What Causes Knowledge Loss in Multilingual Language Models?

Maria Khelli[1], **Samuel Cahyawijaya**[2], **Ayu Purwarianti**[1], **Genta Indra Winata**[3]
[1]Institut Teknologi Bandung [2]Cohere [3]Capital One
khelli07.id@gmail.com, samuelcahyawijaya@cohere.com
ayu@informatika.org, genta.winata@capitalone.com

Abstract

Cross-lingual transfer in natural language processing (NLP) models enhances multilingual performance by leveraging shared linguistic knowledge. However, traditional methods that process all data simultaneously often fail to mimic real-world scenarios, leading to challenges like catastrophic forgetting, where fine-tuning on new tasks degrades performance on previously learned ones. Our study explores this issue in multilingual contexts, focusing on linguistic differences affecting representational learning rather than just model parameters. We experiment with 52 languages using LoRA adapters of varying ranks to evaluate non-shared, partially shared, and fully shared parameters. Our aim is to see if parameter sharing through adapters can mitigate forgetting while preserving prior knowledge. We find that languages using non-Latin scripts are more susceptible to catastrophic forgetting, whereas those written in Latin script facilitate more effective cross-lingual transfer.

1 Introduction

Cross-lingual transfer in natural language processing (NLP) models has demonstrated enhanced performance in multilingual contexts compared to monolingual settings, largely due to the advantages of leveraging cross-lingual knowledge (Hu et al., 2020; FitzGerald et al., 2023; Winata et al., 2023b, 2024). Typically, training occurs only once simultaneously, where all available data is processed in a single training run. However, in real-world applications, data is often received sequentially over time, highlighting the importance of continuous model updates to maintain performance (Rolnick et al., 2019). Unlike humans, who can retain prior knowledge while acquiring new skills, neural network models often struggle to preserve previously learned information when fine-tuned on new tasks, which is known as catastrophic forgetting, a decline in performance on earlier tasks after the model is exposed to new data (Winata et al., 2023a). To mitigate this issue, several studies have investigated continual learning strategies and the implementation of adapters (Badola et al., 2023) as viable solutions. This limitation poses a significant challenge for multilingual NLP, as models must adapt to new languages while retaining previously acquired linguistic knowledge. Without an effective learning strategy, models risk performance degradation, rendering them less suitable for long-term deployment.

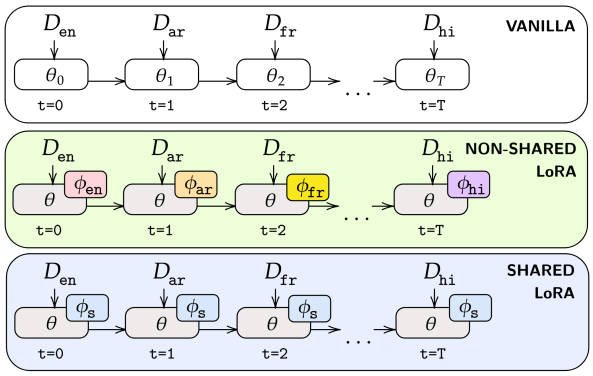

Figure 1: Pipeline for various approaches in lifelong learning. In our lifelong learning framework, we employ a LoRA-based approach where the parameters of the base model, denoted as θ, remain fixed, and for VANILLA, the model parameters are updated at all times. We explore the phenomenon of multilingual knowledge loss by comparing the effects of training with both shared and non-shared parameters.

Lifelong learning is essential for integrating new annotated data across languages without requiring full retraining of systems. As language changes and new data becomes available, models must adapt incrementally to minimize computational costs. This approach helps maintain efficiency and scalability, while addressing the challenge of catastrophic forgetting, which has been explored in various studies (Liu et al., 2021; Winata et al., 2023a; Badola et al., 2023; M'hamdi et al., 2023). However, there is a lack of systematic analysis on this issue in multi-

Proceedings of the Fourth Workshop on NLP Applications to Field Linguistics (FieldMatters), pages 15–25

lingual contexts. This study aims to fill that gap by investigating factors contributing to catastrophic forgetting beyond model parameters, including how linguistic differences can affect representational learning and lead to knowledge erosion when learning multiple languages sequentially.

In this study, we investigate the effects of non-shared, partially shared, and fully shared parameters in a multilingual context, examining 52 languages through the use of LoRA adapters with varying ranks and different sharing model parameter settings as shown in Figure 1. Our primary focus is to assess the impact of parameter sharing on model performance, while also conducting a comprehensive analysis of the role that different languages play in catastrophic forgetting. Additionally, we explore sequential learning to identify when performance drops occur and whether these declines are influenced by the introduction of newly learned languages or the cumulative number of previously learned languages. Our contributions can be summarized as follows:

- We examine the factors contributing to knowledge loss in multilingual language models, focusing on aspects such as language diversity, parameter sharing strategies, and base model selection within a lifelong learning framework for massively multilingual learning.
- We assess cross-lingual transferability and introduce multi-hop metrics to better understand the impact of language skills on model performance.
- We analyze model parameter adaptation to investigate trends in the model's ability to learn languages in a lifelong learning context.

2 Methodology

2.1 Task Setup

A sequence of T tasks is structured as an ordered set of datasets $\mathcal{D} = \{D_1, D_2, \ldots, D_t, \ldots, D_T\}$, where each dataset D_t corresponds to a specific task t, representing a distinct language. The model, parameterized by θ_t, undergoes iterative updates, with parameters at step $t+1$ being derived from those at step t through the function $f(\theta_t, D_t)$. These updates are performed using gradient-based optimization to maximize the log-likelihood over dataset D_t. In this paper, task T is interchangeable with language L.

2.2 Training Methods

We use XLM-R_{BASE} (Conneau et al., 2020) as our base model and compare key methods with E5 instruct (Wang et al., 2024) for evaluating the consistency of the findings. A classification layer is added on top of the encoder model, tailored sequence label of the slot filling. For adapter-based approaches, only the parameters within the adapter modules are updated during training.

MULTI. A single model (or LoRA adapter) is trained on all languages simultaneously, optimizing over the entire dataset $\mathcal{D}$:

$$\theta_{\text{MULTI}} = \arg\max_{\theta} \sum_{t=1}^{T} \log p(D_t \mid \theta). \quad (1)$$

MONO. Each language/task has its own independently trained model θ_t:

$$\theta_t = \arg\max_{\theta} \log p(D_t \mid \theta), \quad (2)$$

$$\forall t \in \{1, \ldots, T\}. \quad (3)$$

VANILLA. A single model is trained incrementally, updating parameters sequentially:

$$\theta_{t+1} \leftarrow f(\theta_t, D_t), \forall t \in \{1, \ldots, T-1\}. \quad (4)$$

SHARED LoRA. A single LoRA adapter ϕ is trained while keeping the base model θ_0 frozen:

$$\phi_s \leftarrow f(\phi'_t, D_t), \theta = \theta_0, \forall t \in \{1, \ldots, T-1\}. \quad (5)$$

NON-SHARED LoRA. Each language has its own separate LoRA adapter ϕ_t, while keeping the base model θ_0 frozen:

$$\phi_t = \arg\max_{\phi} \log p(D_t \mid \theta_0, \phi), \forall t \in \{1, \ldots, T\}. \quad (6)$$

The specific ordering of languages used in the VANILLA is specified in Appendix Table 3.

2.3 Model Parameters Adaptation

We utilize low-rank adapters LoRA (Hu et al., 2021) for training parameters to analyze the effectiveness to have sharing parameters. It is a parameter-efficient fine-tuning method for large pre-trained models leveraging the intrinsic low-dimensionality of parameter updates, reducing the need for full model adaptation. Instead of modifying dense layers directly, it freezes the pre-trained

weights and introduces trainable low-rank matrices, significantly minimizing the number of learnable parameters and enhancing fine-tuning efficiency.

Formally, given a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA constrains the update ΔW to a low-rank decomposition:

$$\Delta W = BA, \tag{7}$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, with rank $r \ll \min(d, k)$. This decomposition ensures that only A and B are updated while W_0 remains fixed. Consequently, the forward pass is expressed as:

$$h = W_0 x + \Delta W x = W_0 x + BAx, \tag{8}$$

where x is the input vector, and h is the output. The low-rank update $\Delta W x$ is scaled by a constant factor $\frac{\alpha}{r}$, analogous to a learning rate, to regulate the magnitude of the update. LoRA offers key advantages: it enhances *memory* and *computational efficiency* by limiting trainable parameters, reducing resource requirements, and enabling modular fine-tuning. Its linear structure ensures *no additional inference latency* and allows seamless integration. By leveraging low-rank adaptation, LoRA enables scalable and efficient model adaptation without compromising previously learned tasks.

3 Experimental Setup

3.1 Datasets

We utilize the MASSIVE, multilingual slot filling dataset (FitzGerald et al., 2023), which encompasses 52 languages and provides structured information, including scenarios, intents, utterances, and annotated utterances. Each language is uniformly represented, with 11.5K training samples, 2.03K validation samples, and 2.97K test samples.

3.2 Hyper-parameters

The training setup employed different configurations depending on whether LoRA was used. For models trained with LoRA, a learning rate of 5×10^{-6} was applied, whereas models without LoRA used a higher learning rate of 5×10^{-5}. The number of training epochs is 100 for models with LoRA, and 50 for those without. Early stopping was implemented in both settings, with a patience of 15 epochs for LoRA and 5 epochs for non-LoRA models, based on the F1-score on validation data. The LoRA configuration included a dropout rate of 0.1, and the scaling factor α was set equal to the rank (32, 64, 256 respectively).

3.3 Evaluation Metrics

We evaluate the performance of the model using average F1 score for the learned tasks and visualized its progression over number of learned languages, as illustrated in Figure 2. Besides that, there are additional metrics, particularly for sequential methods such as `VANILLA` and `SHARED LoRA`.

3.3.1 Performance Shift

This metrics is used to measure the average performance shift, which quantifies the change in a previously learned language performance after training in a new language. Formally, we define the average performance change as follows:

$$\mathcal{P}_{\text{avg}} = \frac{1}{N}\sum_{n=1}^{N}(\mathcal{P}_t - \mathcal{P}_{t+1}), \tag{9}$$

where $\mathcal{P}_t$ and $\mathcal{P}_{t+1}$ represent the average F1 score over all previously encountered tasks at time steps t and $t+1$, respectively. To account for variability in task sequences, the performance changes are averaged over five times ($N = 5$).

3.4 Cross-lingual Transfer

We assess cross-lingual transfer effectiveness using Cross-lingual Forward Transfer (CFT) and Cross-lingual Backward Transfer (CBT) metrics from Winata et al. (2023a) and we introduce a new metric, Multi-Hop Forward Transfer (MFT), and Multi-Hop Backward Transfer (MBT) to measure the multi-hop transfer for each language. Let $R \in \mathbb{R}^{T\times T}$ be a matrix where $R_{i,j}$ represents the test score performance on task t_j after training on the last sample from task t_i. The two types of metrics are defined as follows.

Cross-lingual Forward Transfer (CFT). The metric evaluates the model's ability to generalize to unseen languages by assessing test performance on tasks not encountered during training. It is defined as:

$$CFT = \frac{1}{T-1}\sum_{i=1}^{T-1}\bar{X}_i, \tag{10}$$

where

$$\bar{X}_i = \frac{1}{T-i}\sum_{j=i+1}^{T} R_{i,j}. \tag{11}$$

Here, $\bar{X}_i$ represents the average performance across future tasks ($t_j > t_i$).

Cross-lingual Backward Transfer (CBT). The metric measures the impact of learning a new task t_i on the performance of previously learned tasks. It is formally defined as:

$$CBT = \frac{1}{T-1} \sum_{i=1}^{T-1} (R_{T-1,i} - R_{i,i}) . \quad (12)$$

This metric quantifies the extent of catastrophic forgetting, where adding a new task may negatively impact the performance of past tasks.

Multi-Hop Forward Transfer (MFT). The metric measures the knowledge transfer effect between tasks separated by multiple learning steps. For a hop distance h, MFT is defined as:

$$MFT_h = \frac{1}{|L|} \sum_{l \in L} (\mathcal{P}_{i+h} - \mathcal{P}_{i-1}), \quad (13)$$

where $\mathcal{P}_i$ represents the average performance on tasks seen up to step i. This metric quantifies how learning a language affects performance on another language that will be encountered h steps later in the training sequence.

Multi-Hop Backward Transfer (MBT). The metric similarly evaluates the effect of learning a new task on the performance of tasks encountered several steps earlier. For a hop distance h, MBT is defined as:

$$MBT_h = \frac{1}{|L|} \sum_{l \in L} (\mathcal{P}_i - \mathcal{P}_{i-h-1}). \quad (14)$$

This metric measures how training on a language affects the performance on languages that were learned h steps before in the training sequence. The term *multi-hop* refers to our evaluation across multiple hops, as illustrated in Figure 5. A hop distance of zero corresponds to the performance change metric.

4 Results

Figure 3 illustrates the impact of training different languages sequentially on model performance towards learned language, measured by the average F1 change across 5 different orders.

Performance vs. Model Parameters. Table 1 presents a comparison of training methods in terms of average F1 score and trainable parameters. The `MULTI` method achieves the best overall performance (75.03%) with a much less parameter footprint (278.04M) compared to `MONO`'s, offering an excellent balance between effectiveness and efficiency. On the opposite end, `MONO`, which trains an entirely separate model per language, consumes an enormous parameter budget (14,458.27M) while yielding only moderate performance (72.98%), highlighting the inefficiency of isolated training.

Among parameter-efficient alternatives, LoRA-based approaches exhibit varying trade-offs. `NON-SHARED LoRA` performs competitively (up to 73.16% at rank 256), benefiting from task-specific specialization, albeit with moderate parameter cost (2,228.75M). In contrast, `SHARED LoRA`'s best result dramatically reduces the number of trainable parameters (e.g., 10.72M at rank 256) but suffers heavily in performance, dropping to as low as 61.26%.

Method	Params (M)	F1 (%)	Language Vitality		
			Low	Mid	High
MULTI	278.04	**75.03**	75.42	**75.84**	72.63
$r = 32$	5.36	74.19	74.27	**75.17**	71.83
$r = 64$	10.72	73.79	74.00	**74.56**	71.73
$r = 256$	42.86	74.11	74.16	**74.83**	72.41
MONO	14,458.27	72.98	73.66	**74.11**	69.43
VANILLA	278.04	66.16	65.70	**67.65**	63.46
SHARED LoRA					
$r = 32$	5.36	60.24	59.35	**62.34**	56.75
$r = 64$	10.72	**61.26**	60.55	**63.37**	57.48
$r = 256$	42.86	60.16	59.06	**62.15**	57.22
NON-SHARED LoRA					
$r = 32$	278.04	72.14	72.42	**73.39**	68.89
$r = 64$	557.19	72.38	72.55	**73.48**	69.65
$r = 256$	2,228.75	**73.16**	73.82	**74.26**	69.73

Table 1: Comparison of methods based on trainable parameters (in million parameters) and averaged F1 (%). Lower trainable parameters is better, higher average performance is better.

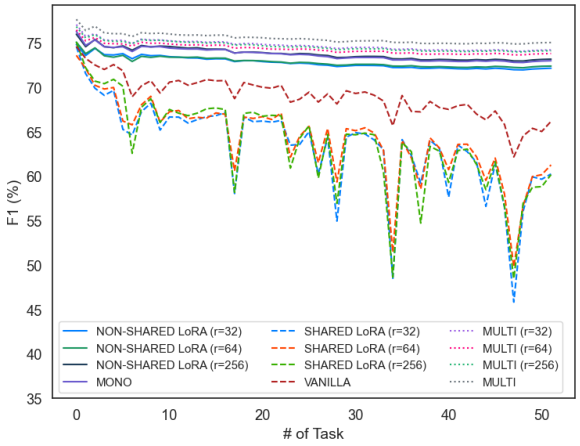

Figure 2: Performance results after training each language over the time.

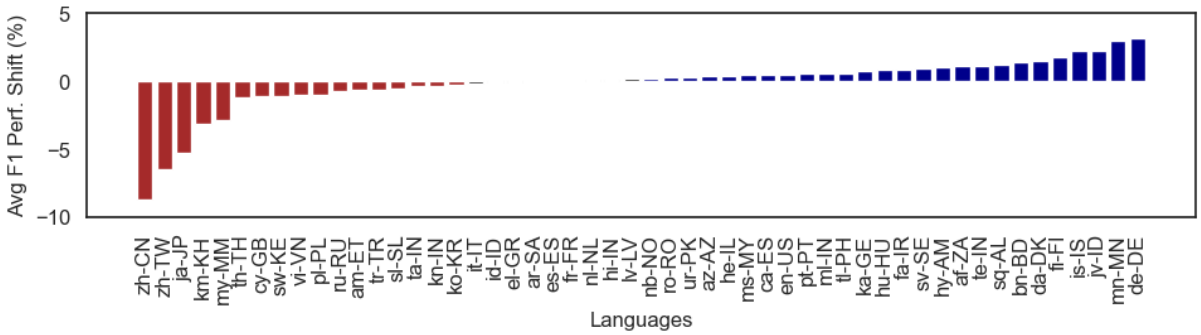

Figure 3: Performance change after training a certain language on x-axis in sequential training (VANILLA). Chinese (zh-CN) exhibits the most significant performance decline, while German (de-DE) serves as the most effective donor language, enhancing overall performance.

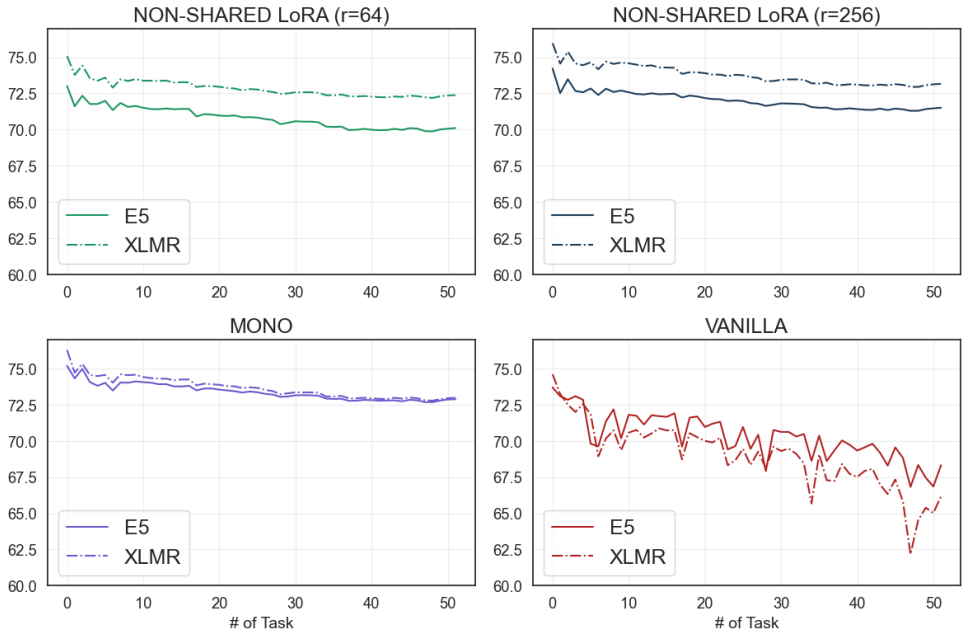

Figure 4: Comparison results between XLM-R and E5 models.

Crucially, increasing the LoRA rank—while expanding the model's capacity—does not substantially improve performance. For instance, MULTI with rank 32 (74.19%) performs nearly as well as at rank 256 (74.11%), and similar diminishing returns are observed across both SHARED and NON-SHARED LoRA. This trend extends to transfer metrics: Table 2 shows that higher rank under SHARED LoRA does not significantly improve forward transfer—CFT remains within the narrow band of 0.51–0.53. These results highlight a key trade-off: higher trainable parameters generally improve performance, but the efficiency of parameter usage varies across methods. The MULTI method provides the best balance between parameter efficiency and performance, while LoRA-based approaches demonstrate potential for parameter-efficient training at the cost of reduced performance. However, it should be noted that the MULTI method might not be trainable in parallel like the NON-SHARED LoRA method. Hence, in some scenarios, the NON-SHARED LoRA method should be considered.

Trends Between Models. Figure 2 illustrates how different training strategies affect performance over time. A key trend is that MULTI method (dotted lines), trained jointly on all languages, exhibit consistent performance, maintaining F1-scores above 73% throughout training. In contrast, sequential learning models show clear signs of degradation as training progresses. The VANILLA model suffers from moderate catastrophic forgetting, with F1-score reductions of 10–15 points. SHARED LoRA

fares worse, degrading by as much as 15–30 points across tasks. Meanwhile, NON-SHARED LoRA offers more stable performance across steps, ranging between 70–73% and demonstrating greater resilience to forgetting.

These observations are further supported by Table 2, which reports backward and forward transfer scores. The VANILLA model achieves a CBT of $-\mathbf{0.08}$ and CFT of $\mathbf{0.55}$, suggesting that while it suffers from forgetting, it still generalizes reasonably well to future tasks. SHARED LoRA, however, shows consistently more negative CBT scores (-0.13 to -0.14), confirming its vulnerability to catastrophic forgetting. This performance is also reflected in CFT, where the scores are also lower than VANILLA method. Together, these findings underscore the importance of balancing task generalization and knowledge retention, particularly in continual cross-lingual setups.

Method	CBT	CFT
VANILLA	**-0.08**	**0.55**
SHARED LoRA		
$r = 32$	-0.13	0.52
$r = 64$	-0.12	0.53
$r = 256$	-0.14	0.51

Table 2: CBT and CFT metrics for VANILLA and SHARED LoRA models — higher values indicate better performance.

Comparison XLM-R and E5 Models. Figure 4 presents a comparison of XLM-R and E5 models across different training methods. Despite variations in methodology, the general pattern of results remains consistent across models. Overall, XLM-R performs better than E5, except in VANILLA method where E5 tends to outperform XLM-R_{BASE}, though performance degradation due to forgetting is still evident. The results suggest that while different methods and model architectures influence the degree of forgetting, the overall trend of performance degradation remains a common characteristic across all settings.

5 Analysis on Languages

To frame our analysis, we interpret MFT as measuring a language's ability to **donate** knowledge to subsequent languages, while MBT reflects how well a language **receives** and retains knowledge after subsequent training steps. This donor-receiver perspective allows us to reason about asymmetries in cross-lingual transfer.

5.1 Languages Affect Forgetting

The results reveal that certain languages significantly impact the model's capacity to retain prior knowledge. Training on Chinese (zh-CN), Japanese (ja-JP), and Traditional Chinese (zh-TW) consistently leads to the most pronounced cases of catastrophic forgetting. This is evidenced by their strongly negative MBT values in Figure 5 and severe performance degradation in Figure 3, particularly when these languages are introduced later in the training sequence. As receivers, these languages appear highly vulnerable to interference from prior tasks. More detailed explanation can be seen in Appendix A. In contrast, languages such as Norwegian (nb-NO), Catalan (ca-ES), Portuguese (pt-PT), and Greek (el-GR) show some of the highest MBT scores across hop distances. These languages maintain stability when trained after others and also preserve prior task performance, indicating they are robust receivers. Interestingly, they may also act as indirect donors by not interfering with earlier knowledge.

However, not all performance trends align perfectly with MBT. For example, German (de-DE) appears beneficial in performance drop metrics (Figure 3), yet does not rank highly in MBT. This suggests that its apparent advantage may be due to its position in the training sequence—e.g., being trained before high-forgetting languages—rather than any inherent ability to preserve earlier knowledge. This underscores an important point: interpreting language influence solely through performance drop can be misleading. MBT offers a more principled, sequence-agnostic perspective on which languages genuinely aid in preserving prior knowledge and resisting catastrophic forgetting.

5.2 Latin vs. Non-Latin Scripts

Script similarity plays a significant role in cross-lingual knowledge transfer. In both MFT and MBT heatmaps, we observe that languages using Latin scripts—such as es-ES, fr-FR, and de-DE—tend to be strong donors and stable receivers. They benefit more from training on other languages and also suffer less from catastrophic forgetting. This likely reflects greater subword token overlap and lexical similarity, which help preserve learned representations under shared tokenization.

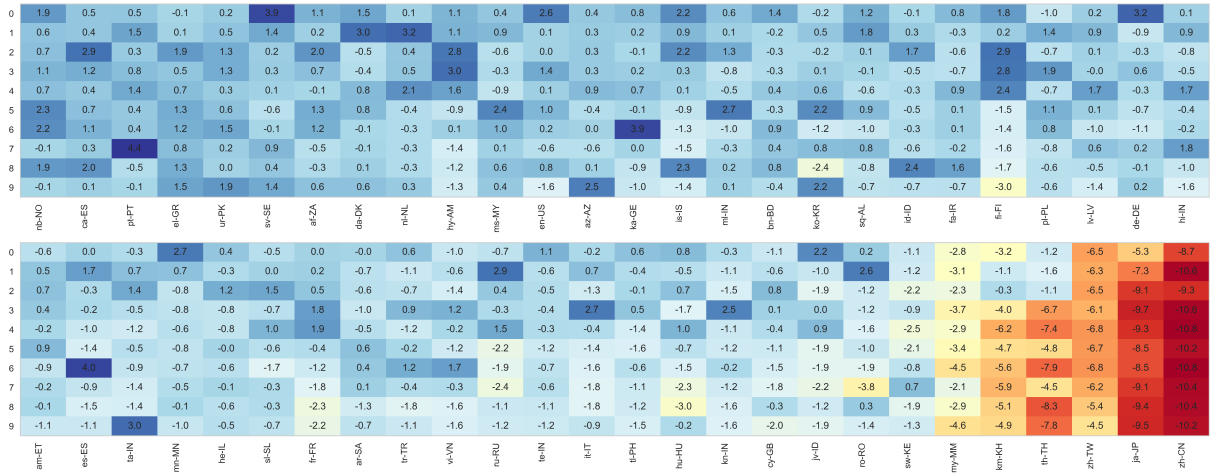

Figure 5: Heatmap of Multi-hop Backward Transfer (MBT), illustrates how training on later languages affects earlier ones over increasing hop distances (y-axis: 0–9). Cooler colors indicate positive backward transfer, while warmer colors reflect degradation in performance. Orders of the language is sorted descending (read from top-left to bottom-right) based on its average over all hops.

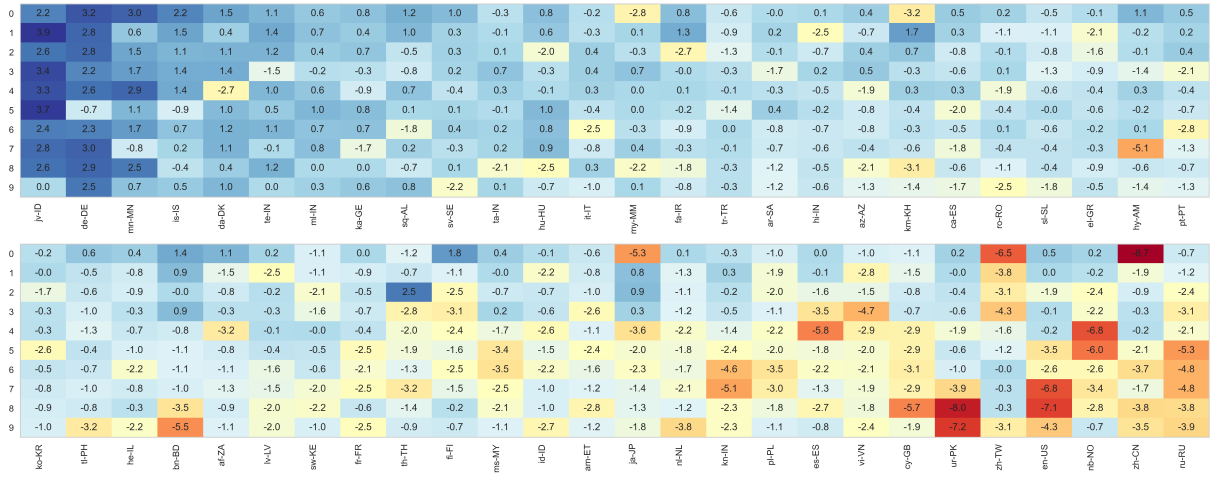

Figure 6: Heatmap of Multi-hop Forward Transfer (MFT), represents each language's ability to donate knowledge to subsequent tasks over increasing hop distances (y-axis: 0–9). Cooler colors indicate stronger positive transfer, while warmer colors reflect limited or negative influence on future learning. Orders of the language is sorted descending (read from top-left to bottom-right) based on its average over all hops.

In contrast, non-Latin script languages, especially those using logographic (e.g., `zh-CN`) or abugida scripts (e.g., `th-TH`, `hi-IN`), tend to be weak donors and vulnerable receivers. These languages show low MFT—suggesting limited forward transfer to other tasks—and highly negative MBT, indicating susceptibility to forgetting. The subword tokenizer, likely optimized for Latin-based alphabets, aggravates this imbalance. This highlights a fundamental challenge for multilingual continual learning: shared vocabulary spaces can lead to representational dominance of Latin-script languages, marginalizing others.

5.3 Language Family

While language family information is not explicitly modeled, typologically or lexically similar languages often demonstrate mutual reinforcement in transfer. Under the donor-receiver lens, we observe that Romance languages such as `es-ES`, `pt-PT`, and `fr-FR` frequently act as strong donors (high MFT) and reliable receivers (stable MBT), especially when positioned near each other in the training sequence. Similarly, Germanic languages like `nl-NL`, `sv-SE`, and `de-DE` show stable transfer interactions.

However, these patterns are not universal. The apparent family-related benefits may arise from

shared scripts and vocabulary rather than deep structural similarity. For instance, several Indo-European languages from different branches perform well together, likely due to orthographic overlap. Conversely, languages from distant families—such as Sino-Tibetan (zh-CN), Austroasiatic (km-KH), or Afro-Asiatic (ar-SA)—often act as poor receivers (low MBT) and limited donors (low MFT), especially when sequenced after typologically dissimilar languages. Future work could explicitly incorporate phylogenetic distances to better disentangle the impact of language family on multilingual continual learning.

5.4 Language Vitality

Language vitality—encompassing speaker population, data availability, and digital presence—also plays a nuanced role in continual learning dynamics. As receivers, high-vitality languages such as zh-CN, ja-JP, and hi-IN (Joshi et al., 2020) show some of the most negative MBT scores, indicating that they are especially vulnerable to forgetting. Surprisingly, they also make relatively poor donors, as reflected in lower MFT scores compared to more typologically compatible mid-vitality languages.

This counterintuitive trend is clarified in Table 1, where mid-vitality languages (Joshi et al., 2020) consistently achieve the highest F1 scores across model variants. These languages appear to strike a balance: they share enough structure with other languages to act as effective donors, while remaining resilient as receivers under sequential training. In contrast, high-vitality languages—despite abundant resources—struggle under parameter-efficient continual learning setups. Their unique token distributions and structural divergence make them harder to adapt to and easier to overwrite. These findings suggest that vitality-aware scheduling or modularization may be critical for improving cross-lingual robustness in continual learning scenarios.

6 Related Work

Catastrophic forgetting is a significant challenge in neural networks, where models lose previously acquired knowledge when fine-tuned on new tasks (McCloskey and Cohen, 1989). This issue is particularly pronounced in multilingual contexts, as models must adapt to new languages without degrading performance on previously learned ones (Winata et al., 2023a). To mitigate this, various strategies have been proposed, including memory replay (Rolnick et al., 2019), regularization techniques (Kirkpatrick et al., 2017), and architectural innovations like progressive networks (Rusu et al., 2016).

Lifelong learning also known as continual learning, is an emerging approach that enables models—particularly LLMs and their agents—to continuously acquire new knowledge while retaining prior capabilities. This knowledge can be integrated into LLMs either by updating model parameters through training or adapters, or by leveraging external sources like Wikipedia or tools without modifying the model itself or knowledge base (Zheng et al., 2024). Recent work extends lifelong learning to agent-based settings, decomposing it into perception, memory, and action modules that together support continuous adaptation (Zheng et al., 2025).

For internal knowledge updates, adapters have proven to be a lightweight and effective solution, introducing small, task-specific modules that can be fine-tuned independently, reducing interference across tasks (Houlsby et al., 2019; Winata et al., 2021; Hu et al., 2021). The MAD-X framework (Pfeiffer et al., 2020b) enhances cross-lingual transfer by separating language and task adaptation, while language-specific adapters balance specialization and sharing (Badola et al., 2023). Additionally, methods like AdapterFusion (Pfeiffer et al., 2020a) combines task-specific adapters through a learned composition layer, promoting parameter sharing and effective transfer learning while minimizing forgetting.

7 Conclusion

Our paper highlights the critical challenges of catastrophic forgetting in cross-lingual transfer for multilingual NLP models with 52 languages. We provide insights into how various parameter-sharing strategies can influence knowledge retention and overall model performance. Our findings indicate that partial parameter sharing can effectively mitigate forgetting while maintaining performance, presenting a promising approach for developing more robust multilingual NLP systems. Additionally, we identify that certain languages during training can negatively impact performance, contributing to catastrophic forgetting. Overall, this research enhances the ongoing efforts to improve the adaptability and efficiency of NLP models in real-world NLP applications.

Limitations

In this paper, we concentrate our investigation on XLM-R model and use E5, rather than exhaustively evaluating every possible model due to resource constraints. This focused approach allows us to provide a more in-depth analysis of these models and their performance in cross-lingual contexts.

Ethical Considerations

In our evaluation of language models for multilingual tasks, we place strong emphasis on transparency and fairness. We carefully design and document our data collection and evaluation methodologies to ensure they are consistent, unbiased, and reproducible. By applying uniform assessment criteria across models, we aim to enable meaningful and equitable comparisons.

References

Kartikeya Badola, Shachi Dave, and Partha Talukdar. 2023. Parameter-efficient finetuning for robust continual multilingual learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9763–9780.

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

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, et al. 2023. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4277–4302.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, et al. 2019. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

Edward 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*.

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

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

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*.

Zihan Liu, Genta Indra Winata, Andrea Madotto, and Pascale Fung. 2021. Preserving cross-linguality of pre-trained models via continual learning. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 64–71, Bangkok, Thailand (Online). Association for Computational Linguistics.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Elsevier.

Meryem M'hamdi, Xiang Ren, and Jonathan May. 2023. Cross-lingual continual learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3908–3943, July 9-14, 2023. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673. Association for Computational Linguistics.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, et al. 2016. Progressive neural networks. In *arXiv preprint arXiv:1606.04671*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Genta Winata, Lingjue Xie, Karthik Radhakrishnan, Shijie Wu, Xisen Jin, Pengxiang Cheng, Mayank Kulkarni, and Daniel Preoţiuc-Pietro. 2023a. Overcoming catastrophic forgetting in massively multilingual continual learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 768–777.

Genta Indra Winata, Alham Fikri Aji, Samuel Cahyawijaya, Rahmad Mahendra, Fajri Koto, Ade Romadhony, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Pascale Fung, et al. 2023b. Nusax: Multilingual parallel sentiment dataset for 10 indonesian local languages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 815–834.

Genta Indra Winata, Frederikus Hudi, Patrick Amadeus Irawan, David Anugraha, Rifki Afina Putri, Yutong Wang, Adam Nohejl, Ubaidillah Ariq Prathama, Nedjma Ousidhoum, Afifa Amriani, et al. 2024. Worldcuisines: A massive-scale benchmark for multilingual and multicultural visual question answering on global cuisines. *arXiv preprint arXiv:2410.12705*.

Genta Indra Winata, Guangsen Wang, Caiming Xiong, and Steven Hoi. 2021. Adapt-and-adjust: Overcoming the long-tail problem of multilingual speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, page 361.

Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. 2024. Towards lifelong learning of large language models: A survey. *arXiv preprint arXiv:2406.06391*.

Junhao Zheng, Chengming Shi, Xidi Cai, Qiuke Li, Duzhen Zhang, Chenxing Li, Dong Yu, and Qianli Ma. 2025. Lifelong learning of large language model based agents: A roadmap. *arXiv preprint arXiv:2501.07278*.

A Detailed Results

A.1 Language Order

Table 3 presents the language orders used in the sequential training experiments. These orders are used to train models in a step-by-step fashion, where each iteration introduces a new language. The results from these training sequences are subsequently used to compute aggregate metrics, as shown in Figure 2 and Figure 4.

The first order is derived based on the amount of language resources available in the XLM-R model (Conneau et al., 2020). This order reflects the relative training data size used during XLM-R's pretraining, with high-resource languages appearing earlier in the sequence. The remaining orders (2 through 5) are randomly shuffled variants to introduce diversity and reduce potential order bias. However, in the fifth order, languages that are found to be particularly destructive—i.e., those that tend to cause performance degradation on previously learned languages—are deliberately placed toward the end of the sequence. This design allows us to analyze how the position of destructive languages affects knowledge retention and transfer in sequential multilingual training.

A.2 Heatmap on `VANILLA` method for first language order

The heatmap on Figure 7 provides a detailed visualization of the model's performance across training iterations (represented by rows) and evaluated languages (represented by columns). In each iteration, the model is trained on a new language. For instance, as shown in the figure, the first iteration trains on `en-US`, the second on `ru-RU`, the third on `id-ID`, and so forth. After training on a language, the model's performance on that language typically improves. This trend is reflected in the heatmap: the lower-left triangle (below the diagonal), corresponding to previously learned languages, tends to display cooler colors, indicating better performance; in contrast, the upper-right triangle (unlearned languages) often exhibits warmer colors, reflecting performance degradation.

This visualization clearly highlights cross-lingual interactions—specifically, how training on a new language can either benefit or harm performance on other languages. For example, in row 18, where the model is trained on `zh-CN`, the corresponding row becomes noticeably warmer compared to previous iterations, suggesting a general decline in performance across many languages. However, for linguistically related languages such as `ja-JP`, where many Kanji characters overlap with Chinese characters (hence vocabulary overlap), performance actually improves. This suggests that while `zh-CN` introduces interference for many languages, it serves as a helpful donor for `ja-JP`—likely due to shared orthographic features, such as the incorporation of Chinese characters in the Japanese writing system.

Order	Languages in ISO 639-1
1	en-US, ru-RU, id-ID, vi-VN, fa-IR, th-TH, ja-JP, de-DE, ro-RO, hu-HU, fr-FR, fi-FI, ko-KR, es-ES, pt-PT, nb-NO, el-GR, zh-CN, da-DK, pl-PL, he-IL, it-IT, nl-NL, ar-SA, tr-TR, hi-IN, zh-TW, ta-IN, sv-SE, sl-SL, ca-ES, ka-GE, lv-LV, ms-MY, bn-BD, ml-IN, az-AZ, ur-PK, hy-AM, sq-AL, te-IN, kn-IN, is-IS, tl-PH, mn-MN, my-MM, sw-KE, km-KH, af-ZA, am-ET, cy-GB, jv-ID
2	tr-TR, ro-RO, ur-PK, es-ES, hi-IN, pl-PL, hy-AM, sv-SE, sl-SL, ta-IN, te-IN, ml-IN, id-ID, ka-GE, el-GR, ko-KR, de-DE, fa-IR, ms-MY, ca-ES, az-AZ, nl-NL, pt-PT, fr-FR, hu-HU, sw-KE, mn-MN, he-IL, zh-CN, fi-FI, ru-RU, is-IS, cy-GB, ja-JP, sq-AL, vi-VN, th-TH, jv-ID, it-IT, my-MM, kn-IN, lv-LV, am-ET, nb-NO, ar-SA, en-US, af-ZA, zh-TW, bn-BD, da-DK, km-KH, tl-PH
3	sv-SE, nl-NL, fi-FI, kn-IN, hu-HU, ms-MY, es-ES, my-MM, is-IS, ko-KR, af-ZA, vi-VN, bn-BD, tr-TR, tl-PH, lv-LV, ru-RU, fr-FR, en-US, ro-RO, am-ET, he-IL, hi-IN, ja-JP, te-IN, id-ID, ta-IN, it-IT, jv-ID, nb-NO, ka-GE, sq-AL, ca-ES, az-AZ, zh-TW, fa-IR, mn-MN, zh-CN, de-DE, da-DK, ml-IN, sw-KE, sl-SL, km-KH, ar-SA, pt-PT, cy-GB, ur-PK, hy-AM, el-GR, pl-PL, th-TH
4	nb-NO, ta-IN, th-TH, fi-FI, ru-RU, af-ZA, vi-VN, ko-KR, ro-RO, km-KH, is-IS, ms-MY, sl-SL, en-US, hi-IN, he-IL, bn-BD, pt-PT, fa-IR, sv-SE, am-ET, kn-IN, az-AZ, tl-PH, ar-SA, nl-NL, cy-GB, hy-AM, it-IT, de-DE, da-DK, te-IN, hu-HU, lv-LV, zh-CN, mn-MN, es-ES, ca-ES, pl-PL, fr-FR, ja-JP, ka-GE, sw-KE, id-ID, zh-TW, jv-ID, sq-AL, el-GR, tr-TR, my-MM, ml-IN, ur-PK
5	mn-MN, ml-IN, is-IS, fa-IR, az-AZ, pl-PL, de-DE, ko-KR, ar-SA, sw-KE, jv-ID, sq-AL, tl-PH, ru-RU, lv-LV, fr-FR, ro-RO, ka-GE, cy-GB, tr-TR, he-IL, sl-SL, af-ZA, nl-NL, my-MM, hu-HU, hi-IN, vi-VN, it-IT, pt-PT, da-DK, ca-ES, am-ET, el-GR, ta-IN, id-ID, te-IN, sv-SE, bn-BD, ur-PK, en-US, kn-IN, ms-MY, nb-NO, es-ES, fi-FI, zh-TW, zh-CN, ja-JP, th-TH, km-KH, hy-AM

Table 3: Language orders in the sequential training experiments.

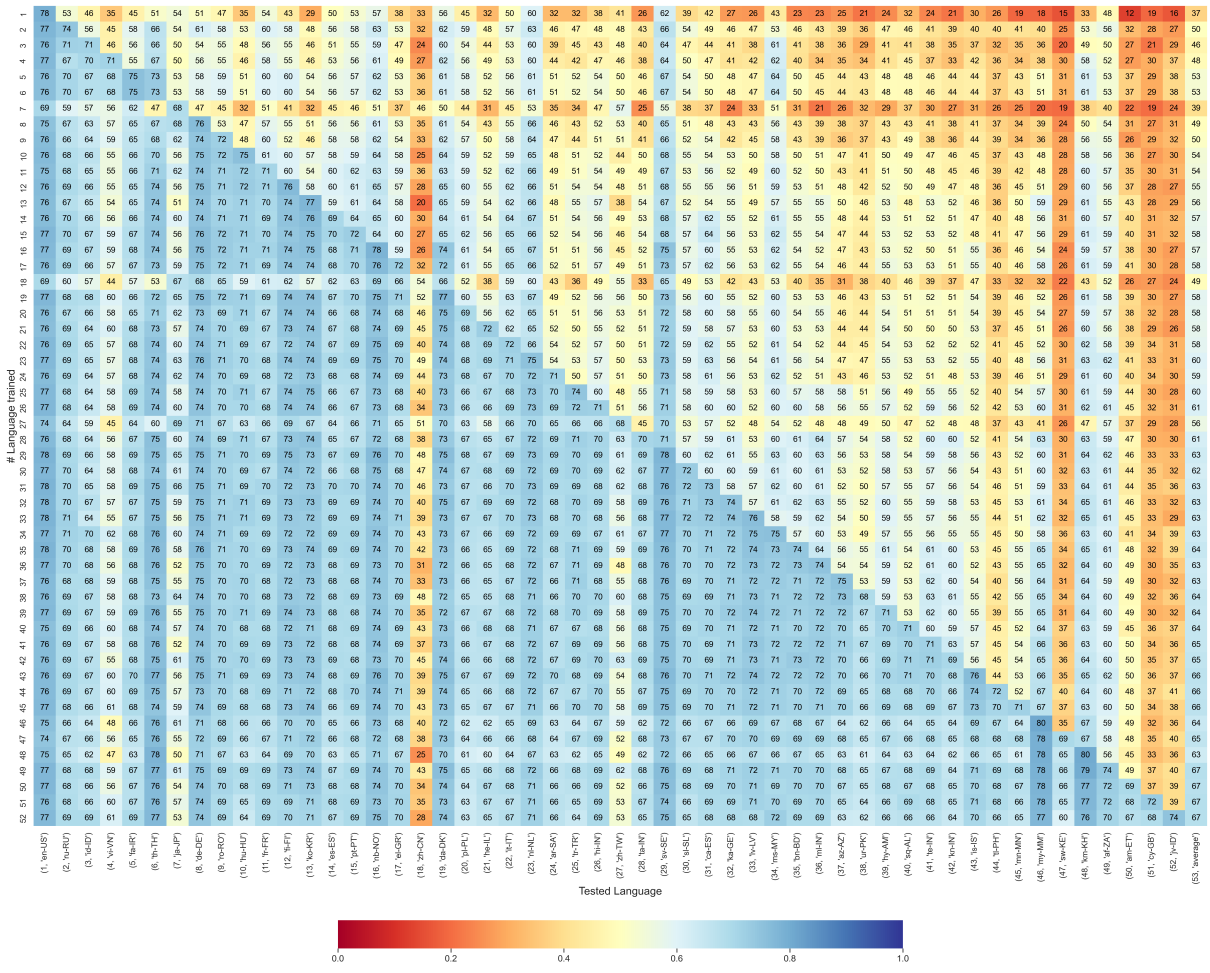

Figure 7: Heatmap on VANILLA method for first language order.