ProxyLM: Predicting Language Model Performance on Multilingual Tasks via Proxy Models

David Anugraha[1], **Genta Indra Winata**[2*], **Chenyue Li**[1],
Patrick Amadeus Irawan[3], **En-Shiun Annie Lee**[1,4]
[1]University of Toronto [2]Capital One
[3]Institut Teknologi Bandung [4]Ontario Tech University
anugraha@cs.toronto.edu, genta.winata@capitalone.com,
{jimschenchen,patrickamadeusirawan}@gmail.com, annie.lee@ontariotechu.ca

Abstract

Performance prediction is a method to estimate the performance of Language Models (LMs) on various Natural Language Processing (NLP) tasks, mitigating computational costs associated with model capacity and data for fine-tuning. Our paper presents PROXYLM, a scalable task- and language-agnostic framework designed to predict the performance of LMs using proxy models. These proxy models act as surrogates, approximating the performance of the LM of interest. By leveraging these proxy models, PROXYLM significantly reduces computational overhead in task evaluations, achieving up to a 37.08× speedup over traditional methods, even with our smallest proxy models. Our results across multiple multilingual NLP tasks and various robustness tests demonstrate that PROXYLM not only adapts well to previously unseen languages in pre-trained LMs, but also generalizes effectively across different datasets, outperforming the state-of-the-art by at least 1.78× in terms of root-mean-square error (RMSE).

1 Introduction

Language Models (LMs) have become increasingly valuable for assessing Natural Language Processing (NLP) tasks (Raffel et al., 2020; Brown et al., 2020; Costa-jussà et al., 2022; Touvron et al., 2023a,b; Workshop et al., 2022). However, fine-tuning and evaluating these models are resource-intensive processes in terms of both computation and time. These costs escalate with model size, especially when experimenting across multiple datasets. As highlighted in Kaplan et al. (2020), there is a scaling law that applies to both model and dataset sizes, and computational demands, indicating that larger models and broader datasets require increased computational resources. Modeling low-resource languages (LRLs) in multilingual contexts presents a range of challenges. One significant challenge is the limited data availability, which hampers effective fine-tuning processes (Gu et al., 2018; Adilazuarda et al., 2024), making model adaptation through fine-tuning a challenging task (Zoph et al., 2016; Liu et al., 2021). Another critical issue is the lack of pre-training data for numerous regional languages, such as Southeast Asian languages (Winata et al., 2022, 2024b; Yong et al., 2024), with many languages being omitted during the pre-training phase of multilingual LMs.

Given the limited academic computational resources for LM fine-tuning and inadequate datasets for LRLs, performance prediction is an efficient method to alleviate the dependency on extensive resources by leveraging past performance records on an NLP task. While linear regression and gradient-boosting hold promise in performance prediction (Birch et al., 2008; Srinivasan et al., 2021; Xia et al., 2020; Ye et al., 2021; Schram et al., 2023; Khiu et al., 2024), existing solutions primarily focus on homogeneous data settings and prioritize high-resource languages using Transformer models (Vaswani et al., 2017). Khiu et al. (2024) examine diverse datasets and LRLs but encounter limitations in the number of experiments, language diversity, and model scope, focusing solely on mBART (Liu et al., 2020). Recent advancements in larger multilingual models, like NLLB (Costa-jussà et al., 2022) and M2M100 (Fan et al., 2021), have significantly improved machine translation capabilities, exceeding those of mBART and other LMs (Zhu et al., 2023).

In this paper, we propose PROXYLM,[1] a scalable task- and language- agnostic framework to predict LM performance by utilizing proxy models. Proxy models are defined as substitute models, wherein the performance of these substitute models are used

[*] The work was conducted outside Capital One.

[1] We release our code at https://github.com/davidanugraha/proxylm.

Findings of the Association for Computational Linguistics: NAACL 2025, pages 1981–2011
April 29 - May 4, 2025

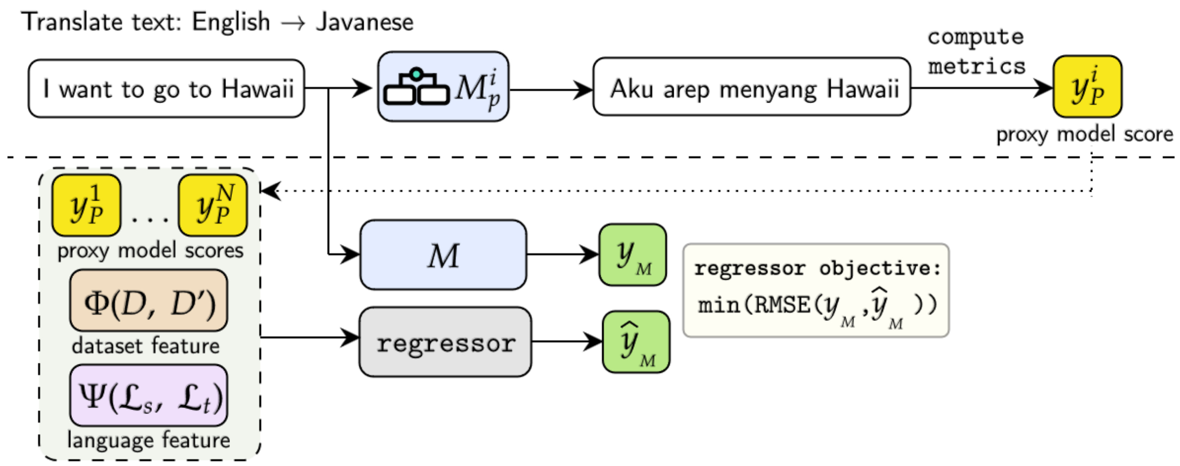


Figure 1: PROXYLM framework for LM performance prediction. **(Top)** The evaluation metric is computed on the test set using **a proxy model** $\mathcal{M}_p^i$. **(Bottom)** The regressor g is trained using proxy model scores as well as dataset and language features by minimizing the RMSE difference of $y_{\mathcal{M}}$ and $\hat{y}_{\mathcal{M}}$.

to estimate the performance of another LM. This other model can be significantly larger than our proxy models. For optimizing the prediction, we utilize much smaller LMs as proxy models and off-the-shelf models without further tuning. This approach is very scalable to multiple proxy models and task-agnostic to any modalities, thus it can be applied to any downstream tasks. This study focuses on three multilingual tasks, covering machine translation (MT), intent classification, and slot filling. Our approach outperforms the existing work from Xia et al. (2020); Ye et al. (2021); Schram et al. (2023); Khiu et al. (2024), which opens a new avenue to employ LMs for model performance prediction. Therefore the contribution of our paper can be summarized in three-fold:

1. We introduce PROXYLM, an efficient and scalable task- and language-agnostic framework designed to predict the performance of LMs. This framework significantly reduces the computational costs associated with fine-tuning and inference during model selection.

2. We demonstrate the effectiveness and robustness of PROXYLM across 34 dataset sources and 56 languages for MT, and 51 languages for intent classification and slot filling, each on two estimated LMs. Our framework substantially outperforms all existing baselines in all NLP tasks, datasets, and settings, including scenarios involving extremely LRLs that remain unseen by pre-trained LMs and across different datasets, surpassing the state-of-the-art performance measured with root-mean-square error (RMSE) by at least 1.78×.

3. We also provide a time analysis comparing the deployment of proxy models for performance prediction to direct LM fine-tuning. Our results indicate that, with our smallest proxy models, we can achieve up to a 37.08× speedup on task evaluation compared to the traditional approach, highlighting the efficiency of our approach.

2 Methodology

In this section, we formally define the LM performance prediction problem and our proposal to improve performance prediction.

2.1 PROXYLM

Recall that performance prediction is a task of estimating a system's performance based on the model and its training strategy, training and test dataset, and language used. Formally, let LM $\mathcal{M}$ be our estimated model. $\mathcal{M}$ is trained over a training dataset $\mathcal{D}$ with source language $\mathcal{L}_s$ and target language $\mathcal{L}_t$, and then tested using dataset $\mathcal{D}'$. $\mathcal{M}$'s performance, denoted $y_{\mathcal{M}}$, can be formulated under function f that relates between these variables:

$$y_{\mathcal{M}} = f(\mathcal{M}, \mathcal{D}, \mathcal{D}', \mathcal{L}_s, \mathcal{L}_t). \quad (1)$$

We can approximate f by transforming Equation 1 into a regression task with a regressor function g, which will be trained on past performance records. Previous works (Xia et al., 2020; Ye et al., 2021;

Schram et al., 2023; Khiu et al., 2024) formulate regressor that takes dataset features $\Phi(\mathcal{D}, \mathcal{D}')$ to identify the characteristics of the training and test datasets, as well as the distribution shift between them. It also takes language features $\Psi(\mathcal{L}_s, \mathcal{L}_t)$ to measure the similarities between the source and target languages. This can be formulated as follows:

$$\hat{y_{\mathcal{M}}} = g(\Phi(\mathcal{D}, \mathcal{D}'); \Psi(\mathcal{L}_s, \mathcal{L}_t)). \tag{2}$$

We present PROXYLM, a framework that leverages the past performance of other models, referred to as proxy models, as additional context for our regressor. Intuitively, proxy models can provide valuable insights that assist in predicting the performance of the estimated model $\mathcal{M}$, which addresses the gap in previous works for not accounting $\mathcal{M}$. Formally, let $\mathcal{M}_p = [\mathcal{M}_p^1, \ldots, \mathcal{M}_p^N]$ be a set of N proxy models. To integrate the information from these proxy models, we modify Equation 2 as follows:

$$\hat{y_{\mathcal{M}}} = g(\hat{y}_{\mathcal{M}_p}; \Phi(\mathcal{D}, \mathcal{D}'); \Psi(\mathcal{L}_s, \mathcal{L}_t)), \tag{3}$$

where $y_{\mathcal{M}_p} = [y_{\mathcal{M}_p^1}, \ldots, y_{\mathcal{M}_p^N}]$ represents the performance records of N proxy models. The advantage of using proxy models arises from their faster fine-tuning and evaluation compared to the estimated model $\mathcal{M}$. This also means that off-the-shelf models can be used directly without additional tuning if they already perform the task adequately, further enhancing efficiency.

2.2 PROXYLM Features

Language Features. We use URIEL Typological Database (Littell et al., 2017) similar to Xia et al. (2020) including geographic, genetic, inventory, syntactic, phonological, and featural distance. The language features are useful to provide a language-specific representation to the regressor.

Dataset Features. We extract six features from the dataset, including train size, vocab size, average sentence length, word overlap, Type-Token Ratio (TTR), and TTR distance from $\mathcal{D}$ and $\mathcal{D}'$ based on Xia et al. (2020). We will refer to these features and language features combined as NLPerf features. Furthermore, we incorporate the distribution shift information between the training and test datasets using Jensen-Shannon Divergence (JSD) as described by Khiu et al. (2024). In addition, we include the cosine similarity of term frequency-inverse document frequency (TF-IDF) representations and sentence embeddings using Sentence-BERT (Reimers and Gurevych, 2019). Details on how these features are computed can be found in Appendix Section A.

Proxy Models Features. We leverage the performance data from proxy models, derived by averaging results from multiple fine-tuning and evaluation iterations on identical datasets and languages. Moreover, we retain the flexibility to adjust the number of proxy models employed, facilitating efficient and adaptable performance estimation.

3 Experimental Setup

In this section, we describe the datasets and LMs used to obtain LMs' performance records. These records are then used to train various regressor models under different experimental settings to investigate our approach to performance predictions. The details of the hyper-parameters for both the LMs and the regressors are provided in B.3.

3.1 Datasets

Machine Translation. We use two types of datasets: English-centric and Many-to-Many Languages. The English-centric dataset involves English serving as either the source or target language. Our English-centric dataset comes from the MT560 (Gowda et al., 2021) dataset, where we curate 32 datasets and select 50 languages out of 500 for evaluation. Furthermore, we use the FLoRes-200 dataset (Costa-jussà et al., 2022) for additional validation and test sets. These datasets consist of translations with varying quality across diverse domains, presenting significant challenges for a robust performance prediction. In contrast, the Many-to-Many Languages dataset allows any language to act as the source or target. We use the NusaTranslation dataset (Cahyawijaya et al., 2023) as our Many-to-Many Languages dataset, which comprises parallel texts in 12 Indonesian regional languages. As many of these languages are absent in pre-trained multilingual models, we analyze 8 out of the 12 languages due to limited data in the remaining 4. Both datasets encompass 56 languages across various domains such as economics, technology, and medicine. Detailed language insights are available in the Appendix Section B.1.

Intent Classification and Slot Filling. We use MASSIVE (FitzGerald et al., 2022) as our dataset encompassing 51 languages. We fine-tune each LM on all languages and evaluate it on all languages. Detailed language insights are available in the Appendix Section B.1.

3.2 Estimated LMs

Machine Translation. We employ two estimated LMs: M2M100 1.2B (Fan et al., 2021) and NLLB 1.3B (Costa-jussà et al., 2022). Each estimated model is fine-tuned using a standard next-token prediction objective on the training set.

Intent Classification and Slot Filling. We employ two decoder-only estimated LMs: LLaMA-3 Instruct (8B) (Dubey et al., 2024) and Aya-23 (8B) (Aryabumi et al., 2024). Both models are fine-tuned using supervised fine-tuning (SFT) combined with Low-Rank Adaptation (LoRA) (Hu et al., 2021) on the training set.

3.3 Proxy Models

Machine Translation. We utilize four different transformer-based models: an encoder-decoder random initialized Transformers (100M) (Vaswani et al., 2017), SMaLL-100 (330M) (Mohammadshahi et al., 2022), M2M100 (Fan et al., 2021), and NLLB (Costa-jussà et al., 2022). For M2M100 and NLLB, we use the models without any additional tuning **(No FT)** in a zero-shot fashion. For simplicity, the term "fine-tuning" will be used throughout this paper to refer to both the process of training from scratch (as in the case of the Transformer (100M) model) and the process of fine-tuning pre-trained LMs. Model details are provided in the Appendix Section B.2. The evaluation is primarily conducted using SentencePiece BLEU (spBLEU) (Goyal et al., 2022), which has proven to be a reliable metric in multilingual and LRLs. For a more comprehensive assessment, we also use COMET-22 (Rei et al., 2022), as it shows a high correlation with human judgments (Freitag et al., 2023). However, COMET-22 is applied to only a subset of the English-centric dataset due to its limited language coverage, especially to LRLs.

Intent Classification and Slot Filling. We utilize three decoder-only models: SmolLM (135M and 360M) (Allal et al., 2024) and BLOOMZ (560M) (Muennighoff et al., 2022). Model details are provided in the Appendix Section B.2. The evaluation is done using accuracy for intent classification and micro-F1 for slot filling.

3.4 Regressor Models

We utilize XGBoost (Chen and Guestrin, 2016), LGBM (Ye et al., 2021), Poly2 (Khiu et al., 2024), and Poly3 (Khiu et al., 2024) as our regressors. In most of our experiments, we apply XGBoost as our default regressor because we find it to be the best-performing model based on the cross-validation during training, while the other regressors serve as baselines. Specifically for MT in the Many-to-Many Languages setting, Matrix Factorization with context features (MF) is used as an additional baseline (Schram et al., 2023). We do not apply MF to our English-centric setting because MF requires the performance records to be structured in two dimensions—one for the source language and one for the target language. In the English-centric setting, this would result in a sparse matrix with only one fully populated row or column, corresponding to English, making MF impractical for this setup.

3.5 Experimental Settings

Each regressor is evaluated using RMSE as our performance metric and evaluated 5 times. For all tasks, we set our experiment settings as follows:

- **Random**: We randomly sample the performance records into training and test sets with a ratio of 7:3. Then, we run 10-fold cross-validation on the training set to find the best hyper-parameters for each regressor. The best-performing regressor would subsequently be evaluated on the test set.
- **Leave-One-Language-Out (LOLO)**: We select one language as the test set, which is not encountered during training.

To test the robustness of ProxyLM, we also provide two additional setups specifically for MT:

- **Unseen**: The performance records for MT can be divided into two categories: (1) records with "seen" languages and (2) records with "unseen" languages. "Unseen" languages refer to languages that are not present in the pre-training LM data, while "seen" languages denote those that are present. In this setting, the regressor is trained using records of "seen" languages and tested using records of "unseen" languages.
- **Cross-Dataset**: The regressor can be trained using performance records from the English-centric dataset and tested using the Many-to-Many Languages dataset. We opt not to reverse this setup as the Many-to-Many dataset exhibits no domain shift and contains fewer performance records.

Models	English-centric					Many-to-Many				
	Random		LOLO		Avg.	Random		LOLO		Avg.
	M2M100↓	NLLB↓	M2M100↓	NLLB↓		M2M100↓	NLLB↓	M2M100↓	NLLB↓	
NLPerf (Xia et al., 2020) with different regressors										
XGBoost	7.69	7.73	9.20	12.92	9.39	2.45	**1.11**	7.83	8.28	4.94
Poly2 (Khiu et al., 2024)	11.21	16.23	15.55	43.02	21.50	4.70	4.68	7.07	7.90	6.09
Poly3 (Khiu et al., 2024)	11.00	15.64	62.29	236.29	81.31	4.60	4.64	7.26	8.01	6.13
LGBM (Ye et al., 2021)	7.88	8.15	9.71	12.81	9.64	3.65	2.60	7.08	7.14	5.12
ProxyLM (Ours)‡ with different proxy models										
Transformer	4.68	7.22	6.18	11.78	7.47	2.56	1.70	5.65	6.24	4.04
SMaLL-100	4.07	6.33	4.59	10.33	6.33	2.56	1.65	4.85	5.14	3.55
SMaLL-100 (No FT)	5.27	6.04	6.28	10.94	7.13	2.44	1.34	6.93	7.25	4.49
Estimated Model (No FT)	5.23	4.15	6.18	5.42	5.25	**2.38**	1.27	5.10	5.50	3.56
Ensemble†	**3.21**	**3.68**	**3.74**	**4.94**	**3.89**	2.41	1.56	**3.73**	**3.79**	**2.90**

Table 1: MT test results on English-centric and Many-to-Many Languages datasets using spBLEU in average RMSE (**lower is better**). **Bold** numbers indicate the best performance, while underlined numbers represent the second-best performance. The columns show the setting and estimated model. "No FT" denotes "no fine-tuning" and the model inference is done in a zero-shot fashion. Avg represents the average of the results across the row for each respective dataset. ‡The reported results use XGBoost as the regressor. †Ensemble denotes combining all four proxy models, the detailed breakdown of this result with the standard deviation can be seen in the Appendix Section C.

4 Results and Analysis

In this section, we present the results of the performance predictions for ProxyLM and baselines under the specified settings. Further, we discuss the robustness, effectiveness, and efficiency of ProxyLM in the context of performance prediction.

4.1 Machine Translation

4.1.1 English-centric Results

Table 1 shows the overall results on the English-centric dataset using spBLEU. ProxyLM remarkably outperforms all existing baselines. We find that incorporating all proxy models (Ensemble) is the most effective for prediction, leading to a 2.41× averaged reduction in RMSE across all experimental settings compared to the best baseline. Note that the significant improvement remains consistent when evaluated using a different metric, such as COMET-22, which yields a 2.00× averaged reduction in RMSE across experimental settings compared to the best baseline, as shown in Table 14 in Appendix Section A. We observe that using the **No FT** estimated model to predict the performance of their fine-tuned models is surprisingly useful in all settings, especially for NLLB, where the model already has decent machine translation quality on LRLs. This observation is supported by our findings within the XGBoost model that the NLLB No FT feature has the highest importance score among all features, as shown in Figure 9 in the Appendix. Furthermore, using SMaLL-100 fine-tuned performance provides useful estimations for settings involving M2M100 as the estimated model. This may indicate that the performance of a model with similar architecture can be a good estimator for the performance of the larger estimated model. In other words, the choice of proxy model to help prediction matters. Feature importance analysis from the XGBoost model supports this, revealing that the SMaLL-100 fine-tuned feature has the highest importance score among all features, as shown in Figure 8 in the Appendix.

4.1.2 Many-to-Many Languages Results

Table 1 presents the performance of different models on the Many-to-Many Languages dataset. The results reveal that the Ensemble model achieves the lowest RMSE, with a 1.70× averaged reduction in RMSE across all experimental settings compared to the best baseline, indicating superior accuracy in performance predictions. An exception occurs in the random NLLB setting, where the model utilizing only NLPerf features outperforms the ensemble model, achieving the best performance. Note that no domain shift occurs within the dataset.

A comparative analysis shows that predicting the performance of the M2M100 model in the random setting presents a greater challenge compared to predicting the NLLB model. This discrepancy suggests that the complexity of performance prediction can vary substantially depending on the specific LM and the conditions under which it is evaluated. A particularly noteworthy finding is the effectiveness of using No FT models for estimating LM performance. The No FT models, which do not re-

Models	Intent Classification					Slot Filling				
	Random		LOLO		Avg.	Random		LOLO		Avg.
	Aya23↓	LLaMA3↓	Aya23↓	LLaMA3↓		Aya23↓	LLaMA3↓	Aya23↓	LLaMA3↓	
NLPerf (Xia et al., 2020) with different regressors										
XGBoost	0.0761	0.0191	0.1573	0.0581	0.0777	0.0693	0.0548	0.1219	0.1093	0.0888
Poly2 (Khiu et al., 2024)	0.1996	0.0979	0.2075	0.0918	0.1492	0.1396	0.1412	0.1418	0.1414	0.1410
Poly3 (Khiu et al., 2024)	0.1990	0.0969	0.2191	0.0925	0.1519	0.1393	0.1401	0.1448	0.1413	0.1414
LGBM (Ye et al., 2021)	0.0839	0.0198	0.1545	0.0558	0.0785	0.0692	0.0557	0.1218	0.1152	0.0905
PROXYLM (Ours)‡ with different proxy models										
SmolLM (135M)	0.0676	0.0171	0.1273	0.0455	0.0644	0.0618	0.0538	0.1004	0.0953	0.0778
SmolLM (360M)	**0.0604**	**0.0157**	<u>0.1118</u>	**0.0441**	**0.0580**	**0.0562**	**0.0506**	<u>0.0844</u>	**0.0868**	**0.0695**
BLOOMZ (560M)	0.0692	0.0179	0.1283	0.0482	0.0659	0.0618	0.0540	0.1023	0.0995	0.0794
Ensemble†	<u>0.0609</u>	<u>0.0164</u>	**0.1112**	<u>0.0442</u>	<u>0.0582</u>	<u>0.0561</u>	<u>0.0508</u>	**0.0830**	<u>0.0884</u>	<u>0.0696</u>

Table 2: Intent classification and slot filling results using accuracy and micro-F1 score, respectively, with average RMSE (**lower is better**). **Bold** numbers indicate the best performance, while <u>underlined</u> numbers represent the second-best performance. Avg represents the average of the results across the row for each respective task. ‡The reported results use XGBoost as the regressor for both intent classification and slot filling. †Ensemble denotes combining all three proxy models. The detailed breakdown of this result with the standard deviation can be seen in the Appendix Section C.

quire any additional fine-tuning, demonstrate high accuracy in their performance predictions. This method offers substantial efficiency benefits, as it eliminates the need for extensive computational resources typically required for model training. In contrast, we find similar results between the LOLO setting for Many-to-Many Languages and English-centric results, where PROXYLM using Ensemble remarkably outperforms all existing baselines. In addition, we find that using SMaLL-100 fine-tuned performance results in better predictions compared to those of the No FT estimated model.

4.2 Intent Classification and Slot Filling

Table 2 presents the overall results for both intent classification and slot filling tasks. In these experiments, PROXYLM employs XGBoost as the regressor for intent classification and LGBM for slot filling, as LGBM exhibited superior performance during training cross-validation for the slot filling task. Consistent with our findings in MT, PROXYLM outperforms all existing baselines in both tasks. Notably, the SmolLM (360M) model emerges as the most effective proxy model, achieving an average RMSE reduction of 1.34× in intent classification and 1.28× in slot filling across all experimental settings when compared to the best baseline. As with MT, the choice of proxy model significantly impacts prediction performance. Based on the feature importance analysis in Figure 10-13 in the Appendix, SmolLM (360M) exhibits the highest importance score among all proxy models in both tasks. This suggests that the size of the proxy models may not be a reliable indicator of their effectiveness.

5 Ablation Study

5.1 Robustness of PROXYLM

Figure 2 illustrates the performance of our models in both the Unseen and Cross-Dataset setups, highlighting the robustness results achieved by PROXYLM. For the Cross-Dataset evaluation, we opted for LGBM instead of XGBoost, as it demonstrated better performance during cross-validation on the training set. PROXYLM with Ensemble shows a significant reduction in RMSE compared to the best baseline: a 1.84× reduction in the Unseen setup, and reductions of 2.15× and 1.78× for M2M100 and NLLB in the Cross-Dataset scenario, respectively. This consistent performance across datasets and languages—including those not encountered during the regressor's training, such as unseen languages for the pre-trained LMs—emphasizes the model's generalization capabilities. We also observe better performance for M2M100 compared to NLLB, which may be attributed to NLLB's reliance on an English-centric dataset containing only seen languages, lacking examples of unseen languages for the regressor. This may indicate the importance of including instances of unseen languages in the regressor training dataset for achieving more robust predictions.

Figure 3 shows the impact of features used in PROXYLM for MT in the LOLO setting with XGBoost. Utilizing proxy models as features leads to a significant reduction in RMSE across all scenarios, showcasing their importance compared to

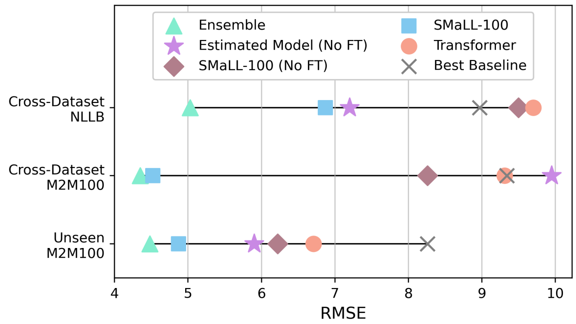


Figure 2: Unseen and Cross-Dataset MT test results on English-centric dataset in average RMSE (**lower is better**). We only show the best-performing baseline for comparison with PROXYLM with different proxy models. "No FT" denotes "no fine-tuning". We only show M2M100 results for the Unseen setting since NLLB covers all languages in the English-centric dataset. The reported results for the Unseen setting use XGBoost, while the Cross-Dataset experiments use LGBM. Ensemble denotes combining all four proxy models. The detailed breakdown of this result with the standard deviation can be seen in the Appendix Section C.

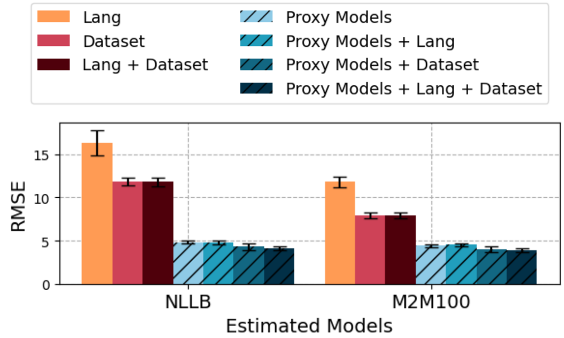
Datasets	Inference	Fine-tuning	
		English-centric	Many-to-Many Langs.
Estimated models			
M2M100	421 s	3.94 hrs (7.04×)	1.42 hrs (7.32×)
NLLB	737 s	12.08 hrs (21.57×)	7.21 hrs (37.08×)
Proxy models			
SMaLL-100	333 s	2.38 hrs (4.25×)	1.03 hrs (5.29×)
Transformer	231 s	0.56 hrs (1×)	0.19 hrs (1×)

Table 3: Comparison of LMs' inference time (in seconds) and fine-tuning time (in hours) for **one MT experimental run**. The multiplier of fine-tuning time is relative to the Transformer model. All times were calculated using the interquartile mean to ignore outliers.

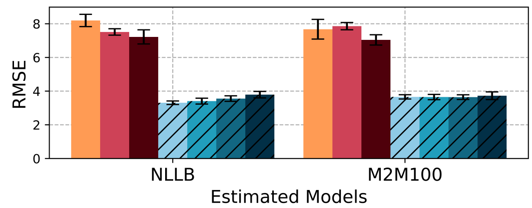
other features. For the English-centric dataset, including language and dataset features alongside proxy models enhances performance. Dataset features alone show better improvement than language features alone, but the combination of both yields the best performance. On the other hand, for the Many-to-Many Languages dataset, the benefits of incorporating dataset and language features are less pronounced, especially for the M2M100, and there may even be a performance dip for the NLLB due to the dataset's lack of domain shift.

5.2 Time Efficiency

Table 3 compares the fine-tuning and inference times required for the estimated and proxy models on MT, while Table 4 compares the fine-tuning and evaluation times required for the estimated and



(a) English-centric test results.



(b) Many-to-Many Languages test results.

Figure 3: Ablation study on the LOLO setting with XGBoost on English-centric and Many-to-Many Languages datasets. **Proxy Models** here indicates **Ensemble**, which is a combination of all proxy models. Proxy Models significantly reduce RMSE across all scenarios.

Datasets	Intent Classification	Slot Filling
Estimated models		
Aya-23	0.68 hrs (6.13×)	5.10 hrs (13.48×)
LLaMA3	0.54 hrs (4.91×)	4.36 hrs (11.53×)
Proxy models		
BLOOMZ-560M	0.12 hrs (1.06×)	0.59 hrs (1.57×)
SmolLM-360M	0.12 hrs (1.09×)	0.40 hrs (1.05×)
SmolLM-150M	0.11 hrs (1.00×)	0.38 hrs (1.00×)

Table 4: Comparison LMs' overall fine-tuning and evaluation time (in hours) for **one intent classification or slot filling experimental run**. The multiplier of the time is relative to SmolLM-135M model as it is the smallest proxy model. All times were calculated using the interquartile mean to ignore outliers.

proxy models on intent classification and slot filling. The results demonstrate that fine-tuning proxy models or direct inference from any model is remarkably faster than fine-tuning all estimated models. Table 5 further illustrates this point, showing only a minimal trade-off in the time needed to train the regressor models. This additional training time is relatively negligible, highlighting the efficiency of using proxy models.

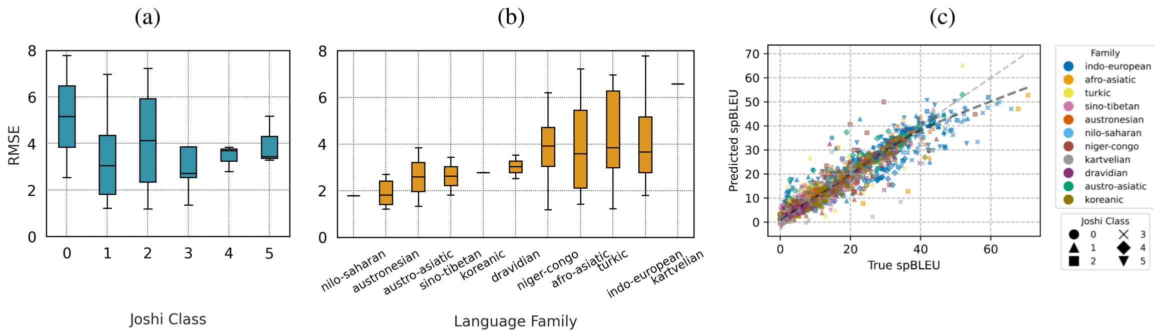


Figure 4: Detailed results of XGBoost with PROXYLM Ensemble on **M2M100** model under the **LOLO** setting on MT using the English-centric dataset from Table 1. The results are grouped by (a) Joshi Class and (b) language family that follows the mapping which is provided in Appendix B.1; (c) shows the scatter plot illustrating the correlation of spBLEU scores between the PROXYLM's prediction and estimated LM, with the light gray dashed line representing the line of equality ($y = x$) with $R^2 = 0.90$ and black dashed line representing Locally Weighted Scatterplot Smoothing (LOWESS) curve to represent the trend.

Regressors	English-centric (MT)	Many-to-Many (MT)	Intent and Slot
XGBoost	2.24 s	0.87 s	6.33 s
Poly2	0.07 s	0.06 s	0.09 s
Poly3	0.06 s	0.06 s	0.10 s
LGBM	90.79 s	27.51 s	118.24 s
MF	N/A	141.69 s	N/A

Table 5: Regressor models training time (in seconds) per one round of cross-validation with 10-folds across all setups. We combine the timing for intent classification and slot filling since they both contain the same amount of training data. All times were calculated using the interquartile mean to ignore outliers.

5.3 Performance by Language Categories

In Figure 4, we present detailed XGBoost results with PROXYLM Ensemble on the M2M100 model under the English-centric LOLO experiment, grouped by language categories. PROXYLM demonstrates relatively stable performance across languages belonging to different Joshi classes and linguistic families. Based on the Locally Weighted Scatterplot Smoothing (LOWESS) (Cleveland, 1979) curve depicted in Figure 4(c), our method consistently maintains unbiased predictions for spBLEU scores below 40 across various language types. However, as the spBLEU score increases, the availability of data points diminishes, leading to our method under-predicting the performance compared to the true spBLEU score. Outliers observed in Kartvelian languages and Indo-European languages with Joshi class 3 may have contributed to this discrepancy in prediction. These observations suggest that increasing the number of data points covering higher spBLEU scores may help mitigate the bias in prediction.

6 Related Work

The prediction performance of machine learning algorithms has been mainly explored in two research directions: (1) predict the model performance during the training runtime, and (2) predict the model performance by providing extracted features from the dataset (Xia et al., 2020).

Performance Prediction During the Training Runtime. The former aims to infer and extrapolate the learning curve to approximate training results using evaluation metric measurements (Kolachina et al., 2012). Domhan et al. (2015) study the quick detection of poor hyper-parameters in probabilistic models after a few steps of Stochastic Gradient Descent (SGD). Adriaensen et al. (2024) extrapolate learning curves from a parametric prior using Markov Chain Monte Carlo (MCMC).

Performance Prediction Using Extracted Features. The latter aims to predict the model performance by learning a correlation between input features and the final evaluation metric. Birch et al. (2008) identify strong predictive features such as the amount of reordering, the morphological complexity of the target language, and the historical relatedness of the two languages. Xia et al. (2020) leverage extracted dataset features and typological database language representations. Ye et al. (2021) introduce the use of confidence intervals and calibration with various regressor algorithms for reliable performance prediction. Schram et al. (2023) apply Bayesian matrix factorization for performance prediction on multilingual NLP tasks. In this work, we focus on exploring the latter. Existing approaches have shown promise using linear regression and gradient-boosting trees (Birch et al., 2008; Xia et al., 2020; Srinivasan et al., 2021; Ye et al., 2021). These studies have considered data size, typological features, and language similarity as factors contributing to the model performance.

Enhancing LLMs through Small Models. Recent studies, as compiled in Chen and Varoquaux (2024), have explored leveraging smaller models to complement LLMs across various tasks. A closely related application involves using smaller models to assist with LLM inference and evaluation (Kuhn et al., 2023; Manakul et al., 2023; Wang et al., 2024; Winata et al., 2024a). In contrast, our work focuses on employing proxy models within a language- and task-agnostic framework to predict LLM performance accurately. This approach offers a cost-effective alternative to fine-tuning and inference during model selection, while achieving high evaluation accuracy and robustness across extreme domain shifts. These include datasets spanning diverse domains, quality levels, and languages, with particular effectiveness demonstrated for extremely LRLs.

7 Conclusion

In this paper, we introduce PROXYLM, a novel framework designed to predict the performance of LMs by leveraging proxy models including for LRLs. By utilizing proxy models as substitutes to estimate the performance of the target model, we strategically employ smaller LMs or off-the-shelf models without additional fine-tuning. This framework is highly scalable to multiple proxy models and is task- and language-agnostic, making it applicable to a wide range of downstream NLP tasks. Our approach showcases substantial advancements in prediction accuracy compared to standard baselines and exhibits strong generalization capabilities across varied scenarios.

Limitations

This paper focuses on the empirical use of various proxy models without delving into the intricacies of the proxy model selection. Specifically, we do not investigate methods for determining which proxy models are the most effective across different contexts without relying on empirical experimentation. This limitation highlights a significant avenue for future research, which could involve a more systematic approach to identifying the most effective proxy models given estimated LMs and NLP tasks. Nevertheless, the use of proxy models consistently outperforms the absence of such models, demonstrating their importance in performance prediction.

Alternatively, developing robust methodologies for collecting and analyzing relevant past performance records could provide invaluable insights that enhance the generalization and accuracy of our predictive framework. Some performance data may offer greater information gain than others, potentially minimizing the number of performance records required to achieve a more robust and accurate predictor. By establishing a clearer understanding of which performance records yield the most informative insights, we can optimize our approach and improve our overall predictive capabilities. Future research in this area may further explore various data collection strategies and analytical techniques to develop a more comprehensive framework for selecting and utilizing proxy models effectively.

Ethical Considerations

We are committed to conducting our evaluations with the utmost standards of transparency and fairness. This commitment involves applying rigorous methodologies that ensure equitable and unbiased assessment processes.

Acknowledgements

We extend our sincere gratitude to Viktoria Schram for providing assistance in reproducing Matrix Factorization as our baseline.

References

Muhammad Farid Adilazuarda, Samuel Cahyawijaya, Alham Fikri Aji, Genta Indra Winata, and Ayu Purwarianti. 2024. Lingualchemy: Fusing typological and geographical elements for unseen language generalization. *arXiv preprint arXiv:2401.06034*.

Steven Adriaensen, Herilalaina Rakotoarison, Samuel Müller, and Frank Hutter. 2024. Efficient bayesian learning curve extrapolation using prior-data fitted networks. *Advances in Neural Information Processing Systems*, 36.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. 2024. Smollm - blazingly fast and remarkably powerful.

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*.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting success in machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 745–754.

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.

Samuel Cahyawijaya, Holy Lovenia, Fajri Koto, Dea Adhista, Emmanuel Dave, Sarah Oktavianti, Salsabil Akbar, Jhonson Lee, Nuur Shadieq, Tjeng Wawan Cenggoro, et al. 2023. Nusawrites: Constructing high-quality corpora for underrepresented and extremely low-resource languages. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 921–945.

Lihu Chen and Gaël Varoquaux. 2024. What is the role of small models in the llm era: A survey. *arXiv preprint arXiv:2409.06857*.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Language resources and evaluation*, 49:375–395.

William S Cleveland. 1979. Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.*, 74(368):829–836.

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

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. 2015. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Twenty-fourth international joint conference on artificial intelligence*.

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*.

Ondrej Dušek, Jan Hajic, Jaroslava Hlaváčová, Michal Novák, Pavel Pecina, Rudolf Rosa, Ales Tamchyna, Zdenka Urešová, and Daniel Zeman. 2014. Machine translation of medical texts in the khresmoi project. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 221–228.

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

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, et al. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv preprint arXiv:2204.08582*.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frédéric Blain, Daniel Deutsch, Craig Stewart, et al. 2023. Results of wmt23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, pages 578–628.

Thamme Gowda, Zhao Zhang, Chris Mattmann, and Jonathan May. 2021. Many-to-english machine translation tools, data, and pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 306–316.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

J Gu, H Hassan, J Devlin, and VOK Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistic.

Barry Haddow and Faheem Kirefu. 2020. Pmindia–a collection of parallel corpora of languages of india. *arXiv preprint arXiv:2001.09907*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*.

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

Eric Khiu, Hasti Toossi, David Anugraha, Jinyu Liu, Jiaxu Li, Juan Armando Parra Flores, Leandro Acros Roman, A Seza Doğruöz, and En-Shiun Annie Lee. 2024. Predicting machine translation performance on low-resource languages: The role of domain similarity. *arXiv preprint arXiv:2402.02633*.

Tom Kocmi, Rachel Bawden, Ondrej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popovic. 2022. Findings of the 2022 conference on machine translation (wmt22). In *Conference on Machine Translation*.

Prasanth Kolachina, Nicola Cancedda, Marc Dymetman, and Sriram Venkatapathy. 2012. Prediction of learning curves in machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22–30.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

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

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

Zihan Liu, Genta Indra Winata, and Pascale Fung. 2021. Continual mixed-language pre-training for extremely low-resource neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2706–2718.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Alireza Mohammadshahi, Vassilina Nikoulina, Alexandre Bérard, Caroline Brun, James Henderson, and Laurent Besacier. 2022. Small-100: Introducing shallow multilingual machine translation model for low-resource languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8348–8359.

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*.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535.

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. *Journal of machine learning research*, 21(140):1–67.

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022. Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Viktoria Schram, Daniel Beck, and Trevor Cohn. 2023. Performance prediction via bayesian matrix factorisation for multilingual natural language processing tasks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1790–1801.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361.

Anirudh Srinivasan, Sunayana Sitaram, Tanuja Ganu, Sandipan Dandapat, Kalika Bali, and Monojit Choudhury. 2021. Predicting the performance of multilingual nlp models. *arXiv preprint arXiv:2110.08875*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218.

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, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xidong Wang, Nuo Chen, Junyin Chen, Yan Hu, Yidong Wang, Xiangbo Wu, Anningzhe Gao, Xiang Wan, Haizhou Li, and Benyou Wang. 2024. Apollo: Lightweight multilingual medical llms towards democratizing medical ai to 6b people. *arXiv preprint arXiv:2403.03640*.

Genta Winata, Shijie Wu, Mayank Kulkarni, Thamar Solorio, and Daniel Preoţiuc-Pietro. 2022. Cross-lingual few-shot learning on unseen languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 777–791.

Genta Indra Winata, David Anugraha, Lucky Susanto, Garry Kuwanto, and Derry Tanti Wijaya. 2024a. Metametrics: Calibrating metrics for generation tasks using human preferences. *arXiv preprint arXiv:2410.02381*.

Genta Indra Winata, Ruochen Zhang, and David Ifeoluwa Adelani. 2024b. Miners: Multilingual language models as semantic retrievers. *arXiv preprint arXiv:2406.07424*.

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

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. 2020. Predicting performance for natural language processing tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8625–8646.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable nlp performance prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3703–3714.

Zheng-Xin Yong, Cristina Menghini, and Stephen H Bach. 2024. Lexc-gen: Generating data for extremely low-resource languages with large language models and bilingual lexicons. *arXiv preprint arXiv:2402.14086*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.

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

A Dataset Features Calculation

In this section, we will describe in detail how the dataset features were computed.

- *Train size*: The number of dataset entries used for training specified NLP task.
- *Vocab size*: The number of unique tokens in the dataset after tokenization and preprocessing, where we use SentencePiece.
- *Average Sentence Length*: The average number of tokens per sentence in the dataset.
- *Word Overlap*:

  $$\frac{|T_1 \cap T_2|}{|T_1| + |T_2|}$$

  where T_1 and T_2 correspond to the unique tokens of two datasets.
- *Type-Token Ratio (TTR)*: The ratio of unique tokens to the total number of tokens in the dataset. This is given by:

  $$\text{TTR} = \frac{|V|}{\sum_{i=1}^{N} |s_i|}$$

  where $|V|$ is the number of unique tokens, and $\sum_{i=1}^{N} |s_i|$ is the total number of tokens in all N sentences.

- *TTR Distance*: The distance between the TTRs of two datasets, calculated as:

$$D_{\text{TTR}} = \left(1 - \frac{\text{TTR}_1}{\text{TTR}_2}\right)^2$$

where TTR_1 and TTR_2 are the TTR values for two datasets.

- *Jensen-Shannon Divergence (JSD)*: Measures the divergence between two token distributions. For two token distributions P and Q, the JSD is calculated as:

$$\text{JSD}(P, Q) = \frac{1}{2}\left[\text{KL}(P \parallel M) + \text{KL}(Q \parallel M)\right]$$

where $M = \frac{1}{2}(P + Q)$ and KL is the Kullback-Leibler divergence:

$$\text{KL}(P \parallel Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

- *TF-IDF Cosine Similarity*: The cosine similarity between two datasets based on their TF-IDF representations. The cosine similarity is given by:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

where A and B are the TF-IDF vectors of the two datasets.

- *Sentence-BERT Similarity*: The cosine similarity between the mean sentence embeddings of two datasets obtained from Sentence-BERT (Reimers and Gurevych, 2019). It is computed as:

$$\text{Cosine Similarity}(E_A, E_B) = \frac{E_A \cdot E_B}{\|E_A\|\|E_B\|}$$

where E_A and E_B are the mean sentence embeddings for datasets A and B, respectively.

B Experimental Details

B.1 Languages Under Study

We list all the languages used in the training from the MT560 (Gowda et al., 2021) and NusaTranslation (Cahyawijaya et al., 2023) datasets in Table 7 and Table 9, respectively. The language code follows *ISO639-3 coding. All languages are also complemented by their †rarity taxonomy based on (Joshi et al., 2020) into two vitality classes: 0-2→low resource language (LRL), 3-4→mid resource language (MRL), and 5→high resource language (HRL). We also provide information about whether the language was part of the pre-trained M2M100 model dataset to highlight the model knowledge coverage.

B.2 Models

The details on the proxy models we use in MT experiments are as follows:

- Transformer (100M) (Vaswani et al., 2017): a standard encoder-decoder transformer-based model with 6 encoder layers and 6 decoder layers with an embedding dimension of 512. We train the model from randomly initialized parameters with the training set.
- SMaLL-100 (330M) (Mohammadshahi et al., 2022):[2] a distilled version of the M2M100 (12B) model. We utilize the model in two ways: fine-tuned on training data and zero-shot inference.
- M2M100 (No FT) (Fan et al., 2021):[3] a pre-trained estimated model of M2M100 (1.2B) without any fine-tuning. We run the model in a zero-shot fashion.
- NLLB (No FT) (Costa-jussà et al., 2022):[4] a pre-trained estimated model of NLLB-200 Distilled (1.3B) without any fine-tuning. We run the model in a zero-shot fashion.

The details on the proxy models we use in intent classification and slot filling experiments are as follows:

- SmolLM (135M and 360M) (Allal et al., 2024):[5] a series of small LMs built on Cosmo-Corpus, a meticulously curated high-quality training dataset. SmolLM models have shown promising results when compared to other models in their size categories across various benchmarks testing common sense reasoning and world knowledge.

[2]SMaLL-100 (330M) is taken from `https://github.com/alirezamshi/small100`.

[3]M2M100 (1.2B) is taken from `https://github.com/facebookresearch/fairseq/tree/main/examples/m2m_100`.

[4]NLLB (1.3B) is taken from `https://github.com/facebookresearch/fairseq/tree/nllb`.

[5]SmolLM 135M and 360M are taken from `https://huggingface.co/HuggingFaceTB/SmolLM-135M` and `https://huggingface.co/HuggingFaceTB/SmolLM-360M` respectively

Datasets	Languages Under Study	Domain
English-centric Dataset		
FLoRes-200 (Costa-jussà et al., 2022)	afr, amh, arz, bak, bel, ceb, cym, deu, dik, ewe, fao, fra, guj, hau, hne, hye, ibo, ind, jav, kan, kat, kaz, khm, kin, kir, kor, lmo, ltz, mar, mri, mya, oci, pan, plt, rus, sin, sna, snd, som, ssw, tam, tat, tgl, tuk, wol, xho, vie, yor, zho, zul	Multi-domain
MT560 (Gowda et al., 2021)		
Europarl	deu	Wiki
DGT	fra	Government
Joshua Indian Corpus	tam	Wiki
Neulab TED Talk (Qi et al., 2018)	bel, deu, fra, hye, kat, kaz, kor, mar, mya, rus, tam, vie, zho	TED
News Commentary	deu, fra, rus, zho	News
News-test WMT22 (Kocmi et al., 2022)	deu, rus, zho	News
General-test WMT22 (Kocmi et al., 2022)	deu, fra, kaz, rus, zho	News
OPUS100 (Zhang et al., 2020)	afr, amh, bel, cym, deu, fra, guj, hau, hye, ibo, ind, kan, kat, kaz, khm, kin, kir, kor, mar, mya, oci, pan, rus, sin, tam, tat, tuk, vie, xho, yor, zho, zul	Multi-domain
OPUS Bible (Christodouloupoulos and Steedman, 2015)	afr, amh, ceb, deu, dik, ewe, fra, guj, hye, ind, kan, kor, mar, mri, mya, plt, rus, sin, sna, som, ssw, tam, tgl, vie, wol, xho, zho, zul	Religion
OPUS OpenSubtitles	afr, hye, kat, kor, sin, tgl, vie, zho	Movies
OPUS Tatoeba (Tiedemann, 2012)	afr, amh, arz, bak, bel, ceb, fao, hau, hye, ibo, ind, jav, kan, kat, kaz, khm, kin, kir, kor, lmo, ltz, mar, mri, mya, pan, sna, tat, tgl, tuk, vie, yor, zul	Conversational
OPUS Tanzil (Tiedemann, 2012)	amh, deu, fra, hau, ind, kor, rus, snd, som, tat, zho	Religion
OPUS Gnome (Tiedemann, 2012)	fao, mri, som, zho	Technology
OPUS GlobalVoices (Tiedemann, 2012)	amh, kor, mya	News
OPUS Wikipedia Health (Tiedemann, 2012)	fra, kor, rus, vie	Health
OPUS Antibiotic (Tiedemann, 2012)	fra	Health
OPUS Tico (Tiedemann, 2012)	fra, rus	Health
OPUS Vaccination (Tiedemann, 2012)	fra	Health
OPUS Ubuntu (Tiedemann, 2012)	vie, zho	Technology
OPUS EU Bookshop (Tiedemann, 2012)	cym	Government
OPUS SPC	afr	Government
OPUS Memat	xho	Medicine
OPUS XhosaNavy	xho	Government
OPUS KDE4	hne	Technology
OPUS infopankki	som	Immigration
OPUS TLDR	zho	General
UN	rus	Government
lindat khresmoi (Dušek et al., 2014)	fra	Health
PMIndia (Haddow and Kirefu, 2020)	guj, kan, mar, pan, sin, tam	Government
Wiki Titles	guj, kaz, tam	Wiki
WikiMatrix (Schwenk et al., 2021)	arz, bak, bel, ceb, fao, fra, ind, jav, kat, kaz, kor, lmo, ltz, mar, oci, sin, tam, tat, tgl, vie, zho	Wiki
Many-to-Many Languages Dataset		
NusaTranslation (Cahyawijaya et al., 2023)	bew, btk, ind, jav, mad, mak, min, sun	Social Media

Table 6: List of datasets under study covering 50 different languages. We only opt for 50 out of 500 languages available in the MT560 dataset, 50 out of 200 languages available in the FLoRes-200 dataset, and 8 out of 12 languages available in the NusaTranslation dataset.

- BLOOMZ (560M) (Muennighoff et al., 2022): a decoder model fine-tuned from BLOOM (Workshop et al., 2022) on a cross-lingual task mixture (xP3) and finds the resulting models capable of cross-lingual generalization to unseen tasks and languages.

B.3 Hyper-parameters

LM Each fine-tuning and evaluation for LMs is done with an NVIDIA Tesla V100 32GB GPU.

The hyper-parameters used during fine-tuning from the English-centric and Many-to-Many Languages datasets for MT task are listed in Table 10, 11, 19, and 20 for SMaLL100, M2M100, NLLB, and Transformer models, respectively.

The common hyper-parameters used during fine-tuning for intent and slot classification tasks are listed in Table 12 for SmolLM-135M, SmolLM-360M, BLOOMZ-560M, Aya23-8B, and LLaMA3-8B respectively. The learning rate used for SmolLM-135M, SmolLM-360M, and BLOOMZ-560M is $1e^{-5}$, while the learning rate used for Aya23-8B and LLaMA3-8B is $1e^{-4}$. Since we are fine-tuning using SFT, we use the "default" template for SmolLM-135M, SmolLM-360M, and BLOOMZ-560M, while we use "cohere" for Aya23-8B and "llama3" for LLaMA3-8B. Additionally, we use LoRA for both Aya23-8B and LLaMA3-8B.

Regressor Each regressor is trained on an AMD Ryzen Threadripper 2990WX with 128 GB of RAM and 16 threads. Regressors' hyper-parameters used are provided in Table 21, 22, 23, 24, 25, and 26 for XGBoost, Poly2/Poly3, LGBM, and MF, respectively. These hyper-parameters were obtained based on the best cross-validation RMSE score on training using 10 folds.

B.4 Regressor Dataset Sizes

We provide the details of the regressor's training and test set size in Table 27.

C More Detailed Results

We provide detailed results for Table 1, 2, and 2 by providing the standard errors of the predictions. The mapping of vitality, Joshi class, and language family follows the classifications in Table 7 and 9. The mapping of all languages in Table 5 until 7.

D Feature Importances

We provide feature importance scores of XGBoost with PROXYLM Ensemble for the random English-centric experiment in Figure 8 and 9. Each combination consists of one most influential feature followed by others with marginal contributions to the model, each with an importance score of 0.12 or less. We observe that proxy models are always the most influential features in prediction. Other feature importances plot are also provided in Figure 11 until 13.

E License for Artifacts

We discuss the license or terms for the use of any artifacts we use in Table 28.

Language	Language Code*	Family	Joshi Class†	Vitality†	Seen by M2M100	Covered by COMET-22
Afrikaans	afr	indo-european	3	MRL	✓	✓
Amharic	amh	afro-asiatic	2	LRL	✓	✓
Armenian	hye	indo-european	1	LRL	✓	✓
Bashkir	bak	turkic	1	LRL	✓	✗
Belarusian	bel	indo-european	3	MRL	✓	✓
Burmese	mya	sino-tibetan	1	LRL	✓	✓
Cebuano	ceb	austronesian	3	MRL	✓	✗
Chhattisgarhi	hne	indo-european	0	LRL	✗	✗
Chinese	zho	sino-tibetan	5	HRL	✓	✓
Dinka	dik	nilo-saharan	1	LRL	✗	✗
Egyptian Arabic	arz	afro-asiatic	3	MRL	✓	✗
Ewe	ewe	niger-congo	1	LRL	✗	✗
Faroese	fao	indo-european	1	LRL	✗	✗
Georgian	kat	kartvelian	3	MRL	✓	✓
German	deu	indo-european	5	HRL	✓	✓
Gujarati	guj	indo-european	1	LRL	✓	✓
French	fra	indo-european	5	HRL	✓	✓
Hausa	hau	afro-asiatic	2	LRL	✓	✓
Igbo	ibo	niger-congo	1	LRL	✓	✗
Indonesian	ind	austronesian	3	MRL	✓	✓
Javanese	jav	austronesian	1	LRL	✓	✓
Kannada	kan	dravidian	1	LRL	✓	✓
Kazakh	kaz	turkic	3	MRL	✓	✓
Khmer	khm	austro-asiatic	1	LRL	✓	✓
Kirghiz	kir	turkic	1	LRL	✗	✓
Kinyarwanda	kin	niger-congo	1	LRL	✗	✗
Korean	kor	koreanic	4	MRL	✓	✓
Lombard	lmo	indo-european	1	LRL	✗	✗
Luxembourgish	ltz	indo-european	1	LRL	✓	✗
Malagasy	plt	austronesian	1	LRL	✓	✓
Maori	mri	austronesian	1	LRL	✗	✗
Marathi	mar	indo-european	2	LRL	✓	✓
Occitan	oci	indo-european	1	LRL	✓	✗
Punjabi	pan	indo-european	2	LRL	✓	✓
Russian	rus	indo-european	4	MRL	✓	✓
Shona	sna	niger-congo	1	LRL	✗	✗
Sindhi	snd	indo-european	1	LRL	✓	✓
Sinhala	sin	indo-european	0	LRL	✓	✓
Somali	som	afro-asiatic	1	LRL	✓	✓
Swati	ssw	niger-congo	1	LRL	✓	✗
Tagalog	tgl	austronesian	3	MRL	✓	✗
Tamil	tam	dravidian	3	MRL	✓	✓
Tatar	tat	turkic	1	LRL	✗	✗
Turkmen	tuk	turkic	1	LRL	✗	✗
Vietnamese	vie	austro-asiatic	4	MRL	✓	✓
Welsh	cym	indo-european	1	LRL	✓	✓
Wolof	wol	niger-congo	2	LRL	✓	✗
Xhosa	xho	niger-congo	2	LRL	✓	✓
Yoruba	yor	niger-congo	2	LRL	✓	✗
Zulu	zul	niger-congo	2	LRL	✓	✗

Table 7: List of languages from the English-centric dataset on MT task, including their rarity category mapping, an indication of whether they are involved in the pre-training process for M2M100, and another indication of whether they are covered by COMET-22. Note that all languages are involved in the pre-training process for NLLB.

Language	Language Code*	Family	Joshi Class†	Vitality†
Afrikaans	afr	indo-european	3	MRL
Amharic	amh	afro-asiatic	2	LRL
Arabic	ara	afro-asiatic	5	HRL
Azerbaijani	aze	turkic	1	LRL
Bengali	ben	indo-european	3	MRL
Welsh	cym	indo-european	1	LRL
Danish	dan	indo-european	3	MRL
German	deu	indo-european	5	HRL
Greek	ell	indo-european	3	MRL
English	eng	indo-european	5	HRL
Spanish	spa	indo-european	5	HRL
Persian	fas	indo-european	4	MRL
Finnish	fin	uralic	4	MRL
French	fra	indo-european	5	HRL
Hebrew	heb	afro-asiatic	3	MRL
Hindi	hin	indo-european	4	MRL
Hungarian	hun	uralic	4	MRL
Armenian	hye	indo-european	1	LRL
Indonesian	ind	austronesian	3	MRL
Icelandic	isl	indo-european	2	LRL
Italian	ita	indo-european	4	MRL
Japanese	jpn	japonic	5	HRL
Javanese	jav	austronesian	1	LRL
Georgian	kat	kartvelian	3	MRL
Khmer	khm	austro-asiatic	1	LRL
Kannada	kan	dravidian	1	LRL
Korean	kor	koreanic	4	MRL
Latvian	lav	indo-european	3	MRL
Malayalam	mal	dravidian	1	LRL
Mongolian	mon	mongolic	1	LRL
Malay	msa	austronesian	3	MRL
Burmese	mya	sino-tibetan	1	LRL
Norwegian Bokmål	nob	indo-european	1	LRL
Dutch	nld	indo-european	4	MRL
Polish	pol	indo-european	4	MRL
Portuguese	por	indo-european	4	MRL
Romanian	ron	indo-european	3	MRL
Russian	rus	indo-european	4	MRL
Slovenian	slv	indo-european	3	MRL
Albanian	sqi	indo-european	1	LRL
Swedish	swe	indo-european	4	MRL
Swahili	swa	niger-congo	2	LRL
Tamil	tam	dravidian	3	MRL
Telugu	tel	dravidian	1	LRL
Thai	tha	kra-dai	3	MRL
Tagalog	tgl	austronesian	3	MRL
Turkish	tur	turkic	4	MRL
Urdu	urd	indo-european	3	MRL
Vietnamese	vie	indo-european	4	MRL
Chinese (Simplified)	zho	sino-tibetan	5	HRL
Chinese (Traditional)	zho	sino-tibetan	5	HRL

Table 8: List of languages from the MASSIVE dataset for intent classification and slot filling, including their rarity category mapping.

Language	Language Code*	Family	Joshi Class†	Vitality†	Seen by M2M100	Seen by NLLB
Indonesian	ind	austronesian	3	MRL	✓	✓
Javanese	jav	austronesian	1	LRL	✓	✓
Betawi	bew	creole	0	LRL	✗	✗
Batak	btk	austronesian	0	LRL	✗	✗
Madurese	mad	austronesian	0	LRL	✗	✗
Makassarese	mak	austronesian	0	LRL	✗	✗
Minangkabau	min	austronesian	0	LRL	✗	✓
Sundanese	sun	austronesian	1	LRL	✓	✓

Table 9: List of languages from the Many-to-Many Languages dataset on MT task along with their rarity category mapping and an indication of whether they are included in the pre-training process for each respective model.

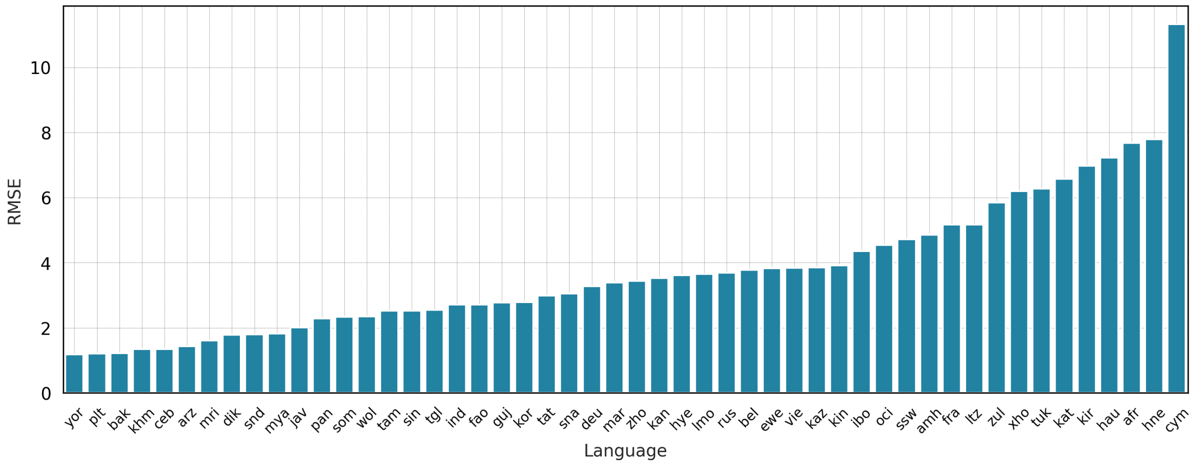


Figure 5: Detailed results of XGBoost with ProxyLM Ensemble on the **M2M100** model under the **LOLO** setting using the **English-centric** dataset on **MT task** from Table 13 per languages.

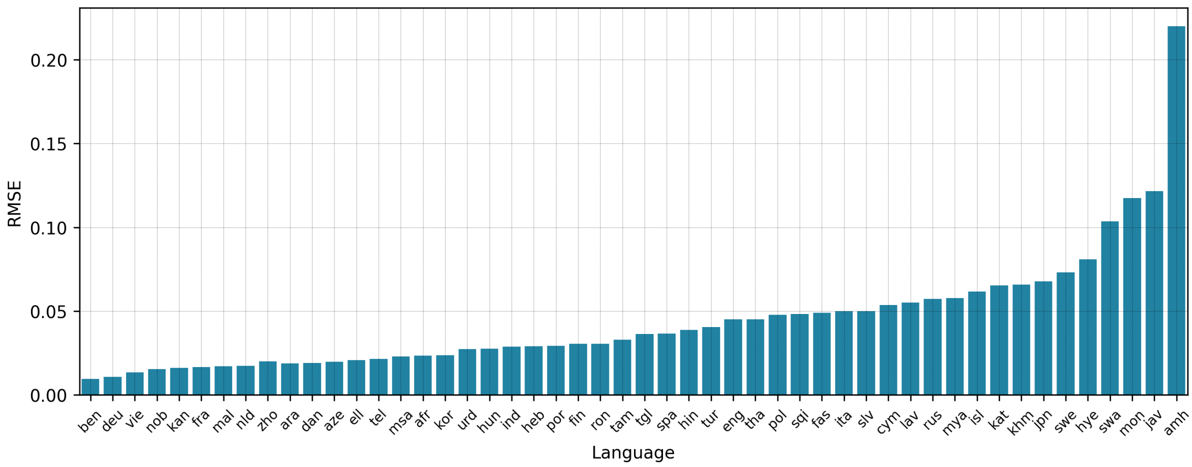


Figure 6: Detailed results of XGBoost with ProxyLM Ensemble on the **LLaMA3** model under the **LOLO** setting on **intent classification** from Table 17 per languages.

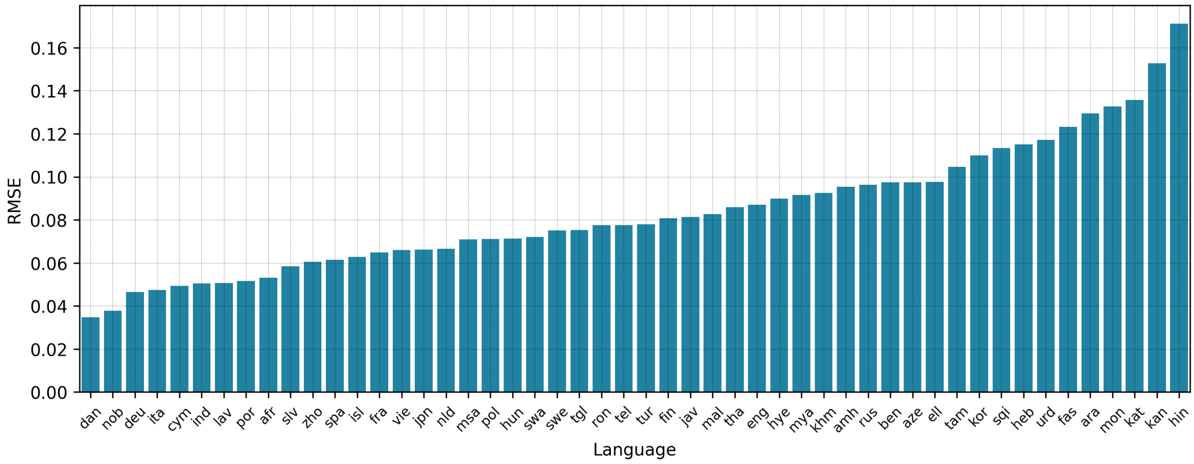


Figure 7: Detailed results of LGBM with PROXYLM Ensemble on the **Aya** model under the **LOLO** setting on **slot classification** from Table 18 per languages.

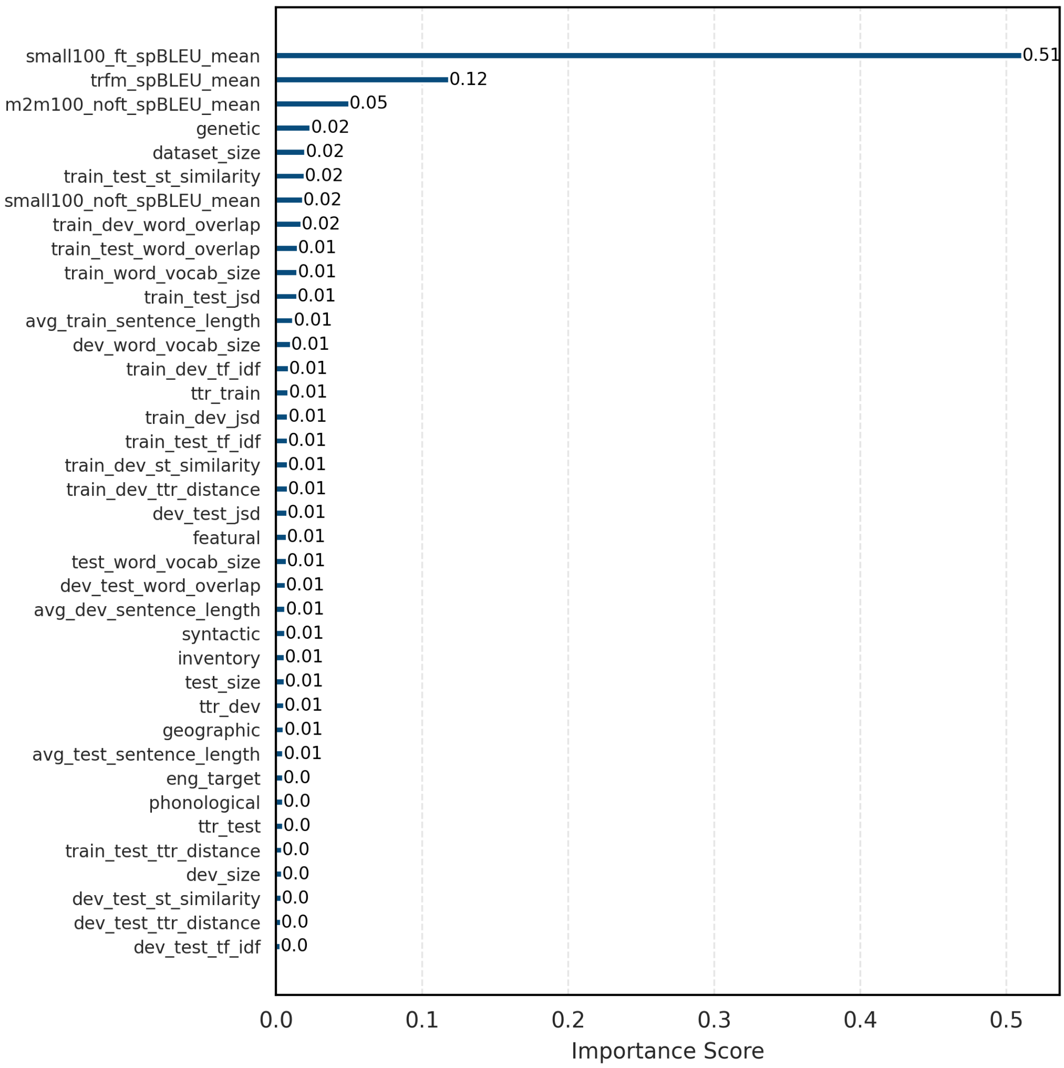


Figure 8: Feature importance analysis of XGBoost with PROXYLM Ensemble on **MT task** using **M2M100** model using the **English-centric** dataset.

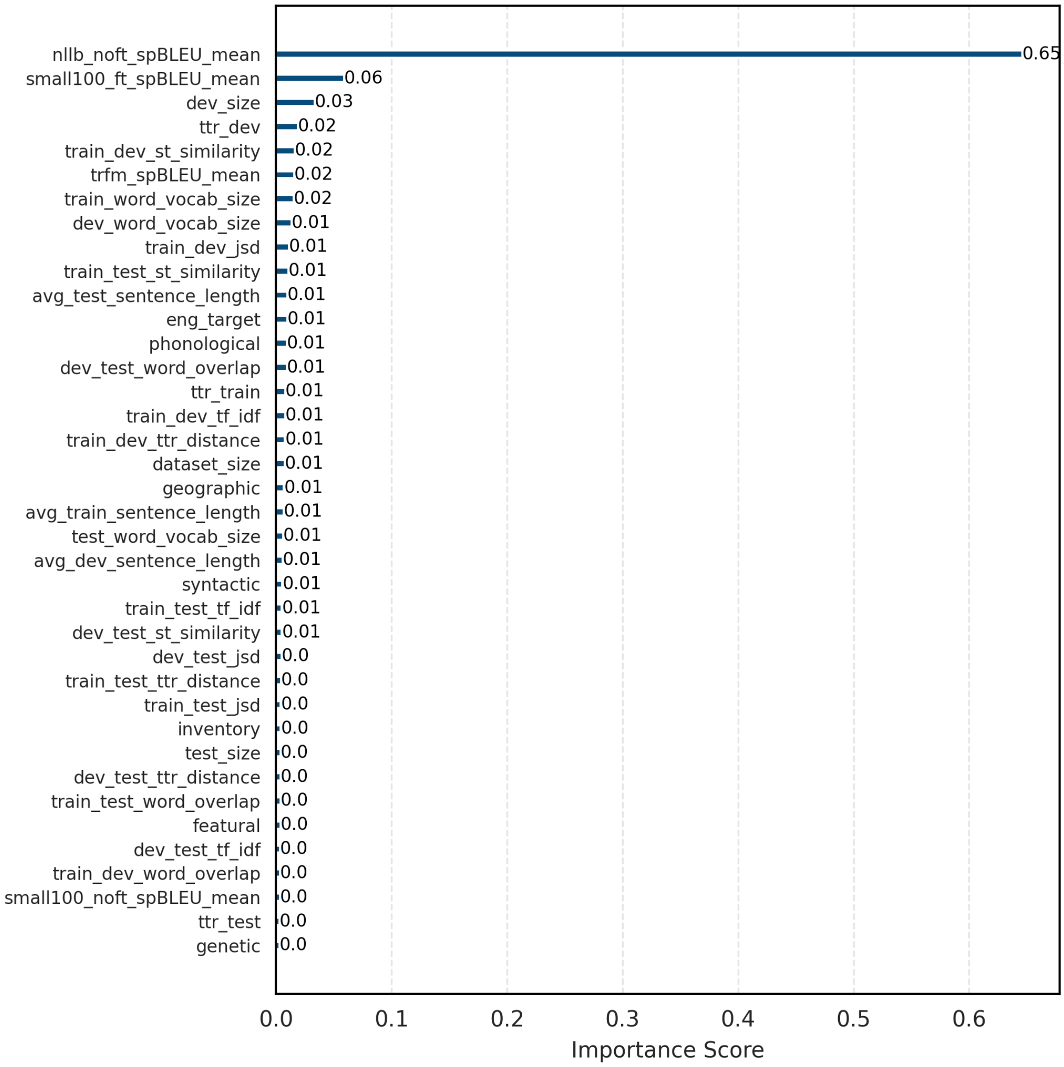


Figure 9: Feature importance analysis of XGBoost with PROXYLM Ensemble on **MT task** using **NLLB** model using the **English-centric** dataset.

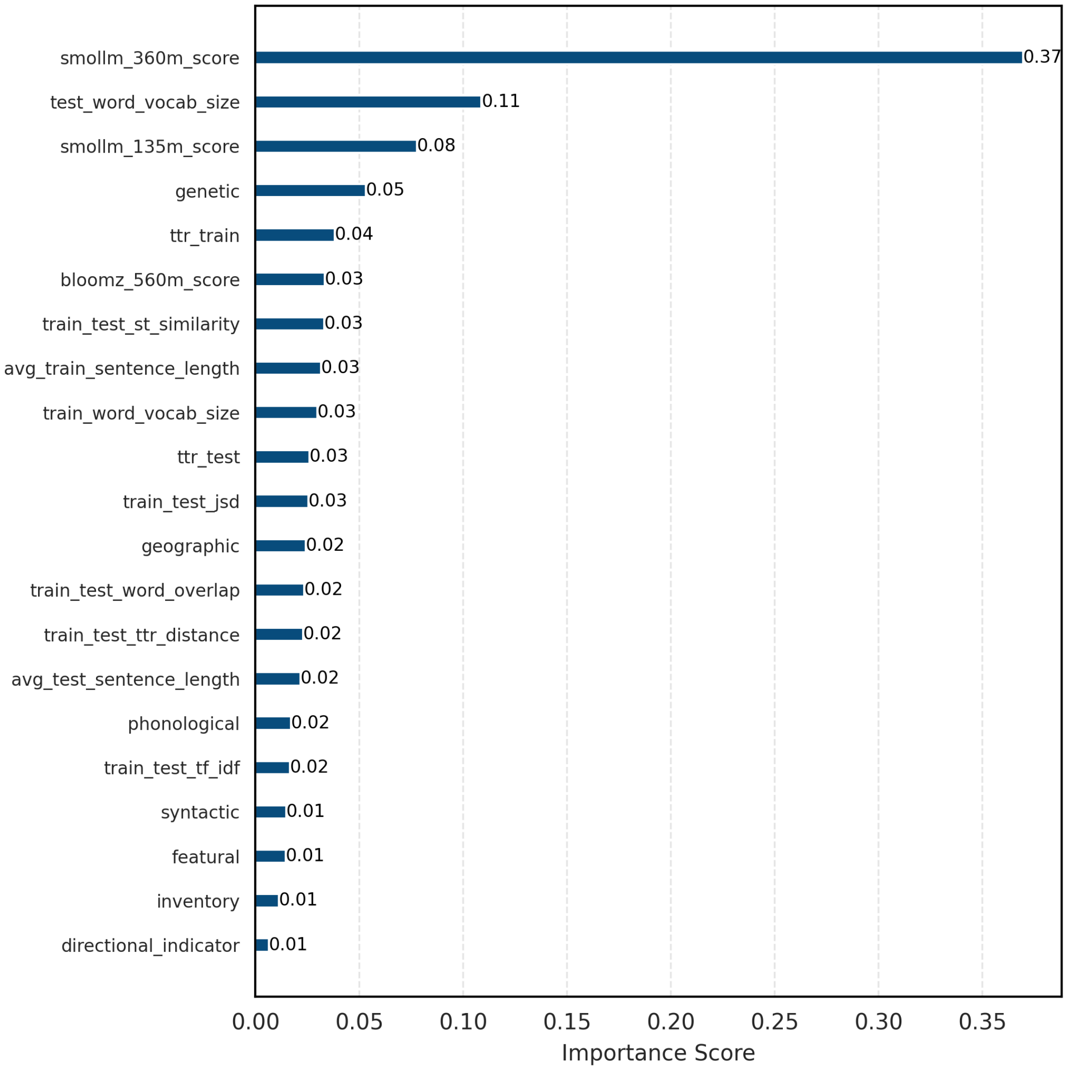


Figure 10: Feature importance analysis of XGBoost with PROXYLM Ensemble on the **Aya-23** on **slot filling** task.

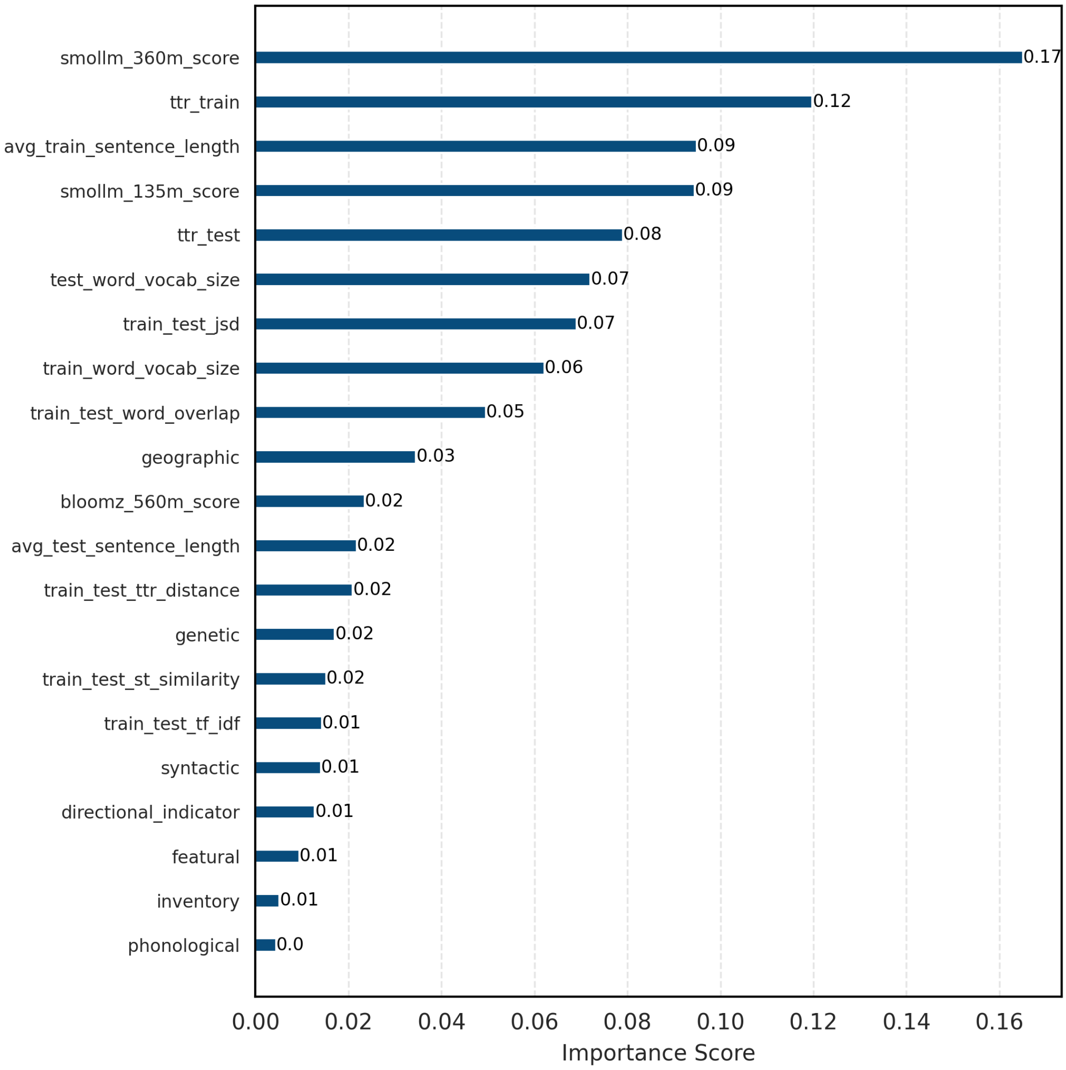


Figure 11: Feature importance analysis of XGBoost with ProxyLM Ensemble on the **LLaMA3** on **slot filling** task.

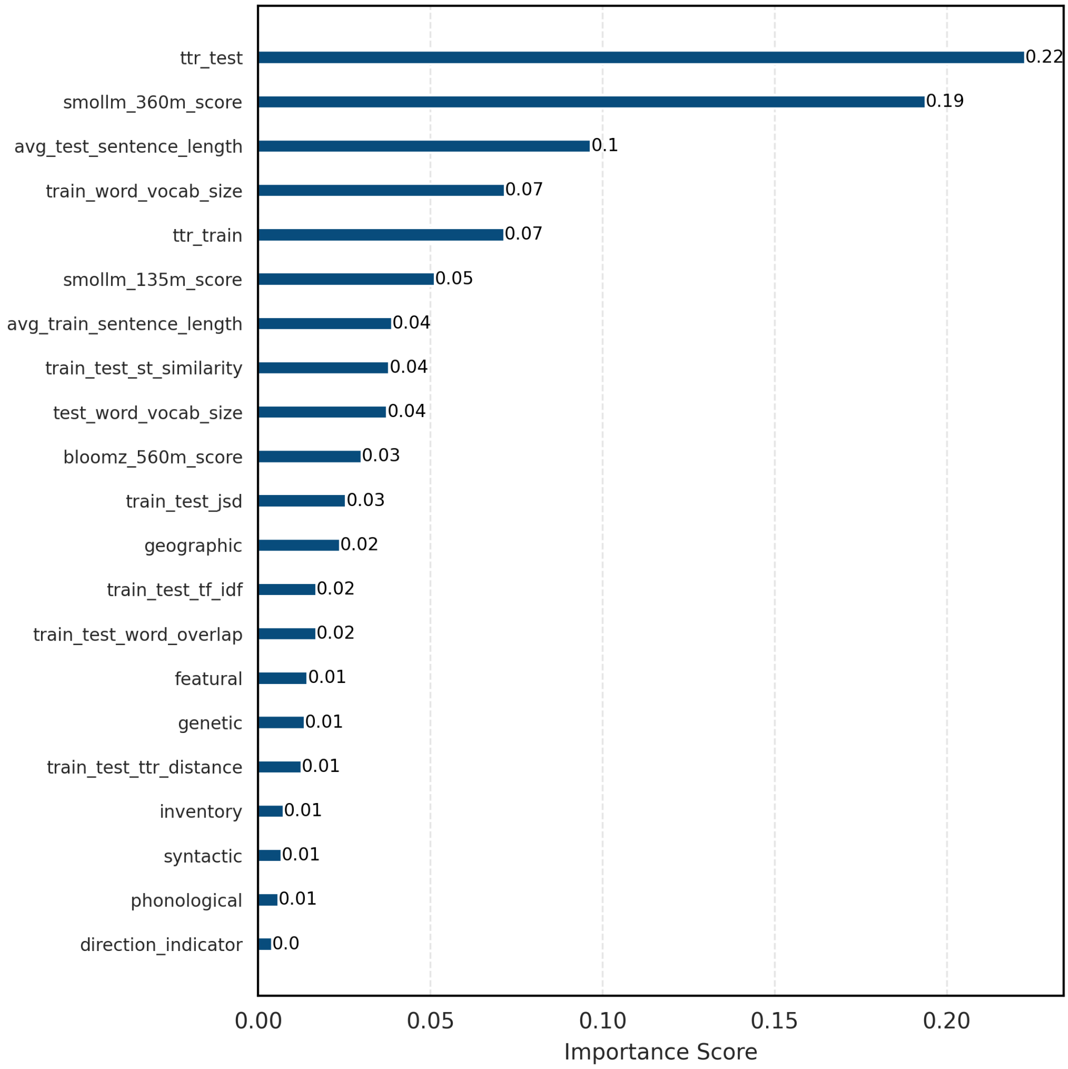


Figure 12: Feature importance analysis of XGBoost with PROXYLM Ensemble on the **Aya-23** on **intent classification** task.

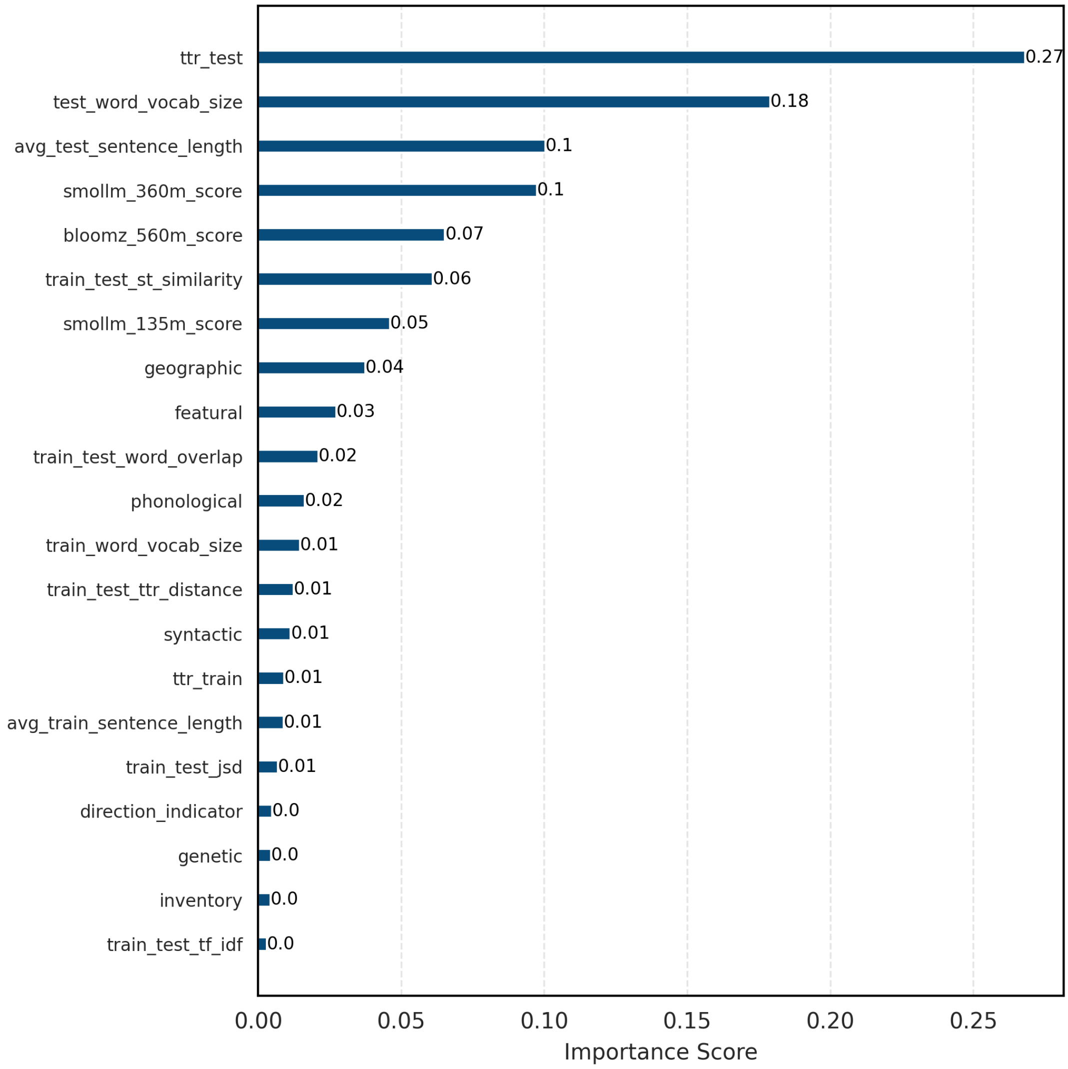


Figure 13: Feature importance analysis of XGBoost with PROXYLM Ensemble on the **LLaMA3** on **intent classification** task.

Hyper-parameter	English-centric	Many-to-Many Langs.
Encoder Layers	12	12
Decoder Layers	3	3
Encoder Embed Dim	1024	1024
Decoder Embed Dim	1024	1024
Encoder FFN Embed Dim	4096	4096
Decoder FFN Embed Dim	4096	4096
Encoder Attention Heads	16	16
Decoder Attention Heads	16	16
Encoder Layerdrop	0.05	0.05
Decoder Layerdrop	0.05	0.05
Optimizer	Adam	Adam
Adam Eps	1e-6	1e-6
Adam Betas	(0.9, 0.98)	(0.9, 0.98)
Patience	6	6
Batch Size	16	16
Dropout	0.1	0.1
Attention Dropout	0.1	0.1
ReLU Dropout	0.1	0.1
Weight Decay	0.0	0.0
Label Smoothing	0.1	0.1
Clip Norm	1.0	1.0
Learning Rate	0.0001	0.0003
Max Tokens (per GPU)	1,000	1,000

Table 10: List of hyper-parameters used for SMaLL100 with English-centric and Many-to-Many Languages datasets.

Hyper-parameter	English-centric	Many-to-Many Langs.
Encoder Layers	24	24
Decoder Layers	24	24
Encoder Embed Dim	1024	1024
Decoder Embed Dim	1024	1024
Encoder FFN Embed Dim	8192	8192
Decoder FFN Embed Dim	8192	8192
Encoder Attention Heads	16	16
Decoder Attention Heads	16	16
Encoder Layerdrop	0.05	0.05
Decoder Layerdrop	0.05	0.05
Optimizer	Adam	Adam
Adam Eps	1e-6	1e-6
Adam Betas	(0.9, 0.98)	(0.9, 0.98)
Patience	6	6
Batch Size	32	32
Dropout	0.1	0.1
Attention Dropout	0.1	0.1
ReLU Dropout	0.1	0.1
Weight Decay	0.0	0.0
Label Smoothing	0.1	0.1
Clip Norm	0.0	0.0
Learning Rate	0.0002	0.0002
Max Tokens (per GPU)	1,792	1,792

Table 11: List of hyper-parameters used for M2M100 with English-centric and Many-to-Many Languages datasets.

Hyper-parameter	Value
Cutoff Length	256
Preprocessing Num Workers	16
Train Batch Size	1
Gradient Accumulation Steps	2
Epochs	3
Learning Rate Scheduler	cosine
Warmup Ratio	0.1
Eval Batch Size	1
Eval Steps	500

Table 12: List of common hyper-parameters used for fine-tuning SmolLM-135M, SmolLM-360M, BLOOMZ-560M, Aya23-8B, and LLaMA3-8B on intent and slot classification task.

Models	Random		LOLO		
	M2M100↓	**NLLB↓**	**M2M100↓**	**NLLB↓**	**Avg.**
NLPerf (Xia et al., 2020) with different regressors					
XGBoost	7.69 ± 0.59	7.73 ± 0.08	9.20 ± 0.40	12.92 ± 0.54	9.16
Poly2 (Khiu et al., 2024)	11.21 ± 0.49	16.23 ± 3.51	15.55 ± 0.00	43.02 ± 0.00	34.62
Poly3 (Khiu et al., 2024)	11.00 ± 0.49	15.64 ± 2.59	62.29 ± 0.00	236.29 ± 0.00	66.97
LGBM (Ye et al., 2021)	7.88 ± 0.65	8.15 ± 0.19	9.71 ± 0.17	12.81 ± 0.23	9.62
PROXYLM (Ours)‡ with different proxy models					
Transformer	4.68 ± 0.41	7.22 ± 0.25	6.18 ± 0.30	11.78 ± 0.50	7.31
SMaLL-100	<u>4.07</u> ± 0.31	6.33 ± 0.11	<u>4.59</u> ± 0.24	10.33 ± 0.41	6.04
SMaLL-100 (No FT)	5.27 ± 0.50	6.04 ± 0.32	6.28 ± 0.32	10.94 ± 0.50	6.95
Estimated Model (No FT)	5.23 ± 0.54	<u>4.15</u> ± 0.24	6.18 ± 0.31	<u>5.42</u> ± 0.27	<u>5.38</u>
Ensemble†	**3.21** ± 0.29	**3.68** ± 0.36	**3.74** ± 0.20	**4.94** ± 0.29	**4.01**

Table 13: English-centric test results using spBLEU in average RMSE ± standard deviation (**lower is better**). **Bold** numbers indicate the best performance, while <u>underlined</u> numbers represent the second-best performance. The columns show the setting and estimated model. “No FT" denotes “no fine-tuning" and the model inference is done in a zero-shot fashion. Avg represents the average of the results across the row. ‡The reported results use XGBoost as the regressor. †Ensemble denotes combining all four proxy models.

Models	Random		LOLO		
	M2M100↓	**NLLB↓**	**M2M100↓**	**NLLB↓**	**Avg.**
NLPerf (Xia et al., 2020) with different regressors					
XGBoost	0.0717 ± 0.0018	0.0556 ± 0.0038	0.1130 ± 0.0054	0.1007 ± 0.0046	0.0825
Poly2 (Khiu et al., 2024)	0.1325 ± 0.0014	0.1503 ± 0.0019	0.1258 ± 0.0000	0.1176 ± 0.0000	0.1316
Poly3 (Khiu et al., 2024)	0.1325 ± 0.0014	0.1520 ± 0.0019	0.1258 ± 0.0000	0.1176 ± 0.0000	0.1320
LGBM (Ye et al., 2021)	0.0752 ± 0.0025	0.0600 ± 0.0033	0.1147 ± 0.0000	0.1019 ± 0.0000	0.0880
PROXYLM (Ours)‡ with different proxy models					
Transformer	0.0543 ± 0.0009	0.0492 ± 0.0041	0.0837 ± 0.0042	0.0957 ± 0.0031	0.0707
SMaLL-100	<u>0.0328</u> ± 0.0012	0.0431 ± 0.0033	<u>0.0547</u> ± 0.0027	0.0824 ± 0.0030	<u>0.0533</u>
SMaLL-100 (No FT)	0.0454 ± 0.0024	0.0393 ± 0.0033	0.0778 ± 0.0030	0.0967 ± 0.0034	0.0648
Estimated Model (No FT)	0.0460 ± 0.0017	<u>0.0348</u> ± 0.0016	0.0777 ± 0.0034	**0.0607** ± 0.0034	0.0548
Ensemble†	**0.0289** ± 0.0012	**0.0293** ± 0.0013	**0.0454** ± 0.0019	<u>0.0613</u> ± 0.0031	**0.0412**

Table 14: English-centric MT test results using COMET-22 in average RMSE ± standard deviation (**lower is better**). **Bold** numbers indicate the best performance, while <u>underlined</u> numbers represent the second-best performance. The columns show the setting and estimated model. “No FT" denotes “no fine-tuning" and the model inference is done in a zero-shot fashion. There are no results for the Unseen setting since COMET-22 score does not cover the unseen languages to M2M100. Furthermore, all languages are covered by NLLB in English-centric dataset. Avg represents the average of the results across the row. ‡The reported results use XGBoost as the regressor. †Ensemble denotes combining all four proxy models, the detailed breakdown of this result can be seen in Section C in the Appendix.

Models	Random		LOLO		
	M2M100↓	**NLLB↓**	**M2M100↓**	**NLLB↓**	**Avg.**
NLPerf (Xia et al., 2020) with different regressors					
XGBoost	2.45 $\pm$ 0.30	**1.11** $\pm$ 0.06	7.83 $\pm$ 0.23	8.28 $\pm$ 0.31	4.94
Poly2 (Khiu et al., 2024)	4.70 $\pm$ 0.40	4.68 $\pm$ 0.51	7.07 $\pm$ 0.00	7.90 $\pm$ 0.00	6.09
Poly3 (Khiu et al., 2024)	4.60 $\pm$ 0.41	4.64 $\pm$ 0.49	7.26 $\pm$ 0.00	8.01 $\pm$ 0.00	6.13
LGBM (Ye et al., 2021)	2.66 $\pm$ 0.22	1.76 $\pm$ 0.29	7.91 $\pm$ 0.01	8.06 $\pm$ 0.00	5.10
MF (Schram et al., 2023)	3.65 $\pm$ 0.26	2.60 $\pm$ 0.39	7.08 $\pm$ 0.23	7.14 $\pm$ 0.22	5.12
ProxyLM[‡] (Ours) with different proxy models					
Transformer	2.56 $\pm$ 0.43	1.70 $\pm$ 0.20	5.65 $\pm$ 0.23	6.24 $\pm$ 0.34	4.04
SMaLL-100	2.56 $\pm$ 0.33	1.65 $\pm$ 0.44	<u>4.85</u> $\pm$ 0.36	<u>5.14</u> $\pm$ 0.46	<u>3.55</u>
SMaLL-100 (No FT)	2.44 $\pm$ 0.21	1.34 $\pm$ 0.38	6.93 $\pm$ 0.34	7.25 $\pm$ 0.37	4.49
Estimated Model (No FT)	**2.38** $\pm$ 0.36	<u>1.27</u> $\pm$ 0.03	5.10 $\pm$ 0.28	5.50 $\pm$ 0.26	3.56
Ensemble[†]	<u>2.41</u> $\pm$ 0.28	1.56 $\pm$ 0.35	**3.73** $\pm$ 0.23	**3.79** $\pm$ 0.19	**2.90**

Table 15: Many-to-Many Languages test results in average RMSE $\pm$ standard deviation (**lower is better**). **Bold** numbers indicate the best performance, while <u>underlined</u> numbers represent the second-best performance. The columns show the setting and estimated model. "No FT" denotes "no fine-tuning". Avg represents the average of the results across the row. [‡]The reported results are experiments using XGBoost regressor. [†]Ensemble denotes combining all four proxy models.

Models	Unseen	Cross-Dataset	
	M2M100↓	**M2M100↓**	**NLLB↓**
NLPerf (Xia et al., 2020) with different regressors			
XGBoost	8.26 $\pm$ 0.53	12.90 $\pm$ 1.01	12.02 $\pm$ 1.02
Poly2 (Khiu et al., 2024)	9.51 $\pm$ 0.00	11.02 $\pm$ 0.00	8.97 $\pm$ 0.00
Poly3 (Khiu et al., 2024)	9.64 $\pm$ 0.00	11.06 $\pm$ 0.00	10.98 $\pm$ 0.00
LGBM (Ye et al., 2021)	9.56 $\pm$ 0.97	9.34 $\pm$ 0.00	10.58 $\pm$ 0.00
ProxyLM[‡] (Ours) with different proxy models			
Transformer	6.71 $\pm$ 0.37	9.31 $\pm$ 0.00	9.70 $\pm$ 0.00
SMaLL-100	<u>4.87</u> $\pm$ 0.34	<u>4.52</u> $\pm$ 0.05	<u>6.87</u> $\pm$ 0.11
SMaLL-100 (No FT)	6.22 $\pm$ 0.38	8.26 $\pm$ 0.59	9.50 $\pm$ 0.04
Estimated Model (No FT)	5.90 $\pm$ 0.40	9.95 $\pm$ 0.40	7.20 $\pm$ 0.24
Ensemble[†]	**4.48** $\pm$ 0.23	**4.35** $\pm$ 0.02	**5.03** $\pm$ 0.48

Table 16: Unseen and Cross-Dataset MT test results on English-centric dataset in average RMSE $\pm$ standard deviation (**lower is better**). **Bold** numbers indicate the best performance, while <u>underlined</u> numbers represent the second-best performance. The columns show the setting and estimated model. "No FT" denotes "no fine-tuning". We only show M2M100 results for the Unseen setting since NLLB covers all languages in the English-centric dataset. [‡]The reported results for the Unseen setting use XGBoost, while the Cross-Dataset experiments use LGBM. [†]Ensemble denotes combining all four proxy models.

Models	Random		LOLO		
	Aya-23↓	**LLaMA3↓**	**Aya-23↓**	**LLaMA3↓**	**Avg.**
NLPerf (Xia et al., 2020) with different regressors					
XGBoost	0.0761 ± 0.0008	0.0191 ± 0.0008	0.1573 ± 0.0039	0.0581 ± 0.0021	0.0777
Poly2 (Khiu et al., 2024)	0.1996 ± 0.0017	0.0979 ± 0.0033	0.2075 ± 0.0000	0.0918 ± 0.0000	0.1492
Poly3 (Khiu et al., 2024)	0.1990 ± 0.0011	0.0969 ± 0.0015	0.2191 ± 0.0000	0.0925 ± 0.0000	0.1519
LGBM (Ye et al., 2021)	0.0839 ± 0.0030	0.0198 ± 0.0023	0.1545 ± 0.0000	0.0558 ± 0.0006	0.0785
PROXYLM (Ours)‡ with different proxy models					
SmolLM (135M)	0.0676 ± 0.0013	0.0171 ± 0.0008	0.1273 ± 0.0033	0.0455 ± 0.0020	0.0644
SmolLM (360M)	**0.0604** ± 0.0015	**0.0157** ± 0.0003	<u>0.1118</u> ± 0.0031	**0.0441** ± 0.0017	**0.0580**
BLOOMZ (560M)	0.0692 ± 0.0025	0.0179 ± 0.0009	0.1283 ± 0.0038	0.0482 ± 0.0023	0.0659
Ensemble†	<u>0.0609</u> ± 0.0010	<u>0.0164</u> ± 0.0008	**0.1112** ± 0.0032	<u>0.0442</u> ± 0.0019	<u>0.0582</u>

Table 17: Intent classification results using accuracy in average RMSE ± standard deviation (**lower is better**). **Bold** numbers indicate the best performance, while <u>underlined</u> numbers represent the second-best performance. Avg represents the average of the results across the row. ‡The reported results use XGBoost as the regressor. †Ensemble denotes combining all four proxy models.

Models	Random		LOLO		
	Aya-23↓	**LLaMA3↓**	**Aya-23↓**	**LLaMA3↓**	**Avg.**
NLPerf (Xia et al., 2020) with different regressors					
XGBoost	0.0693 ± 0.0009	0.0548 ± 0.0034	0.1219 ± 0.0035	0.1093 ± 0.0027	0.0888
Poly2 (Khiu et al., 2024)	0.1396 ± 0.0008	0.1412 ± 0.0017	0.1418 ± 0.0000	0.1414 ± 0.0000	0.1410
Poly3 (Khiu et al., 2024)	0.1393 ± 0.0008	0.1401 ± 0.0017	0.1448 ± 0.0000	0.1413 ± 0.0000	0.1414
LGBM (Ye et al., 2021)	0.0692 ± 0.0017	0.0557 ± 0.0029	0.1218 ± 0.0000	0.1152 ± 0.0000	0.0905
PROXYLM (Ours)‡ with different proxy models					
SmolLM (135M)	0.0618 ± 0.0018	0.0538 ± 0.0022	0.1004 ± 0.0028	0.0953 ± 0.0024	0.0778
SmolLM (360M)	**0.0562** ± 0.0011	**0.0506** ± 0.0023	<u>0.0844</u> ± 0.0024	**0.0868** ± 0.0018	**0.0695**
BLOOMZ (560M)	0.0618 ± 0.0027	0.0540 ± 0.0019	0.1023 ± 0.0033	0.0995 ± 0.0028	0.0794
Ensemble†	<u>0.0561</u> ± 0.0016	<u>0.0508</u> ± 0.0021	**0.0830** ± 0.0022	<u>0.0884</u> ± 0.0025	<u>0.0696</u>

Table 18: Slot filling results using micro-F1 score in average RMSE ± standard deviation (**lower is better**). **Bold** numbers indicate the best performance, while <u>underlined</u> numbers represent the second-best performance. Avg represents the average of the results across the row. ‡The reported results use LGBM as the regressor. †Ensemble denotes combining all four proxy models.

Hyper-parameter	English-centric	Many-to-Many Langs.
Encoder Layers	24	24
Decoder Layers	24	24
Encoder Embed Dim	1024	1024
Decoder Embed Dim	1024	1024
Encoder FFN Embed Dim	8192	8192
Decoder FFN Embed Dim	8192	8192
Encoder Attention Heads	16	16
Decoder Attention Heads	16	16
Encoder Layerdrop	0.05	0.05
Decoder Layerdrop	0.05	0.05
Optimizer	Adam	Adam
Adam Eps	1e-6	1e-6
Adam Betas	(0.9, 0.98)	(0.9, 0.98)
Patience	6	6
Batch Size	32	32
Dropout	0.1	0.1
Attention Dropout	0.1	0.1
ReLU Dropout	0.0	0.0
Weight Decay	0.01	0.01
Label Smoothing	0.1	0.1
Clip Norm	1.0	1.0
Learning Rate	0.00002	0.0001
Max Tokens (per GPU)	1,000	1,000

Table 19: List of hyper-parameters used for NLLB with English-centric and Many-to-Many Languages datasets.

Hyper-parameter	English-centric	Many-to-Many Langs.
Encoder Layers	6	6
Decoder Layers	6	6
Encoder Embed Dim	512	512
Decoder Embed Dim	512	512
Encoder FFN Embed Dim	2048	2048
Decoder FFN Embed Dim	2048	2048
Encoder Attention Heads	8	8
Decoder Attention Heads	8	8
Encoder Layerdrop	0.05	0.05
Decoder Layerdrop	0.05	0.05
Optimizer	Adam	Adam
Adam Eps	1e-6	1e-6
Adam Betas	(0.9, 0.98)	(0.9, 0.98)
Patience	6	6
Batch Size	32	32
Dropout	0.1	0.1
Attention Dropout	0.1	0.1
ReLU Dropout	0.1	0.1
Weight Decay	0.0001	0.0001
Label Smoothing	0.1	0.1
Clip Norm	0	0
Learning Rate	0.001	0.0005
Max Tokens (per GPU)	1,000	1,000

Table 20: List of hyper-parameters used for Transformer with English-centric and Many-to-Many Languages datasets.

Hyper-parameter	Using spBLEU		Using COMET	
	M2M100	NLLB	M2M100	NLLB
max n_estimators	5000	5000	5000	5000
max eta	0.1	0.1	0.1	0.1
min_child_weight	5.0	4.2	3.2	1.1
max_depth	5	4	3	5
gamma	0	0	0	0
subsample	0.6	0.94	0.6	1
colsample_bytree	0.83	0.82	0.9	0.86
reg_alpha	0.2	0.32	0.11	0
reg_lambda	0.1	0.37	0.48	0.05

Table 21: List of hyper-parameters used for XGBoost Regressor on MT task with M2M100 and NLLB models trained with English-centric dataset.

Hyper-parameter	M2M100	NLLB
max n_estimators	2000	2000
max eta	0.1	0.1
min_child_weight	5	2.5
max_depth	3	3
gamma	0	0
subsample	0.7	0.9
colsample_bytree	0.6	0.6
reg_alpha	0	0
reg_lambda	0.35	0.15

Table 22: List of hyper-parameters used for XGBoost Regressor on MT task with M2M100 and NLLB models trained with Many-to-Many Languages dataset.

Hyper-parameter	Aya-23	LLaMA3
max n_estimators	5000	5000
max eta	0.1	0.1
min_child_weight	3.0	3.0
max_depth	3	3
gamma	0.1	0.1
subsample	0.85	0.6
colsample_bytree	1.0	0.95
reg_alpha	0.1	0.1
reg_lambda	0.2	0.5

Table 23: List of hyper-parameters used for XGBoost Regressor on intent classification and slot filling tasks with Aya-23 and LLaMA-3 models.

Hyper-parameter	Value
alpha	0.1
l1_ratio	0.9

Table 24: List of hyper-parameters used for Poly2/Poly3 Regressor for all tasks.

Hyper-parameter	Value
max learning_rate	0.3
max num_leaves	64
n_estimators	100
max_bin	200000
max_depth	10
min_child_weight	0.001
min_child_samples	20
min_split_gain	0.0
colsample_bytree	1.0
subsample	1.0
reg_alpha	0.1
reg_lambda	0.1

Table 25: List of hyper-parameters used for LGBM Regressor for all tasks. "Max" indicates the maximum value set for the hyper-parameter during the hyper-parameter search.

Hyper-parameter	Specification
max alpha	0.01
beta_w	0.1
beta_h	0.1
beta_z	0.01
beta_s	0.01
beta_t	0.01
lr_decay	0.001
iterations	2000

Table 26: List of hyper-parameters used for MF Regressor with M2M100 and NLLB models trained with Many-to-Many Languages datasets. "Max" indicates the maximum value set for the hyper-parameter during the hyper-parameter search.

Experimental Settings	Train Size	Test Size
Random (English-centric)	1,367	587
Random (Many-to-Many Langs.)	156	68
Random (Intent Classification)	1,820	781
Random (Slot Filling)	1,820	781
Unseen	1,853	101
Cross-Dataset	1,954	224

Table 27: Regressor's training and test set size on different experimental settings. The total MT experimental records for English-centric and Many-to-Many Languages datasets are 1,954 and 224, respectively. On the other hand, the total experimental records for intent classification and slot filling are both 2,601.

Datasets	**URL Link**	**License**
MT560 (Gowda et al., 2021)	`https://opus.nlpl.eu/MT560`	Unknown
FLoRes (Costa-jussà et al., 2022)	`Muennighoff/flores200`	CC-BY-SA 4.0
NusaTranslation (Cahyawijaya et al., 2023)	`https://huggingface.co/datasets/indonlp/nusatranslation_mt`	Apache 2.0
MASSIVE (FitzGerald et al., 2022)	`https://huggingface.co/datasets/AmazonScience/massive`	CC-BY 4.0

Table 28: List of datasets under study with their licenses.