

FuxiTranyu: A Multilingual Large Language Model Trained with Balanced Data

Haoran Sun, Renren Jin, Shaoyang Xu, Leiyu Pan, Supryadi,
Menglong Cui, Jiangcun Du, Yikun Lei, Lei Yang,
Ling Shi, Juesi Xiao, Shaolin Zhu and Deyi Xiong*

TJUNLP Lab, College of Intelligence and Computing, Tianjin University
{hrsun,rrjin,dyxiong}@tju.edu.cn

Abstract

Large language models (LLMs) have demonstrated prowess in a wide range of tasks. However, many LLMs exhibit significant performance discrepancies between high- and low-resource languages. To mitigate this challenge, we present **FuxiTranyu**, an open-source multilingual LLM, which is designed to satisfy the need of the research community for balanced and high-performing multilingual capabilities. The base model, FuxiTranyu-8B, features 8 billion parameters and is trained from scratch on meticulously balanced multilingual data that contains 600 billion tokens covering 43 natural languages and 16 programming languages. We also develop two instruction-tuned models: FuxiTranyu-8B-SFT which is fine-tuned on a diverse multilingual instruction dataset, and FuxiTranyu-8B-DPO which is further refined with DPO on a preference dataset for enhanced alignment ability. Extensive experiments on a wide range of multilingual benchmarks demonstrate the competitive performance of FuxiTranyu against existing multilingual LLMs, e.g., BLOOM-7B, PolyLM-13B, and Mistral-7B-Instruct. Both neuron and representation interpretability analyses reveal that FuxiTranyu achieves consistent multilingual representations across languages. To promote further research into multilingual LLMs, we release both the base and instruction-tuned FuxiTranyu models together with 58 pre-training checkpoints at HuggingFace¹ and Github.²

1 Introduction

A well-pretrained base model is crucial for facilitating research and applications of large language models. However, training a base LLM from scratch typically demands a substantial amount of data and significant computational resources, posing a barrier to the development of new LLMs. The

majority of LLMs are usually tailored to specific languages such as English (Touvron et al., 2023a,b) or Chinese (Bai et al., 2023), neglecting the growing demand for multilingual capabilities, especially from low-resource languages. While certain LLMs like Mistral models (Jiang et al., 2023a) demonstrate multilingual capabilities, their coverage is limited, restricting the exploration in massively multilingual settings.

Recent efforts have been dedicated towards mitigating such language-specific constraints through supervised fine-tuning, as exemplified by Okapi (Lai et al., 2023). However, as highlighted by the alignment hypothesis in LIMA (Zhou et al., 2024), the knowledge of LLMs is predominantly derived from pre-training, while supervised fine-tuning primarily aligns model behavior to instructions, which is a narrow subset of the pre-training data. This makes fine-tuning less effective for boosting multilingual abilities when pre-training is dominated by a few languages.

Other initiatives have focused on pre-training multilingual LLMs, such as BLOOM (Scao et al., 2022a) and PolyLM (Wei et al., 2023). Nevertheless, these efforts are hindered by their performance, which does not measure up to that of current trending LLMs. BLOOM suffers from outdated training data, while PolyLM is undermined by imbalanced language distribution, with English data accounting for approximately 70% and Chinese for ~20%, potentially leading to insufficient learning of under-represented languages. Previous studies (Xu et al., 2024) disclose three traits of multilingual LLMs caused by imbalanced language resources: cross-lingual inconsistency, distorted linguistic relationships, and unidirectional transfer between high- and low-resource languages, emphasizing the need for balanced data distribution.

Recently introduced multilingual LLMs, e.g., Aya 23 models (Aryabumi et al., 2024), have demonstrated remarkable performance on multiple

*Corresponding author.

¹<https://huggingface.co/TJUNLP/FuxiTranyu-8B>

²<https://github.com/tjunlp-lab/FuxiTranyu>

LLMs	Pre-training Tokens	Languages	Base Model Available	Pretraining Checkpoints Available
BLOOM-7B1 (Scao et al., 2022a)	300B	46 NLs + 13 PLs	✓	✓
Aya 23-8B (Aryabumi et al., 2024)	Unknown	23 NLs	×	×
PolyLM-13B (Wei et al., 2023)	638B	18 NLs	✓	×
FuxiTranyu-8B	606B	43 NLs + 16 PLs	✓	✓

Table 1: Comparison between trending multilingual large language models and FuxiTranyu, where NL stands for natural language while PL for programming language.

multilingual benchmarks. They are derived from the CommandR series of models³ by performing supervised fine-tuning. However, only the weights of Aya 23 have been released, with its base model remaining undisclosed.

In this work, we present **FuxiTranyu**, a family of multilingual LLMs supporting 43 natural languages and 16 programming languages. The FuxiTranyu initiative aims to mitigate the aforementioned challenges of multilingual LLMs. The base model comprises 8 billion parameters and has been trained from scratch using approximately 600 billion multilingual tokens. To ensure balanced learning across all supported languages, we have manually controlled the sampling ratio of pre-training data for different languages, striving for as balanced distribution as possible. In line with our commitment to advancing research in multilingual LLMs, we have also released 58 pre-training checkpoints, resonating with the efforts of LLM360 (Liu et al., 2023). Table 1 compares FuxiTranyu with currently available multilingual LLMs from different perspectives.

In addition to the base model, we develop two instruction-tuned models: FuxiTranyu-8B-SFT, fine-tuned on a collected high-quality multilingual instruction dataset, and FuxiTranyu-8B-DPO, further tuned on preferences with DPO for enhanced alignment ability.

Our evaluations focus on knowledge, capability and alignment dimensions categorized by Guo et al. (2023). Evaluation results on multilingual discriminative tasks such as multilingual ARC, HellaSwag, and MMLU (Lai et al., 2023), XWinograd (Muennighoff et al., 2022; Tikhonov and Ryabinin, 2021), XCOPA (Ponti et al., 2020), XStoryCloze (Lin et al., 2021), and multilingual generative tasks including WMT and IWSLT translation benchmarks (Bojar et al., 2016; Cettolo et al., 2017) and XL-Sum summarization benchmark (Hasan et al., 2021), demonstrate FuxiTranyu’s superior performance compared to BLOOM-7B1

and PolyLM-13B, as detailed in Section 5. The instruction-tuned models, FuxiTranyu-8B-SFT and FuxiTranyu-8B-DPO, also outperform Llama-2-Chat-7B, Mistral-7B-Instruct-v0.1, BLOOMZ-7B1, PolyLM-MultiAlpaca-13B on translation and summarization benchmarks.

To further understand the multilingual capabilities of FuxiTranyu models, we have conducted neuron- and representation-level analysis, revealing that FuxiTranyu-8B learns more language-agnostic representations compared to BLOOM-7B1 (Scao et al., 2022a), which can be attributed to the balanced pre-training data. However, languages with extremely limited resources, such as Bengali and Tamil, are allocated with fewer neurons. Additionally, different layers and components of FuxiTranyu-8B handle multilingual text differently, with deep layers being more language-specific and the importance of attention and MLP components varying across layers.

2 Related Work

Recent advanced LLMs (Touvron et al., 2023a,b; Dubey et al., 2024; Bai et al., 2023; Yang et al., 2024; Young et al., 2024; Jiang et al., 2023a; Team et al., 2024a,b) have excelled in NLP and cross-modal tasks, sparking increased research on multilingual LLMs (Scao et al., 2022a; Chowdhery et al., 2022; Wei et al., 2023), which aim at broader language support. There are three main approaches to building multilingual LLMs: pre-training from scratch, continual pre-training, and post-training (e.g., supervised fine-tuning and reinforcement learning from human feedback).

Pre-training from scratch, like PaLM 1&2 (Chowdhery et al., 2022; Anil et al., 2023), BLOOM (Scao et al., 2022a), and PolyLM (Wei et al., 2023), leverages extensive training corpora from diverse sources, enabling the incorporation of new knowledge. However, pre-training poses a variety of challenges, such as the need for vast computing resources, which can hinder the development of new multilingual LLMs.

³<https://cohere.com/command>

Additionally, it also suffers from the curse of multilinguality (Conneau et al., 2019; Chai et al., 2022; Dubey et al., 2024; Gurgurov et al., 2024), where the performance of individual languages deteriorates as the number of languages increases. On the other hand, continual pre-training, like Aurora-M (Nakamura et al., 2024), LLaMAX (Lu et al., 2024), is more efficient but risks catastrophic forgetting of previously learned knowledge.

Supervised fine-tuning (SFT) often leverages multilingual instruction data or incorporates translation tasks to address data scarcity (Shen et al., 2023a; Lai et al., 2023; Wang et al., 2022). However, both continual pre-training and SFT rely heavily on high-quality, diverse datasets, which are often limited to many languages. Reinforcement Learning from Human Feedback (RLHF) is increasingly used to align models with human preferences (Shen et al., 2023b). In multilingual LLMs, multilingual RLHF data are used to train multilingual reward models (Chen et al., 2024). However, RLHF typically relies on human-annotated data, which can be expensive and time-consuming to collect, especially for under-resourced languages. While these methods can achieve impressive performance, they can also be computationally expensive and may not generalize well to unseen languages.

3 Pretraining

We elaborate on the sources and domains of our pre-training data and the efforts we have made in the pre-processing stage in Section 3.1. Next, we discuss the details of our FuxiTranyu architecture in Section 3.2. We present the strategy we used to determine which languages should be supported by the FuxiTranyu series of models in Appendix A, the details of our tokenizer training in Appendix B, and the pre-training settings in Appendix C.1.

3.1 Data Collection

The quantity, diversity, and quality of data have proven the most crucial factors determining the performance of a pre-trained base model (Hoffmann et al., 2022; Touvron et al., 2023a,b). In pursuit of these objectives, we collect a substantial volume of multilingual data to ensure there are enough tokens for pre-training, in line with scaling laws. Our data collection encompasses a broad spectrum of domains, including public web documents, encyclopedic content, reports, books, scientific articles, and codes. To ensure the quality of the collected

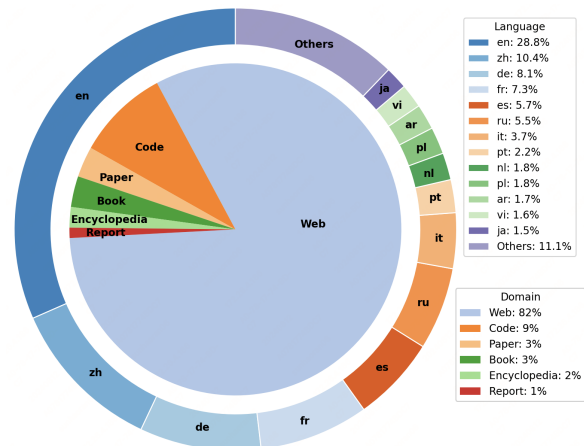


Figure 1: Languages and domains distribution in the pre-training data of FuxiTranyu.

corpora, we have employed heuristic quality filters, learned quality filters, and deduplication processes. The composition of the pre-training data mixture is illustrated in Figure 1, and we will delve into the specifics of data collection and pre-processing in the remaining of this section.

A significant portion of our multilingual data comprises web documents, a common approach in open-sourced LLMs (Touvron et al., 2023a; Bai et al., 2023; Cai et al., 2024; Young et al., 2024). We opt to utilize CulturaX (Nguyen et al., 2023), a filtered subset of OSCAR (Ortiz Su’arez et al., 2020; Suárez et al., 2019) (itself a subset of Common Crawl) and mC4 (Raffel et al., 2020) datasets. To improve quality and diversity, we supplement these with data from ROOTS (Laurençon et al., 2022), MultiUN (Eisele and Chen, 2010; Chen and Eisele, 2012), and OpenSubtitles (Lison and Tiedemann, 2016), focusing on languages in our language list. Additionally, we incorporate data from encyclopedias, reports, books, and articles, drawing inspiration from Phi series models (Gunasekar et al., 2023) that achieve strong results using high-quality textbooks. We have collected approximately 500GB of article data from Semantic Scholar (S2ORC) (Lo et al., 2020), and around 10GB of Chinese books from the Fudan Cbook dataset.⁴ We also source multilingual book data from Project Gutenberg, though it forms a small portion of the final corpus.

Additionally, we collect 535GB of code data from open-source datasets, primarily from Starcoder data,⁵ a subset of the Stack dataset (Kocetkov

⁴<https://github.com/FudanNLPLAB/CBook-150K>

⁵<https://huggingface.co/datasets/bigcode/starcoderdata>

et al., 2022) used to train the StarCoder model (Li et al., 2023). We also include a subset of Github code from the RedPajama dataset.⁶

At the filtering stage, we primarily employ three methods similar to prior works (Scao et al., 2022a; Almazrouei et al., 2023; Bai et al., 2023; Young et al., 2024). The initial filtering phase uses heuristic rules to exclude undesired documents. This involves filtering out documents containing black-listed URLs or words, such as stop words or flagged words. Subsequently, we filter documents based on statistical information, including the ratio or the number of repeated n-gram characters or words, as well as the document length. Following this, we apply a learned quality filter method based on specific metrics, such as perplexity. In line with the approach taken in BLOOM (Scao et al., 2022a), we utilize KenLM (Heafield, 2011) to compute the perplexity of the documents and subsequently filter out those exceeding a predefined threshold.

Upon completion of the quality filter stage, significant efforts are dedicated to data deduplication, as previous studies have emphasized its importance for LLM performance (Lee et al., 2022). We employ MinHash for fuzzy-match deduplication. However, due to the memory-intensive nature of deduplication, processing the entire dataset at once on a server with limited memory is unfeasible. Yet, processing only a portion of the data will not achieve complete deduplication. To address this challenge, we apply a strategy of multi-turn micro-deduplication. We split large documents into chunks and store them in a chunk pool. In each turn, we randomly select chunks from the pool, assemble them back into documents, and perform deduplication on these assembled documents. After processing, the deduplicated documents are again split into chunks and reintegrated into the pool. This process is repeated multiple times until the number of filtered-out documents drops below 1%. This approach is used for high-resource languages, while low-resource languages are processed in memory due to their smaller dataset size. In the case of code data, we also utilize the MinHash algorithm for data deduplication. Specifically, we leverage the implementation from the bigcode project.⁷

⁶<https://huggingface.co/datasets/togethercomputer/RedPajama-Data-1T>

⁷https://github.com/bigcode-project/bigcode-dataset/blob/main/near_deduplication/minhash_deduplication.py

3.2 Model Architecture

The architecture of FuxiTranyu has been crafted using a modified GPT-2 style framework, drawing inspiration from successful open-source LLMs such as BLOOM, LLaMA, and Qwen. Our modifications are as follows:

- **Untied Embeddings.** We opt to separate the weights of the input and output embeddings to enhance performance, despite the resulting increase in total model parameters and memory usage.
- **Linear Bias.** In contrast to prior approaches (Chowdhery et al., 2022; Touvron et al., 2023a), we choose not to eliminate the linear bias of the linear projection layers in self-attention and feed-forward layers.
- **Position Encodings.** To extend the model’s ability to handle long context, we adopt RoPE (Su et al., 2021), replacing the original absolute or relative position embedding method utilized in T5 (Raffel et al., 2020). RoPE has demonstrated promising results in managing long context situations and has been widely employed in LLMs (Touvron et al., 2023a; Inc., 2023; Bai et al., 2023).
- **Normalization.** Given the significance of pre-training stability in training large LMs with a substantial number of tokens, we implement pre-normalization due to its superior stability compared to post-normalization (Xiong et al., 2020). Furthermore, we incorporate the widely used RMSNorm (Zhang and Sennrich, 2019; Jiang et al., 2023b) to enhance training efficiency.
- **Activation Function.** While SwiGLU (Shazeer, 2020) has been a popular choice for activation functions due to its performance improvements (Scao et al., 2022b), it introduces an additional linear function into the activation process, resulting in a 50% increase in parameters in the feed-forward layer. Considering this, we decide to use the GeLU (Hendrycks and Gimpel, 2016) activation function. GeLU has been shown to achieve similar performance to SwiGLU, as reported in (Scao et al., 2022b).

4 Post-training

To develop a model capable of following instructions and engaging in conversational interactions with humans, we have adopted the instruction fine-

tuning and reinforcement learning (RL) approach outlined in Ouyang et al. (2022).

During the instruction fine-tuning phase, we curate a diverse and high-quality open-source instruction dataset. Given the abundance of instruction-following datasets that have demonstrated exceptional alignment results with various models, manually selecting and fine-tuning the mixture rates for each dataset becomes a challenging task. Consequently, we opt to designate a primary dataset and supplement it with additional datasets. In this context, we select the OpenHermes 2.5 data collection (Teknium, 2023) as our base dataset, which is composed of multiple datasets covering a wide range of instructions and yielding excellent results when fine-tuned with Mistral-7B-v0.1. We make modifications to the original OpenHermes 2.5 dataset by replacing Airoboros 2.2 with Airoboros 3.2.⁸ Additionally, we incorporate the Aya dataset (Singh et al., 2024) to enhance the multilingual capabilities of our base model. We filter out the instructions where language is not included in our pre-training language list. To bolster the model’s proficiency in Chinese, we include the COIG-CQIA (Bai et al., 2024), ruozhiba-gpt4,⁹ and in-house Chinese multidisciplinary instruction data as supplementary datasets. To enhance math and coding abilities, we use the dart-math-hard (Tong et al., 2024) and Magicoder-Evol-Instruct¹⁰ (Luo et al., 2023) datasets. The involved languages in the supervised fine-tuning stage can be found in Appendix C.2.

In the RL training stage, we opt to use DPO (Rafailov et al., 2023) as our RL algorithm instead of RLHF (Ouyang et al., 2022; Schulman et al., 2017), as it requires less GPU memory than RLHF, which utilizes PPO as the RL algorithm. We use UltraFeedback (Cui et al., 2023) for the DPO training, since this dataset focuses on general alignment ability and has been successfully utilized by Zephyr (Tunstall et al., 2023) to train the DPO model.

We detail the settings of post-training in Appendix C.2.

5 Experiments

We conducted extensive experiments to evaluate the capabilities of FuxiTranyu (both the base

⁸<https://huggingface.co/datasets/jondurbin/airoboros-3.2>

⁹https://huggingface.co/datasets/hfl/ruozhiba_gpt4

¹⁰<https://huggingface.co/datasets/ise-uiuc/Magicoder-Evol-Instruct-110K>

model and instruction-tuned models) under the multilingual setting. We compared FuxiTranyu against strong baselines, including both English-centric and multilingual models. For English-centric models, we used Llama-2 (Llama-2-7B, Llama-2-chat-7B) (Touvron et al., 2023b) and Mistral (Mistral-7B-v0.1, Mistral-7B-instruct-v0.1) (Jiang et al., 2023a) as baseline. For multilingual models, we compared FuxiTranyu with BLOOM (BLOOM-7B1, BLOOMZ-7B1) (Scao et al., 2022a; Muennighoff et al., 2022), PolyLM (PolyLM-13B, PolyLM-MultiAlpaca-13B) (Wei et al., 2023), and LLaMAX2 (LLaMAX2-7B, LLaMAX2-7B-Alpaca) (Lu et al., 2024).¹¹ We used the LM Evaluation Harness framework (Gao et al., 2023) for all evaluation experiments.

Discriminative Tasks For evaluating discriminative tasks, we used ARC (Clark et al., 2018), Hellaswag (Zellers et al., 2019), MMLU (Hendrycks et al., 2020), XWinograd (Tikhonov and Ryabinin, 2021), XCOPA (Ponti et al., 2020), and XStoryCloze (Lin et al., 2021) datasets. Specifically for the multilingual evaluation, we utilized the multilingual version of ARC, HellaSwag and MMLU datasets (Lai et al., 2023) and selected 15 languages for the evaluation (ar, bn, de, en, es, fr, hu, id, it, pt, ru, sk, ta, vi, zh). For XWinograd, XCOPA, and XStoryCloze datasets, we utilized all of the languages provided in the datasets.

Generative Tasks We evaluated the performance towards generative tasks, especially in translation and summarization tasks. For the translation task, we employed WMT14 in en-fr translation direction (Bojar et al., 2014), WMT16 in en-de and en-ro translation directions (Bojar et al., 2016) and IWSLT 2017 (Cettolo et al., 2017) in en-ar translation direction for measuring the translation performance in our models and benchmark models. For the summarization task, we used XL-Sum (Hasan et al., 2021) dataset. We selected 15 languages for the evaluation (ar, en, es, fr, gu, hi, id, mr, pt, ru, sr, ta, uk, vi, zh).

5.1 Base Model Evaluation

First, we report the experiment results of our base models vs. baseline models. We focus on evaluating the capabilities of LLMs towards discriminative tasks. Evaluation results are shown in Table 2.

¹¹LLaMAX series models are continual pre-trained on the Llama-2 model to support beyond 100 languages.

Models	m-ARC (25-shot)	m-Hellaswag (10-shot)	m-MMLU (5-shot)	XWinograd (5-shot)	XCOPA (0-shot)	XStoryCloze (0-shot)
Llama-2-7B	35.5	48.6	35.4	78.0	58.9	55.6
Mistral-7B-v0.1	40.7	54.5	46.7	80.5	55.8	57.2
BLOOM-7B1	31.8	43.4	27.1	70.0	56.9	58.2
PolyLM-13B	30.6	46.0	26.4	73.4	58.9	56.4
LLaMAX2-7B	33.1	50.3	26.7	76.9	54.5	58.8
FuxiTranyu-8B	32.7	51.8	26.6	76.1	60.5	58.9

Table 2: Average performance of FuxiTranyu-8B base model compared to BLOOM-7B1, PolyLM-13B, Llama-2-7B, Mistral-7B-v0.1, and LLaMAX2-7B on multilingual discriminative and generative tasks.

Models	m-ARC (25-shot)	m-Hellaswag (10-shot)	m-MMLU (5-shot)	XWinograd (5-shot)	XCOPA (0-shot)	XStoryCloze (0-shot)	Translation (BLEU, 0-shot)	Summarization (ROUGE, 0-shot)
Llama-2-Chat-7B	36.4	46.3	36.0	74.8	55.9	56.5	22.1	4.6
Mistral-7B-Instruct-v0.1	36.3	45.5	39.0	74.0	54.5	53.4	19.1	2.2
BLOOMZ-7B1	31.2	38.0	25.8	64.0	53.3	49.8	14.7	4.4
PolyLM-MultiAlpaca-13B	28.6	39.1	25.9	70.9	59.9	57.0	-	-
LLaMAX2-Alpaca-7B	38.7	52.5	35.4	77.4	56.6	62.0	29.1	0.3
FuxiTranyu-8B-SFT	32.8	49.2	26.9	74.7	61.2	57.4	28.3	9.2
FuxiTranyu-8B-DPO	34.2	47.9	27.4	69.1	61.8	57.6	26.8	7.1

Table 3: Average performance of FuxiTranyu-8B instruct and chat models compared to BLOOMZ-7B1, Llama-2-Chat-7B, and Mistral-7B-Instruct-v0.1 on multilingual discriminative and generative tasks.

Our model achieves the best performance on the XCOPA and XStoryCloze tasks. For other tasks, our model is significantly better than multilingual models like BLOOM-7B and PolyLM-13B. When compared to LLaMAX2-7B, the evaluation results of our model are almost comparable, with no significant difference from the evaluation results of LLaMAX2-7B. But compared with English-centric models, our model is still worse than Llama-2-7B and Mistral-7B-v0.1 due to the limited training data used for English.

5.2 Instruction-Tuned Model Evaluation

We further compared our instruction-tuned models with other instruction-tuned models. We evaluated these models on both discriminative and generative tasks. Results are shown in Table 3. On discriminative tasks, our models achieve the best result on XCOPA. For m-Hellaswag, XWinograd, and XStoryCloze, our models outperform the English-centric models but slightly underperform the multilingual models compared with LLaMAX2-7B. Our models still underperform in m-ARC and m-MMLU tasks due to the limited training data used.

In generative tasks, our models excel on the summarization task, outperforming all baseline models. For the translation task, our models outperform the English-centric models but slightly underperform the multilingual model like LLaMAX2-Alpaca-7B.

More details of our evaluations are discussed in Appendix D, where we report the results for each language tested.

6 Analysis and Interpretability

We further conducted an interpretability analysis of FuxiTranyu to provide a deep understanding of the underlying mechanisms driving its multilingual capabilities. To ensure a comprehensive analysis and consistency with prior research, we investigated our models from both the neuron (Wu et al., 2023; Shi et al., 2024; Leng and Xiong, 2024; Zhang et al., 2024; Tang et al., 2024; Liu et al., 2024; Kojima et al., 2024) and representation (Conneau et al., 2020; TiyaJamorn et al., 2021; Chang et al., 2022; Rajae and Pilehvar, 2022; Xu et al., 2023; Dong et al., 2024; Xie et al., 2024) perspectives. Specifically, our neuron analysis explores the importance of different neurons to the multilingual abilities of the model, while the representation analysis examines the characteristics of multilingual representations learned by the model. Here, we first introduce the details and results of our neuron analysis, while the representation analysis is discussed in Appendix E.1.

6.1 Neuron Analysis

Neurons in a neural network are the basic computational units of the model. Different inputs may fire neurons in different regions, leading to varied outputs. This computational process can be understood from another perspective: different sets of neurons in the model hold varying degrees of importance for the inputs, thus producing different responses and outputs. To better understand

why models generate specific outputs for specific inputs in a multilingual context, we aim to reveal the model’s internal mechanisms by evaluating the importance of neurons. Specifically, we assess the importance of different neurons for various linguistic inputs to determine which neurons play a key role in processing particular languages.

We draw on the approach of assessing parameter sensitivity in model pruning, where the basic idea is that a parameter is considered sensitive or important if removing it, by setting the representation produced by that parameter to zero, significantly affects the loss function (Zhang et al., 2024). Specifically, the model can be represented as a parameter set $\theta = [\theta_1, \theta_2, \dots, \theta_n]$, where $\theta_i \in \mathbb{R}^d$ is the i -th neuron in the model. Let \mathbf{h}_i denote the representation produced by neuron θ_i . The importance of neuron θ_i , denoted as $\Phi(i)$, is defined as the change in the loss function \mathcal{L} before and after setting representation \mathbf{h}_i to zero. Formally, $\Phi(i)$ can be estimated as follows:

$$\Phi(i) = |\Delta\mathcal{L}(\mathbf{h}_i)| = |\mathcal{L}(\mathbf{H}, \mathbf{h}_i = \mathbf{0}) - \mathcal{L}(\mathbf{H}, \mathbf{h}_i)| \quad (1)$$

where \mathbf{H} is the representation produced by a neuron other than θ_i in the same structure as the θ_i .

Calculating the importance of each neuron in the model using the aforementioned method is very time-consuming, as it requires traversing each neuron. However, based on prior studies, we can simplify these calculations using a Taylor expansion, as shown in Equation (2):

$$\Phi(i) = |\mathcal{L}(\mathbf{H}, \mathbf{h}_i = \mathbf{0}) - (\mathcal{L}(\mathbf{H}, \mathbf{h}_i = \mathbf{0}) + \frac{\partial\mathcal{L}(\mathbf{H}, \mathbf{h}_i)}{\partial\mathbf{h}_i}\mathbf{h}_i + R_1(\mathbf{h}_i))| \quad (2)$$

After ignoring the term $R_1(\mathbf{h}_i)$, the neuron importance evaluation function is simplified to $\frac{\partial\mathcal{L}(\mathbf{H}, \mathbf{h}_i)}{\partial\mathbf{h}_i}\mathbf{h}_i$, which is the product of the gradient and the representation. This enables parallel computation of each neuron’s importance.

Furthermore, to measure the significance of a specific parameter set $\alpha = [\theta_l, \theta_{l+1}, \dots, \theta_k] \subseteq \theta$, we compute the importance of each neuron in the set using the following equation:

$$\Phi(\alpha) = \sum_{i=l}^k \Phi(i) \quad (3)$$

where $\Phi(\alpha)$ denotes the importance of the parameter set α . The set α can represent a component or a layer of the model, with the neuron indices in α generally being continuous.

6.2 Neuron Analysis Setup

We chose the Flores-200 dataset (Costa-jussà et al., 2022) to evaluate the importance of neurons. By selecting the languages ar, bn, es, fr, id, pt, ta, vi, zh, en, de, hu, it, ru, and sk, we analyzed the significance of different model components and layers in response to various linguistic inputs.

6.3 Neuron Analysis Results

We analyzed the varying importance of different layers across diverse language inputs, as shown in Figure 4 (Appendix E.2). Our findings indicate that universally, shallow layers exhibit low significance while deep layers demonstrate great importance. Notably, languages such as *bn* and *ta* exhibit a notably diminished importance in deep layers compared to others, aligning with our evaluation results where these languages perform poorly. This discrepancy may stem from their relatively limited representation learning in the pre-training data.

We then analyzed the significance of various components across different language inputs, depicted in Figure 5 (Appendix E.2), with 8 components per layer. Our findings mirror previous conclusions: components in shallow layers exhibit low importance, whereas those in deep layers show high significance. Moreover, a more detailed observation reveals that MLP components hold greater importance in shallow layers, whereas attention components are more critical in deep layers.

7 Conclusion

In this paper, we have presented FuxiTranyu to address the need for open-source multilingual LLMs. Along with the base model, FuxiTranyu-8B, we also present instruction-tuned models fine-tuned on multilingual supervised fine-tuning and preference data, FuxiTranyu-8B-SFT and FuxiTranyu-8B-DPO. Evaluations on multilingual benchmarks show FuxiTranyu outperforms previous multilingual and monolingual LLMs. Furthermore, interpretability analyses underscore the efficacy of the multilingual capabilities embedded in FuxiTranyu.

Acknowledgements

The present research was supported by the National Key Research and Development Program of China (Grant No. 2023YFE0116400). The computing resources used in this project were supported by the Scientific Computing Center of the College of Intelligence and Computing, Tianjin University. We would like to thank the anonymous reviewers for their insightful comments.

References

- Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Hestlow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: An open large language model with state-of-the-art performance.
- Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. PaLM 2 technical report. *arXiv preprint arXiv:2305.10403*.
- Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Kelly Marchisio, Sebastian Ruder, et al. 2024. Aya 23: Open weight releases to further multilingual progress. *arXiv preprint arXiv:2405.15032*.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Yuelin Bai, Xinrun Du, Yiming Liang, Yonggang Jin, Ziqiang Liu, Juntong Zhou, Tianyu Zheng, Xincheng Zhang, Nuo Ma, Zekun Wang, et al. 2024. [Coig-cqia: Quality is all you need for chinese instruction fine-tuning](#).
- Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. [Findings of the 2016 conference on machine translation](#). In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.
- Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleksandra Tamchyna. 2014. [Findings of the 2014 workshop on statistical machine translation](#). In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.
- Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. [Overview of the IWSLT 2017 evaluation campaign](#). In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan. International Workshop on Spoken Language Translation.
- Yekun Chai, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, and Hua Wu. 2022. Ernie-code: Beyond english-centric cross-lingual pretraining for programming languages. *arXiv preprint arXiv:2212.06742*.
- Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. 2022. [The geometry of multilingual language model representations](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 119–136. Association for Computational Linguistics.
- Du Chen, Yi Huang, Xiaopu Li, Yongqiang Li, Yongqiang Liu, Haihui Pan, Leichao Xu, Dacheng Zhang, Zhipeng Zhang, and Kun Han. 2024. [Orion-14b: Open-source multilingual large language models](#). *CoRR*, abs/2401.12246.
- Yu Chen and Andreas Eisele. 2012. [MultiUN v2: UN documents with multilingual alignments](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2500–2504, Istanbul, Turkey. European Language Resources Association (ELRA).
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. [Think you have solved question answering? try arc, the AI2 reasoning challenge](#). *CoRR*, abs/1803.05457.

- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Emerging cross-lingual structure in pretrained language models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6022–6034. Association for Computational Linguistics.
- Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Y. Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. [No language left behind: Scaling human-centered machine translation](#). *CoRR*, abs/2207.04672.
- Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. [Ultrafeedback: Boosting language models with high-quality feedback](#).
- Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.
- Weilong Dong, Xinwei Wu, Renren Jin, Shaoyang Xu, and Deyi Xiong. 2024. [Contrans: Weak-to-strong alignment engineering via concept transplantation](#). *CoRR*, abs/2405.13578.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Andreas Eisele and Yu Chen. 2010. [MultiUN: A multilingual corpus from united nation documents](#). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonnell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. [A framework for few-shot language model evaluation](#).
- Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.
- Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Linhao Yu, Yan Liu, Jiakuan Li, Bojian Xiong, Deyi Xiong, et al. 2023. Evaluating large language models: A comprehensive survey. *arXiv preprint arXiv:2310.19736*.
- Daniil Gurgurov, Tanja Bäuml, and Tatiana Anikina. 2024. Multilingual large language models and curse of multilinguality. *arXiv preprint arXiv:2406.10602*.
- Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. [XL-sum: Large-scale multilingual abstractive summarization for 44 languages](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.
- Kenneth Heafield. 2011. [KenLM: Faster and smaller language model queries](#). In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Dan Hendrycks and Kevin Gimpel. 2016. [Bridging nonlinearities and stochastic regularizers with Gaussian error linear units](#). *CoRR*, abs/1606.08415.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.
- Baichuan Inc. 2023. [Baichuan-7B: A large-scale 7B pretraining language model developed by BaiChuan-Inc](#).
- Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, et al. 2023. Neftune: Noisy embeddings improve instruction finetuning. *arXiv preprint arXiv:2310.05914*.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego

- de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Zixuan Jiang, Jiaqi Gu, Hanqing Zhu, and David Z. Pan. 2023b. [Pre-RMSNorm and Pre-CRMSNorm transformers: Equivalent and efficient pre-LN transformers](#). *CoRR*, abs/2305.14858.
- Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. 2022. [The stack: 3 TB of permissively licensed source code](#). *CoRR*, abs/2211.15533.
- Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. [On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons](#). *CoRR*, abs/2404.02431.
- Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 318–327.
- Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826.
- Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. [Deduplicating training data makes language models better](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.
- Yongqi Leng and Deyi Xiong. 2024. [Towards understanding multi-task learning \(generalization\) of llms via detecting and exploring task-specific neurons](#). *CoRR*, abs/2407.06488.
- Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy V, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Moustafa-Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailley Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. [StarCoder: May the source be with you!](#) *CoRR*, abs/2305.06161.
- Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Nanaman Goyal, Shrutit Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O’Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. 2021. [Few-shot learning with multilingual language models](#). *CoRR*, abs/2112.10668.
- Pierre Lison and Jörg Tiedemann. 2016. [OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).
- Weize Liu, Yinlong Xu, Hongxia Xu, Jintai Chen, Xuming Hu, and Jian Wu. 2024. [Unraveling babel: Exploring multilingual activation patterns within large language models](#). *CoRR*, abs/2402.16367.
- Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. 2023. Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*.
- Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. [S2ORC: The semantic scholar open research corpus](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Yinquan Lu, Wenhao Zhu, Lei Li, Yu Qiao, and Fei Yuan. 2024. Llamax: Scaling linguistic horizons of llm by enhancing translation capabilities beyond 100 languages. *arXiv preprint arXiv:2407.05975*.
- Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. WizardCoder: Empowering code large language models with evolve-instruct. *arXiv preprint arXiv:2306.08568*.
- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao,

- M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.
- Taishi Nakamura, Mayank Mishra, Simone Tedeschi, Yekun Chai, Jason T Stillerman, Felix Friedrich, Prateek Yadav, Tanmay Laud, Vu Minh Chien, Terry Yue Zhuo, et al. 2024. Aurora-m: The first open source multilingual language model red-teamed according to the us executive order. *arXiv preprint arXiv:2404.00399*.
- Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. **Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages**.
- Pedro Javier Ortiz Suarez, Laurent Romary, and Benoit Sagot. 2020. **A monolingual approach to contextualized word embeddings for mid-resource languages**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. **Training language models to follow instructions with human feedback**. In *NeurIPS*.
- Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. **XCOPA: A multilingual dataset for causal commonsense reasoning**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Sara Rajae and Mohammad Taher Pilehvar. 2022. **An isotropy analysis in the multilingual BERT embedding space**. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1309–1316. Association for Computational Linguistics.
- Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.
- Teven Le Scao, Angela Fan, Christopher Akiki, Elie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022a. **BLOOM: A 176B-parameter open-access multilingual language model**. *arXiv preprint arXiv:2211.05100*.
- Teven Le Scao, Thomas Wang, Daniel Hesslow, Lucile Saulnier, Stas Bekman, M Saiful Bari, Stella Biderman, Hady Elsahar, Niklas Muennighoff, Jason Phang, et al. 2022b. What language model to train if you have one million gpu hours? *arXiv preprint arXiv:2210.15424*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Noam Shazeer. 2020. **GLU variants improve transformer**. *CoRR*, abs/2002.05202.
- Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Y. Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou. 2023a. **Flan-moe: Scaling instruction-finetuned language models with sparse mixture of experts**. *CoRR*, abs/2305.14705.
- Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. 2023b. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*.
- Dan Shi, Renren Jin, Tianhao Shen, Weilong Dong, Xinwei Wu, and Deyi Xiong. 2024. **IRCAN: mitigating knowledge conflicts in LLM generation via identifying and reweighting context-aware neurons**. *CoRR*, abs/2406.18406.
- Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.
- Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Matciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Vu Minh Chien, Sebastian Ruder, Surya Guthikonda, Emad A. Alghamdi, Sebastian Gehrmann, Niklas Muennighoff,

- Max Bartolo, Julia Kreutzer, Ahmet Üstün, Marzieh Fadaee, and Sara Hooker. 2024. [Aya dataset: An open-access collection for multilingual instruction tuning](#).
- Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.
- Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache.
- Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. [Language-specific neurons: The key to multilingual capabilities in large language models](#). *CoRR*, abs/2402.16438.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024a. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024b. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Teknium. 2023. [Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants](#).
- Alexey Tikhonov and Max Ryabinin. 2021. [It’s all in the heads: Using attention heads as a baseline for cross-lingual transfer in commonsense reasoning](#).
- Nattapong Tiyajamorn, Tomoyuki Kajiwaru, Yuki Arase, and Makoto Onizuka. 2021. [Language-agnostic representation from multilingual sentence encoders for cross-lingual similarity estimation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7764–7774. Association for Computational Linguistics.
- Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *arXiv preprint arXiv:2407.13690*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.
- Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. [Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5085–5109. Association for Computational Linguistics.
- Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, et al. 2023. Polylm: An open source polyglot large language model. *arXiv preprint arXiv:2307.06018*.
- Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. [DEPN: detecting and editing privacy neurons in pre-trained language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2875–2886. Association for Computational Linguistics.

- Zhihui Xie, Handong Zhao, Tong Yu, and Shuai Li. 2024. [Discovering low-rank subspaces for language-agnostic multilingual representations](#). *CoRR*, abs/2401.05792.
- Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tiejian Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR.
- Shaoyang Xu, Weilong Dong, Zishan Guo, Xinwei Wu, and Deyi Xiong. 2024. Exploring multilingual human value concepts in large language models: Is value alignment consistent, transferable and controllable across languages? *arXiv preprint arXiv:2402.18120*.
- Shaoyang Xu, Junzhuo Li, and Deyi Xiong. 2023. [Language representation projection: Can we transfer factual knowledge across languages in multilingual language models?](#) In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3692–3702. Association for Computational Linguistics.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.
- Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [HellaSwag: Can a machine really finish your sentence?](#) In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.
- Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.
- Zhihao Zhang, Jun Zhao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. [Unveiling linguistic regions in large language models](#). *CoRR*, abs/2402.14700.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Supported Languages in FuxiTranyu

Our language selection strategy primarily stems from two distinct perspectives: the availability of pre-training data and geographical considerations. We initially approach language selection from the perspective of available pre-training data. Given that the majority of our pre-training data is sourced from web documents, e.g., CulturaX, we determine the languages for pre-training FuxiTranyu based on the statistical information derived from CulturaX. We select the top 21 languages based on the number of available tokens in descending order. Subsequently, we manually incorporate Asian languages, encompassing those from Southeast Asia, West Asia, and Central Asia, resulting in a total of 43 languages. The complete list can be found in Table 4.

In terms of programming languages, we initially consider all 13 languages included in BLOOM (Scao et al., 2022a), such as Java, JavaScript, and Python. Additionally, we include three programming languages (SQL, Assembly, and Visual Basic) due to their high popularity, as indicated by the TIOBE index.¹² The complete list of programming languages is provided in Table 5.

B Tokenization

We implement the Byte-level Byte-Pair Encoding (BBPE) algorithm using the Hugging Face tokenizer library. Our tokenizer is initiated from GPT-2’s tokenizer, incorporating both pre-tokenization and post-tokenization processes. Notably, we opt not to split numbers into digits. In line with the approach outlined in BLOOM (Scao et al., 2022a), we expand the vocabulary size to 250,680 to accommodate multilingual scenarios, thereby mitigating the risk of over-segmentation in low-resource languages.

For training the tokenizer, we randomly sample 1 million documents for each language from our collected data. It’s worth noting that for languages with a total document count being less than 1 million, we utilize all available documents in the training data for the tokenizer.

Following the approach used in BLOOM, we also evaluate the performance of our tokenizer using the fertility metric. To assess its efficacy, we conduct a comparative analysis with the Llama-2 and BLOOM tokenizers. This evaluation involves

¹²<https://www.tiobe.com/tiobe-index/>

computing fertility on the same set of documents across different languages. Results are presented in Figure 2, which indicate that the FuxiTranyu tokenizer is more efficient than the others in most languages. Based on our evaluations and interpretability analysis, we believe that the fertility of the tokenizer positively correlates with the model’s performance in specific languages. In the fertility test, we observe that Bengali (bn), Hindi (hi), and Tamil (ta) exhibit high fertility, indicating lower tokenization efficiency in these languages compared to others. Consequently, the performance and importance of neurons of these languages in our base model are also suboptimal. Further details are discussed in Section 6.3.

C Training Details

C.1 Pre-training Details

The training procedure for the FuxiTranyu model adheres to the standard autoregressive language model framework, utilizing the next-token prediction loss as detailed in (Brown et al., 2020). To enhance pre-training efficiency, we employ a document packing method similar to that described in (Raffel et al., 2020). This involves randomly shuffling documents, merging them, and then truncating them into multilingual chunks that adhere to a maximum context length of 4096 tokens during the pre-training phase.

To mitigate memory consumption and further improve training efficiency, we leverage ZeRO-2 (Rajbhandari et al., 2020) and Flash-Attention V2 (Dao, 2024) technologies. For optimization, the standard AdamW optimizer (Loshchilov and Hutter, 2017) is utilized with hyper-parameters set to $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\epsilon = 10^{-8}$. We employ the cosine learning rate scheduler, starting with a maximum learning rate of $3e-4$ and decaying to a minimum of 10% of the maximum rate. Notably, after encountering divergence issues when training approximately 241 billion tokens, we reduced the maximum learning rate to $1e-4$ to match with the learning rate used in BLOOM, given the multilingual context of both models.

Our FuxiTranyu-8B model is trained using the Megatron-LM (Shoeybi et al., 2019) framework on a setup of 32 A800 GPUs, processing a total of 606 billion tokens. The training utilizes FP16 mixed precision to ensure stability. Detailed training parameters and configurations are provided in Table 6.

ISO-931	Language	Language Family	ISO-931	Language	Language Family
ar	Arabic	Afro-Asiatic	ky	Kyrgyz	Turkic
bg	Bulgarian	Indo-European	lo	Lao	Kra-Dai
bn	Bengali	Indo-European	ms	Malay	Austronesian
ca	Catalan	Indo-European	my	Burmese	Sino-Tibetan
cs	Czech	Indo-European	nl	Dutch	Indo-European
de	German	Indo-European	pl	Polish	Indo-European
el	Greek	Indo-European	pt	Portuguese	Indo-European
en	English	Indo-European	ro	Romanian	Indo-European
es	Spanish	Indo-European	ru	Russian	Indo-European
fa	Persian	Indo-European	sv	Swedish	Indo-European
fi	Finnish	Uralic	ta	Tamil	Dravidian
fr	French	Indo-European	tg	Tajik	Indo-European
he	Hebrew	Afro-Asiatic	th	Thai	Kra-Dai
hi	Hindi	Indo-European	tk	Turkmen	Turkic
hu	Hungarian	Indo-European	tl	Filipino	Austronesian
id	Indonesia	Austronesian	tr	Turkish	Turkic
it	Italian	Indo-European	uk	Ukrainian	Indo-European
ja	Japanese	Japanic	ur	Urdu	Indo-European
kk	Kazakh	Turkic	uz	Uzbek	Turkic
km	Khmer	Austroasiatic	vi	Vietnamese	Austroasiatic
ko	Korean	Koreanic	zh	Chinese	Sino-Tibetan
ku	Kurdish	Indo-European			

Table 4: The list of 43 natural languages supported by FuxiTranyu.

Language	Size (GB)	Ratio (%)	Language	Size (GB)	Ratio (%)
Java	96	17.94	Go	26	4.86
JavaScript	70	13.08	SQL	11	2.06
Python	63	11.77	Rust	9.1	1.70
PHP	59	11.02	Ruby	7.9	1.48
C	53	9.90	Scala	5.1	0.95
C++	52	9.72	Lua	3.0	0.56
C#	48	8.97	Assembly	1.6	0.30
TypeScript	29	5.42	Visual Basic	1.5	0.28

Table 5: The list of 16 programming languages covered in FuxiTranyu, including the sizes and ratios of each language.

C.2 Post-Training Details

The instruction datasets collected do not cover all languages used during pre-training. For the current version of FuxiTranyu, we provide support for the following languages: Arabic, Bengali, Burmese, Chinese, Dutch, English, Filipino, Finnish, French, German, Greek, Hindi, Hungarian, Indonesian, Italian, Japanese, Korean, Kyrgyz, Malay, Persian, Polish, Portuguese, Russian, Spanish, Swedish, Tamil, Thai, Turkish, Ukrainian, Urdu, and Vietnamese.

During the instruction tuning phase, we executed the fine-tuning process on 8 A800 80GB GPUs, leveraging the TRL framework for instruction fine-tuning and DPO training. Throughout both stages, we employed the ChatML format¹³ for the chat template, and designated <PAD> as the pad token. We used AdamW (Loshchilov and Hutter, 2017)

¹³<https://github.com/openai/openai-python/blob/release-v0.28.0/chatml.md>

optimizer, complemented by a cosine learning rate scheduler. The maximum sequence length was set to 4096 for both stages.

In the SFT stage, we configured the maximum learning rate to $2e-5$, with a warmup phase spanning 10% of the total steps. The global batch size was set to 512, and the model was trained for 2 epochs. To optimize memory usage, we enabled Flash-Attention V2 (Dao, 2024), ZeRO stage 2 (Rajbhandari et al., 2020), and gradient checkpointing. Additionally, we employed NEFTune (Jain et al., 2023), which introduces noise to embedding weights to enhance the final performance of our instruction-tuned model.

In the subsequent DPO training stage, we adhered to the latest hyper-parameters specified for reproducing the results of Zephyr, as provided by the alignment-handbook.¹⁴ The beta value for DPO

¹⁴[alignment_handbook2023](#)

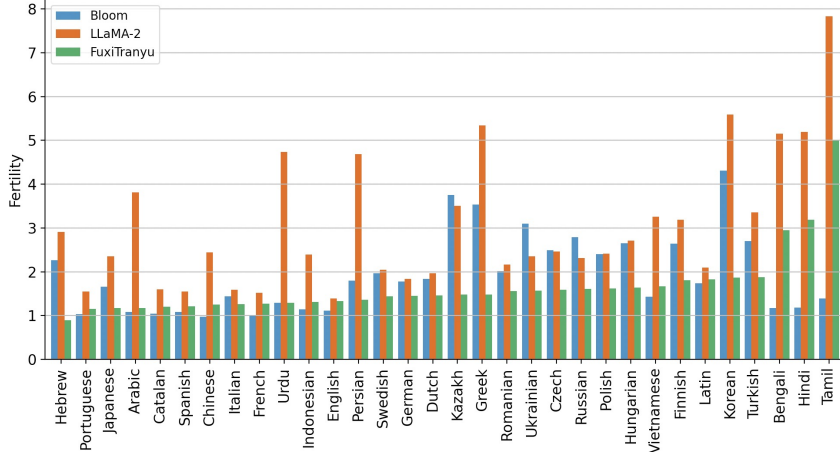


Figure 2: Fertility test results of the tokenizers for FuxiTranyu, Llama-2, and BLOOM.

Pre-Training		SFT		DPO	
# Params	8B	Learning Rate	2e-5	Learning Rate	5e-7
Hidden Size	4,096	Warmup Ratio	10%	Warmup Ratio	10%
Intermediate Size	16,384	Batch Size	512	Batch Size	512
Heads	32	Epochs	2	Epochs	1
Layers	30	NEFTune	✓	beta	0.01
FlashAttn V2	✓	FlashAttn V2	✓	FlashAttn V2	✓
Training Tokens	606B	# Instances	1M	# Instances	61.1k
Position Embed	4,096				
Vocab Size	250,752				
Learning Rate	3e-4 → 1e-4				
Batch Size	2M → 4M				
Context Length	4,096				

Table 6: Model size and hyper-parameters. We append 72 dummy tokens to the vocabulary to make the embedding size be divisible by 128.

was set to 0.01, and the training took 1 epoch on UltraFeedback. The maximum learning rate was set to $5e-7$, with a warmup phase covering 10% of the total training steps. Similar to the SFT stage, the global batch size was maintained at 512, and we activated Flash-Attention V2 and gradient checkpointing to optimize memory usage. To accommodate the policy and reference model within memory constraints, we utilized ZeRO stage 3 for the policy model and omitted ZeRO for the reference model.

D Detailed Evaluation Results

We provide detailed evaluation results for each language in this section. First, we present the results for all 15 tested languages on the multilingual ARC in Table 7, comparing base models and instruction-tuned models. In base models, the results show that our models perform better in 1 of the 15 tested lan-

guages for the ARC task. In instruction-tuned models, our models outperforms in ar and vi languages. We speculate that our models still underperformed on this task due to the relatively small amount of training data used.

Next, we present the results for all 15 tested languages on multilingual HellaSwag in Table 8, comparing base models and instruction-tuned models. Despite our FuxiTranyu-8B model being trained on only about 600B tokens, it achieves remarkable performance. Comparing base models, our models outperforms other models in ar, bn, hu, and vi languages. Our model still lag behind compared with Mistral-7B-v0.1, but outperform other baseline models, except sk language. The SFT and RL-trained models, FuxiTranyu-8B-SFT and FuxiTranyu-8B-DPO, also deliver promising results across all languages, even competing with

powerful monolingual LLMs like Llama-2-7B and Mistral-7B-v0.1, with English and Spanish as an exception.

We report results on multilingual MMLU in [Table 9](#). Our models still underperform baseline models for all languages. It is in line with the number of training tokens utilized in the pre-training process.

Results on XWinograd are depicted in [Table 10](#). In base models, although our models still underperformed compared to Mistral-7B-v0.1, they outperforms previous multilingual LLMs like BLOOM-7B1 and PolyLM-13B across all languages. Notably, our FuxiTranyu SFT and DPO models achieve better results in Chinese.

Results on XCOPA and XStoryCloze are shown in [Table 11](#) and [Table 12](#). For XCOPA, our base models achieve better results in sw, ta, tr, and vi. When compared to instruction-tuned models, our models achieve better results in more languages, specifically in it, ta, th, tr, and vi. On the XStoryCloze task, our base models achieve better results in three languages: ar, my, and ru. However, for instruction-tuned models, our models outperform other baseline models only in my.

We present our evaluation results for generative tasks in [Table 13](#) and [Table 14](#). On the XL-Sum task, our models significantly outperform all baseline models across all evaluated languages, demonstrating the potential of our models on summarization task, particularly in a multilingual context. For the translation tasks in WMT14, WMT16, and IWSLT2017, our models excel in the en-ro and en-de translation directions. However, they still lag behind other baseline models in the ro-en, de-en, fr-en, ar-en, and en-ar translation directions. This indicates that our models perform significantly better for out-of-English translation directions. Although our models underperformed in the en-fr and en-ar directions compared to LLaMAX2-Alpaca, they still achieve notably better results than other models.

E Additional Analysis and Interpretability

E.1 Representation Analysis

Language models encode textual symbols into high-dimensional representations with rich semantic information. For a multilingual language model, due to parameter sharing mechanisms, it encodes textual symbols from different languages into a unified representation space. Furthermore, through

multilingual joint training, the model learns multilingual representations, which encode the intrinsic characteristics of languages and the relationships between different languages. Here, we explore the multilingual characteristics of the model from the perspective of the multilingual representations it learns. Specifically, we calculate the similarity of representations across different languages.

To quantitatively evaluate the similarity between different language representations, we choose cosine similarity for its simplicity and effectiveness. To mitigate the impact of semantic differences on our analysis, we collect multilingual text data from open-source parallel corpora. For a language l , we input its corresponding text data into the model and collect text representations from the last token of each respective text. We then compute the average of these text representations to obtain the language representation v_l for language l . Finally, we calculate the similarity between two language representations as $\text{sim}(l_1, l_2) = \frac{v_1^\top v_2}{\|v_1\| \|v_2\|}$. It's important to note that we extract language representations and compute similarity across each layer of the model.

E.1.1 Analysis Setup

We selected the Flores-200 dataset ([Costa-jussà et al., 2022](#)) as our parallel data source, which includes 2009 sentences for each language. For the explored languages, we chose en, zh, de, fr, es, ru, it, pt, nl, pl, ja, vi, cs, tr, hu, el, sv, ro, uk, and hi, based on their highest language proportions in our pre-training data. For comparison, we also analyzed the BLOOM-7B1 model ([Scao et al., 2022a](#)). For this model, we considered en, zh, fr, es, ru, pt, nl, pl, ja, vi, cs, tr, hu, el, sv, ro, uk, hi, fi, and th.

E.1.2 Results

[Figure 3](#) illustrates the similarities distribution of multilingual representations in the intermediate layers of two models, with languages ordered according to the amount of language resources. It is apparent that for the BLOOM-7B, lower multilingual representation similarities tend to occur between the top 10 languages with higher resource availability and the bottom 10 languages with lower resource availability. In contrast, our model learn more consistent multilingual representations for all the languages we explored. This indicates that our model possesses a higher degree of multilingual balance, which is also reflected in our multilingual evaluation results and pre-training corpus.

Furthermore, we calculate the average similar-

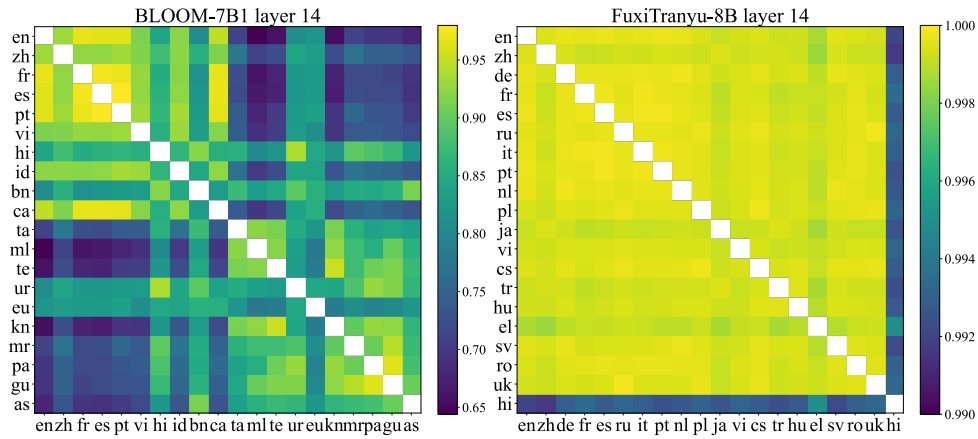


Figure 3: Similarity distribution of multilingual representations in the intermediate layers of BLOOM-7B1 and FuxiTranyu-8B, with languages sorted based on their percentages in the pre-training data.

ity for each layer of the two models, as shown in Figure 6 (Appendix E.2). For our model, it can be observed that there is a significant increase in similarity from the embedding layer to layer 0, reaching a very high level. As the depth of the model increases, the similarity continues to rise, indicating that the model learns richer multilingual alignment information in these layers. Subsequently, there is a sharp decrease in similarity from layer 28 to layer 29, suggesting that language-specific multilingual representations in the final layer are learned to predict the diverse multilingual vocabulary. For BLOOM-7B1, the trend of similarity changes across layers is similar, initially increasing and then decreasing, but the changes are more gradual in magnitude.

E.2 Detailed Analysis Results

We present the varying importance of different layers across diverse language inputs in Figure 4. Figure 5 shows the significance of various components across different language inputs, with 8 components per layer. Furthermore, we calculate the average similarity of multilingual representations across model layers, as shown in Figure 6.

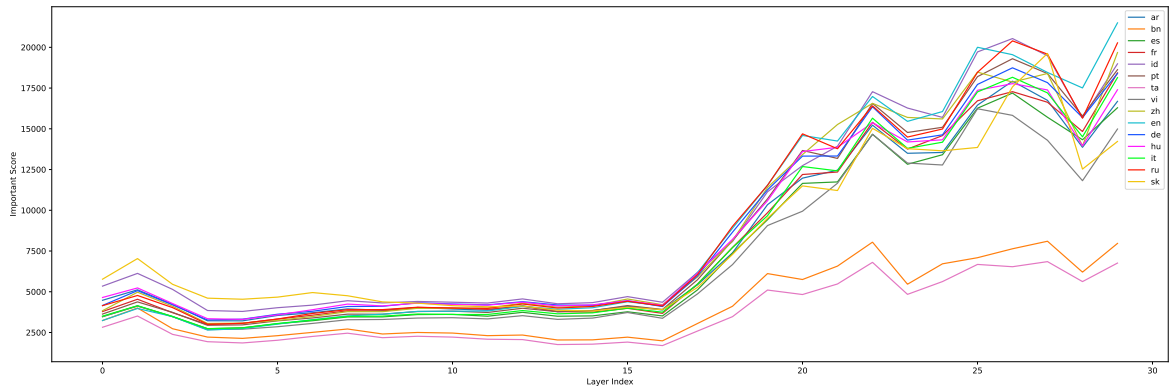


Figure 4: Importance of model layers across various language settings.

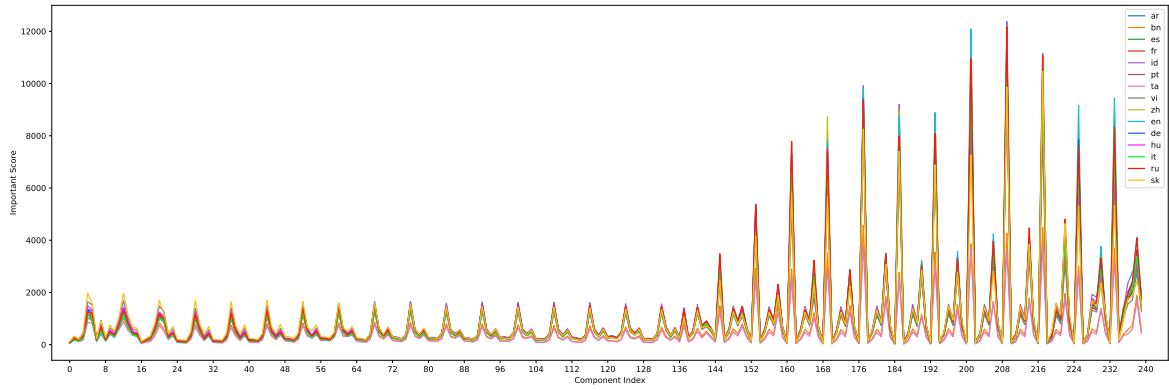


Figure 5: Importance of model components across various language settings.

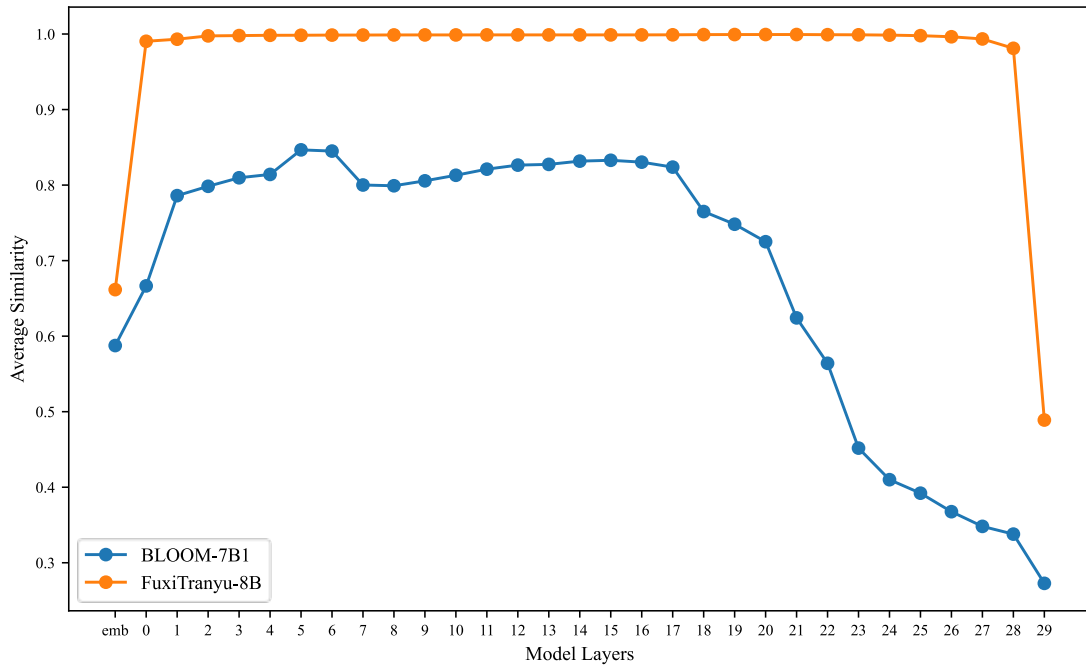


Figure 6: Averaged similarity distribution of multilingual representations for each layer of BLOOM-7B1 and FuxiTranyu-8B, with “emb” denoting the embedding layer.

Models	ar	bn	de	en	es	fr	hu	id
Base Model								
Llama-2-7B	24.9	24.2	<u>37.0</u>	<u>52.5</u>	<u>42.1</u>	<u>43.1</u>	31.7	<u>36.1</u>
Mistral-7B-v0.1	30.5	23.4	43.1	60.0	52.5	47.7	38.7	39.0
BLOOM-7B1	<u>31.4</u>	26.2	27.3	40.0	38.1	36.7	25.9	36.0
PolyLM-13B	27.3	22.4	32.8	41.8	33.2	32.7	23.6	32.8
LLaMAX2-7B	24.4	24.1	35.1	48.7	38.7	38.8	31.6	31.4
FuxiTranyu-8B	31.5	<u>25.8</u>	36.0	38.3	35.3	35.5	<u>32.0</u>	33.3
Instruction-tuned Model								
Llama-2-Chat-7B	26.2	23.9	39.8	53.6	43.0	42.5	32.4	35.4
Mistral-7B-Instruct-v0.1	23.3	24.3	42.5	49.7	<u>45.2</u>	46.5	<u>34.1</u>	30.0
BLOOMZ-7B1	31.2	26.2	25.4	42.7	37.2	37.6	22.8	35.9
PolyLM-MultiAlpaca-13B	27.4	18.4	30.5	38.2	32.9	32.8	18.6	30.2
LLaMAX2-7B-Alpaca	<u>32.4</u>	27.9	<u>42.2</u>	<u>53.5</u>	45.9	<u>44.2</u>	35.6	38.6
FuxiTranyu-8B-SFT	31.1	26.3	33.9	38.9	35.4	<u>36.3</u>	31.5	35.1
FuxiTranyu-8B-DPO	33.3	<u>27.4</u>	35.1	39.3	38.0	37.0	33.7	<u>36.9</u>
Models	it	pt	ru	sk	ta	vi	zh	
Base Model								
Llama-2-7B	<u>40.7</u>	<u>41.8</u>	<u>36.9</u>	29.5	25.0	30.7	36.2	
Mistral-7B-v0.1	49.9	47.2	42.1	37.1	25.9	31.3	42.8	
BLOOM-7B1	29.0	38.6	27.5	24.9	24.2	33.7	<u>37.3</u>	
PolyLM-13B	32.0	34.0	32.8	23.3	<u>25.8</u>	29.2	34.9	
LLaMAX2-7B	36.5	37.4	33.6	<u>30.8</u>	24.1	28.7	32.6	
FuxiTranyu-8B	34.1	36.3	34.7	<u>27.1</u>	24.1	<u>32.4</u>	34.9	
Instruction-tuned Model								
Llama-2-Chat-7B	41.5	<u>43.3</u>	39.9	29.6	26.9	31.5	37.1	
Mistral-7B-Instruct-v0.1	43.3	45.0	<u>39.5</u>	<u>31.1</u>	<u>25.8</u>	26.8	<u>37.7</u>	
BLOOMZ-7B1	27.5	38.7	25.5	22.5	24.2	<u>33.5</u>	37.0	
PolyLM-MultiAlpaca-13B	32.6	32.7	32.5	20.3	20.5	<u>28.8</u>	32.5	
LLaMAX2-7B-Alpaca	<u>42.8</u>	42.7	39.4	36.4	25.5	33.7	39.2	
FuxiTranyu-8B-SFT	33.6	35.2	32.2	29.0	23.5	32.5	36.8	
FuxiTranyu-8B-DPO	36.8	37.1	33.8	29.1	25.1	33.7	37.4	

Table 7: Performance of FuxiTranyu-8B models compared to Llama-2-7B, Mistral-7B-v0.1, BLOOM-7B, PolyLM-13B, and LLaMAX2-7B models on multilingual ARC (25-shot).

Models	ar	bn	de	en	es	fr	hu	id
Base Model								
Llama-2-7B	33.7	28.7	54.0	<u>78.9</u>	60.4	59.1	40.7	48.5
Mistral-7B-v0.1	40.9	31.1	61.1	83.4	67.3	66.5	<u>47.9</u>	53.2
BLOOM-7B1	<u>43.3</u>	<u>32.8</u>	32.4	62.1	56.7	56.6	30.1	49.5
PolyLM-13B	39.6	28.4	49.5	71.3	55.8	54.8	29.3	50.1
LLaMAX2-7B	<u>43.3</u>	32.3	53.8	75.4	59.0	58.1	44.1	51.0
FuxiTranyu-8B	46.7	33.0	<u>56.2</u>	69.2	<u>60.9</u>	<u>60.8</u>	48.2	<u>52.7</u>
Instruction-tuned Model								
Llama-2-Chat-7B	31.4	28.3	50.7	78.6	<u>58.1</u>	57.0	39.0	44.5
Mistral-7B-Instruct-v0.1	31.2	28.7	52.2	70.1	<u>58.1</u>	57.6	39.8	38.1
BLOOMZ-7B1	39.5	31.5	33.1	46.6	48.7	45.7	29.8	42.0
PolyLM-MultiAlpaca-13B	34.0	25.7	40.7	66.0	43.5	43.1	26.7	40.0
LLaMAX2-7B-Alpaca	44.7	33.4	56.8	<u>77.3</u>	62.3	61.4	45.9	53.2
FuxiTranyu-8B-SFT	<u>45.1</u>	31.9	<u>53.4</u>	<u>64.9</u>	57.5	<u>57.9</u>	45.1	<u>49.2</u>
FuxiTranyu-8B-DPO	45.2	<u>33.1</u>	51.4	57.1	55.0	<u>55.2</u>	<u>45.5</u>	48.7
Models	it	pt	ru	sk	ta	vi	zh	
Base Model								
Llama-2-7B	56.0	56.7	49.9	39.2	28.4	45.7	48.7	
Mistral-7B-v0.1	63.0	65.1	58.2	<u>46.6</u>	29.0	47.1	57.2	
BLOOM-7B1	40.8	56.0	32.5	<u>29.8</u>	29.4	<u>48.3</u>	51.2	
PolyLM-13B	51.4	53.7	48.7	30.1	28.0	46.8	52.0	
LLaMAX2-7B	56.1	56.8	51.1	47.8	30.0	47.2	49.3	
FuxiTranyu-8B	<u>58.4</u>	<u>59.3</u>	<u>54.4</u>	43.7	<u>29.9</u>	51.3	<u>52.9</u>	
Instruction-tuned Model								
Llama-2-Chat-7B	53.7	54.0	47.6	36.4	28.8	41.2	45.1	
Mistral-7B-Instruct-v0.1	54.6	55.8	49.6	37.4	27.7	36.1	45.9	
BLOOMZ-7B1	40.3	37.3	33.1	29.6	29.5	40.6	42.6	
PolyLM-MultiAlpaca-13B	40.8	42.4	40.0	27.1	25.2	38.2	53.5	
LLaMAX2-7B-Alpaca	58.7	59.4	53.5	50.3	30.0	49.3	<u>51.9</u>	
FuxiTranyu-8B-SFT	<u>55.2</u>	<u>55.9</u>	<u>51.2</u>	<u>41.1</u>	29.5	<u>48.7</u>	51.3	
FuxiTranyu-8B-DPO	52.9	54.7	51.0	<u>41.1</u>	<u>29.9</u>	<u>48.7</u>	49.3	

Table 8: Performance of FuxiTranyu-8B models compared to Llama-2-7B, Mistral-7B-v0.1, BLOOM-7B, PolyLM-13B, and LLaMAX2-7B models on multilingual HellaSwag (10-shot).

Models	ar	bn	de	en	es	fr	hu	id
Base Model								
Llama-2-7B	<u>29.0</u>	27.5	<u>38.8</u>	<u>46.0</u>	<u>39.9</u>	<u>39.6</u>	<u>33.3</u>	<u>37.0</u>
Mistral-7B-v0.1	35.8	32.2	51.7	60.7	53.7	53.5	46.8	46.9
BLOOM-7B1	27.5	<u>28.2</u>	28.1	25.3	28.9	27.4	26.9	26.9
PolyLM-13B	26.7	26.3	26.1	27.2	26.9	27.2	26.4	24.9
LLaMAX2-7B	25.5	26.2	27.0	28.3	27.0	26.7	26.9	26.8
FuxiTranyu-8B	26.3	25.5	27.6	27.1	27.1	27.5	26.4	26.2
Instruction-tuned Model								
Llama-2-Chat-7B	28.5	27.0	<u>39.5</u>	<u>47.4</u>	<u>40.8</u>	<u>40.3</u>	34.9	<u>35.8</u>
Mistral-7B-Instruct-v0.1	<u>29.9</u>	<u>29.2</u>	42.2	51.9	44.3	44.0	<u>39.3</u>	36.5
BLOOMZ-7B1	24.4	25.9	25.6	22.7	27.1	27.7	26.1	26.3
PolyLM-MultiAlpaca-13B	25.9	26.6	26.2	25.9	26.5	26.3	25.2	25.4
LLaMAX2-7B-Alpaca	30.0	30.4	36.4	43.0	37.2	36.9	47.6	35.5
FuxiTranyu-8B-SFT	26.2	26.8	27.5	28.2	28.1	27.6	26.0	25.9
FuxiTranyu-8B-DPO	27.3	27.8	27.7	28.2	27.9	27.3	26.8	26.6
Models	it	pt	ru	sk	ta	vi	zh	
Base Model								
Llama-2-7B	<u>38.5</u>	<u>38.7</u>	<u>35.7</u>	<u>33.1</u>	<u>27.2</u>	<u>32.8</u>	<u>33.9</u>	
Mistral-7B-v0.1	52.7	53.4	49.8	45.4	29.7	41.5	46.0	
BLOOM-7B1	25.7	25.3	26.2	26.1	26.6	28.1	29.1	
PolyLM-13B	27.5	24.5	26.3	27.4	26.4	25.3	26.8	
LLaMAX2-7B	27.0	26.9	27.0	26.6	26.2	26.8	26.1	
FuxiTranyu-8B	27.1	26.8	27.7	26.0	26.3	26.3	26.0	
Instruction-tuned Model								
Llama-2-Chat-7B	<u>39.7</u>	<u>40.2</u>	<u>36.8</u>	<u>33.7</u>	27.0	32.7	<u>35.2</u>	
Mistral-7B-Instruct-v0.1	42.5	43.4	41.6	37.8	<u>27.7</u>	34.0	40.1	
BLOOMZ-7B1	25.8	22.8	25.4	26.3	26.7	26.3	27.2	
PolyLM-MultiAlpaca-13B	25.9	26.2	26.2	25.5	25.5	25.7	26.1	
LLaMAX2-7B-Alpaca	37.5	35.7	32.6	33.0	28.4	<u>33.6</u>	33.4	
FuxiTranyu-8B-SFT	26.2	25.9	27.9	26.6	27.0	26.4	26.8	
FuxiTranyu-8B-DPO	28.6	27.1	28.2	26.7	26.8	26.7	28.0	

Table 9: Performance of FuxiTranyu-8B models compared to Llama-2-7B, Mistral-7B-v0.1, BLOOM-7B, PolyLM-13B, and LLaMAX2-7B models on multilingual MMLU (5-shot).

Models	fr	pt	zh	en	ru	jp
Base						
Llama-2-7B	81.9	74.9	74.4	<u>90.4</u>	<u>72.1</u>	74.0
Mistral-7B-v0.1	81.9	80.6	80.0	90.6	72.4	77.5
BLOOM-7B1	71.1	76.8	74.4	82.2	56.8	58.5
PolyLM-13B	73.5	74.9	76.6	84.6	65.1	65.7
LLaMAX-7B	77.1	76.8	75.4	87.8	69.8	<u>74.4</u>
FuxiTranyu-8B	<u>78.3</u>	<u>77.2</u>	<u>76.8</u>	85.4	66.4	72.4
Instruction-tuned Model						
Llama-2-Chat-7B	<u>79.5</u>	71.9	62.9	<u>88.3</u>	67.6	<u>70.7</u>
Mistral-7B-Instruct-v0.1	77.1	71.5	<u>74.0</u>	89.8	<u>70.5</u>	<u>67.5</u>
BLOOMZ-7B1	68.7	65.4	71.0	83.5	53.7	56.4
PolyLM-MultiAlpaca-13B	71.1	72.2	73.6	83.9	67.9	65.2
LLaMAX-7B-Alpaca	81.9	76.8	72.2	<u>88.3</u>	71.8	73.7
FuxiTranyu-8B-SFT	75.9	<u>76.4</u>	75.2	83.7	68.3	68.7
FuxiTranyu-8B-DPO	77.1	67.3	66.7	73.9	62.9	66.5

Table 10: Performance of FuxiTranyu-8B models compared to Llama-2-7B, Mistral-7B-v0.1, BLOOM-7B1, PolyLM-13B, and LLaMAX2-7B models on XWinograd (5-shot).

Models	et	ht	it	id	qu	sw	ta	th	tr	vi	zh
Base											
Llama-2-7B	48.6	50.6	<u>65.8</u>	62.4	51.4	52.2	53.4	56.4	54.8	63.0	65.0
Mistral-7B-v0.1	47.0	<u>51.4</u>	<u>65.8</u>	58.2	48.6	51.2	53.8	57.0	56.8	58.8	65.2
BLOOM-7B1	48.2	50.8	52.8	<u>69.8</u>	<u>50.8</u>	51.6	<u>59.2</u>	55.4	51.2	70.8	65.2
PolyLM-13B	49.8	50.4	66.0	70.2	50.4	51.8	55.0	58.6	<u>57.8</u>	<u>70.8</u>	67.0
LLaMAX-7B	<u>49.2</u>	52.6	52.6	53.8	51.4	<u>54.0</u>	58.0	57.2	53.0	53.0	63.4
FuxiTranyu-8B	<u>49.2</u>	51.2	71.4	69.6	49.6	55.4	60.0	<u>58.0</u>	62.4	72.8	<u>65.8</u>
Instruction-tuned Model											
Llama-2-Chat-7B	47.8	51.4	67.0	62.4	50.8	52.2	50.6	54.8	55.6	61.6	61.2
Mistral-7B-Instruct-v0.1	48.2	51.2	65.4	54.0	49.2	<u>54.6</u>	55.2	53.2	52.2	53.2	63.4
BLOOMZ-7B1	49.2	51.4	51.8	58.2	<u>52.2</u>	53.2	54.6	54.4	53.0	55.8	52.8
PolyLM-MultiAlpaca-13B	47.8	50.4	65.0	70.0	51.0	52.4	55.6	<u>59.0</u>	59.8	73.4	74.8
LLaMAX-7B-Alpaca	51.2	54.2	61.0	57.2	52.4	55.0	57.0	56.4	55.4	55.4	67.6
FuxiTranyu-8B-SFT	49.4	<u>51.8</u>	<u>71.6</u>	66.8	50.6	53.0	<u>62.0</u>	60.8	<u>63.6</u>	<u>73.6</u>	69.6
FuxiTranyu-8B-DPO	<u>49.6</u>	51.2	75.6	<u>69.2</u>	48.6	52.6	63.0	<u>59.0</u>	65.6	74.6	<u>70.6</u>

Table 11: Performance of FuxiTranyu-8B models compared to Llama-2-7B, Mistral-7B-v0.1, BLOOM-7B1, PolyLM-13B, and LLaMAX2-7B models on XCOPIA (0-shot).

Models	ar	es	eu	hi	id	my	ru	sw	te	zh
Base										
Llama-2-7B	49.6	<u>67.4</u>	50.4	53.7	59.3	48.1	62.9	50.5	54.3	59.5
Mistral-7B-v0.1	53.1	69.0	51.2	55.4	59.2	48.7	<u>66.7</u>	51.6	54.1	63.3
BLOOM-7B1	58.6	66.1	57.2	60.6	64.5	49.0	52.7	<u>53.9</u>	57.4	61.9
PolyLM-13B	56.5	65.6	51.6	48.8	<u>63.9</u>	47.3	64.1	<u>49.3</u>	<u>53.7</u>	63.3
LLaMAX2-7B	<u>58.8</u>	65.3	<u>54.5</u>	58.2	60.6	<u>52.2</u>	61.2	57.2	59.3	60.8
FuxiTranyu-8B	59.2	66.1	52.1	<u>59.4</u>	63.8	56.9	67.6	49.0	52.5	<u>62.1</u>
Instruction-tuned Model										
Llama-2-Chat-7B	50.1	<u>67.1</u>	51.0	54.4	60.2	48.8	65.3	<u>52.1</u>	53.7	62.4
Mistral-7B-Instruct-v0.1	47.1	<u>63.3</u>	50.0	49.8	52.3	47.6	62.3	<u>49.6</u>	51.8	59.7
BLOOMZ-7B1	47.9	51.0	48.6	50.8	51.0	47.4	46.9	50.4	<u>54.0</u>	50.0
PolyLM-MultiAlpaca-13B	57.2	66.0	51.2	49.0	<u>65.3</u>	47.2	<u>65.5</u>	48.4	53.1	66.8
LLaMAX2-7B-Alpaca	60.4	70.6	54.8	62.1	66.5	53.8	67.4	60.1	59.3	<u>65.3</u>
FuxiTranyu-8B-SFT	57.6	65.4	<u>51.4</u>	56.8	61.5	54.7	63.7	49.8	53.3	59.4
FuxiTranyu-8B-DPO	<u>60.2</u>	64.9	49.8	<u>58.1</u>	62.3	<u>54.6</u>	63.7	49.0	52.2	61.0

Table 12: Performance of FuxiTranyu-8B models compared to Llama-2-7B, Mistral-7B-v0.1, BLOOM-7B1, PolyLM-13B, and LLaMAX2-7B models on XStoryCloze (0-shot).

Models	ar	en	es	fr	gu	hi	id	mr	pt	ru	sr	ta	uk	vi	zh
Llama-2-Chat-7B	0.5	<u>11.0</u>	11.0	9.8	0.5	0.2	6.1	0.2	8.9	2.8	3.2	0.8	<u>2.3</u>	10.1	<u>1.0</u>
Mistral-7B-Instruct-v0.1	0.1	<u>11.0</u>	3.0	3.4	0.3	0.2	3.1	0.6	3.2	0.4	2.1	0.2	0.3	4.6	0.6
BLOOMZ-7B1	0.3	7.6	<u>13.7</u>	<u>13.1</u>	0.4	0.0	1.2	0.0	<u>13.1</u>	0	1.7	0.0	0.0	15.4	0.0
LLaMAX2-7B-Alpaca	0.0	1.7	0.5	0.7	0.0	0.0	0.3	0.0	0.2	0.0	0.5	0.1	0.1	0.2	0.0
FuxiTranyu-8B-SFT	<u>1.9</u>	11.8	16.3	16.6	<u>0.7</u>	<u>1.6</u>	17.8	<u>2.1</u>	17.5	<u>6.4</u>	6.1	<u>1.3</u>	5.3	27.7	5.6
FuxiTranyu-8B-DPO	2.8	9.5	11.1	11.0	0.9	2.4	<u>10.7</u>	3.2	12.3	6.5	<u>4.0</u>	2.8	5.3	<u>18.3</u>	5.6

Table 13: Performance of FuxiTranyu-8B models compared to Llama-2-7B, Mistral-7B-v0.1, BLOOM-7B1, and LLaMAX2-7B models on XL-Sum (0-shot).

Models	WMT16 (EN-RO)		WMT16 (RO-EN)		WMT16 (EN-DE)		WMT16 (DE-EN)	
	BLEU	CHRF	BLEU	CHRF	BLEU	CHRF	BLEU	CHRF
Llama-2-Chat-7B	17.18	44.20	31.43	58.00	20.01	48.31	35.41	60.78
Mistral-7B-Instruct-v0.1	13.66	41.47	24.58	53.04	19.41	49.25	30.19	58.27
BLOOMZ-7B1	1.88	20.09	11.35	36.22	3.76	23.27	22.30	46.69
LLaMAX2-7B-Alpaca	24.52	51.94	36.02	<u>60.85</u>	<u>26.31</u>	53.95	37.05	61.90
FuxiTranyu-8B-SFT	25.64	53.07	<u>34.96</u>	61.33	27.03	56.4	<u>35.91</u>	61.55
FuxiTranyu-8B-DPO	<u>24.8</u>	<u>53.06</u>	<u>32.9</u>	59.97	25.57	56.42	33.52	<u>60.43</u>
Models	WMT14 (EN-FR)		WMT14 (FR-EN)		IWSLT2017-AR-EN		IWSLT2017-EN-AR	
	BLEU	CHRF	BLEU	CHRF	BLEU	CHRF	BLEU	CHRF
Llama-2-Chat-7B	24.97	52.34	34.49	60.89	12.51	36.18	1.15	17.73
Mistral-7B-Instruct-v0.1	24.24	52.08	31.40	59.50	9.13	32.64	0.31	13.31
BLOOMZ-7B1	17.73	41.02	31.07	56.03	25.25	47.64	4.58	25.05
LLaMAX2-7B-Alpaca	32.86	59.53	36.00	61.64	29.76	<u>52.68</u>	10.47	40.27
FuxiTranyu-8B-SFT	<u>32.82</u>	<u>59.57</u>	34.07	61.1	<u>28.83</u>	52.79	<u>7.15</u>	<u>31.14</u>
FuxiTranyu-8B-DPO	31.98	59.64	32.27	<u>60.19</u>	<u>27.05</u>	51.5	6.5	29.41

Table 14: Performance of FuxiTranyu-8B models compared to Llama-2-7B, Mistral-7B-v0.1, BLOOM-7B1, and LLaMAX2-7B models on WMT14, WMT16, and IWSLT2017 (0-shot).