Cross-lingual Lifelong Learning

Anonymous ACL submission

Abstract

The longstanding goal of multi-lingual learning has been to develop a universal cross-lingual model that can withstand the changes in multi-lingual data distributions. However, most existing models assume full access to the target languages in advance, whereas in realistic scenarios this is not often the case, as new languages can be incorporated later on. In this paper, we present the **C**ross-lingual **L**ifelong **L**earning (CLL) challenge, where a model is continually fine-tuned to adapt to emerging data from different languages. We provide insights into what makes multilingual sequential learning particularly challenging. To surmount such challenges, we benchmark a representative set of cross-lingual continual learning algorithms and analyze their knowledge preservation, accumulation, and generalization capabilities compared to baselines on carefully curated datastreams. The implications of this analysis include a recipe for how to measure and balance between different cross-lingual continual learning desiderata, which goes beyond conventional transfer learning.

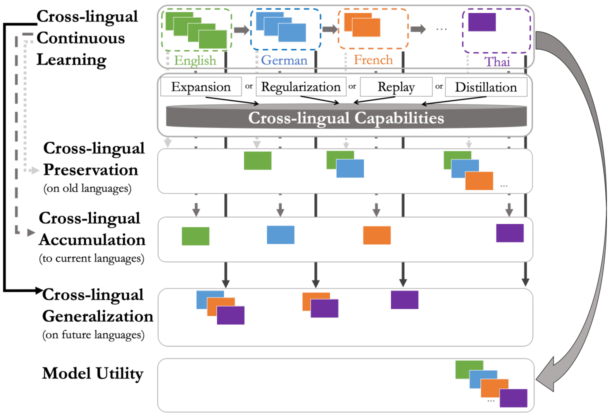


Figure 1: An overview of CLL: We use an example of a non-stationary datastream moving from high to low resource languages. To support this problem setting, we *evaluate the cross-lingual capabilities of continual approaches* such as model expansion, regularization, replay, and distillation. Those capabilities include knowledge **preservation** on old languages, **accumulation** to the current language, **generalization** to unseen languages, and **model utility** at the end of training.

1 Introduction

With more than 7,000 languages spoken around the globe, downstream applications still lack proper linguistic resources across languages (Joshi et al., 2020), necessitating the use of *transfer learning* techniques that take advantage of data that is mismatched to the application. In an effort to simplify architecture complexity and energy consumption, it is desirable to unify multi-lingual performance into a single, parameter- and memory-constrained model, and to allow this model to evolve, learning on multi-lingual training data as it becomes available. Such is the longstanding goal of language representation learning. Existing multi-lingual representations such as M-BERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) are strong pillars in cross-lingual transfer learning, but if care is not taken when choosing how to train, they can neglect to maximize transfer and are subject to *forgetting* (French, 1993), where performance decreases after exposure to some new task or language.

Most previous work that attempts to deal with the challenge of transfer exploitation and forgetting mitigation focuses on the problem of sequentially learning over different NLP downstream tasks or domains (Sun et al., 2020; Han et al., 2020; Madotto et al., 2021), rather than on language shifts. Indeed, the current literature for learning over sequences of languages is rather scarce, and mostly focuses on cross-lingual transfer learning between a pair of languages. Previous works that fall into that category include Liu et al. (2021) and Garcia et al. (2021). Liu et al. pre-train a (parent) language model and then fine-tune it on a downstream task in one of several different (child) languages. This "two-hop" case conflates task transfer and language transfer, and confuses analysis – the interference between the pre-trained language model 'task' and the fine-tuned task along with the parent and child languages cannot be disentangled. Garcia et al. focus on sequentially learning over two sets of parent and children language pairs in machine translation. However, this still focuses on the 'two-hop' case; the effect of multiple shifts in the datastream is not trivially generalizable to more than two hops. Garcia et al. also constrain their focus to the mitigation of forgetting with the objective of adapting better to new languages. This is an almost exclusive focus in continual learning literature (Lopez-Paz and Ranzato, 2017; Hayes et al., 2018). However, there is more than forgetting while sequentially learning over multiple languages. We need a more **robust and balanced evaluation between different cross-lingual continual learning desiderata that balance the dynamics of transfer and generalization** in addition to forgetting.

In this paper, we prescribe a *multi-hop* continual learning evaluation that simulates sequentially learning a single task, as the multi-lingual model is exposed to training data from *different languages*. We formulate the Cross-lingual Lifelong Learning challenge and experiment with *balanced streams* of n data scenarios for $n > 2$. Unlike previous work, this paper defines comprehensive goals including knowledge preservation, accumulation, generalization, and model utility as guidelines for analyzing the cross-lingual capabilities of multilingual sequential training. To measure them, we **define evaluation metrics and tweak data distributions and language permutations** to investigate (1) the capabilities and obstacles of a multi-lingual language model in preserving and accumulating knowledge across different languages and (2) the effectiveness of different continual learning algorithms in mitigating those challenges.

We apply this test bed to a six-language task-oriented dialogue task and analyze a wide variety of successful continual learning algorithms in that context. We cover a representative set of approaches spanning over: (a) model-expansion approaches (Pfeiffer et al., 2020b), (b) regularization-based (Kirkpatrick et al., 2017), (c) memory replay (Chaudhry et al., 2019b), and (d) distillation-based (Hinton et al., 2015; Aguilar et al., 2020). Our findings confirm the need for a multi-hop analysis and the effectiveness of continual learning algorithms, especially model expansion and memory replay approaches, in enhancing knowledge preservation and accumulation of M-BERT. We additionally demonstrate the robustness of different continual learning approaches to variations in individual data setup choices that would be misleading if presented in a traditional manner.

Our **main contributions** are: (1) We are the first to explore and analyze cross-lingual continual fine–tuning[1] across multiple hops and show the importance of this multi-hop analysis in reaching clearer conclusions with greater confidence compared to conventional cross-lingual transfer learning (§4.5). (2) We evaluate the aggregated effectiveness of a range of different continual learning approaches (Figure 1) at reducing forgetting and improving transfer (§4.2). (3) We show that that the order of languages and data set size impacts the knowledge preservation and accumulation of multi-lingual sequential fine-tuning and that certain continual learning approaches bridge that gap (§4.3). (4) We make concrete recommendations on model design to balance transfer and final model performance with forgetting (§4.2). (5) We analyze zero-shot generalization trends and their correlation with forgetting (§4.4).

2 Cross-lingual Continual Learning

We first formally define cross-lingual lifelong learning, its goals and challenges, the downstream tasks and datastreams, the analysis setup goals, and the evaluation protocols that support them.

2.1 Problem Formulation

We define cross-lingual lifelong learning as the problem of sequentially fine-tuning the Transformer-based model θ for a particular downstream task over a cross-lingual data stream. Let $\mathscr{L} = \{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2 \cdots \boldsymbol{\ell}_N\}$ be a set of labeled *languages*, let $\mathfrak{S}(\mathscr{L})$ be the set of all *permutations* of $\mathscr{L}$, and without loss of generality let $p \in \mathfrak{S}(\mathscr{L})$ be one such permutation and let $p[i] \in \mathscr{L}$ be the ith language in p . In this case, a training data *stream* is made of N labeled and distinct datasets $\mathcal{D}_{1 \cdots N}$, each consisting of separate train and test portions. The language of $\mathcal{D}_i$ is $p[i]$. Let *hop* i be the stage in cross-lingual lifelong learning where θ_{i-1} is optimized to θ_i via exposure to $\mathcal{D}_i$. Let $\mathcal{D}_{<i}$ and $\mathcal{D}_{>i}$ refer to a sequence of dataset (train or test portions, depending on context) used in hops from 1 to i and i to N (excluding i), respectively.

[1] To encourage future research in this direction, we release our github repository in the camera-ready version.

2.2 Goals

We define the goals for our study of cross-lingual lifelong learning as follows (also depicted in Figure 1): 1) *Cross-lingual preservation.* This is the ability to retain previous knowledge on seen languages. 2) *Cross-lingual accumulation.* This is the ability to accumulate knowledge learned from previous languages to benefit the learning on current language. 3) *Cross-lingual generalization.* This goes beyond learning for the current languages towards generalizing uniformly well to unseen languages. 4) *Model utility.* This tests how well we can use one final model for all languages.

2.3 Challenges

Learning sequentially from a non-stationary data distribution (i.e., task datasets coming from different languages) can impose considerable challenges on the goals defined earlier: 1) *Catastrophic forgetting.* This happens when fine-tuning a model on $\mathcal{D}_{\geq i}$ leads to a decrease in the performance on $\mathcal{D}_{<i}$. 2) *Negative transfer.* This happens when fine-tuning a model up to $\mathcal{D}_{\leq i}$ leads to a lower performance on $\mathcal{D}_i$ than training on it alone. 3) *Low zero-shot transfer.* This happens when fine-tuning on $\mathcal{D}_{\leq i}$ gives a low performance than on unseen $\mathcal{D}_{>i}$. 4) *Low final performance.* This happens when fine-tuning on all $\mathcal{D}_{\leq N}$ gives a low performance when tested on $\mathcal{D}_{\leq N}$ at the end of training.

2.4 Downstream Tasks and Datastreams

Here, we describe the downstream tasks and multi-lingual sequential datastreams used.

Downstream Tasks. We choose task-oriented dialogue parsing as a use case and consider the multi-lingual task-oriented parsing (MTOP) benchmark (Li et al., 2021). Task-oriented dialogue parsing provides a rich testbed for analysis, as it encompasses two subtasks: *intent classification* and *slot filling*, thus allowing us to test different task capabilities in cross-lingual continual learning.

Data Stream Construction. For a set of N languages $\mathscr{L}$, our study considers a permutation subset $P \subset \mathfrak{S}(\mathscr{L})$ according to the following properties:[2]

- $|P| = |\mathscr{L}| = N$, where $\forall \boldsymbol{\ell}_i \in P$ appears exactly once in each stream.
- $\forall \boldsymbol{\ell}_i \in \mathscr{L}, \forall j \in 1 \ldots N$, there exists some $p \in P$ such that $p[j] = \boldsymbol{\ell}_i$.

[2]Details of the different language permutations used for the data streams can be found in Appendix B.1.

Lang	ISO	Train / Dev / Test Original Version	Train / Dev / Test Balanced Version
English	EN	15,667 / 2,235 / 4,386	9,219 / 1,285 / 2,299
German	DE	13,424 / 1,815 / 3,549	
French	FR	11,814 / 1,577 / 3,193	
Hindi	HI	11,330 / 2,012 / 2,789	
Spanish	ES	10,934 / 1,527 / 2,998	
Thai	TH	10,759 / 1,671 / 2,765	

Table 1: Statistics of MTOP per language and split.

- high2low $\in P$, the permutation from most high-resource to most low-resource fine-tuning data sets, based on the training dataset size.
- low2high $\in P$, the permutation from most low-resource to most high-resource fine-tuning data sets, based on the training dataset size.

We use MTOP which is a multi-lingual dataset covering 6 typologically diverse languages and spanning over 11 domains. In this evaluation, we use only the decoupled representation. We use the original data for most experiments. For one additional ablation study, we fix the distribution of the training, development, and testing sentences following a balanced distribution over the intents for all languages. Table 1 shows a summary of the statistics per language and split for both versions.

2.5 Analysis Setup

We provide an extensive analysis in the form of different ablation studies. These revolve around the continual learning goals, described in §2.2.

Q1. Can a multi-lingual language model learn to preserve and accumulate knowledge across different languages? Specifically, we investigate whether multi-lingual sequential fine-tuning can accumulate and retain knowledge and how well its final checkpoint can be used for all languages at the same time. This is a fundamental question to help us determine if the use of continual learning is needed at all to perform sequential cross-lingual fine-tuning. We investigate the performance of the baseline and reference models (§3.1) using the meta-metrics (§2.6), on the average over language permutations and the original version of the dataset set shown in Table 1 (§2.4).

Q2. Are continual learning algorithms effective in boosting knowledge preservation and accumulation compared to naive sequential fine-tuning? We compare different continual learning algorithms, analyze their accumulation capabilities and final model utility in reaching a compromise between them and retaining previous knowledge. For that purpose, we compare the performance of the algorithms (§3.2) using the second and third metrics (§2.6) and analyze their relationship to knowledge preservation (first metric), taking the average over language permutations (§2.4).

Q3. Which language permutations impose more challenges on knowledge preservation and accumulation? We wish to understand the role of language order in knowledge preservation, accumulation, and final model utility of multilingual sequential fine-tuning and which continual learning approaches bridge the gap between different language permutations. We use the same experiment plan as in questions Q1 and Q2 with respect to different languages permutations and the original version of the dataset (§2.4). For additional ablation studies on the role of fine-tuning data set size, we use the balanced dataset.

Q4. How do different continual learning models generalize to unseen languages? We analyse the zero-shot generalization to unseen languages in the stream. For that purpose, we look at several continual learning models and compare them to the baseline over the average of different language permutations in terms of the last metric (§2.6). We also analyze the relationship between generalization and preservation to check for any correlations or trade-offs.

Q5. How is a multi-hop different from two-hop continual learning analysis? Finally, we wish to investigate which insights a multi-hop analysis over multiple languages in the stream provides us with that is different from the conventional two-hop cross-lingual continual transfer learning from a source to a target language. For this purpose, we conduct several experiments involving only the first and last language in each stream (§2.4) to compare them to the corresponding full stream involving the remaining languages in between.

2.6 Evaluation Protocols

Let R be some metric for evaluating K and $R_{i,\leq j}$ be the evaluation on test set for language ℓ_i using a model trained on $\mathcal{D}_{1\cdots j}$, we define the following *meta-metrics* (which are inspired, but slightly different from the metrics defined in Lopez-Paz and Ranzato (2017) and Chaudhry et al. (2019a)):

- **Forgetting (F)** $\downarrow$. This is the average forgetting *over all hops* (excluding the first hop as no forgetting occurred yet) computed as: $F = \frac{1}{N-1}\sum_{j=2}^{N} F_{\leq j}$ (1), such that $F_{\leq j} = \frac{1}{j-1}\sum_{i=1}^{j-1} F_{i,\leq j}$ (2) is the average forgetting that occurred at hop i. We compute $F_{i,\leq j} = \max_{k\subseteq[1,j-1]} R_{i,\leq k} - R_{i,\leq j}$ (3), where $F_{i,\leq j}$ is the degree to which performance on $\mathcal{D}_i$ has suffered by continuing to train up to $\mathcal{D}_j$ instead of stopping before $\mathcal{D}_{j-1}$.
- **Transfer (T)** $\uparrow$. This is the average forward transfer computed as: $T = \frac{1}{N-1}\sum_{i=2}^{N} T_i$ (4), such that $T_i = R_{i,\leq i} - R_i$ (5), where R_i denotes evaluation of a model fine-tuned *only* on $\mathcal{D}_i$. T_i is thus the incremental impact of sequentially training on datasets prior to seeing $\mathcal{D}_i$.
- **Final performance (FP)** $\uparrow$. This is the average performance after training on all datasets in the studied stream: $FP = \frac{1}{N}\sum_{i=1}^{N} R_{i,\leq N}$. (6)

To measure *generalization to new languages*, we add a **zero-shot transfer** ($T^0\uparrow$) metric, which is measured as: $T^0 = \frac{1}{N-1}\sum_{i=2}^{N} T_i^0$ (7), where $T_i^0 = \frac{1}{i-1}\sum_{j=1}^{i-1} R_{i,\leq j} - \bar{R}_i$ (8) is the average performance of a model on the forward transfer to a language ℓ_i after seeing all datasets before and not including it compared to the random performance $\bar{R}_i$ before even fine-tuning on any language.

3 Methods

We use the same architecture as in Castellucci et al. (2019); M'hamdi et al. (2021) to jointly learn intent classification and slot filling subtasks on top of M-BERT.[3] In this section, we describe several baselines and continual learning algorithms of how this architecture is trained sequentially or jointly on multiple languages.

3.1 Baselines & Reference Models

Before delving into continual learning approaches, we consider simple baselines,[4] which either train in a sequential multi-hop or a joint one-hop manner and are either language-specific or multi-lingual.

Lower-bound Baseline. This consists of *naive sequential fine-tuning* (*Naive Seq FT*), which sequentially fine-tunes with no continual learning.

Upper-bound Models. These are stronger reference models, as they either train from scratch for

[3]More details about the architecture can be found in Appendix A.

[4]All those baselines and reference models use the same base model architecture and its loss with no further additions or special optimizations to the architecture.

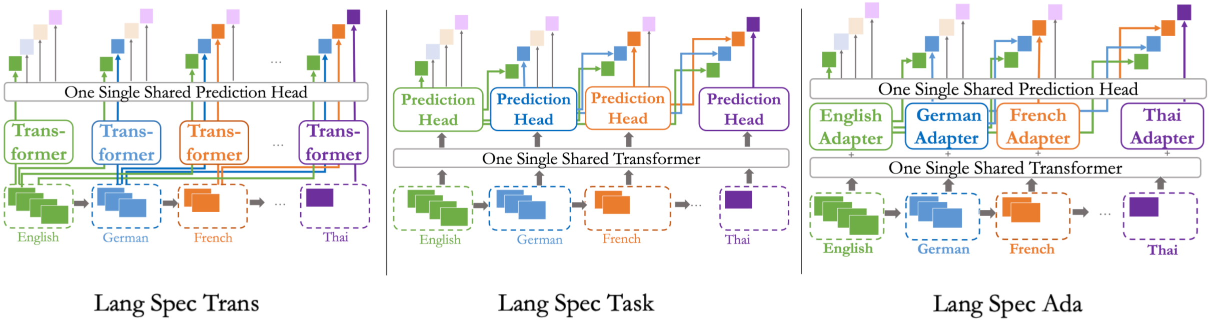


Figure 2: A comparison between different variants of model expansion for this problem setting: either at the side of the input (*Lang-Spec Trans*), the output (*Lang-Spec Task*), or using adapters (*Lang-Spec Ada*).

each new language or have access to all languages:

- *Language-specific fine-tuning* (*Lang-Spec FT*). This is the baseline that trains a model on the data set for each language D_{ℓ_i} independently.
- *Multi-lingual learning* (*Multilingual*). This trains one model jointly across all data sets $\mathcal{D}_{1\cdots N}$.
- *Incremental joint learning (*Inc Joint*)*. This incrementally trains adding the data set for each language in the stream. This consists of the following hops: 1) D_{ℓ_1}, 2) $\mathcal{D}_{\ell_1,\ell_2}$, $\cdots$, and N) $\mathcal{D}_{1\cdots N}$.

3.2 Continual Learning Approaches

To continually fine-tune on different tasks, we establish several strong approaches from the following categories:[5]

Model Expansion. We consider the following approaches shown in Figure 2. We either expand on the input side, i.e. M-BERT representations, (*Lang-Spec Trans*) or on the output side, i.e. the task-specific prediction heads (*Lang-Spec Task*) for each language, while sharing the rest in each case (the output and input respectively). We also separately add MAD-X adapters (Pfeiffer et al., 2020b). We either fine-tune the adapter layers and freeze the rest of M-BERT (*Lang-Spec Ada(F)*) or tune them both (*Lang-Spec Ada(T)*).

Regularization. We focus on elastic weight consolidation (*EWC*) (Kirkpatrick et al., 2017), which tackles catastrophic forgetting by reducing the changes in parameters that are deemed critical to past tasks. We use the online version of EWC (*EWC-Online*) for efficiency purposes.

Memory Replay. We use experience replay (*ER*) (Chaudhry et al., 2019b), which alleviates forgetting by maintaining a fixed-size memory equally balanced between the different languages and regularly drawing examples from the memory to replay.

Distillation-based. On top of ER, we distill dark knowledge from previous model checkpoints. We explore two variants: logit distillation (*KD-Logit*) (Hinton et al., 2015) and representation distillation (*KD-Rep*) (Aguilar et al., 2020), which optimize the minimum square error loss between either the output logits or M-BERT representations between the current and previous models.

4 Results & Analysis

In this section, we present our results and findings for the different analysis questions raised in §2.5. For §4.1, scores are reported using accuracy (Acc) and F1-score (F1) for intent classification and slot filling, respectively.[6] All experiments are run for one single seed and then bootstrap sampling is used to compute the average and confidence intervals over either just the random shuffling of the test data (§4.3) or also averaging over language permutations. More details can be found in Appendix B.3.

4.1 Multi-lingual Sequential Learning

Model	Acc	F1
Naive Seq FT	90.52 ±1.42	69.10 ±1.24
Lang-Spec FT	93.20 ±0.08	73.59 ±0.81
Inc Joint	94.20 ±0.15	74.97 ±0.51
Multilingual	**94.25** ±0.07	**76.34** ±0.82

Table 2: The average final performance across different language permutations for *the baseline compared to reference models*. We highlight the best scores in bold and underline the second best across models.

Our analysis begins with an investigation of how well different baselines and reference models learn to preserve and accumulate knowledge across different languages, by looking at the average over language permutations (Q1 in §2.5). Since not all reference models are sequential, we start by comparing them all to the baseline using their final

[5]More details about the approaches can be found in Appendix A and the hyperparameters used can be found in B.2.

[6]For the remaining sections, all results are reported for intent classification for space efficiency and more results for slot filling can be found in Appendix C.

performances. The final performance is indicative of how well a single final model can encapsulate the knowledge across languages. From Table 2, we notice that *Naive Seq FT* and *Multilingual* have the worst and best final performances, respectively. This suggests that **a multilingual joint model is more beneficial than sequential models**, but in practical scenarios having access to all languages at the same time might be **costly or prohibitive**. While *Lang-Spec FT* improves only over *Naive Seq FT* by 2.68% and 4.49%, it falls behind *Inc Joint* by 1% and 1.38% and *Multilingual* by 1.05% and 2.75% on intent classification and slot filling, respectively. Therefore, **training sequentially is more beneficial than training a model from scratch**, to exploit cross-lingual transfer capabilities.

Model	F↓		T↑	
	Acc	F1	Acc	F1
Naive Seq FT	2.99 ±1.20	6.22 ±0.95	0.76 ±0.09	**1.42** ±0.33
Inc Joint	**0.15** ±0.10	**0.93** ±0.38	**0.85** ±0.12	1.33 ±0.83

Table 3: Forgetting (F) and transfer (T) performance averaged across different language permutations for *sequential baseline and reference models*. We highlight the best models in bold and underline the second best.

We focus, thereafter, more on *Naive Seq FT* and its forgetting and transfer trends compared to *Inc Joint*, which is a sequential variant of the reference model *Multilingual*. *Inc Joint* exhibits significantly less forgetting which also causes its final performance to be higher than *Naive Seq FT*. This suggests that **recalling previously used training data is helpful in knowledge preservation**. However, the difference between the two, in terms of their transfer performance, is not statistically significant.[7] We hypothesize that this could be due to exposing *Inc Joint* to all resources from previously seen languages, so it is **likely that the data distribution between all these languages may distract the model from learning on the new one**.

4.2 The Effectiveness of Continual Learning

To investigate the effectiveness of continual learning approaches in improving knowledge preservation and accumulation, we compare them to the baseline using the average over language permutations (Q2 in §2.5). We show, in scatter plots 3 and 4, the transfer and final performances of different approaches, respectively, as functions of their negative forgetting. In general, we observe that continual learning approaches mitigate forgetting, improve transfer, and final performance compared to *Naive Seq FT* (except for *EWC-Online*, where even the small improvement in transfer is not statistically significant (Appendix D)).

From Figure 4, we notice that model expansion approaches[8](*Lang-Spec Trans* and *Lang-Spec Enc[0-8]*) are the best in mitigating forgetting and improving the final performance unlike *Lang-Spec Task*. This proves that M-BERT, when trained in a language specific manner, is responsible for encapsulating the cross-lingual representations necessary for enabling knowledge preservation, whereas any changes to the downstream task-specific layers do not make much of a difference. This implies that in cross-lingual continual learning **more attention should be paid to how to train those representations in a language-specific manner efficiently**. *Lang-Spec Ada(T)* are one way to do it more efficiently, but its performance still lags behind. *ER* achieves a performance closer to *Lang-Spec Trans* and *Lang-Spec Enc[0-8]*[9] and this suggests that **even tiny bits of memory are beneficial**.

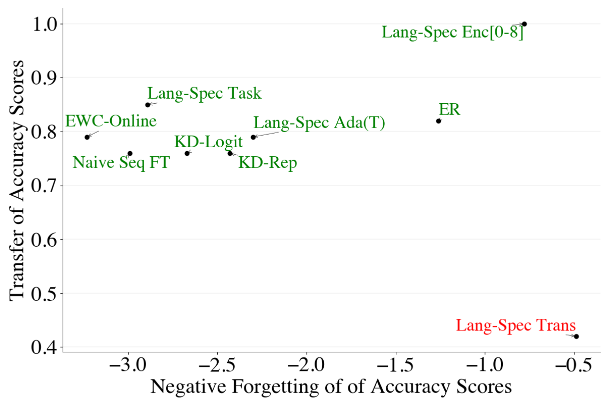


Figure 3: Transfer versus negative forgetting for intent classification task. Outliers are in red.

In the baseline approach which suffers from the lowest forgetting, we also notice the lowest transfer and final performance in Figures 3 and 4. As continual learning approaches reduce forgetting, they also improve the final performance and some of them also improve transfer but not to the same degree. This suggests that **the lower the forgetting a model can achieve, the easier it gets for it to**

[7]We report the p-values from pairwise Tukey's HSD analysis to gain a reliable unified view that individual t-tests may fail to convey. More explanation can be found in Appendix B.3.

[8]We include a full analysis of the expansion over several subsets of M-BERT components in Appendix C.2.

[9]This trains M-BERT encoder layers $\in 1 \dots 9$ in a language-specific manner, while sharing the embeddings, the rest of the layers $\in 10 \dots 12$, and prediction heads.

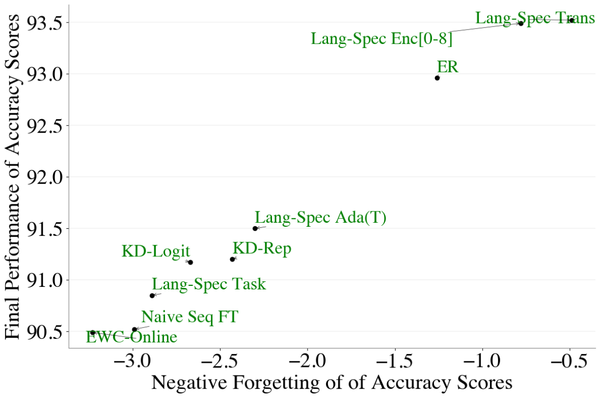


Figure 4: Final performance versus negative forgetting for intent classification task.

accumulate knowledge. There are some outliers like *Lang-Spec Trans* which is the best model in terms of reducing forgetting but also the worst in terms of transfer. This could be due to the fact that *Lang-Spec Trans* exhibits a similar behavior to *Lang-Spec FT* thus the transfer, which is the difference with *Lang-Spec FT*, is almost null.

4.3 Analysis across Different Language Permutations

Model	F ↓		T ↑		FP ↑	
	high2low	low2high	high2low	low2high	high2low	low2high
Naive Seq FT	**1.74** ±0.02	5.42 ±0.04	0.83 ±0.02	**0.85** ±0.01	**91.87** ±0.02	87.65 ±0.02
Lang-Spec Trans	**0.39** ±0.01	0.62 ±0.02	**0.71** ±0.02	0.28 ±0.02	**93.86** ±0.01	93.38 ±0.01
Lang-Spec Enc[0-8]	***0.59*** ±0.01	*1.08* ±0.02	***1.13*** ±0.01	*0.95* ±0.01	***93.77*** ±0.01	*93.16* ±0.01
Lang-Spec Task	**1.55** ±0.01	5.47 ±0.04	**0.98** ±0.02	0.63 ±0.01	**91.97** ±0.02	87.66 ±0.02
Lang-Spec Ada(T)	**1.13** ±0.01	4.73 ±0.04	**0.94** ±0.02	0.74 ±0.01	**92.44** ±0.01	88.91 ±0.02
Lang-Spec Ada(F)	**0.95** ±0.02	1.18 ±0.04	**3.30** ±0.02	2.10 ±0.02	**90.87** ±0.02	89.84 ±0.02
EWC-Online	**2.01** ±0.02	6.35 ±0.04	**0.94** ±0.02	0.59 ±0.01	**90.72** ±0.02	87.79 ±0.02
ER	**0.93** ±0.02	1.81 ±0.03	**0.87** ±0.01	0.56 ±0.02	**93.24** ±0.01	92.68 ±0.02
KD-Logit	**1.82** ±0.02	4.57 ±0.04	**0.76** ±0.02	**0.76** ±0.02	**91.25** ±0.02	89.53 ±0.02
KD-Rep	**1.87** ±0.02	3.78 ±0.04	0.80 ±0.02	**0.93** ±0.01	**90.86** ±0.02	89.75 ±0.02

Table 4: Performance on intent classification comparison between the baseline and continual learning algorithms across two language permutations. We highlight the lowest forgetting (F), highest transfer (T), and final performance (FP) of accuracy scores among high2low and low2high in bold, whereas the best and second best scores across approaches for high2low and low2high separately are underlined and italicized, respectively.

So far our analysis has focused on the average over different language permutations, but are the same patterns observed for different language permutations? To shed the light on that, we analyze the performance of different continual learning algorithms and baselines in terms of their forgetting, transfer, and final performance over high2low and low2high permutations (Q3 in §2.5), in Table 4.[10] In general, we observe that for ***Naive Seq FT*** **and some continuous learning approaches, it is more challenging to learn from low to high resource languages**, as there is a huge difference in forgetting and final performance and to a lesser degree a decrease in transfer. On the other hand, **model expansion and memory replay approaches reduce the forgetting and final gap between language permutations**. We hypothesize that low2high being more challenging than high2low could be due to the fine-tuning training data size that is different between languages.

Model	F ↓		T ↑		FP ↑	
	high2low	low2high	high2low	low2high	high2low	low2high
Original Data	**1.74** ±0.02	5.42 ±0.04	0.83 ±0.02	**0.85** ±0.01	**91.87** ±0.02	87.65 ±0.02
Balanced Data	**1.25** ±0.02	5.81 ±0.05	**0.89** ±0.02	0.75 ±0.02	89.33 ±0.02	85.81 ±0.02

Table 5: Performance on intent classification comparison between two versions of the data: original data version and balanced data for *Naive Seq FT* across the same permutations as (Table 4). We embolden the best among high2low and low2high for each metric.

To verify this hypothesis, we dig deeper to check if the differences among training fine-tuning data sizes between languages is the main factor by performing an ablation study on that. Therefore, we use the same amount of training resources for each language and report the results on *Naive Seq FT* in Table 5. We can see that there is still a gap between these two language permutations for forgetting and final performance. This suggests that **the difference in fine-tuning training data size is not what accounts for the differences between the two language permutations**. There are perhaps biases in the pre-training or other linguistic artifacts that need to be studied in future work.

4.4 Zero-Shot Generalization in Cross-lingual Continual Learning

To analyze the zero-shot transfer to unseen languages, we plot the performance on zero-shot transfer as a function of negative forgetting for the baseline and continual learning approaches, to investigate any relationship between generalization and preservation (Q4 in 2.5). In Figure 5, we infer that **most continual learning approaches don't substantially improve the generalization compared to *Naive Seq FT***. We notice that model expansion approaches (in red), in particular, hurt the generalization performance even if they significantly reduce forgetting. This **zero-shot transfer versus interference trade-off** is referred to as the **stability-plasticity dilemma** (Mermillod et al., 2013), where the weights responsible for improving on new tasks are often responsible for the forgetting on previous tasks. If we exclude model

[10]Full results for slot filling, more language permutations, and the balanced data can be found in Appendix C.3.

expansion approaches (sub-figure on the right), we notice that approaches which reduce forgetting also improve generalization compared to *Naive Seq FT*. Better approaches to balance between the two can be investigated in future work.

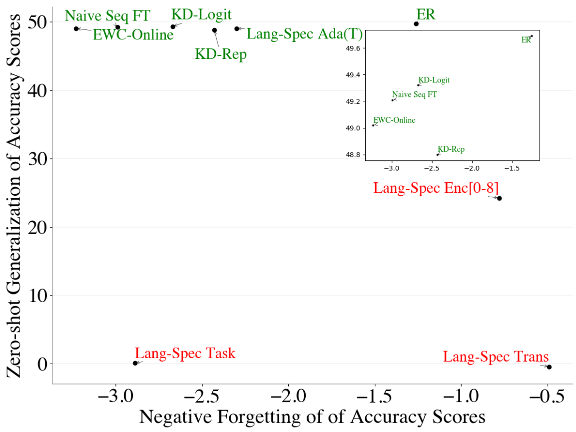


Figure 5: Zero-shot generalization versus negative forgetting for intent classification. Outliers are highlighted in red. We zoom over the rest of the models in the upper right corner subplot.

4.5 Multi-Hop vs Two-Hop Cross-lingual Continual Learning

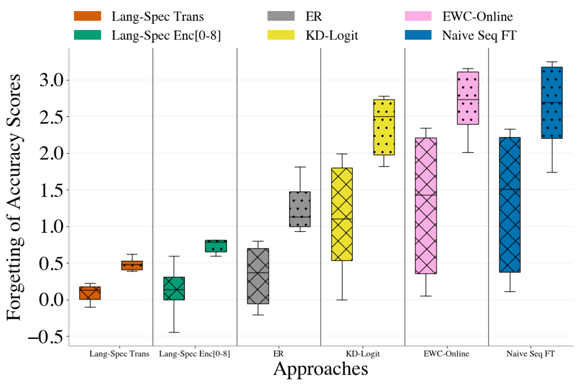


Figure 6: Comparison between forgetting trends for intent classification using two-hop (crossed boxplots) and multi-hop analysis (dotted boxplots), on the left and right respectively for each approach, showing the variance over different language permutations.

To motivate this cross-lingual continual learning work further, we dig deeper into how a multi-hop analysis is different from a conventional transfer learning analysis (Q5 in §2.5). Figure 6 shows a comparison between the two in terms of forgetting for different approaches aggregated over different language permutations. More results for slot filling and other metrics can be found in Figure 11 in Appendix C.5. *Lang-Spec Trans* tends to have the least forgetting and *Naive Seq FT* the most, but importantly **the variance for a multi-hop analysis is much smaller than that for two-hop analysis**.

5 Related Work

Continual learning approaches have found favor especially among the computer vision community, including regularization-based (Kirkpatrick et al., 2017; Zenke et al., 2017; Li and Hoiem, 2016; Ritter et al., 2018), memory-based (Shin et al., 2017; Chaudhry et al., 2019b,a), etc. Only recently, it has started gaining more interest in the NLP community. Current approaches often fail to effectively retain previous knowledge and adapt to new information simultaneously (Biesialska et al., 2020; Han et al., 2020; de Masson d'Autume et al., 2019).

Existing continual learning work for cross-lingual NLP is even more scarce, either focusing on proposing cross-lingual approaches that indirectly support lifelong learning, such as Artetxe et al. (2020), on the transfer-ability of monolingual models. Other approaches derive a cross-lingual continual learning problem directly from cross-lingual transfer learning, such as Garcia et al. (2021), which investigate a lexical approach for cross-lingual continual machine translation. Liu et al. (2021) explore continual techniques to fine-tune on downstream applications for new languages, while preserving the original cross-lingual ability of the pre-trained model. However, they focus on a two-hop analysis from high to low resource language pairs or from pre-training to fine-tuning tasks, unlike our work, which analyzes across multiple hops.

6 Conclusion

We formulate the cross-lingual lifelong learning problem setup. We show that simple naive sequential fine-tuning is prone to catastrophic forgetting and has poor accumulation and generalization capabilities sensitive to different language permutations. To address these issues, we provide the first benchmark to compare the effectiveness of different continual learning algorithms for the cross-lingual case. We show that continual learning models improve cross-lingual knowledge preservation, which also contributes to facilitating knowledge accumulation, but to a lesser degree on generalization. We also discuss the challenges of sequentially training for certain language permutations. We hope that this study will encourage more analyses in the same spirit to gain more insights that go beyond conventional cross-lingual transfer learning.

References

Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. 2020. Knowledge distillation from internal representations. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7350–7357. AAAI Press.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4623–4637. Association for Computational Linguistics.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6523–6541. International Committee on Computational Linguistics.

Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multilingual intent detection and slot filling in a joint bert-based model. *CoRR*, abs/1907.02884.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019a. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet Kumar Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. 2019b. Continual learning with tiny episodic memories. *CoRR*, abs/1902.10486.

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

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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

Robert M. French. 1993. Catastrophic interference in connectionist networks: Can it be predicted, can it be prevented? In *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pages 1176–1177. Morgan Kaufmann.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1184–1192, Online. Association for Computational Linguistics.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 745–758. Association for Computational Linguistics.

Tyler L. Hayes, Ronald Kemker, Nathan D. Cahill, and Christopher Kanan. 2018. New metrics and experimental paradigms for continual learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2031–2034. Computer Vision Foundation / IEEE Computer Society.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

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.

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

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

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 388–395. ACL.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2950–2962. Association for Computational Linguistics.

Zhizhong Li and Derek Hoiem. 2016. Learning without forgetting. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, volume 9908 of *Lecture Notes in Computer Science*, pages 614–629. Springer.

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, Online. Association for Computational Linguistics.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6467–6476.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul A. Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. 2021. Continual learning in task-oriented dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7452–7467. Association for Computational Linguistics.

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. 2013. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4:504.

Meryem M'hamdi, Doo Soon Kim, Franck Dernoncourt, Trung Bui, Xiang Ren, and Jonathan May. 2021. X-METRA-ADA: Cross-lingual meta-transfer learning adaptation to natural language understanding and question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3617–3632, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulic, 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 2020, Online, November 16-20, 2020*, pages 7654–7673. Association for Computational Linguistics.

Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online structured laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. LAMOL: language modeling for lifelong language learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Gido M. van de Ven and Andreas S. Tolias. 2019. Three scenarios for continual learning. *CoRR*, abs/1904.07734.

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

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR.

A More Details about Approaches

A.1 Base Model Architecture

We use the same architecture as in Castellucci et al. (2019); M'hamdi et al. (2021) to jointly learn intent classification and slot filling subtasks. As shown in Figure 7, we leverage features from Transformer (Vaswani et al., 2017) encoder and add classification prediction heads on top of it. More specifically, a multi-lingual pre-trained model is used to encode the input. Then, to predict the intent and slot spans, we add task-specific prediction heads. For intent prediction, this takes the form of a linear layer plus softmax on top of the $[CLS]$ token representation. For slot filling, we use a sequence labeling layer in the form of a linear layer plus CRF respectively. We use the sum of both intent and CRF based slot losses to optimize the model parameters.

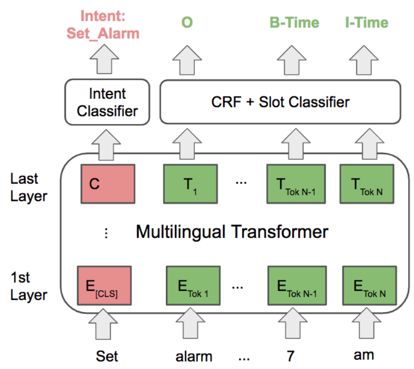


Figure 7: Architecture of base-task oriented dialogue.

A.2 Adapters

Adapters consist of downsampling layers followed by upsampling layers inserted between layers of our Transformer encoder in addition to their invertible components. We don't add task-specific adapters which, according to our ablation studies, didn't prove beneficial. We add adapter components to every encoder layer following MAD-X configuration and using their pre-trained weights obtained from AdapterHub (Pfeiffer et al., 2020a).[11] We either fine-tune the weights for the languages available in AdapterHub or train from scratch for languages for which there are no pretraining adapter weights. At inference time, we use adapter layers fine-tuned independently for each language in the datastream.

[11] https://adapterhub.ml/explore/text_lang/

A.3 Online Elastic Weight Consolidation (EWC-Online)

To penalize changes in the parameters crucial to previous languages, we use EWC, which adds a regularization term to the loss applied only after the first data set $\mathcal{D}_i$ in the language stream is seen. $\forall i \in 2 \dots N$, we compute the total loss as follows:

$$\mathcal{L}^i_{total} = \mathcal{L}^i_{cur} + \lambda \mathcal{L}^i_{reg}, \quad (9)$$

where $\mathcal{L}_{cur}$ is the usual loss of the downstream task on the current data $\mathcal{D}_i$ and $\mathcal{L}_{reg}$ is the regularization term and λ is a hyperparameter to control the regularization strength. For efficiency purposes, we use the online version of EWC (*EWC-Online*), which number of quadratic terms in the regularization terms doesn't increase with the number of languages seen so far. Following that, our regularization term is computed as, based on the formulation in van de Ven and Tolias (2019):

$$\mathcal{L}^i_{reg} = \sum_{j=1}^{N_p} \tilde{F}^{(i-1)}_{jj} (\theta_j - \theta^k_j)^2, \quad (10)$$

where θ are the parameters of the transformers model in addition to the downstream prediction heads, N_p is the total number of parameters, and $\tilde{F}^{(i-1)}_{jj}$ is the Fisher information matrix on the last language just before training on $\mathcal{D}_i$. This is computed as the running sum of the i^{th} diagonal elements of the Fisher Information matrices of $\mathcal{D}_j$, for all $j \in 1 \dots (i-1)$. $\tilde{F}^{(i)}_{jj} = \gamma \tilde{F}^{(i-1)}_{jj} + F^i_{jj}$ and $\tilde{F}^1_{jj} = F^1_{jj}$. In practice, F^i is simply the gradients all parameters flattened into one single matrix.

A.4 Experience Replay (ER)

After training for each $\mathcal{D}_i$ for all $i \in 1 \dots N$, we populate the memory with randomly sampled examples from $\mathcal{D}_i$. For each $\mathcal{D}_i$ for all $i \in 2 \dots N$, after training for every $k = 100$ mini-batches and optimizing for the current loss separately, the model randomly samples an equal batch from the memory for each $\mathcal{D}_j$ such that $j \in 1 \dots (i-1)$ and replays them using the current model checkpoint used for training on $\mathcal{D}_i$. We retrieve an equal amount of memory from each language and at each step and hop. The loss from the current $\mathcal{D}_i$ and the loss on the memory on the $\mathcal{D}_j$ are interleaved as the replay on the memory only happens every k steps. This prioritization of the current language helps make the training more stable without over-fitting on the small memory from previous languages.

A.5 Knowledge Distillation (KD-Logit & KD-Rep)

We use the same strategy explained in §A.4 to select the memory to be replayed using a knowledge distillation loss. For each $\mathcal{D}_i$ for all $i \in 2 \dots N$, after training for every $k = 100$ mini-batches, we randomly samples an equal batch from the memory for each $\mathcal{D}_j$ such that $j \in 1 \dots (i-1)$. We also load the model checkpoints for each hop j and use that model and the memory for $\mathcal{D}_j$ to compute either the intent and slot logits in the case of *KD-Logit* or the multilingual representations of M-BERT in the case of *KD-Rep*. We do the same thing using the current model checkpoint this time. Then, we use the minimum square error loss to minimize the distance between the intent logits obtained using the previous and current model checkpoints and do the same thing for slot logits for *KD-Logit*. Then, we take the same over intent and slot distillation losses across different language retrieved from the memory. The same is done for computing the distillation loss over the multilingual representations in *KD-Rep*.

B Experimental Setup Details

B.1 Datastreams

Order 1	Order 2	Order 3	Order 4	Order 5	Order 6
English	Thai	Spanish	French	Hindi	German
German	Spanish	Hindi	Thai	English	French
French	Hindi	English	German	Spanish	Thai
Hindi	French	German	English	Thai	Spanish
Spanish	German	Thai	Hindi	French	English
Thai	English	French	Spanish	German	Hindi

Table 6: Simulated language permutations.

We use the following data streams for all our experiments as summarized in Table 6. The MTOP dataset has been released by Facebook (Li et al., 2021) under Creative Commons Attribution-ShareAlike 4.0 International Public License which allows its usage.

B.2 Implementation Details

For all experiments, we use M-BERT(bert-base-multilingual-cased)[12] with 12 layers as our pre-trained Transformer model. We use the dev set to pick the hyperparameters of the optimizer to be used. We perform a search for the most optimal learning rate over a range $[1e-4, 3e-4, 1e-5, 3e-5]$ for Adam optimizer (Kingma and Ba, 2015) and finally fix the learning rate to $3e-5$ for all experiments for a fair comparison. We use $\epsilon = 1e-8$, $\beta_1 = 0.9$, $\beta_2 = 0.99$, batch size of 16, $\gamma = 0.1$ for EWC Online, 6000 memory size for ER and knowledge distillation. For all experiments, we run for 10 epochs maximum and pick the best model based on dev data. We also fix a seed of 42 for the random initialization of numpy, random, and torch over all experiments. All experiments are run using the same computing infrastructure Pytorch version 1.7.1, using *one* Tesla P100 GPU of 16280 MiB of memory CUDA version 11.2.

The runtime and the number of parameters depend on the approach used and the mode of training are detailed in Table 7. With the exception of model expansion approaches, all approaches have the same number of parameters coming from the sum of M-BERT and prediction head parameters. *Lang-Spec Trans* has the highest number of parameters which is six times more than *Naive Seq FT* but only requires two times more runtime as only one $frac16$ part of language-specific M-BERT is updated at each hop for each whereas the rest is used in evaluation mode only. *Lang-Spec Ada(F)* has the smallest number of parameters which around 24% and takes 2 times less than the usual runtime of *Naive Seq FT* (while exhibiting lower forgetting and higher transfer than *Naive Seq FT*, as shown in Table 8). Memory replay and knowledge distillation approaches have more runtime (slightly more than *Lang-Spec Trans*) as they store and handle memory and compute the replay or distillation losses interleaved with the main loss which makes them time-consuming.

B.3 Bootstrap Sampling & Statistical Significance

We run all experiments over one fixed seed of 42. We then use bootstrap sampling (Koehn, 2004) to compute the mean and confidence intervals for each of the metrics described in §2.6 over a single approach. For each language permutation, and for each $R_{i,\leq j}$, representing some performance metric on language ℓ_i after training on $\mathcal{D}_{1\dots j}$, we sample with replacement 600 sentences from the testing data over 600 iterations. By using this number of iterations and sampling sentences, we ensure and also double check that all sentences in the test set

[12] github.com/huggingface/transformers version 3.4.0 pre-trained on 104 languages, including all languages evaluated on in this paper.

Model	Runtime	# Param
Naive Seq FT	3h16min	178,081,402
Lang-Spec FT	52min	178,081,402
Inc Joint	1d22h51min	178,081,402
Multilingual	16h45min	178,081,402
Lang-Spec Embed	7h46min	639,123,322
Lang-Spec Enc[0-2]	7h52min	284,399,482
Lang-Spec Enc[3-5]	7h12min	284,399,482
Lang-Spec Enc[6-8]	7h8min	284,399,482
Lang-Spec Enc[9-11]	7h20min	284,399,482
Lang-Spec Enc[0-8]	8h1min	497,035,642
Lang-Spec Trans	7h15min	1,067,348,602
Lang-Spec Enc[0-11]	7h53min	603,353,722
Lang-Spec Enc[0-5]	7h16min	390,717,562
Lang-Spec Enc[6-11]	7h10min	390,717,562
Lang-Spec Task	6h18min	179,221,212
Lang-Spec Ada(T)	4h34min	222,301,402
Lang-Spec Ada(F)	1h57min	44,447,962
EWC-Online	1d3h17min	178,081,402
ER	8h55min	178,081,402
KD-Logit	7h23min	178,081,402
KD-Rep	8h	178,081,402

Table 7: Runtime and parameters statistics.

are covered in the evaluation ensuring a uniform evaluation across approaches. Let x be the list of results we get for each iteration independently. Then, we compute the mean and standard deviation $\bar{x}$ and $std(x)$ respectively and the 95% confidence interval size CI using the following equation:

$$CI = \frac{1.9639 \times std(x)}{\sqrt{600}}, \\ std(x) = \sqrt{\frac{\sum(x-\bar{x})^2}{600}}. \tag{11}$$

This computes x and CI for each language permutation separately. To aggregate this across different language permutations, we simply take the average and the standard deviation.

To compute the statistical significance between different approaches, we use ANOVA and perform a multiple pairwise comparisons analysis using Tukey's honestly significant difference (HSD) test[13] over different language permutations for each metric.

C More Results & Analysis

C.1 Full Average Results

Table 8 shows the full results and confidence intervals for different continual learning approaches. Compared to intent classification, we observe a higher forgetting and slightly higher transfer but a lower zero-shot transfer and final performance in the case of slot filling. This could be due to the nature of the task of slot filling which is more challenging to learn. In general, we can observe the same forgetting, transfer, zero-shot transfer, and final performance trends between intent classification and slot filling. In other words, if a model a has higher forgetting of intent classification than model b then the same thing applied to slot filling. Some exceptions include *ER* which the highest zero-shot transfer on slot filling, while having not the highest but the second highest zero-shot transfer on intent classification. This could be due to the transfer between intent classification and slot filling that is maximized when training them jointly.

C.2 Per M-BERT Components Analysis

Table 9 shows ablation studies for the analysis of M-BERT components following four different categories: groups of 12 layers with or without embeddings, groups of 3 layers, 6 layers, and 9 layers at a time trained in a language specific manner and the rest shared between languages. We notice that training the full *Lang-Spec Trans* has the best in terms of forgetting. Training only the first 8 encoder layers *Lang-Spec Enc[0-8]*, excluding embeddings, in a language-specific manner comes next with the second lowest forgetting, a better transfer, an even better one for zero-shot forward transfer, but a slightly better final performance. Another good model reaching a good compromise between zero-shot transfer and forgetting with less language-specific layers is *Lang-Spec Enc[0-5]*. *Naive Seq FT* is still the best compared to those model-expansion approaches in terms of zero-shot performance, but has a lower final performance and higher forgetting. We also notice the same trend for language-specific embeddings *Lang-Spec Embed* which reaches the second best zero-shot transfer performance, but with also a high forgetting. This suggests that language-specific knowledge is less likely to be encoded in the embeddings and more at the encoder layers. This shows that there is a real plasticity-stability tradeoff between zero-shot transfer and knowledge preservation (which we explain in more details in §4.3).

C.3 Full Results on Language Permutations

Full results for all language permutations can be found in Tables 10, 11, and 12. By looking at additional language permutations, low2high (Thai → Spanish → Hindi → French → German → English) is still the most challenging one in terms

[13] We use bioinfokit library https://github.com/reneshbedre/bioinfokit

Model	F↓ Acc	F↓ F1	T↑ Acc	T↑ F1	T^0↑ Acc	T^0↑ F1	FP↑ Acc	FP↑ F1
Shared {Trans, Task} Baselines								
Naive Seq FT	2.99 ±1.20	6.22 ±0.95	0.76 ±0.09	1.42 ±0.33	49.21 ±3.21	36.10 ±2.15	90.52 ±1.42	69.10 ±1.24
Lang-Spec FT							93.20 ±0.08	73.59 ±0.81
Lang-Spec FT + Ada(T)							93.26 ±0.08	73.01 ±0.86
Lang-Spec FT + Ada(F)							88.81 ±0.13	65.79 ±0.90
Inc Joint	**0.15** ±0.10	**0.93** ±0.38	0.85 ±0.12	1.33 ±0.83	**50.12** ±2.50	36.34 ±2.59	94.20 ±0.15	74.97 ±0.51
Multilingual							**94.25** ±0.07	**76.34** ±0.82
Model Expansion Baselines								
Lang-Spec Trans	0.49 ±0.08	1.28 ±0.21	0.42 ±0.16	1.26 ±0.15	-0.43 ±0.15	0.42 ±0.06	93.52 ±0.18	74.71 ±0.15
Lang-Spec Enc[0-8]	0.78 ±0.16	1.95 ±0.48	1.00 ±0.09	1.74 ±0.64	24.23 ±1.75	12.33 ±1.25	93.49 ±0.21	74.16 ±0.85
Lang-Spec Task	2.89 ±1.24	5.27 ±1.02	0.85 ±0.12	1.50 ±1.05	0.10 ±0.25	0.07 ±0.02	90.85 ±1.47	69.48 ±1.54
Lang-Spec Ada(T)	2.30 ±1.18	4.68 ±0.86	0.79 ±0.07	1.87 ±0.72	49.04 ±3.10	35.80 ±2.27	91.50 ±1.27	70.25 ±1.78
Lang-Spec Ada(F)	1.04 ±0.19	2.85 ±0.96	**2.64** ±0.39	**4.74** ±0.49	8.36 ±1.19	3.63 ±0.81	90.32 ±0.34	67.98 ±0.73
Other continual Learning Algorithms								
EWC-Online	3.23 ±1.45	6.16 ±1.03	0.79 ±0.12	1.54 ±0.31	49.02 ±2.98	36.06 ±2.23	90.49 ±1.35	69.34 ±1.58
ER	1.26 ±0.32	3.20 ±0.39	0.82 ±0.13	1.92 ±0.54	49.69 ±3.28	**36.58** ±2.09	92.96 ±0.21	73.37 ±0.74
KD-Logit	2.67 ±0.92	5.83 ±0.81	0.76 ±0.11	1.62 ±0.55	49.32 ±2.95	36.20 ±2.34	91.17 ±0.80	69.54 ±1.34
KD-Rep	2.43 ±0.62	5.60 ±0.72	0.76 ±0.09	1.67 ±0.56	48.80 ±3.01	36.15 ±2.23	91.20 ±0.74	69.64 ±1.56

Table 8: A summary of results for different continual learning approaches over the average across language order. For each metric and score, we highlight the best score in bold and underline the second best score.

Model	F↓ Acc	F↓ F1	T↑ Acc	T↑ F1	T^0↑ Acc	T^0↑ F1	FP↑ Acc	FP↑ F1
Naive Seq FT	2.99 ±1.20	6.22 ±0.95	0.76 ±0.09	1.42 ±0.33	49.21 ±3.21	36.10 ±2.15	90.52 ±1.42	69.10 ±1.24
Lang-Spec FT							93.20 ±0.08	73.59 ±0.81
Lang-Spec Trans	**0.49** ±0.08	**1.28** ±0.21	0.42 ±0.16	1.26 ±0.15	-0.43 ±0.15	0.42 ±0.06	**93.52** ±0.18	**74.71** ±0.15
Lang-Spec Enc[0-11]	0.48 ±0.07	1.32 ±0.16	0.43 ±0.19	1.08 ±0.27	-0.30 ±0.18	0.57 ±0.08	93.51 ±0.13	74.50 ±0.25
Lang-Spec Embed	3.12 ±1.34	5.89 ±0.95	0.95 ±0.16	1.62 ±0.68	50.66 ±2.97	36.61 ±1.89	90.68 ±1.28	69.59 ±1.26
Lang-Spec Enc[0-2]	1.90 ±0.77	4.33 ±0.66	0.97 ±0.13	1.61 ±0.56	**52.18** ±3.26	**37.41** ±1.99	92.25 ±0.75	71.56 ±1.51
Lang-Spec Enc[3-5]	1.46 ±0.64	2.90 ±0.35	*0.98* ±0.19	**1.95** ±0.4	47.82 ±2.98	34.65 ±1.77	92.72 ±0.67	73.04 ±0.95
Lang-Spec Enc[6-8]	1.43 ±0.55	3.08 ±0.57	0.89 ±0.15	*1.64* ±0.41	38.33 ±3.01	23.67 ±2.35	92.44 ±0.76	72.25 ±1.08
Lang-Spec Enc[9-11]	2.21 ±0.88	4.10 ±0.87	0.67 ±0.2	1.63 ±0.55	41.37 ±2.13	20.05 ±1.92	91.41 ±1.06	71.16 ±1.13
Lang-Spec Enc[0-5]	1.29 ±0.67	2.99 ±0.65	**1.07** ±0.11	**1.95** ±0.56	45.25 ±2.56	31.22 ±2.19	92.90 ±0.52	73.30 ±1.07
Lang-Spec Enc[6-11]	1.66 ±0.36	3.33 ±0.67	0.51 ±0.3	0.96 ±0.59	6.04 ±1.13	4.52 ±0.96	91.97 ±0.38	71.62 ±1.18
Lang-Spec Enc[0-8]	*0.78* ±0.16	*1.95* ±0.48	1.00 ±0.09	1.74 ±0.64	24.23 ±1.75	12.33 ±1.25	*93.49* ±0.21	*74.16* ±0.85
Lang-Spec Enc[9-11]	2.21 ±0.88	4.10 ±0.87	0.67 ±0.2	1.63 ±0.55	41.37 ±2.13	20.05 ±1.92	91.41 ±1.06	71.16 ±1.13

Table 9: Per group layer analysis: ablation studies of different M-BERT's components. Best, second best, and third best scores for each metric are emboldened, underlined, and italicized respectively.

of knowledge preservation, accumulation, generalization, and model utility. High2low (English → German → French → Hindi → Spanish → Thai) is still the easiest to learn. Order 5(Hindi → English → Spanish → Thai → French → German) is the second most challenging language permutation to train. In general, the same trends regarding the more challenging nature of training for certain language permutations are observed for both intent classification and slot filling uniformly. Table 13 includes the results for more language permutations for the balanced data.

C.4 Per Language Analysis

Tables 14, 15, and 16 show the full results for forgetting, transfer, and zero-shot transfer respectively, across different languages averaged over different language permutations. We notice that languages like English, German, French, and Spanish have constantly lower forgetting than languages like Hindi and Thai for both intent classification and slot filling for *Naive Seq FT* compared to the reference model *Inc Joint* for which the forgetting is low and nearly equal. Approaches like *Lang-Spec Trans*, *Lang-Spec Enc[0-8]*, *Lang-Spec Ada(F)*, and to a certain degree *ER* also reduce that gap. We also notice that approaches that lower forgetting for a particular languages do so uniformly for all languages. The performance in terms of zero-shot transfer is significantly lower in the case of Thai.

Model	high2low				low2high			
	Test Intent Accuracy On							
	F ↓	T ↑	T^0 ↑	FP ↑	F ↓	T ↑	T^0 ↑	FP ↑
Shared {Trans, Task} Baselines								
Naive Seq FT	**1.74** ±0.02	0.83 ±0.02	**49.1** ±0.03	**91.87** ±0.02	5.42 ±0.04	**0.85** ±0.01	44.73 ±0.02	87.65 ±0.02
Lang-Spec FT				93.20 ±0.08				93.20 ±0.08
Lang-Spec FT + Ada(T)				93.26 ±0.08				93.26 ±0.08
Lang-Spec FT + Ada(F)				88.81 ±0.13				88.81 ±0.13
Inc Joint	0.28 ±0.01	**0.98** ±0.02	**50.61** ±0.03	94.04 ±0.01	**0.13** ±0.01	0.93 ±0.01	45.84 ±0.03	**94.31** ±0.01
Multilingual				94.25 ±0.07				94.25 ±0.07
Model Expansion Baselines								
Lang-Spec Trans	**0.39** ±0.01	**0.71** ±0.02	**-0.48** ±0.00	**93.86** ±0.01	0.62 ±0.02	0.28 ±0.02	-0.53 ±0.00	93.38 ±0.01
Lang-Spec Enc[0-8]	**0.59** ±0.01	**1.13** ±0.01	21.97 ±0.02	**93.77** ±0.01	1.08 ±0.02	0.95 ±0.01	**22.49** ±0.01	93.16 ±0.01
Lang-Spec Task	**1.55** ±0.01	**0.17** ±0.00	**0.98** ±0.02	**91.97** ±0.02	5.47 ±0.04	-0.11 ±0.00	0.63 ±0.01	87.66 ±0.02
Lang-Spec Ada(T)	**1.13** ±0.01	**0.94** ±0.02	**49.27** ±0.03	**92.44** ±0.01	4.73 ±0.04	0.74 ±0.01	43.79 ±0.02	88.91 ±0.02
Lang-Spec Ada(F)	**0.95** ±0.02	**3.30** ±0.02	**9.21** ±0.01	**90.87** ±0.02	1.18 ±0.04	2.10 ±0.02	8.63 ±0.01	89.84 ±0.02
Other continual Learning Algorithms								
EWC-Online	**2.01** ±0.02	**0.94** ±0.02	**49.77** ±0.03	**90.72** ±0.02	6.35 ±0.04	0.59 ±0.01	44.26 ±0.02	87.79 ±0.02
ER	**0.93** ±0.02	**0.87** ±0.01	**49.15** ±0.03	**93.24** ±0.01	1.81 ±0.03	0.56 ±0.02	44.37 ±0.02	92.68 ±0.02
KD-Logit	**1.82** ±0.02	**0.76** ±0.02	**49.17** ±0.03	**91.25** ±0.02	4.57 ±0.04	0.76 ±0.02	44.45 ±0.02	89.53 ±0.02
KD-Rep	**1.87** ±0.02	**0.80** ±0.02	**49.34** ±0.03	**90.86** ±0.02	3.78 ±0.04	0.93 ±0.01	43.91 ±0.03	89.75 ±0.02
	Test Slot Filling On							
	F ↓	T ↑	T^0 ↑	FP ↑	F ↓	T ↑	T^0 ↑	FP ↑
Shared {Trans, Task} Baselines								
Naive Seq FT	**4.63** ±0.23	**1.36** ±0.16	**37.02** ±0.06	**69.46** ±0.14	7.73 ±0.25	0.89 ±0.19	32.64 ±0.04	66.94 ±0.14
Lang-Spec FT				73.59 ±0.81				73.59 ±0.81
Lang-Spec FT + Ada(T)				73.01 ±0.86				73.01 ±0.86
Lang-Spec FT + Ada(F)				65.79 ±0.9				65.79 ±0.9
Inc Joint	1.11 ±0.14	**2.16** ±0.18	**37.66** ±0.05	**75.82** ±0.13	**0.25** ±0.12	-0.12 ±0.16	32.75 ±0.03	75.15 ±0.14
Multilingual				76.34 ±0.82				76.34 ±0.82
Model Expansion Baselines								
Lang-Spec Trans	**0.99** ±0.12	**1.12** ±0.17	0.33 ±0.00	74.76 ±0.14	1.14 ±0.14	1.05 ±0.17	**0.39** ±0.00	**74.77** ±0.13
Lang-Spec Enc[0-8]	**2.37** ±0.15	**2.08** ±0.16	10.58 ±0.01	72.59 ±0.13	1.97 ±0.15	0.93 ±0.18	**12.67** ±0.01	**74.08** ±0.14
Lang-Spec Task	**4.09** ±0.18	**0.06** ±0.00	**2.08** ±0.17	**68.99** ±0.13	7.24 ±0.24	**0.06** ±0.00	-0.40 ±0.18	66.39 ±0.14
Lang-Spec Ada(T)	**4.15** ±0.20	**2.74** ±0.19	**37.66** ±0.05	**70.11** ±0.13	6.29 ±0.22	1.41 ±0.17	31.69 ±0.03	67.21 ±0.13
Lang-Spec Ada(F)	**2.25** ±0.18	**4.93** ±0.18	**4.44** ±0.00	**68.35** ±0.15	4.93 ±0.24	3.82 ±0.18	2.52 ±0.00	66.43 ±0.15
Other continual Learning Algorithms								
EWC-Online	**4.77** ±0.2	**1.22** ±0.17	**37.71** ±0.06	**67.61** ±0.12	8.12 ±0.27	1.14 ±0.18	32.61 ±0.03	66.80 ±0.14
ER	**2.58** ±0.15	**1.92** ±0.15	**38.08** ±0.06	72.44 ±0.13	3.69 ±0.25	0.96 ±0.18	33.40 ±0.03	**73.0** ±0.13
KD-Logit	**4.65** ±0.20	**1.71** ±0.16	**37.91** ±0.06	**68.30** ±0.13	6.91 ±0.25	0.62 ±0.16	32.42 ±0.03	67.77 ±0.13
KD-Rep	**4.35** ±0.18	**1.29** ±0.17	**37.85** ±0.06	**68.49** ±0.14	6.85 ±0.25	0.7 ±0.19	32.80 ±0.03	67.04 ±0.13

Table 10: Per language permutation view: a pairwise comparison between high2low (English → German → French → Hindi → Spanish → Thai) and low2high (Thai → Spanish → Hindi → French → German → English). We highlight the best forgetting (lowest), transfer (highest), zero-shot transfer (highest), and final performance (highest) of accuracy and f1 scores among those two orders for each approach in bold, whereas the best scores across approaches for the two orders separately are underlined.

C.5 More Analysis

Figures 8a, 8b, and 8c plot transfer, final performance, and zero-shot transfer versus negative forgetting for the subtask of slot filling. The same trends observed for intent classification can also be observed for slot filling. Figures 9a and 9b show how *Naive Seq FT* intent classification accuracy score and slot filling F1 score, respectively, change for each language separately after different hops of training. We can see that although performance increases as more hops are seen for high-resource Latin-script languages like English, Spanish and to some degree French, the same cannot be said for low-resource languages Thai and Hindi, which also suffer from being script isolates.

To analyze the zero-shot generalization to unseen languages, we analyze the performance of each model across different hops. In other words, we consider the average performance after seeing from 1 to 5 languages, enabled by the balanced datastreams we carefully curated 2.4. We can check the performance after training on each x language(s) from exactly one datastream. Figures 10a and 10b show a comparison between different approaches across different hops of training using zero-shot transfer metric for intent classification and slot filling, respectively. In general, we can observe that the average performance of the zero-shot transfer after seeing n languages, where $n \in [1\,..\,5]$. In this case, after seeing one language,

Model	Spanish → Hindi → English → German → Thai → French				French → Thai → German → English → Hindi → Spanish			
	Test Intent Accuracy On							
	F ↓	T ↑	T^0 ↑	FP ↑	F ↓	T ↑	T^0 ↑	FP ↑
Shared {Trans, Task} Baselines								
Naive Seq FT	**2.12** ±0.02	**0.83** ±0.01	**52.17** ±0.03	**91.63** ±0.02	2.95 ±0.03	0.72 ±0.01	51.93 ±0.02	91.29 ±0.02
Lang-Spec FT				93.20 ±0.08				93.20 ±0.08
Lang-Spec FT + Ada(T)				93.26 ±0.08				93.26 ±0.08
Lang-Spec FT + Ada(F)				88.81 ±0.13				88.81 ±0.13
Inc Joint	**0.10** ±0.01	**0.79** ±0.02	**53.85** ±0.03	94.03 ±0.01	0.22 ±0.01	0.72 ±0.01	50.53 ±0.02	**94.11** ±0.01
Multilingual				94.25 ±0.07				94.25 ±0.07
Model Expansion Baselines								
Lang-Spec Trans	**0.44** ±0.01	0.37 ±0.01	**-0.37** ±0.00	93.45 ±0.01	0.53 ±0.02	**0.52** ±0.01	-0.49 ±0.00	**93.65** ±0.01
Lang-Spec Enc[0-8]	**0.62** ±0.01	0.88 ±0.01	**26.36** ±0.02	**93.67** ±0.01	0.81 ±0.02	**0.92** ±0.01	25.25 ±0.02	93.57 ±0.01
Lang-Spec Task	**2.24** ±0.03	**0.47** ±0.00	0.81 ±0.02	**91.70** ±0.02	2.98 ±0.03	-0.09 ±0.00	**0.94** ±0.01	90.93 ±0.02
Lang-Spec Ada(T)	**1.33** ±0.02	**0.76** ±0.02	51.01 ±0.02	**92.92** ±0.02	2.35 ±0.03	0.75 ±0.01	**51.76** ±0.02	91.86 ±0.02
Lang-Spec Ada(F)	0.92 ±0.02	**2.76** ±0.02	6.34 ±0.01	**90.38** ±0.02	**0.91** ±0.03	2.28 ±0.02	**9.35** ±0.01	89.96 ±0.02
Other continual Learning Algorithms								
EWC-Online	**2.36** ±0.02	**0.78** ±0.02	**51.81** ±0.03	**91.88** ±0.02	3.16 ±0.03	0.72 ±0.01	51.16 ±0.02	91.00 ±0.02
ER	**1.01** ±0.02	0.77 ±0.01	**52.80** ±0.03	**93.13** ±0.01	1.55 ±0.02	**0.88** ±0.02	52.48 ±0.02	92.72 ±0.02
KD-Logit	**1.83** ±0.02	**0.77** ±0.01	**52.57** ±0.03	**92.08** ±0.01	2.42 ±0.03	0.54 ±0.01	51.09 ±0.02	91.63 ±0.02
KD-Rep	**2.08** ±0.02	**0.72** ±0.01	**52.04** ±0.03	**92.10** ±0.02	2.36 ±0.03	0.66 ±0.02	50.55 ±0.02	91.46 ±0.02
	Test Slot Filling On							
	F ↓	T ↑	T^0 ↑	FP ↑	F ↓	T ↑	T^0 ↑	FP ↑
Shared {Trans, Task} Baselines								
Naive Seq FT	**5.80** ±0.26	**1.47** ±0.16	**37.92** ±0.04	**70.88** ±0.13	6.47 ±0.25	1.24 ±0.18	36.64 ±0.04	68.19 ±0.15
Lang-Spec FT				73.59 ±0.81				73.59 ±0.81
Lang-Spec FT + Ada(T)				73.01 ±0.86				73.01 ±0.86
Lang-Spec FT + Ada(F)				65.79 ±0.90				65.79 ±0.90
Inc Joint	**0.83** ±0.14	1.60 ±0.17	37.46 ±0.04	74.82 ±0.14	0.95 ±0.13	**2.32** ±0.17	**37.57** ±0.04	**75.25** ±0.15
Multilingual				76.34 ±0.82				76.34 ±0.82
Model Expansion Baselines								
Lang-Spec Trans	1.47 ±0.16	**1.42** ±0.16	**0.49** ±0.00	**74.72** ±0.14	**1.37** ±0.15	1.26 ±0.16	0.47 ±0.00	74.59 ±0.15
Lang-Spec Enc[0-8]	1.83 ±0.16	**2.11** ±0.15	13.30 ±0.01	**75.06** ±0.13	**1.28** ±0.15	0.76 ±0.18	**13.57** ±0.01	74.59 ±0.13
Lang-Spec Task	4.93 ±0.22	**0.11** ±0.00	**2.35** ±0.16	**71.13** ±0.13	**4.69** ±0.21	0.06 ±0.00	1.72 ±0.17	70.61 ±0.14
Lang-Spec Ada(T)	**3.76** ±0.21	**2.40** ±0.16	**37.50** ±0.04	**72.62** ±0.15	4.67 ±0.19	1.14 ±0.16	36.77 ±0.04	68.93 ±0.13
Lang-Spec Ada(F)	**2.18** ±0.17	**5.29** ±0.18	3.81 ±0.00	68.20 ±0.14	2.58 ±0.18	4.54 ±0.16	**4.34** ±0.00	**68.26** ±0.14
Other continual Learning Algorithms								
EWC-Online	6.16 ±0.28	1.38 ±0.16	**37.89** ±0.05	**70.93** ±0.13	**6.10** ±0.24	**1.97** ±0.17	36.25 ±0.04	69.58 ±0.14
ER	**3.13** ±0.19	**1.84** ±0.17	38.39 ±0.04	**73.56** ±0.12	3.30 ±0.21	1.83 ±0.17	36.89 ±0.04	72.67 ±0.15
KD-Logit	**5.06** ±0.24	1.58 ±0.16	**38.31** ±0.05	**71.26** ±0.14	6.40 ±0.25	**1.67** ±0.18	36.13 ±0.04	69.21 ±0.14
KD-Rep	**5.52** ±0.27	**2.19** ±0.17	**37.83** ±0.04	**71.33** ±0.13	5.67 ±0.26	2.05 ±0.16	35.94 ±0.04	69.92 ±0.14

Table 11: Per language permutation view: a pairwise comparison between Order 3 (Spanish → Hindi → English → German → Thai → French) and Order 4 (French → Thai → German → English → Hindi → Spanish). We highlight the best forgetting (lowest), transfer (highest), zero-shot transfer (highest), and final performance (highest) of accuracy and f1 scores among those two orders for each approach in bold, whereas the best scores across approaches for the two orders separately are underlined.

the performance is equivalent to conventional transfer learning involving two hops, whereas the performance after seeing $n >= 2$ is for multi-hop continual learning. We notice that as we increase the number of hops, the transfer capabilities decrease nearly uniformly across most approaches, making the problem more challenging and different from conventional transfer learning. Figures 10c and 10d show the generalization trends for different continual learning approaches compared to the baselines for intent classification and slot filling, respectively. We can see that most continual learning approaches improve over *Naive Seq FT* and the gap increases mainly as more languages are seen (except at hop 4). After 5 hops, there is a clear gap between *Naive Seq FT* and continual learning approaches on top of them *Lang-Spec Ada(T)* and *KD-Logit*. Figure 11 show more results for multi-hop versus two-hop analysis for more metrics and tasks. In general, we can observe the same trend, whereby multi-hop boxplots analysis has smaller confidence intervals than two-hop boxplots

D Statistical Significance

We show in Figures 12 and 13 the results for different approaches with a p-value lower than 0.05 for confidence intervals of 95%, thus rejecting the null hypothesis that they are drawn from the same distribution. Figures 12a, 13a, 12c, 12b, 13a, 12d, 12e, and 12f show confusion plots of statistical signif-

Model	Hindi → English → Spanish → Thai → French → German				German → French → Thai → Spanish → English → Hindi			
	Test Intent Accuracy On							
	F ↓	T ↑	T^0 ↑	FP ↑	F ↓	T ↑	T^0 ↑	FP ↑
Shared {Trans, Task} Baselines								
Naive Seq FT	3.25 ±0.03	**0.68** ±0.02	45.12 ±0.03	90.13 ±0.02	**2.44** ±0.02	0.62 ±0.02	**52.18** ±0.03	**90.53** ±0.02
Lang-Spec FT				93.20 ±0.08				93.20 ±0.08
Lang-Spec FT + Ada(T)				93.26 ±0.08				93.26 ±0.08
Lang-Spec FT + Ada(F)				88.81 ±0.13				88.81 ±0.13
Inc Joint	0.20 ±0.01	**0.99** ±0.01	48.41 ±0.03	**94.42** ±0.01	**-0.02** ±0.01	0.69 ±0.02	**51.47** ±0.02	94.26 ±0.01
Multilingual				94.25 ±0.07				94.25 ±0.07
Model Expansion Baselines								
Lang-Spec Trans	**0.40** ±0.02	0.25 ±0.02	-0.58 ±0.00	**93.40** ±0.01	0.52 ±0.01	**0.41** ±0.02	**-0.11** ±0.00	93.36 ±0.01
Lang-Spec Enc[0-8]	0.81 ±0.02	0.99 ±0.01	23.17 ±0.02	93.33 ±0.01	**0.76** ±0.02	**1.10** ±0.02	**26.12** ±0.02	**93.42** ±0.01
Lang-Spec Task	2.77 ±0.03	**0.92** ±0.01	-0.20 ±0.00	91.16 ±0.02	**2.32** ±0.02	0.85 ±0.01	**0.36** ±0.00	**91.70** ±0.02
Lang-Spec Ada(T)	2.31 ±0.03	**0.82** ±0.02	46.13 ±0.03	91.42 ±0.02	**1.96** ±0.02	0.74 ±0.01	**52.26** ±0.02	**91.43** ±0.02
Lang-Spec Ada(F)	**0.88** ±0.03	2.58 ±0.02	7.15 ±0.01	**90.50** ±0.02	1.41 ±0.04	**2.84** ±0.02	**9.45** ±0.01	90.35 ±0.02
Other continual Learning Algorithms								
EWC-Online	2.97 ±0.03	0.77 ±0.02	45.63 ±0.03	89.98 ±0.02	**2.5** ±0.02	**0.95** ±0.02	**51.51** ±0.02	**91.55** ±0.02
ER	1.25 ±0.02	**0.99** ±0.02	46.63 ±0.03	**93.08** ±0.01	**0.99** ±0.02	0.82 ±0.02	**52.69** ±0.02	92.93 ±0.01
KD-Logit	2.78 ±0.03	**0.89** ±0.01	46.61 ±0.03	91.11 ±0.02	**2.58** ±0.03	0.84 ±0.02	**52.03** ±0.02	**91.41** ±0.02
KD-Rep	2.27 ±0.03	**0.80** ±0.01	45.64 ±0.03	**91.59** ±0.02	**2.20** ±0.02	0.66 ±0.02	**51.35** ±0.03	91.43 ±0.02
	Test Slot Filling On							
	F ↓	T ↑	T^0 ↑	FP ↑	F ↓	T ↑	T^0 ↑	FP ↑
Shared {Trans, Task} Baselines								
Naive Seq FT	6.78 ±0.25	**1.94** ±0.14	33.81 ±0.04	69.51 ±0.13	**5.91** ±0.24	1.64 ±0.17	**38.58** ±0.06	**69.64** ±0.14
Lang-Spec FT				73.59 ±0.81				73.59 ±0.81
Lang-Spec FT + Ada(T)				73.01 ±0.86				73.01 ±0.86
Lang-Spec FT + Ada(F)				65.79 ±0.90				65.79 ±0.90
Inc Joint	**0.89** ±0.13	**1.29** ±0.16	32.92 ±0.03	**74.51** ±0.15	1.53 ±0.14	0.75 ±0.18	**39.67** ±0.05	74.29 ±0.14
Multilingual				76.34 ±0.82				76.34 ±0.82
Model Expansion Baselines								
Lang-Spec Trans	1.58 ±0.16	1.21 ±0.16	0.37 ±0.00	74.47 ±0.14	**1.15** ±0.14	**1.48** ±0.19	**0.47** ±0.00	**74.95** ±0.13
Lang-Spec Enc[0-8]	**1.54** ±0.12	**2.28** ±0.15	10.64 ±0.01	**74.94** ±0.13	2.71 ±0.2	2.27 ±0.17	**13.25** ±0.02	73.70 ±0.15
Lang-Spec Task	5.87 ±0.22	**2.63** ±0.17	**0.06** ±0.00	**70.07** ±0.16	**4.82** ±0.23	0.66 ±0.17	**0.06** ±0.00	69.68 ±0.14
Lang-Spec Ada(T)	5.21 ±0.22	**2.55** ±0.14	33.77 ±0.04	**71.64** ±0.13	**4.01** ±0.24	0.95 ±0.17	**37.43** ±0.04	70.99 ±0.15
Lang-Spec Ada(F)	**2.25** ±0.17	4.67 ±0.17	2.54 ±0.00	**68.68** ±0.17	2.90 ±0.2	**5.20** ±0.18	**4.15** ±0.00	67.96 ±0.14
Other continual Learning Algorithms								
EWC-Online	6.37 ±0.26	1.72 ±0.16	33.53 ±0.04	70.49 ±0.15	**5.44** ±0.24	**1.83** ±0.17	**38.39** ±0.05	**70.61** ±0.15
ER	3.60 ±0.18	**2.76** ±0.16	34.09 ±0.04	73.96 ±0.14	**2.89** ±0.21	2.20 ±0.16	**38.62** ±0.05	**74.56** ±0.14
KD-Logit	6.42 ±0.29	**2.53** ±0.17	33.85 ±0.04	**71.26** ±0.14	**5.54** ±0.26	1.59 ±0.18	**38.57** ±0.05	69.45 ±0.13
KD-Rep	5.62 ±0.24	**2.29** ±0.17	33.70 ±0.04	**71.54** ±0.15	**5.58** ±0.26	1.51 ±0.18	**38.78** ±0.05	69.5 ±0.14

Table 12: Per language permutation view: a pairwise comparison between Order 5(Hindi → English → Spanish → Thai → French → German) and Order 6 (German → French → Thai → Spanish → English → Hindi). We highlight the best forgetting (lowest), transfer (highest), zero-shot transfer (highest), and final performance (highest) of accuracy and f1 scores among those two orders for each approach in bold, whereas the best scores across approaches for the two orders separately are underlined.

Model	F ↓		T ↑		FP ↑	
	Acc	F1	Acc	F1	Acc	F1
Order 1	1.25 ±0.02	**3.60** ±0.18	0.89 ±0.02	1.76 ±0.17	89.33 ±0.02	65.59 ±0.13
Order 2	5.81 ±0.05	7.89 ±0.28	0.75 ±0.02	0.11 ±0.17	85.81 ±0.02	64.18 ±0.14
Order 3	1.68 ±0.02	4.43 ±0.21	0.77 ±0.02	2.20 ±0.17	89.57 ±0.02	68.88 ±0.14
Order 4	2.70 ±0.04	4.62 ±0.23	0.71 ±0.02	1.22 ±0.17	88.59 ±0.02	68.07 ±0.14
Order 5	1.83 ±0.01	5.74 ±0.24	6.64 ±0.01	**4.89** ±0.15	96.00 ±0.01	71.75 ±0.13
Order 6	**1.08** ±0.01	4.44 ±0.20	**7.09** ±0.01	4.86 ±0.15	**96.40** ±0.01	**71.81** ±0.13

Table 13: Impact of language order across the balanced dataset for *Naive Seq FT*. Best and second best scores for each language for intent classification and slot filling independently across approaches are highlighted in bold and underlined, respectively.

icance p-values for different metrics (forgetting, transfer, and final performance) for intent classification and slot filling, respectively. For example, for forgetting, we notice that improvements or losses from approaches are statistically significant with 95% confidence more than 49% and 61% of the time for intent classification and slot filling. For zero-shot transfer, we notice 60% and 56% of

Model	Test Intent Accuracy On German	English	French	Spanish	Hindi	Thai
			Shared {Trans, Task} Baselines			
Naive Seq FT	1.52 ±0.14	1.13 ±0.10	1.75 ±0.16	1.71 ±0.13	3.26 ±0.50	5.09 ±1.24
Inc Joint	**0.32** ±0.05	**0.13** ±0.04	**0.25** ±0.05	**0.18** ±0.04	0.15 ±0.07	0.30 ±0.07
			Model Expansion Baselines			
Lang-Spec Trans	0.33 ±0.06	0.30 ±0.04	0.43 ±0.07	0.34 ±0.06	**0.41** ±0.08	0.47 ±0.09
Lang-Spec Enc[0-8]	0.54 ±0.07	0.46 ±0.05	0.50 ±0.08	0.57 ±0.06	0.65 ±0.10	0.91 ±0.15
Lang-Spec Task	1.22 ±0.12	0.93 ±0.09	1.47 ±0.14	1.39 ±0.12	3.17 ±0.38	5.44 ±1.62
Lang-Spec Ada(T)	1.11 ±0.10	0.74 ±0.07	1.10 ±0.12	0.94 ±0.09	1.88 ±0.23	5.00 ±1.35
Lang-Spec Ada(F)	0.66 ±0.12	0.51 ±0.07	0.81 ±0.14	0.63 ±0.09	1.00 ±0.14	1.49 ±0.19
			Other continual Learning Algorithms			
EWC-Online	1.49 ±0.14	1.13 ±0.09	1.70 ±0.17	1.83 ±0.14	3.31 ±0.42	5.89 ±1.95
ER	0.84 ±0.07	0.56 ±0.06	0.69 ±0.09	0.70 ±0.06	1.00 ±0.11	2.37 ±0.25
KD-Logit	1.46 ±0.14	0.89 ±0.08	1.77 ±0.16	1.65 ±0.13	2.47 ±0.28	4.75 ±0.84
KD-Rep	1.49 ±0.13	1.14 ±0.09	1.52 ±0.13	1.75 ±0.16	2.52 ±0.24	4.10 ±0.53
	Test Slot Filling On German	English	French	Spanish	Hindi	Thai
			Shared {Trans, Task} Baselines			
Naive Seq FT	3.93 ±1.38	4.11 ±1.18	3.39 ±1.00	2.9 ±0.92	6.12 ±1.91	9.00 ±3.47
Inc Joint	1.19 ±0.88	1.15 ±0.69	**0.70** ±0.68	**0.60** ±0.66	1.75 ±0.73	**0.74** ±0.56
			Model Expansion Baselines			
Lang-Spec Trans	**0.84** ±0.70	**0.94** ±0.60	1.09 ±0.67	1.21 ±0.71	**1.28** ±0.72	1.07 ±0.68
Lang-Spec Enc[0-8]	1.91 ±0.97	1.92 ±0.82	0.97 ±0.72	1.26 ±0.65	1.84 ±0.76	2.01 ±0.78
Lang-Spec Task	3.30 ±1.38	3.05 ±0.94	2.80 ±0.95	2.69 ±0.87	6.91 ±2.03	8.01 ±3.01
Lang-Spec Ada(T)	2.69 ±1.03	3.47 ±1.02	2.40 ±0.81	2.72 ±0.99	5.06 ±1.32	7.08 ±2.50
Lang-Spec Ada(F)	1.46 ±0.82	2.12 ±0.81	1.63 ±0.81	1.63 ±0.96	2.55 ±1.00	4.5 ±1.47
			Other continual Learning Algorithms			
EWC-Online	4.12 ±1.38	4.11 ±1.28	3.01 ±0.97	3.71 ±1.05	6.31 ±1.73	8.58 ±3.39
ER	2.36 ±1.00	2.68 ±0.79	1.45 ±0.98	1.58 ±0.73	3.52 ±1.03	3.69 ±1.20
KD-Logit	3.68 ±1.27	4.20 ±1.08	2.80 ±1.04	3.41 ±1.03	5.67 ±1.56	8.81 ±2.88
KD-Rep	3.93 ±1.31	3.97 ±1.24	3.05 ±0.97	3.12 ±0.97	5.49 ±1.53	8.40 ±2.66

Table 14: CLL per language analysis of forgetting. Best and second best scores for each language are highlighted in bold and underlined respectively.

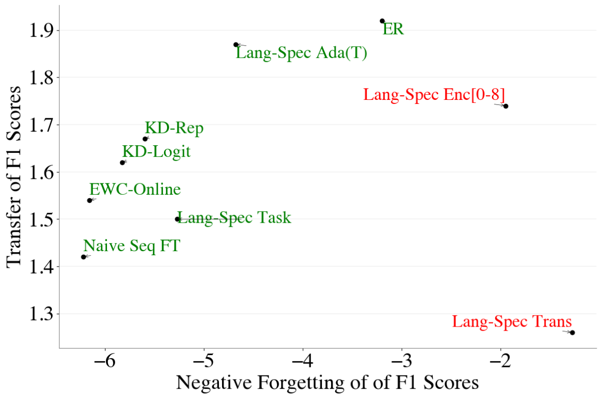
pairwise comparisons are statistically significant for intent classification and slot filling. For final performance, we notice 47% and 49% of pairwise comparisons are statistically significant for intent classification and slot filling. For transfer, we notice that improvements or degradation over transfer of intent classification are not statistically significant with the exceptions of *Lang-Spec Trans* which the lowest in terms of transfer *Lang-Spec Ada(F)* which exhibit high transfer. The same can be said for *Lang-Spec Ada(F)* in slot filling. Overall, model expansion approaches exhibit the highest statistical significance, whereas *EWC-Online* and knowledge distillation are among the lowest.

Model	**Test Intent Accuracy On** German	English	French	Hindi	Spanish	Thai
Shared {Trans, Task} Baselines						
Naive Seq FT	0.8 ±0.07	0.52 ±0.06	1.35 ±0.09	0.83 ±0.07	0.57 ±0.09	0.46 ±0.11
Inc Joint	1.01 ±0.07	0.68 ±0.06	1.48 ±0.08	<u>0.94 ±0.07</u>	0.49 ±0.10	0.5 ±0.11
Model Expansion Baselines						
Lang-Spec Trans	0.25 ±0.08	0.56 ±0.06	0.85 ±0.09	0.57 ±0.08	0.09 ±0.10	0.23 ±0.10
Lang-Spec Enc[0-8]	<u>1.04 ±0.07</u>	<u>0.93 ±0.06</u>	<u>1.54 ±0.07</u>	0.76 ±0.07	0.70 ±0.11	1.01 ±0.10
Lang-Spec Task	-0.25 ±0.12	0.39 ±0.01	0.63 ±0.06	-0.66 ±0.02	0.60 ±0.03	-0.10 ±0.01
Lang-Spec Ada(T)	0.86 ±0.08	0.61 ±0.05	1.16 ±0.08	0.12 ±0.08	0.56 ±0.11	<u>1.44 ±0.12</u>
Lang-Spec Ada(F)	**1.12** ±0.12	**1.72** ±0.09	**3.37** ±0.11	**2.20** ±0.11	**2.77** ±0.18	**4.68** ±0.32
Other continual Learning Algorithms						
EWC-Online	0.79 ±0.07	0.72 ±0.06	1.42 ±0.10	0.82 ±0.07	0.64 ±0.09	0.36 ±0.10
ER	0.88 ±0.07	0.63 ±0.06	1.46 ±0.08	0.78 ±0.08	0.59 ±0.12	0.55 ±0.10
KD-Logit	0.64 ±0.08	0.56 ±0.06	1.36 ±0.08	0.76 ±0.07	<u>0.75 ±0.09</u>	0.48 ±0.10
KD-Rep	0.72 ±0.07	0.75 ±0.05	1.23 ±0.08	0.81 ±0.07	0.67 ±0.10	0.38 ±0.10
	Test Slot Filling On German	English	French	Hindi	Spanish	Thai
Shared {Trans, Task} Baselines						
Naive Seq FT	1.18 ±0.92	1.51 ±0.87	0.36 ±0.93	2.18 ±0.95	-0.19 ±0.9	<u>3.48 ±0.83</u>
Inc Joint	0.68 ±0.95	0.7 ±0.87	0.03 ±0.91	2.25 ±0.95	0.91 ±1.06	3.44 ±0.79
Model Expansion Baselines						
Lang-Spec Trans	0.79 ±0.92	2.0 ±0.77	0.63 ±0.87	1.35 ±0.97	0.4 ±0.87	2.36 ±0.76
Lang-Spec Enc[0-8]	0.88 ±0.87	1.33 ±1.04	0.79 ±0.81	2.16 ±0.94	1.57 ±0.87	3.71 ±0.87
Lang-Spec Task	0.07 ±0.00	0.15 ±0.00	0.07 ±0.00	0.04 ±0.00	-0.02 ±0.00	0.09 ±0.00
Lang-Spec Ada(T)	**3.00** ±0.86	-0.08 ±0.76	<u>2.00 ±1.01</u>	1.21 ±1.03	<u>2.06 ±0.93</u>	3.0 ±0.78
Lang-Spec Ada(F)	<u>2.96 ±1.04</u>	**4.55** ±0.89	**4.38** ±1.02	**4.34** ±1.13	**4.14** ±0.98	**8.07** ±1.01
Other continual Learning Algorithms						
EWC-Online	0.93 ±0.93	1.40 ±0.83	0.93 ±0.83	2.95 ±0.94	0.16 ±0.93	2.89 ±0.82
ER	1.61 ±0.96	1.94 ±0.78	1.11 ±0.86	<u>3.09 ±0.95</u>	0.77 ±0.97	2.97 ±0.85
KD-Logit	0.98 ±0.95	1.32 ±0.81	0.39 ±0.88	2.9 ±1.04	1.09 ±0.87	3.04 ±0.86
KD-Rep	1.36 ±0.95	1.64 ±0.77	0.87 ±0.97	2.98 ±1.04	-0.15 ±0.91	3.32 ±0.79

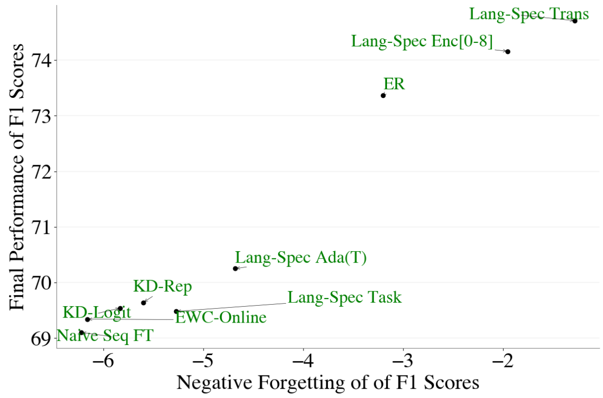
Table 15: CLL per language analysis of transfer. Best and second best scores for each language are highlighted in bold and underlined respectively.

Model	Test Intent Accuracy On					
	German	English	French	Hindi	Spanish	Thai
Shared {Trans, Task} Baselines						
Naive Seq FT	56.68 ±1.55	67.54 ±16.07	60.56 ±3.11	59.15 ±23.1	33.24 ±1.2	18.07 ±0.29
Inc Joint	57.50 ±1.75	70.07 ±12.61	61.55 ±2.89	61.23 ±19.88	32.62 ±2.67	17.73 ±0.29
Model Expansion Baselines						
Lang-Spec Trans	-1.43 ±0.00	0.44 ±0.01	-0.01 ±0.01	-0.95 ±0.01	-0.15 ±0.00	-0.46 ±0.00
Lang-Spec Enc[0-8]	26.14 ±7.42	33.21 ±10.85	25.51 ±7.04	27.18 ±18.12	21.82 ±2.33	11.51 ±0.76
Lang-Spec Task	0.88 ±0.07	0.72 ±0.06	1.55 ±0.08	0.76 ±0.07	0.64 ±0.09	0.59 ±0.09
Lang-Spec Ada(T)	56.76 ±1.41	67.41 ±13.26	60.15 ±4.27	59.04 ±24.16	**35.03** ±4.41	15.83 ±0.59
Lang-Spec Ada(F)	6.39 ±0.09	9.86 ±1.38	9.72 ±0.5	13.41 ±1.18	8.86 ±0.57	1.90 ±0.39
Other continual Learning Algorithms						
EWC-Online	56.99 ±1.76	67.02 ±15.33	60.43 ±2.99	58.6 ±22.11	32.70 ±1.04	18.39 ±0.18
ER	**57.54** ±1.05	68.01 ±17.34	**60.97** ±3.17	**60.05** ±23.77	33.37 ±1.47	18.19 ±0.61
KD-Logit	57.26 ±1.62	**68.06** ±16.59	60.56 ±3.49	59.81 ±23.36	31.31 ±1.12	**18.91** ±0.22
KD-Rep	56.14 ±1.35	67.53 ±16.01	60.22 ±3.17	59.10 ±22.14	31.82 ±1.26	18.01 ±0.55
	Test Slot Filling On					
	German	English	Hindi	Spanish	Thai	
Shared {Trans, Task} Baselines						
Naive Seq FT	44.23 ±1.99	47.92 ±9.98	47.13 ±2.32	46.40 ±15.52	19.10 ±0.31	11.84 ±0.18
Inc Joint	44.49 ±1.53	48.66 ±10.86	**47.85** ±2.25	46.58 ±17.42	18.36 ±0.4	12.09 ±0.24
Model Expansion Baselines						
Lang-Spec Trans	0.45 ±0.00	0.76 ±0.01	0.33 ±0.00	0.83 ±0.01	0.00 ±0.00	0.15 ±0.00
Lang-Spec Enc[0-8]	14.86 ±3.81	15.48 ±6.11	16.09 ±4.06	16.13 ±8.9	6.63 ±1.29	4.82 ±0.34
Lang-Spec Task	1.41 ±1.13	0.62 ±0.81	0.46 ±1.05	2.13 ±1.25	1.58 ±0.97	2.84 ±0.82
Lang-Spec Ada(T)	43.96 ±1.77	46.73 ±8.95	47.32 ±2.83	44.97 ±17.98	**21.23** ±1.24	10.62 ±0.17
Lang-Spec Ada(F)	4.31 ±0.08	4.14 ±0.30	4.44 ±0.29	5.53 ±1.14	2.65 ±0.10	0.73 ±0.03
Other continual Learning Algorithms						
EWC-Online	44.01 ±2.02	47.75 ±9.49	47.10 ±2.47	45.91 ±14.96	19.17 ±0.32	12.45 ±0.14
ER	**44.81** ±1.53	**48.70** ±10.39	47.82 ±2.17	**46.70** ±16.27	19.37 ±0.32	12.08 ±0.20
KD-Logit	44.4 ±2.33	48.13 ±10.07	47.38 ±2.65	46.22 ±15.32	18.93 ±0.50	12.13 ±0.17
KD-Rep	44.14 ±1.86	48.29 ±10.07	47.43 ±2.53	46.06 ±15.25	18.80 ±0.38	**12.21** ±0.16

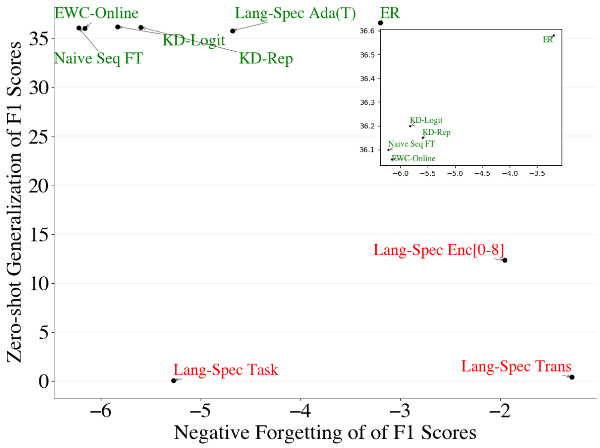
Table 16: CLL per language zero-shot forward transfer. Best and second best scores for each language for intent classification and slot filling independently across approaches are highlighted in bold and underlined respectively.



(a) Transfer versus negative forgetting.

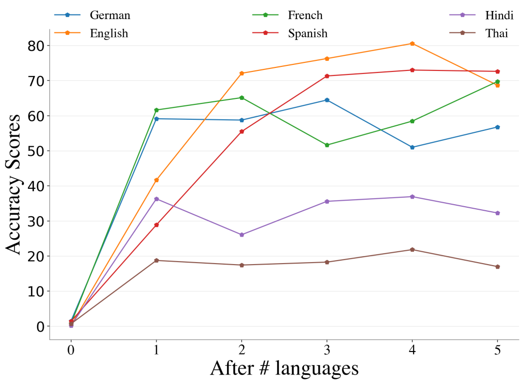


(b) Final performance versus negative forgetting.

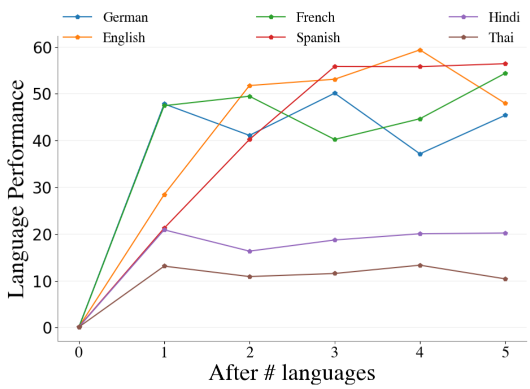


(c) Zero-shot transfer versus negative forgetting.

Figure 8: Transfer, final performance, and zero-shot transfer versus negative forgetting for slot filling task.

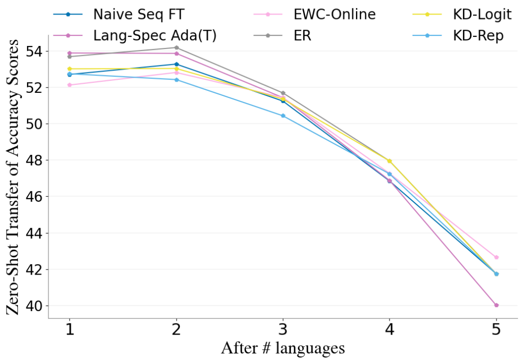


(a) Accuracy for intent classification.

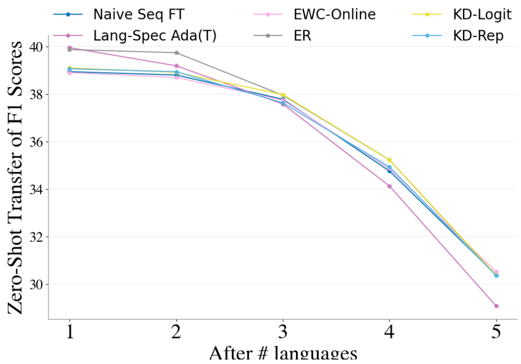


(b) F1 score for slot filling.

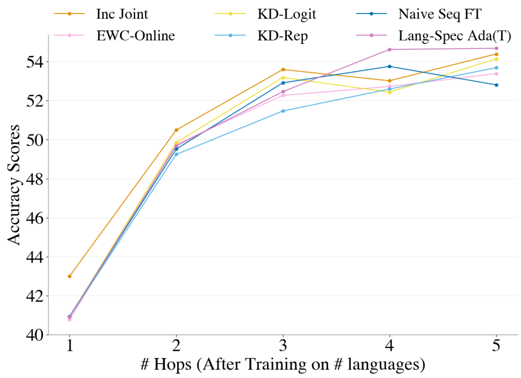
Figure 9: Comparing cross-lingual generalization of *Naive Seq FT* across many hops and different languages for intent classification and slot filling.



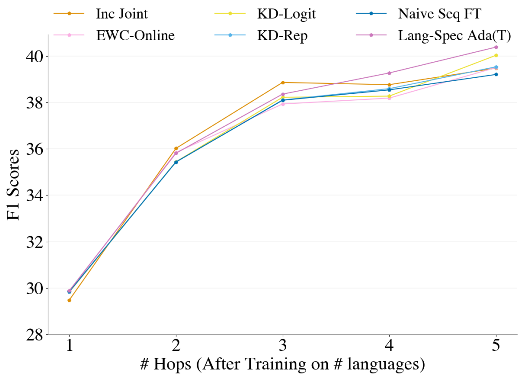
(a) Zero-shot transfer of accuracy for intent classification.



(b) Zero-shot transfer of f1 score for slot filling.

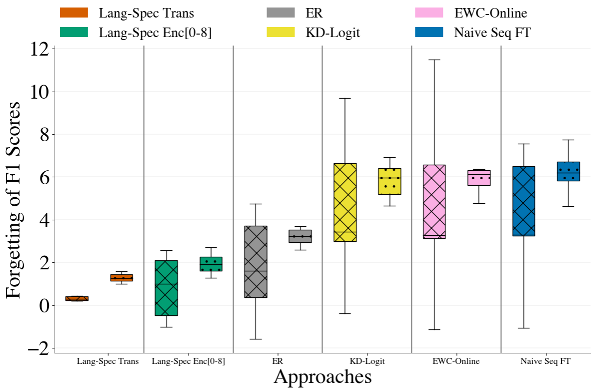


(c) Accuracy for intent classification.

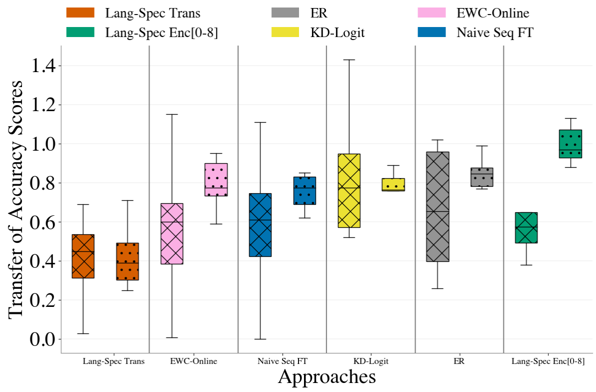


(d) F1 score for slot filling.

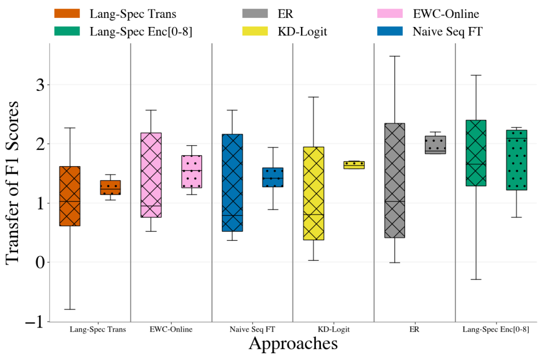
Figure 10: Measuring cross-lingual generalization to new languages across many hops for intent classification and slot filling. This is both in terms of zero-shot transfer metric and plain accuracy and f1 scores.



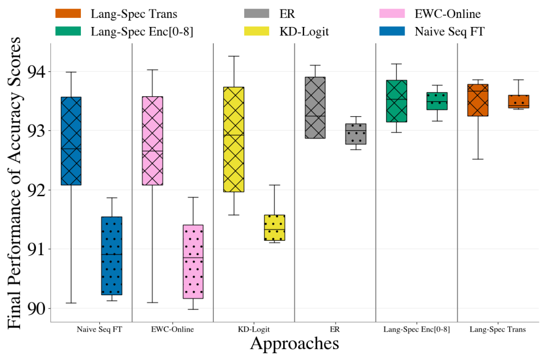
(a) Forgetting for slot filling.



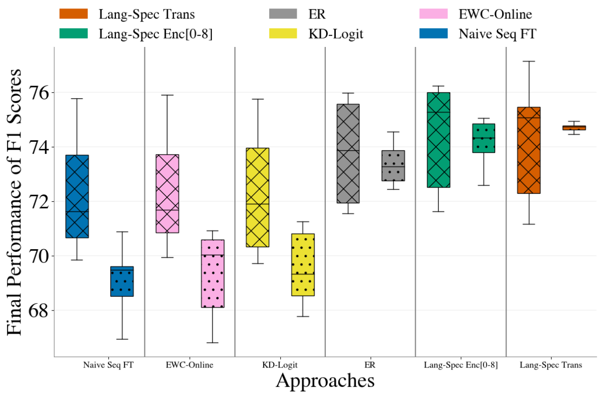
(b) Transfer for intent classification.



(c) Transfer for slot filling.

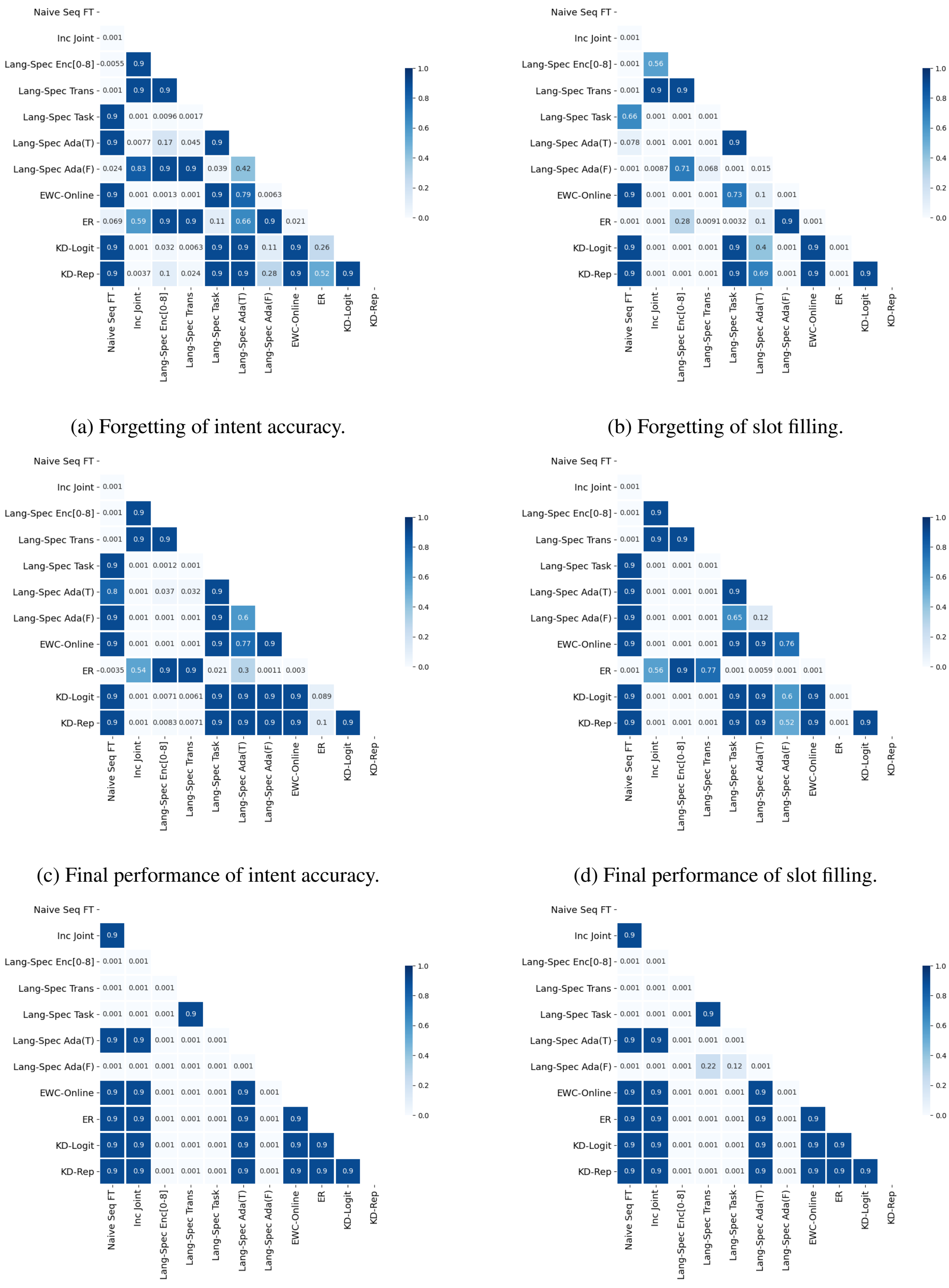


(d) Final performance for intent classification.



(e) Final performance for slot filling.

Figure 11: Comparison between different metrics using two-hop (crossed boxplots) and multi-hop analysis (dotted boxplots), on the left and right respectively for each approach.

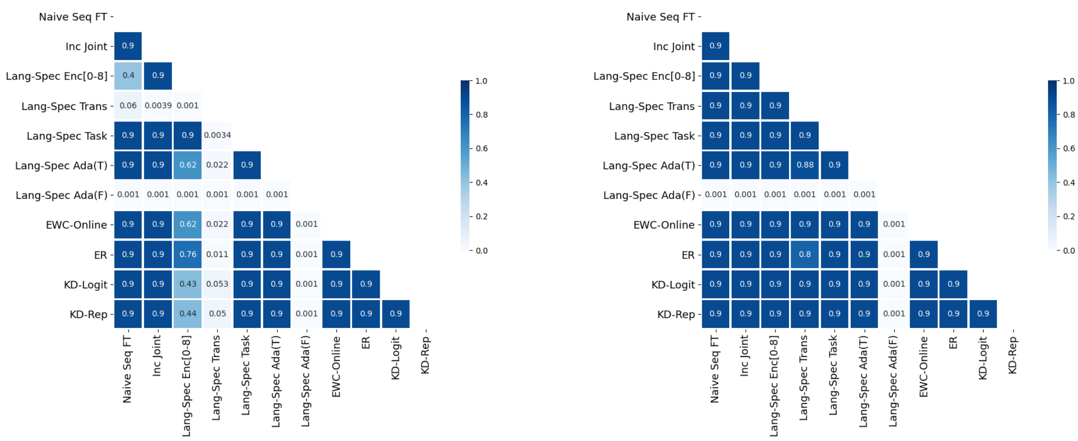


(a) Forgetting of intent accuracy. (b) Forgetting of slot filling.

(c) Final performance of intent accuracy. (d) Final performance of slot filling.

(e) Zero-shot transfer of intent accuracy. (f) Zero-shot transfer of slot filling.

Figure 12: P-values for different pairwise comparison of different continual learning approaches using Tukey's honestly significant difference (HSD) test.



(a) Transfer of intent accuracy.

(b) Transfer of slot filling.

Figure 13: P-values for different pairwise comparison of different continual learning approaches using Tukey's honestly significant difference (HSD) test (Cont.).