45.6	42.7	59.9	**70.7**	53.3	73.9	70.5	73.1	66.8	73.8	**77.3**
zom_Latn	4.3	6.2	20.0	**21.4**	19.9	16.2	11.4	50.6	54.8	**57.4**	46.4	**57.4**
zsm_Latn	6.1	42.3	49.5	67.2	**69.3**	50.4	**72.9**	67.6	70.0	69.7	71.0	68.8
zul_Latn	4.9	20.9	12.9	38.6	**45.2**	34.8	25.9	53.3	**61.4**	52.0	56.8	59.3

Table 16: F1 scores of baselines and models initialized with OFA on **Taxi1500** (Part IV).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
ace_Latn	27.5	36.3	39.6	36.1	**41.1**	40.4	33.7	42.4	42.5	**42.8**	40.1	40.0
afr_Latn	47.0	72.6	68.4	72.8	**75.1**	73.6	75.7	**76.7**	74.9	73.5	76.6	75.7
als_Latn	43.1	76.1	75.1	76.9	80.3	**80.6**	61.8	81.5	78.7	81.4	**82.7**	80.8
amh_Ethi	0.0	36.6	37.8	**42.9**	42.1	29.3	41.8	30.9	45.5	36.4	**54.0**	44.1
ara_Arab	2.6	25.1	35.8	39.1	**40.9**	37.9	45.4	50.4	48.9	51.6	50.1	**51.7**
arg_Latn	59.8	76.4	74.5	77.9	**79.9**	75.9	73.7	**82.3**	76.6	80.1	77.3	78.4
arz_Arab	2.1	35.2	42.4	**51.0**	49.4	43.8	48.0	53.8	**58.7**	53.0	55.7	55.2
asm_Beng	1.3	34.0	**64.7**	59.7	63.8	54.5	53.3	65.5	**66.7**	66.1	64.9	64.0
ast_Latn	58.1	81.9	79.9	81.1	83.5	**84.4**	80.2	83.7	83.7	82.3	**84.5**	84.4
aym_Latn	36.8	45.2	48.9	44.9	**49.3**	40.2	36.0	**47.7**	43.2	39.1	**47.7**	46.4
aze_Latn	22.5	48.0	41.8	53.7	**61.3**	59.9	63.6	64.9	60.6	61.2	**68.6**	67.5
bak_Cyrl	0.0	53.3	40.7	**62.8**	58.0	59.1	36.6	57.3	52.4	55.5	**65.5**	61.5
bar_Latn	49.6	67.6	62.9	67.0	**69.8**	68.8	57.5	**73.1**	67.3	69.8	69.2	72.4
bel_Cyrl	1.9	60.4	63.9	**70.6**	69.9	66.4	73.2	**75.7**	72.6	71.2	73.5	75.0
ben_Beng	0.6	41.5	60.1	57.0	**64.9**	52.7	65.5	69.6	64.8	**70.2**	67.3	69.6
bih_Deva	2.7	37.1	38.8	47.8	**54.7**	53.1	50.0	56.0	52.5	57.7	**61.7**	56.7
bod_Tibt	0.0	13.9	28.9	33.0	**34.1**	20.9	0.0	23.9	**33.1**	32.5	31.2	28.9
bos_Latn	36.6	66.5	64.1	66.6	**68.4**	64.7	74.5	73.2	71.4	71.3	73.1	**75.3**
bre_Latn	36.5	58.4	54.8	56.7	**60.2**	57.8	59.5	62.7	59.0	62.4	**64.7**	63.6
bul_Cyrl	4.5	68.6	66.9	72.0	**73.9**	69.0	77.2	75.5	71.3	73.9	75.5	**77.5**
cat_Latn	66.7	81.9	77.8	79.8	**82.2**	81.1	81.8	83.1	81.1	82.4	82.8	**84.5**
cbk_Latn	**46.2**	45.2	43.8	42.8	44.5	42.7	52.9	**54.6**	48.0	54.3	51.9	54.1
ceb_Latn	43.3	51.1	**52.8**	50.5	47.8	47.5	54.9	62.5	55.1	45.5	**67.5**	57.0
ces_Latn	49.1	69.9	69.4	72.5	**73.6**	70.4	77.7	77.3	75.3	76.1	78.7	**79.0**
che_Cyrl	1.6	22.2	44.2	**54.1**	28.3	25.8	15.3	64.7	**67.8**	39.3	32.7	44.5
chv_Cyrl	0.0	37.4	61.4	**75.1**	66.0	49.7	58.7	77.4	77.4	75.8	**81.6**	75.6
ckb_Arab	1.1	41.8	62.3	61.0	**69.3**	57.4	33.7	73.6	70.6	**74.6**	70.2	73.9
cos_Latn	50.7	**58.8**	56.7	55.3	58.5	57.9	56.5	55.5	54.4	54.5	**61.1**	59.6
crh_Latn	28.6	43.5	34.7	41.5	47.9	**51.2**	40.7	**55.6**	50.3	47.2	54.1	53.1
csb_Latn	33.8	55.2	57.2	54.9	55.9	**57.5**	54.1	57.1	61.3	60.5	**64.7**	57.9
cym_Latn	31.6	50.7	53.1	48.6	55.2	**58.6**	58.4	62.1	58.7	62.7	**63.4**	62.7
dan_Latn	49.2	76.9	75.6	77.7	**78.6**	76.0	81.1	80.6	78.4	79.0	80.8	**81.5**
deu_Latn	46.3	70.3	69.2	71.7	**74.1**	73.5	74.7	75.7	72.0	74.3	75.7	**76.8**
diq_Latn	21.5	**50.2**	35.3	43.4	42.1	46.3	43.7	52.7	58.4	54.2	**59.8**	53.7
div_Thaa	0.0	24.0	28.8	**43.4**	41.9	29.0	0.0	42.5	47.7	50.9	**57.0**	43.1
ell_Grek	6.3	45.1	53.6	58.5	**61.5**	54.4	**73.7**	71.3	63.0	67.4	69.0	73.3
eml_Latn	29.8	30.2	37.7	38.9	**40.5**	30.9	33.5	38.4	40.5	43.5	**44.8**	39.9
eng_Latn	81.9	**83.3**	82.1	83.0	83.0	83.2	82.5	83.3	82.6	83.0	83.1	**83.5**
epo_Latn	41.0	59.6	63.6	64.1	**65.9**	62.1	64.5	**69.4**	66.1	66.7	66.7	68.6
est_Latn	39.4	64.0	60.4	67.5	**68.2**	66.9	72.2	71.9	71.1	71.4	**74.2**	73.8
eus_Latn	29.4	42.9	37.9	42.7	46.7	**49.0**	59.2	61.5	47.0	53.2	**66.9**	57.2
ext_Latn	27.5	45.0	40.0	43.1	45.6	**45.7**	39.1	44.7	42.6	44.4	**51.8**	46.9
fao_Latn	34.0	61.7	69.0	65.4	**69.7**	66.4	60.2	**71.7**	67.9	64.6	69.2	**71.7**
fas_Arab	0.4	24.0	29.3	**42.9**	36.3	32.9	**51.0**	45.2	47.8	44.1	46.5	49.1
fin_Latn	52.9	67.7	64.6	**70.4**	70.3	67.6	75.6	75.1	73.1	74.5	76.0	**76.6**
fra_Latn	61.8	75.1	75.6	76.4	**78.4**	75.5	77.3	**77.9**	77.4	76.7	76.8	76.5
frr_Latn	38.1	51.6	55.8	**56.3**	55.1	53.4	46.8	53.8	53.8	55.8	**58.9**	55.7
fry_Latn	45.0	71.8	69.5	71.4	74.4	**75.3**	74.0	77.0	74.5	73.3	**77.9**	77.5
fur_Latn	32.2	**56.5**	53.3	52.4	55.0	54.8	42.1	57.7	59.0	53.0	**63.0**	56.3
gla_Latn	40.2	52.6	54.6	56.6	**64.2**	56.7	50.6	59.2	56.4	61.7	**66.1**	53.1
gle_Latn	39.0	58.8	55.3	**65.4**	65.1	62.1	69.3	73.9	65.0	71.0	72.0	**74.0**
glg_Latn	60.3	77.0	75.7	77.9	**78.8**	76.5	**80.2**	79.6	78.5	79.2	79.3	78.7
grn_Latn	36.2	45.6	41.4	43.2	47.4	**50.9**	39.1	52.4	50.9	**58.1**	55.7	52.3
guj_Gujr	0.7	44.0	51.4	49.0	**55.0**	53.2	**60.8**	58.8	53.9	57.1	59.7	60.7
hbs_Latn	42.2	56.0	61.8	65.3	**68.3**	56.7	61.6	58.9	57.8	**66.4**	63.4	65.3
heb_Hebr	3.4	16.5	24.8	30.4	**37.2**	23.5	51.4	46.5	39.3	40.9	46.5	**51.6**
hin_Deva	2.8	44.8	49.0	58.4	**64.2**	59.9	68.5	68.0	66.3	69.3	**70.4**	70.3
hrv_Latn	43.5	72.0	71.7	73.5	**74.5**	73.0	77.0	77.0	75.3	76.0	77.8	**78.2**
hsb_Latn	36.7	58.8	**71.0**	59.8	70.4	65.7	64.0	74.3	73.5	76.5	**79.4**	70.9
hun_Latn	39.1	61.7	57.7	63.5	**67.1**	63.6	76.1	75.6	70.1	72.5	74.5	**77.4**
hye_Armn	3.3	37.0	40.9	45.6	41.6	**48.4**	52.7	50.9	42.8	53.2	53.4	**56.4**
ibo_Latn	34.2	48.1	52.0	47.3	51.5	**52.1**	36.4	52.7	50.4	54.0	**56.5**	52.8
ido_Latn	59.3	80.0	79.7	78.3	**81.7**	80.9	59.8	75.7	**85.1**	80.3	76.4	79.5
ilo_Latn	67.9	67.5	74.3	78.5	**81.3**	70.6	55.2	77.2	72.8	75.3	**83.3**	80.2
ina_Latn	42.1	**58.9**	53.3	55.3	57.5	56.5	53.2	55.7	58.6	56.4	**59.3**	58.4
ind_Latn	35.4	46.7	**54.1**	51.6	52.0	49.2	47.8	**60.5**	49.4	50.2	52.2	55.4
isl_Latn	28.5	60.5	60.6	61.0	**66.3**	59.7	68.8	70.8	67.7	69.7	71.0	**73.2**
ita_Latn	61.8	76.0	75.7	76.3	**78.1**	76.3	76.9	77.3	77.0	76.4	78.4	**79.4**
jav_Latn	38.0	51.5	50.0	49.6	52.1	**55.3**	**58.7**	54.3	56.7	52.8	55.6	57.1
jbo_Latn	24.7	22.7	15.3	18.8	**25.5**	22.9	19.2	25.6	21.1	**32.6**	25.2	25.9
jpn_Jpan	3.4	5.6	13.4	11.9	**16.9**	14.0	19.3	**19.5**	15.3	15.0	17.9	19.2
kan_Knda	5.9	25.8	42.3	42.7	**52.1**	42.7	57.1	59.3	56.4	50.9	58.7	**66.1**
kat_Geor	11.6	43.9	49.6	58.2	**61.9**	57.0	65.7	65.8	63.0	63.6	68.0	**69.2**
kaz_Cyrl	2.7	47.0	44.2	**50.8**	43.9	48.8	42.7	49.2	48.9	52.1	**53.3**	52.7
khm_Khmr	3.8	23.7	34.0	36.1	39.4	**41.5**	39.8	38.6	39.6	42.3	41.7	**43.8**
kin_Latn	46.6	**62.1**	59.4	59.8	61.9	59.5	58.3	63.0	67.3	**70.2**	69.3	67.3
kir_Cyrl	1.4	36.7	35.2	39.0	44.1	**46.8**	45.0	44.4	45.1	42.5	**47.4**	44.8
kor_Hang	6.7	18.2	32.3	38.8	**43.6**	24.7	49.5	48.3	44.2	47.9	**51.0**	48.0
ksh_Latn	26.3	55.8	56.8	57.8	**62.2**	59.9	42.4	60.1	58.0	56.1	**63.1**	60.7
kur_Latn	23.6	54.6	50.4	54.4	**59.6**	56.0	62.2	60.8	60.9	60.9	**67.3**	66.4
lat_Latn	48.1	62.7	69.3	72.7	**77.2**	76.4	69.1	**78.4**	73.1	67.9	71.4	78.1
lav_Latn	36.2	67.9	61.9	**68.5**	67.9	66.4	73.8	71.3	70.6	70.8	73.8	**76.6**

Table 17: F1 scores of baselines and models initialized with OFA on **NER** (Part I).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
lij_Latn	24.6	41.4	36.8	40.0	**45.5**	39.9	38.7	43.4	40.7	42.3	**47.8**	41.2
lim_Latn	42.4	66.7	64.9	**68.1**	63.8	66.7	62.6	67.3	71.4	71.1	67.0	**72.3**
lin_Latn	33.7	51.1	39.0	41.9	**52.3**	46.7	37.1	52.1	53.3	53.9	**60.7**	53.7
lit_Latn	41.2	64.9	61.7	63.8	**66.5**	64.1	71.9	72.8	70.6	71.4	73.0	**73.8**
lmo_Latn	38.6	68.9	65.1	66.8	70.3	**71.2**	67.3	74.6	70.4	**75.6**	75.4	71.0
ltz_Latn	36.5	63.8	61.2	66.5	**68.8**	68.2	49.0	**69.1**	64.6	66.6	68.6	68.0
lzh_Hani	1.1	9.2	**14.6**	14.4	12.2	12.9	15.7	11.9	15.3	10.7	**21.2**	9.5
mal_Mlym	3.2	23.6	36.8	51.6	**56.6**	32.0	**62.8**	56.1	47.2	58.9	60.4	61.9
mar_Deva	3.6	43.7	37.6	44.7	**58.3**	53.9	60.7	61.2	57.6	55.6	58.8	**66.6**
mhr_Cyrl	5.7	51.5	41.1	57.0	49.2	**58.8**	43.4	62.0	55.0	63.1	63.3	**64.3**
min_Latn	28.9	36.3	38.7	35.2	37.5	**47.5**	42.3	**47.7**	38.3	40.8	45.5	40.4
mkd_Cyrl	5.3	59.4	61.8	73.7	**77.1**	66.6	75.8	75.1	71.4	75.6	**77.4**	71.9
mlg_Latn	46.0	49.5	**52.0**	43.0	50.5	47.9	54.6	52.1	49.3	56.7	**60.7**	53.2
mlt_Latn	33.5	54.9	65.6	67.7	**77.5**	59.5	42.4	**78.4**	72.2	72.0	73.9	72.1
mon_Cyrl	10.7	50.6	54.0	59.8	**62.4**	51.7	68.7	67.8	61.9	**69.3**	62.7	66.4
mri_Latn	13.6	48.3	46.0	47.7	**51.8**	45.6	16.0	**55.7**	52.1	**55.7**	41.0	54.6
msa_Latn	42.4	65.5	**67.5**	65.5	67.2	67.1	60.2	68.2	67.5	65.9	**69.3**	68.1
mwl_Latn	27.3	44.6	39.5	42.5	43.5	**47.2**	44.7	45.7	50.6	44.1	**53.7**	51.3
mya_Mymr	0.0	24.9	30.8	**49.1**	41.7	31.2	50.4	54.6	46.7	56.7	50.0	**58.5**
mzn_Arab	0.0	36.8	34.1	41.2	**46.6**	38.9	39.7	45.6	42.9	44.6	**56.0**	47.1
nan_Latn	42.8	69.7	58.5	63.4	**76.1**	57.2	42.3	79.2	66.7	**87.5**	84.0	84.2
nap_Latn	39.1	52.3	50.8	52.3	**63.8**	51.7	50.9	**64.4**	53.7	56.4	58.9	60.4
nds_Latn	30.7	71.7	71.5	**79.7**	76.4	75.0	62.5	74.6	72.1	71.7	75.6	**76.5**
nep_Deva	3.8	41.4	57.6	61.4	**65.1**	50.6	63.5	58.8	59.8	55.2	**63.8**	58.5
nld_Latn	55.7	76.7	73.8	77.4	**79.1**	78.5	79.8	79.7	77.6	78.9	80.7	**81.7**
nno_Latn	35.6	71.1	73.1	73.7	**75.7**	73.6	77.1	77.0	76.6	74.8	76.9	**77.5**
nor_Latn	44.5	69.6	70.4	70.8	**74.6**	73.3	76.7	76.3	75.1	74.8	76.9	**77.9**
oci_Latn	48.2	64.0	68.2	**70.0**	68.8	64.9	63.9	**72.8**	68.8	65.4	67.6	65.7
ori_Orya	2.7	22.5	22.9	26.3	23.4	**28.9**	33.0	30.2	27.4	32.6	**35.4**	31.2
oss_Cyrl	0.0	45.8	44.8	**58.7**	51.7	49.8	31.8	53.4	53.9	52.0	59.8	**61.0**
pan_Guru	3.3	21.3	34.8	27.9	**39.8**	37.1	**49.3**	47.7	48.3	45.9	49.1	47.9
pms_Latn	51.6	78.0	76.9	**81.2**	77.6	74.6	72.1	79.1	77.5	79.5	**83.0**	77.3
pnb_Arab	1.5	46.1	54.8	65.8	**69.4**	43.3	57.8	62.2	60.8	62.8	**72.2**	69.0
pol_Latn	50.4	72.8	70.4	71.4	**74.6**	72.4	77.4	77.4	75.2	76.1	78.3	**78.6**
por_Latn	63.7	73.6	72.8	75.7	**77.0**	73.4	78.1	78.0	76.7	77.0	76.5	**78.9**
pus_Arab	7.1	26.6	31.3	33.2	**37.5**	36.6	33.8	38.1	36.4	42.7	**43.7**	41.1
que_Latn	53.3	54.4	58.8	62.6	**69.5**	64.2	56.2	63.8	63.9	66.1	**66.9**	64.1
roh_Latn	38.1	57.6	**58.9**	48.8	58.2	58.7	51.9	64.4	56.4	59.9	**65.6**	63.5
ron_Latn	49.0	69.1	**70.5**	69.2	64.4	69.9	**75.0**	67.2	74.1	67.4	71.1	70.8
rus_Cyrl	8.3	55.1	53.6	59.6	**61.7**	60.2	64.5	66.4	65.6	66.1	66.6	**69.7**
sah_Cyrl	11.4	60.0	62.6	56.8	63.3	**69.4**	45.8	69.1	73.7	**74.5**	65.2	71.3
san_Deva	1.4	21.5	26.8	23.7	**32.8**	25.5	**41.9**	38.1	33.5	38.2	34.5	36.9
scn_Latn	42.5	61.3	54.2	**63.1**	61.5	57.1	54.4	**69.7**	63.8	**69.7**	66.4	69.4
sco_Latn	68.5	79.7	84.4	75.2	**89.2**	83.2	80.6	84.7	84.4	82.9	**86.2**	85.3
sgs_Latn	26.8	49.2	39.7	48.7	**58.9**	54.8	44.2	**67.8**	58.2	56.3	64.7	62.9
sin_Sinh	14.5	9.6	28.1	**49.3**	48.7	36.5	52.2	52.4	44.3	46.6	53.1	**55.2**
slk_Latn	45.8	69.0	68.0	70.9	**72.8**	68.4	76.3	77.4	75.9	74.3	77.4	**78.2**
slv_Latn	56.8	75.0	73.7	74.2	**77.0**	74.7	78.8	79.4	78.4	79.2	78.7	**80.5**
snd_Arab	4.3	19.7	33.1	36.7	35.0	**38.8**	39.1	37.6	39.2	44.0	**45.8**	40.7
som_Latn	35.2	46.7	41.6	45.8	47.8	**49.2**	56.0	53.3	54.9	51.2	**57.8**	54.1
spa_Latn	50.6	70.6	72.4	73.6	**74.4**	71.1	73.4	**75.9**	75.6	75.0	**75.9**	71.8
sqi_Latn	59.7	71.5	68.9	70.8	70.4	**74.6**	74.9	76.5	71.3	73.9	76.6	**78.0**
srp_Cyrl	4.7	49.1	54.6	62.1	**63.3**	55.7	59.6	62.9	60.7	62.8	62.7	**64.8**
sun_Latn	24.0	41.6	37.5	40.0	41.5	**43.5**	43.7	54.5	51.1	49.8	53.7	**55.7**
swa_Latn	44.6	62.4	65.6	64.2	**68.5**	67.1	60.3	58.4	62.0	66.9	**70.3**	68.4
swe_Latn	46.3	61.1	60.2	59.7	64.6	**69.2**	71.6	69.5	74.6	69.6	**77.0**	66.0
szl_Latn	34.3	55.5	57.6	**63.6**	58.9	54.9	57.9	69.5	68.4	61.5	**69.8**	**69.8**
tam_Taml	2.2	19.2	39.0	**46.6**	46.2	29.5	55.1	49.3	48.2	54.0	**56.8**	54.6
tat_Cyrl	7.7	43.6	54.5	61.8	**65.5**	49.9	39.6	59.5	63.0	70.0	**70.8**	58.2
tel_Telu	5.3	18.7	27.0	36.9	**39.1**	30.7	49.4	43.4	45.0	44.3	**50.8**	48.8
tgk_Cyrl	3.4	46.4	51.2	52.4	**66.7**	50.7	26.3	60.6	63.8	67.3	**74.9**	69.9
tgl_Latn	63.7	73.0	73.3	68.2	**77.2**	71.8	69.6	**76.6**	75.0	74.7	75.8	74.1
tha_Thai	0.5	**4.2**	3.7	0.4	3.7	0.7	3.8	2.0	0.6	1.8	0.5	**6.0**
tuk_Latn	36.3	55.5	52.8	51.7	**62.0**	56.8	45.3	56.2	58.7	58.0	**60.3**	57.9
tur_Latn	40.5	64.3	58.4	62.8	**68.4**	66.6	74.8	74.3	68.1	73.2	**76.7**	76.6
uig_Arab	4.9	20.8	28.4	33.4	**45.1**	35.4	45.5	46.7	42.8	47.4	**53.5**	48.4
ukr_Cyrl	5.4	59.0	59.9	63.6	66.1	**66.9**	**76.8**	72.0	67.3	69.9	71.7	76.7
urd_Arab	0.4	26.9	33.2	52.1	**58.1**	37.5	56.3	60.0	46.2	61.1	**65.3**	59.9
uzb_Latn	53.2	66.8	66.4	69.7	**70.2**	69.9	70.7	74.7	71.1	75.4	**75.5**	72.1
vec_Latn	43.4	59.6	58.7	**64.2**	63.9	**64.2**	57.5	**70.8**	63.7	64.8	69.8	66.4
vep_Latn	40.2	64.9	**69.6**	64.9	65.1	65.6	57.6	67.2	71.0	65.1	**75.8**	73.2
vie_Latn	45.4	55.6	54.7	61.7	**65.6**	57.6	66.9	67.5	62.1	70.4	**71.7**	69.8
vls_Latn	38.3	71.3	71.2	73.1	**73.5**	72.1	63.2	73.0	71.6	70.8	73.1	**74.6**
vol_Latn	59.4	**62.0**	56.0	60.0	57.1	61.0	**60.0**	59.4	59.0	**60.0**	59.4	59.7
war_Latn	62.0	**71.5**	70.6	65.2	68.8	67.0	59.6	**70.8**	66.1	**70.8**	63.9	66.7
wuu_Hani	1.8	38.8	42.2	**43.1**	34.2	38.2	28.9	33.8	27.5	31.2	**44.7**	35.9
xmf_Geor	5.5	41.2	56.4	**63.2**	61.7	59.9	50.6	**62.3**	52.2	57.3	62.0	61.7
yid_Hebr	0.0	25.9	35.1	37.0	**47.8**	39.1	46.2	49.2	46.6	51.7	**55.2**	50.6
yor_Latn	37.9	43.5	44.2	38.4	**53.8**	51.7	40.7	58.3	47.9	62.6	62.2	**64.0**
yue_Hani	1.2	10.5	**21.6**	21.5	20.5	21.0	23.4	20.3	17.5	20.6	**25.8**	24.2
zea_Latn	49.6	54.0	**58.1**	53.9	54.8	57.7	68.1	66.0	66.0	67.8	**69.5**	66.4
zho_Hani	1.6	10.0	19.5	19.0	**19.9**	19.5	24.3	21.6	18.7	19.5	**26.4**	25.6

Table 18: F1 scores of baselines and models initialized with OFA on **NER** (Part II).

Language-script	RoBERTa	RoBERTa-rand	OFA-mono-100	OFA-mono-200	OFA-mono-400	OFA-mono-768	XLM-R	XLM-R-rand	OFA-multi-100	OFA-multi-200	OFA-multi-400	OFA-multi-768
afr_Latn	29.7	79.9	80.5	81.5	**83.7**	82.4	**89.3**	88.1	87.3	87.3	84.6	88.5
ajp_Arab	20.8	50.5	52.5	55.3	**63.6**	54.9	63.0	68.0	67.9	67.4	**72.0**	71.7
aln_Latn	16.0	39.0	40.0	39.3	**46.9**	42.1	**54.1**	48.8	49.6	50.2	50.6	51.9
amh_Ethi	5.7	47.5	43.0	47.7	**58.6**	47.0	63.9	63.2	58.9	63.8	**65.8**	**65.8**
ara_Arab	16.5	53.5	55.8	57.0	**63.8**	61.8	**67.9**	65.6	65.6	66.9	67.3	66.9
bam_Latn	23.3	**40.6**	38.4	33.1	33.5	37.2	25.1	**44.0**	32.2	35.0	37.9	40.5
bel_Cyrl	23.5	77.3	81.3	80.6	**84.6**	81.2	86.0	84.7	85.2	**86.4**	84.8	86.0
ben_Beng	12.9	47.3	63.8	68.6	**76.1**	63.0	82.0	**84.2**	81.6	81.5	79.8	83.3
bre_Latn	23.3	56.9	54.2	51.7	**58.6**	57.5	61.3	60.8	57.8	58.6	**63.4**	63.0
bul_Cyrl	21.2	79.8	82.5	83.2	**86.1**	84.7	**88.6**	87.8	86.9	87.9	87.5	**88.6**
cat_Latn	35.6	84.4	80.6	83.7	**85.1**	84.8	86.6	86.8	86.9	**88.0**	87.3	87.2
ceb_Latn	31.1	56.0	56.7	57.0	**63.4**	59.1	50.1	62.6	61.5	61.8	**67.9**	65.7
ces_Latn	36.2	75.4	78.6	74.9	**81.6**	78.8	**84.4**	84.0	83.1	83.1	83.3	84.3
cym_Latn	18.7	54.9	56.0	53.4	**61.9**	60.5	**65.8**	64.7	61.0	64.3	65.1	65.4
dan_Latn	36.4	85.5	86.3	86.3	**88.3**	86.7	**90.3**	90.1	89.7	89.3	89.6	**90.3**
deu_Latn	50.7	82.5	82.5	83.2	**84.0**	82.4	**88.4**	87.1	86.6	86.9	86.6	86.9
ell_Grek	18.8	72.4	75.3	73.5	**79.8**	73.6	**88.0**	84.5	84.1	86.1	83.9	84.3
eng_Latn	**96.0**	95.9	95.7	95.9	**96.0**	**96.0**	**96.3**	96.1	95.8	95.9	95.9	96.0
est_Latn	30.9	64.5	65.4	67.0	**75.0**	67.9	**85.9**	82.2	79.2	80.6	82.2	83.2
eus_Latn	30.6	43.6	41.8	43.0	**49.4**	44.4	**71.2**	61.1	61.9	59.4	64.0	64.3
fao_Latn	27.8	83.3	84.6	84.8	**87.0**	84.4	77.6	88.3	88.1	87.8	88.7	**88.8**
fas_Arab	12.2	49.8	65.8	67.2	**71.9**	67.2	70.3	70.1	70.8	70.5	71.9	**72.3**
fin_Latn	32.5	55.0	60.8	60.6	**67.9**	59.7	**85.1**	78.2	73.6	76.4	78.6	80.4
fra_Latn	41.0	**80.2**	77.0	79.6	74.2	78.6	85.9	84.9	80.6	82.9	82.7	**86.0**
gla_Latn	16.6	53.4	48.5	57.1	**59.4**	58.9	58.4	59.2	55.9	59.4	**59.9**	58.7
gle_Latn	26.1	57.1	53.3	59.9	65.0	**65.3**	**66.1**	64.9	62.0	62.3	62.7	64.2
glg_Latn	42.8	78.8	77.1	**83.0**	80.6	81.0	82.7	81.9	82.3	**85.3**	84.4	81.3
glv_Latn	24.2	54.9	50.0	47.3	**55.3**	53.8	27.2	**54.2**	52.4	50.8	53.8	51.8
grc_Grek	12.5	33.7	50.9	38.0	**57.8**	38.6	64.7	71.2	67.9	71.5	69.7	**71.9**
grn_Latn	7.3	20.1	17.0	22.9	**24.1**	21.7	10.5	22.6	20.2	23.8	26.6	**27.9**
gsw_Latn	24.7	69.7	69.9	64.9	**73.9**	68.4	49.1	78.0	77.2	77.0	**80.5**	79.8
hbo_Hebr	3.3	2.5	20.8	17.6	**35.4**	19.6	40.3	36.0	36.6	44.0	**47.5**	45.6
heb_Hebr	27.3	40.8	50.2	52.7	**62.0**	50.2	67.5	67.2	64.4	65.7	66.9	**68.9**
hin_Deva	3.5	42.3	51.9	49.3	54.3	**58.3**	**73.2**	67.8	70.3	72.6	72.3	65.3
hrv_Latn	40.7	83.3	82.9	81.4	84.9	**85.3**	85.2	85.0	85.5	**85.8**	85.2	85.6
hsb_Latn	37.3	73.5	75.6	75.0	**80.9**	77.9	72.1	82.3	82.9	83.0	82.1	**83.1**
hun_Latn	34.9	66.6	65.7	68.7	**74.3**	69.8	**82.3**	80.0	78.2	78.9	80.2	81.4
hye_Armn	22.9	67.7	69.7	67.2	**77.0**	68.3	**84.7**	82.8	83.0	84.3	84.3	84.2
hyw_Armn	18.0	62.6	68.2	66.5	**73.3**	67.0	79.0	81.2	79.6	81.7	**82.8**	81.5
ind_Latn	32.4	76.7	78.9	78.9	**81.7**	80.6	**83.7**	**83.7**	83.2	82.6	83.1	83.5
isl_Latn	20.2	70.6	73.8	74.0	**78.5**	72.8	**84.4**	81.9	80.8	81.2	82.5	82.7
ita_Latn	44.8	79.3	80.0	83.6	**87.0**	82.1	87.4	87.6	87.2	**88.8**	88.7	88.3
jav_Latn	37.9	57.7	67.2	64.9	**73.0**	69.3	73.4	**75.8**	72.2	73.4	74.3	74.6
jpn_Jpan	**16.3**	10.5	8.5	14.8	11.5	15.7	14.8	21.8	20.1	20.8	24.0	**30.6**
kaz_Cyrl	30.6	50.4	58.8	62.5	**69.1**	61.5	**77.2**	74.5	73.9	74.7	76.1	75.6
kmr_Latn	22.9	47.5	67.6	59.6	**69.7**	62.3	73.5	73.7	74.2	73.0	**76.5**	74.5
kor_Hang	24.1	33.7	42.7	42.2	**50.6**	42.6	53.6	52.4	53.1	52.8	**53.7**	51.5
lat_Latn	26.7	54.3	50.3	55.2	**63.0**	54.4	**75.6**	69.0	64.4	69.3	71.5	71.0
lav_Latn	31.5	68.3	71.4	70.1	**75.9**	72.5	**85.8**	82.2	80.6	81.6	82.0	83.4
lij_Latn	25.5	67.3	69.0	66.5	**72.7**	66.0	47.0	77.0	77.2	77.0	77.0	**77.3**
lit_Latn	32.4	59.2	64.2	65.6	**71.5**	66.9	**84.2**	79.8	77.2	78.9	80.1	80.8
lzh_Hani	2.4	8.9	7.8	**16.7**	14.0	14.5	14.5	20.5	**23.1**	17.7	21.3	22.3
mal_Mlym	29.1	55.3	72.3	75.9	**80.0**	68.6	**86.3**	85.4	**86.3**	82.5	85.9	82.4
mar_Deva	1.5	43.3	56.8	60.8	**68.1**	55.7	82.5	81.3	79.6	79.0	81.0	**82.9**
mlt_Latn	20.2	63.0	73.9	72.4	**75.7**	72.1	21.5	80.8	78.0	79.3	**81.5**	79.5
myv_Cyrl	29.1	50.6	52.2	55.3	**63.9**	53.9	39.2	64.3	62.7	66.0	**70.1**	64.9
nap_Latn	16.7	35.3	37.5	25.0	**47.1**	35.3	58.8	58.8	82.4	55.6	**88.9**	70.6
nds_Latn	27.9	71.2	72.2	68.8	**75.2**	71.8	57.3	77.9	77.3	75.8	**78.2**	76.9
nld_Latn	43.1	84.7	84.7	85.4	**87.1**	85.7	**88.6**	88.4	87.5	88.0	88.4	88.4
nor_Latn	31.8	84.0	84.3	84.7	**87.4**	84.4	**88.3**	87.8	86.5	87.3	87.6	88.1
pcm_Latn	42.8	**56.2**	53.9	54.5	56.1	55.5	46.7	**57.0**	55.9	53.7	56.2	**57.0**
pol_Latn	37.0	74.5	75.8	73.8	**81.2**	78.4	**83.1**	82.0	82.0	81.9	81.2	**83.1**
por_Latn	47.6	84.4	83.6	**86.8**	85.0	86.0	88.3	87.6	87.1	88.4	**88.5**	88.3
quc_Latn	26.5	54.7	62.3	55.7	**63.9**	**63.9**	28.7	**62.7**	57.5	60.6	59.0	60.1
ron_Latn	37.7	69.8	71.0	72.8	**76.3**	72.6	**83.6**	80.0	78.3	79.8	80.1	81.5
rus_Cyrl	24.7	79.6	83.5	83.5	**86.3**	84.4	**89.0**	88.0	87.0	88.4	88.0	88.4
sah_Cyrl	17.2	34.7	47.0	44.0	**72.1**	50.6	22.3	73.4	74.7	76.8	**80.5**	78.4
san_Deva	2.3	8.8	**13.3**	13.0	11.3	8.8	19.1	22.2	13.4	16.6	25.3	**26.9**
sin_Sinh	18.2	28.8	38.6	40.1	**47.4**	40.9	**58.5**	50.7	54.9	54.4	55.4	49.7
slk_Latn	36.5	75.5	77.7	76.1	**81.9**	76.1	84.1	**84.8**	84.1	83.7	82.8	84.6
slv_Latn	28.8	65.6	67.5	66.5	**71.4**	67.1	**78.1**	74.5	73.7	74.4	75.1	75.3
sme_Latn	24.9	56.6	56.9	57.8	**68.8**	63.6	29.8	73.7	72.9	74.4	**78.1**	74.6
spa_Latn	51.8	86.1	84.9	86.7	85.8	**87.5**	88.2	87.8	88.1	**89.0**	88.8	87.3
sqi_Latn	50.1	71.5	74.7	77.2	77.6	**77.7**	**78.5**	76.0	76.3	77.8	74.7	78.3
srp_Latn	41.5	84.2	83.5	81.3	84.9	**85.4**	85.8	84.7	85.7	**85.9**	85.7	84.9
swe_Latn	31.0	85.1	85.6	85.5	**89.2**	86.8	**93.4**	91.4	90.6	91.3	91.5	92.0
tam_Taml	25.7	41.9	57.0	63.9	**70.1**	59.4	**75.6**	73.9	72.0	73.1	73.2	74.3
tat_Cyrl	26.8	51.3	59.3	60.0	**66.8**	62.7	45.6	68.9	72.1	72.0	**72.3**	69.2
tel_Telu	29.9	46.6	57.9	63.0	**74.0**	63.2	**85.7**	80.7	76.5	80.8	79.2	79.7
tgl_Latn	31.0	65.4	69.1	68.4	**73.6**	72.3	73.3	75.2	72.7	75.9	75.6	**76.5**
tha_Thai	4.9	36.4	40.0	45.7	**45.9**	37.2	44.3	55.0	54.7	**55.2**	51.1	54.6
tur_Latn	28.2	45.6	53.9	52.7	**60.5**	53.6	**73.0**	69.1	66.2	66.1	69.0	70.8
uig_Arab	17.1	38.6	44.2	49.7	**58.9**	47.2	68.3	67.4	67.0	66.6	**70.1**	69.0
ukr_Cyrl	25.2	76.3	78.4	78.6	**82.4**	80.3	**85.5**	84.6	83.7	85.2	83.9	84.9
urd_Arab	4.1	31.5	48.8	44.6	**50.1**	43.9	59.6	59.2	65.3	**67.4**	65.5	58.3
vie_Latn	21.0	47.5	50.4	54.0	**61.3**	53.3	**70.4**	67.2	65.1	66.1	65.3	69.1
wol_Latn	23.6	50.6	43.6	43.9	**60.3**	51.3	25.6	**59.5**	55.6	57.9	57.2	56.8
xav_Latn	4.7	**12.6**	11.8	9.6	6.4	7.9	6.2	10.5	10.7	11.1	6.7	**18.3**
yor_Latn	18.8	47.1	56.7	51.0	**63.7**	56.4	22.7	65.3	64.4	**66.6**	64.4	63.8
yue_Hani	15.8	24.4	20.5	28.0	25.7	**34.4**	27.7	34.6	36.7	**42.8**	40.7	42.5
zho_Hani	16.8	22.4	20.0	26.1	24.1	**29.4**	24.6	31.7	39.3	40.6	39.5	**43.1**

Table 19: F1 scores of baselines and models initialized with OFA on **POS**.