Cross-lingual Continual Learning

Meryem M'hamdi **Xiang Ren** **Jonathan May**
Information Sciences Institute
University of Southern California
{meryem, xiangren, jonmay}@isi.edu

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. There has been a large amount of work to adapt such multi-lingual models to unseen target languages. However, the majority of work in this direction focuses on the standard one-hop transfer learning pipeline from source to target languages, whereas in realistic scenarios, new languages can be incorporated at any time in a sequential manner. In this paper, we present a principled **C**ross-lingual **C**ontinual **L**earning (CCL) evaluation paradigm, where we analyze different categories of approaches used to continually 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 different cross-lingual continual learning desiderata, which go beyond conventional transfer learning.

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 without having to pre-train or fine-tune from scratch. 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 fine-tune them, they can neglect to maximize *transfer* (Ruder et al., 2019) to new tasks or languages and are subject to *forgetting* (French, 1993), where performance decreases after exposure to new task or language.

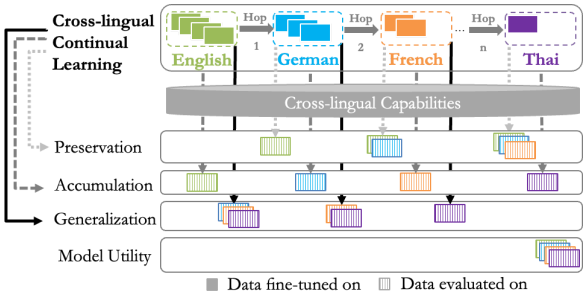


Figure 1: An overview of CCL: We use an example of a non-stationary datastream moving from high to low resource languages. Each bold and dashed box represents either a training or test data instance being fine-tuned or evaluated on, respectively. To support this problem setup, we evaluate the cross-lingual capabilities of *continual approaches*. Those capabilities include knowledge **preservation** on old languages, **accumulation** to the current language, and **generalization** to unseen languages at each point of the training. In addition to that, we evaluate **model utility** at the end of continual learning.

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 is mostly reduced to cross-lingual transfer learning between a pair of languages (Liu et al., 2021; Garcia et al., 2021; Muller et al., 2021; Pfeiffer et al., 2021; Minixhofer et al., 2022). Liu et al. pre-

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 3908–3943

train a (parent) language model and then fine-tune it on a downstream task in one of several different (child) languages. This conflates task 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. propose an adaptation scheme to each new language pair independently while retaining the translation quality on the parent language pairs. Similarly, Muller et al. (2021) and Pfeiffer et al. (2021) propose lexical and semantic level techniques to adapt to target languages. However, all these mentioned works still focus on the 'one-hop' case, consisting of two steps: (1) training on initial parent language(s) (pairs), then (2) adapting to new children language(s) (pairs); the effect of multiple shifts in the datastream is not trivially generalizable to more than one hop. More recently, Pfeiffer et al. (2022) propose an approach for language-specific modules based on adapters and evaluate that on sequential streams of languages. However, they only focus on adapters and two desiderata of continual learning: interference mitigation and transfer maximization. We need a more robust and comprehensive fine-grained evaluation that balances the dynamics between different cross-lingual continual learning desiderata.

In this paper, we pave the way for a more comprehensive multi-hop continual learning evaluation that simulates the sequential learning of a single task over a stream of input from different languages. This evaluation paradigm requires experimentation over *balanced streams* of n data scenarios for $n > 2$. Unlike previous work, this paper concretely defines the following comprehensive goals along with their evaluation metrics as guidelines for analyzing the cross-lingual capabilities of multilingual sequential training: knowledge preservation, accumulation, generalization, and model utility as shown in Figure 1. We apply our test bed to a six-language task-oriented dialogue benchmark and comprehensively analyze a wide variety of successful continual learning algorithms, from previous literature investigated in continual learning contexts different from the cross-lingual context, including (a) model-expansion (Pfeiffer et al., 2020b), (b) regularization (Kirkpatrick et al., 2017), (c) memory replay (Chaudhry et al., 2019b), and (d) distillation-based approaches (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 in enhancing knowledge preservation and accumulation of our multilingual language model. 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 (§4.1). (2) We demonstrate the aggregated effectiveness of a range of different continual learning approaches (Figure 1) at reducing forgetting and improving transfer (§4.3) compared to multilingual sequential baselines (§4.2). (3) We make concrete recommendations on model design to balance transfer and final model performance with forgetting (§4.3). (4) We show that the order of languages and data set size impacts the knowledge preservation and accumulation of multi-lingual sequential fine-tuning and identify the continual learning approaches that are most robust to this variation (§4.4). (5) We analyze zero-shot generalization trends and their correlation with forgetting and show that current continual learning approaches do not substantially improve the generalization (§4.5).

2 Cross-lingual Continual Learning

In this section, we formally define cross-lingual continual learning, describe its goals and challenges, and introduce the downstream tasks, datastreams, and evaluation protocols used. Although we are not the first to define or investigate continual learning for languages, we are to the best of our knowledge the first to define and study cross-lingual continual learning where continual learning is focused on languages only. Thus, we formally define cross-lingual continual learning as learning over a set of languages seen sequentially in multiple hops which is truer to the term of cross-lingual and continual learning, respectively. We distinguish that from 'cross-lingual cross-task cross-stage continual learning' which continually learns over a set of pretraining and downstream tasks sampled from different languages (Liu et al., 2021) and 'cross-

[1]Our code is available at https://github.com/meryemmhamdi1/x-continuous-learning.

lingual one-hop transfer learning' (Garcia et al., 2021).

2.1 Problem Formulation

We define cross-lingual continual learning as the problem of sequentially fine-tuning a model θ for a particular downstream task K over a cross-lingual datastream. In this case, a cross-lingual data *stream* is made of N labeled and distinct datasets $\mathcal{D}_{1\cdots N}$, each one sampled from a distinct language and consisting of separate train and test portions. Let hop_i be the stage in cross-lingual continual learning where θ_i is optimized to θ_{i+1} via exposure to $\mathcal{D}_i$. 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 $p[i] \in \mathscr{L}$ be the ith language in p. The language of $\mathcal{D}_i$ is $p[i]$. Therefore, by default, the number of languages used is equal to the number of datasets. Let $\mathcal{D}_{<i}$ and $\mathcal{D}_{>i}$ refer to a sequence of datasets (train or test portions, depending on context) used in hops from 1 to $i-2$ and i to $N-1$, respectively; we generalize these terms to $\mathcal{D}_{\leq i}$ and $\mathcal{D}_{\geq i}$ by including hop $i-1$ as well at the end or, respectively, beginning of the sequence.

2.2 Goals

We define the goals,[2] necessarily dependent on each other, for our study of cross-lingual continual learning as follows (also depicted in Figure 1):

- *Cross-lingual preservation.* This is the ability to retain previous knowledge on seen languages.
- *Cross-lingual accumulation.* This is the ability to accumulate knowledge learned from previous languages to benefit learning on the current language.
- *Cross-lingual generalization.* This is the ability to generalize uniformly well to unseen languages which goes beyond accumulating knowledge up to the current languages.
- *Model utility.* This is the ability of the fully trained model to perform equally well on all languages.

In this paper, we wish to understand the relationships between these goals. Our aim is to come up with a recipe for a more systematic cross-lingual continual learning. Thus, we need to understand if the goals are aligned with each other or if maximizing some goals lead to minimizing other goals.

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:

- *Catastrophic forgetting.* This happens when fine-tuning a model on $\mathcal{D}_{\geq i}$ leads to a decrease in the performance on $\mathcal{D}_{<i}$.
- *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.
- *Low zero-shot transfer.* This happens when fine-tuning on $\mathcal{D}_{\leq i}$ gives a lower performance than random on unseen $\mathcal{D}_{>i}$.
- *Low final performance.* This is when fine-tuning on all $\mathcal{D}_{\leq N}$ gives an uneven performance between languages 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 multilingual 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.

Datastream Construction. For a set of N languages $\mathscr{L}$, our study considers a permutation subset $P \subset \mathfrak{S}(\mathscr{L})$ with the following properties:[3]

- $|P| = |\mathscr{L}| = N$, i.e. P consists of N permutations, each of which is a sequence of N datasets in each of the N languages in $\mathscr{L}$.
- $\forall \boldsymbol{\ell} \in \mathscr{L}, \forall j \in 1 \ldots N$, there exists some $p \in P$ such that $p[j] = \boldsymbol{\ell}$.
- *H2L* $\in P$, the permutation from most high-resource to most low-resource fine-tuning data sets, based on the training split dataset size.
- *L2H* $\in P$, the reverse of *H2L*.

In our experiments, we use MTOP (Li et al., 2021), which is a multi-lingual task-oriented dialogue

[2] To the best of our knowledge, those goals were never synthesized for the context of cross-lingual continual learning.

[3] Details of the different language permutations used for the datastreams can be found in Appendix C.1.

dataset that covers six typologically diverse languages and spans over 11 domains and 117 intents. We chose MTOP since it is the largest scale dataset available for task-oriented dialogue, and because it covers languages that have varying amounts of data resources available. We use only the flat representation of slots (without nesting) to simplify our evaluation. We use the original data for most experiments. Table 1 shows a summary of the number of sentences (dialogue utterances) per language and split.

Lang	ISO	Train	Dev	Test
English	EN	15,667	2,235	4,386
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: Number of sentences in MTOP per language and split.

2.5 Evaluation Protocols

For each language permutation, we train on each dataset in sequence, but continually evaluate on all languages. Let R be some success metric for evaluating a downstream task K and $R_{i,\leq j}$ be the evaluation on the test set for language $\boldsymbol{\ell}_i$ fine-tuning K on $\mathcal{D}_{\leq j}$. We define the following *meta-metrics* (which are inspired, but slightly different from the metrics in Lopez-Paz and Ranzato (2017) and Chaudhry et al. (2019a)):

- **Forgetting (F ↓).** This is the average forgetting *over all datasets* (excluding the first dataset) computed as:

$$F = \frac{1}{N-1}\sum_{j=2}^{N} F_{\leq j}, \qquad F_{\leq j} = \frac{1}{j-1}\sum_{i=1}^{j-1} F_{i,\leq j}, \tag{1}$$

  where $F_{\leq j}$ is the average forgetting that occurred at the point of training $\mathcal{D}_j$. We compute $F_{i,\leq j} = \max_{k\in[1,j-1]} R_{i,\leq k} - R_{i,\leq j}$. $F_{i,\leq j}$ is the degree to which performance on $\mathcal{D}_i$ has suffered by continuing to train on $\mathcal{D}_{\leq j}$ instead of stopping before covering $\mathcal{D}_j$.
- **Transfer (T ↑).** This is the average forward transfer computed as:

$$T = \frac{1}{N-1}\sum_{i=2}^{N} T_i, \qquad T_i = R_{i,\leq i} - R_i, \tag{2}$$

  where R_i denotes evaluation of a model fine-tuned *only* on $\mathcal{D}_i$. Then, T_i is the incremental impact of sequential training on datasets prior to seeing $\mathcal{D}_i$. To measure *generalization to new languages*, we add a **zero-shot transfer ($\mathbf{T^0}$ ↑)** metric measured as:

$$T^0 = \frac{1}{N-1}\sum_{i=2}^{N} T_i^0, \qquad T_i^0 = \frac{1}{i-1}\sum_{j=1}^{i-1} R_{i,\leq j} - R_i^0, \tag{3}$$

  where T_i^0 is the average performance of a model on the forward transfer to a language $\boldsymbol{\ell}_i$ after training on $\mathcal{D}_{<i}$ compared to the random performance R_i^0 before even fine-tuning on any language (i.e. using fixed pre-trained M-BERT weights and randomly initialized weights for the output layer).
- **Final performance (FP ↑).** This is the average performance after training on all datasets in the studied stream, computed as:

$$FP = \frac{1}{N}\sum_{i=1}^{N} R_{i,\leq N}. \tag{4}$$

3 Methods

For our base model, we use the same M-BERT-based architecture as was used in Castellucci et al. (2019) and M'hamdi et al. (2021) to jointly learn the intent classification and slot filling subtasks of MTOP.[4] On top of that, we define baselines, non-continual learning reference models, and continual learning algorithms.

3.1 Baseline & Reference Models

Before delving into continual learning approaches, we consider a simple lower-bound baseline. In addition to that, we design reference models either trained from scratch for each new language, in a joint manner, or in a sequential multi-hop manner. Those are upper-bound non-continual learning models that are used to assess the performance of different models trained with continual learning methodologies. Those reference models can be in general superior to continual learning models but can also be less efficient and not feasible. For a fair comparison, all models use the same base model architecture and its loss with no further additions or special optimizations to the architecture.

[4]More details about the base model can be found in Appendix B.1.

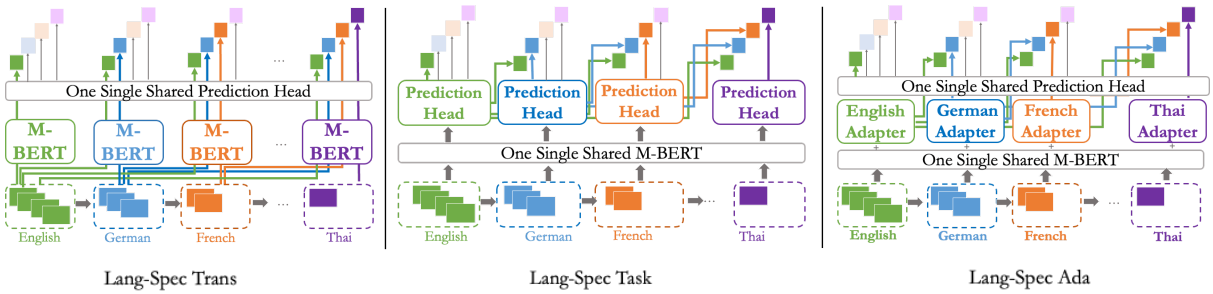


Figure 2: A comparison between different variants of model expansion for this problem setting: either at the side of the input (*Lang-Spec Trans*), the output (*Lang-Spec Task*), or using adapters (*Lang-Spec Ada*).

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

Non-continual Learning Upper-bound Models. These are stronger upper-bound models used as reference points of performance. However, they are either not efficient or prohibitive in the context of cross-lingual continual learning. Some of them require training from scratch for each language which is not efficient. Others require having access to all languages either at the same time or incrementally. Having such access can be restrictive due to privacy or storage efficiency concerns.

- *Language-specific fine-tuning* (*Lang-Spec FT*). This trains independent models on the data set for each language ℓ_i using only $\mathcal{D}_i$.
- *Multi-lingual learning* (*Multilingual*). This trains one single model jointly across all data sets $\mathcal{D}_{\leq N}$.
- *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) $\mathcal{D}_{\leq 1}$, 2) $\mathcal{D}_{\leq 2}$, $\cdots$, and N-1) $\mathcal{D}_{\leq N-1}$. This is the only sequential reference model.

3.2 Continual Learning Approaches

To continually fine-tune on different languages, we establish a representative set of strong approaches[5] spanning the following categories inspired by previous evaluation paradigms such as Jin et al. (2022) lifelong language model domain-incremental pertaining. To the best of our knowledge, we are the first to exhaustively investigate such approaches for the context of cross-lingual continual learning, whereas different approaches were investigated separately for different problem definitions.

Model Expansion. We consider the following approaches, that add hop-specific parameters, as shown in Figure 2. We expand on either the input (i.e. M-BERT representations) or the output side (i.e. task-specific prediction heads). For the former (*Lang-Spec Trans*), the transformer layers are replicated for each hop while sharing the prediction heads. To expand on the output side (*Lang-Spec Task*), we use different prediction heads across hops but share the M-BERT layers. We additionally consider *Lang-Spec Enc[1-9]* which trains M-BERT encoder layers $\in 1 \ldots 9$ in a language-specific manner, while sharing the rest. 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)*).[6]

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

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 (Kariya, 2018) 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 squared error loss on either the output logits or M-BERT representations between the current and previous models.

[5]More details about the approaches and their hyperparameters can be found in Appendix B and C.2, respectively.

[6]More details on adapters and how zero-shot evaluation works for model expansion approaches are in Appendix B.2.

4 Results & Analysis

In this section, we provide an extensive analysis in the form of different ablation studies. We ask critical analysis questions that revolve around the continual learning goals described in §2.2. For §4.2, scores are reported using accuracy (Acc) and F1-score (F1) for intent classification and slot filling, respectively. For the remaining sections, all results are reported for intent classification only, slot filling results, for which the same trends are observed, can be found in Appendix D. Bootstrap sampling (over test data shuffling) is used to compute the average and 95% confidence intervals (averaged over all language permutations except for §4.4). More details can be found in Appendix C.3. We also separately repeat key experiments over 3 different seeds and obtain similar findings which can be found in Appendix E. We decide to report the results using bootstrap sampling since they have tighter confidence intervals.

4.1 How is a Multi-Hop Analysis Different from its One-Hop Counterpart?

To motivate our cross-lingual continual learning evaluation paradigm, we start by investigating how a multi-hop analysis is different from a conventional one-hop transfer learning analysis. Figure 3 shows a comparison between the two in terms of forgetting (Eq. 1) for different approaches aggregated over different language permutations. More results for slot filling and other metrics can be found in Figure 9 in Appendix D.5.

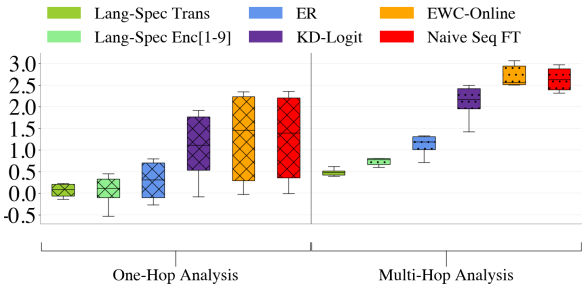


Figure 3: Comparison between forgetting trends for intent classification using one-hop (crossed boxplots on the left) and multi-hop analysis (dotted boxplots on the right), showing the variance over different language permutations. One-hop analysis exhibits higher variance than its multi-hop counterpart.

Lang-Spec Trans tends to have the least forgetting and *Naive Seq FT* the most, but importantly **the variance for the multi-hop analysis is much smaller than that for the one-hop analysis**. Having larger confidence intervals, the one-hop analysis also tends to be misleading in the sense that certain models are depicted as having a good performance while it is not truly the case. For example, *Naive Seq FT*, according to the one-hop analysis, shows a range of forgetting, from very little (0.5) to a lot (2.0). So in some circumstances, it has little forgetting thus a good performance under the one-hop analysis. But according to the multi-hop analysis, it clearly has a lot of forgetting with more confidence. Therefore, the multi-hop analysis leads to a more conclusive analysis. We conjecture that averaging over more hops and balanced diversified datastreams is what leads to narrower confidence intervals. This agrees with the well-known fact that larger sample sizes lead to narrower confidence intervals (Hazra, 2017).

4.2 Can a Multi-lingual Language Model Learn to Preserve and Accumulate Knowledge across Different Languages?

Given the conclusiveness of the multi-hop analysis in §4.1, we follow that type of analysis thereafter. In this section, we investigate how well the baseline and different non-continual learning reference models learn to preserve and accumulate knowledge across different languages, by looking at the average over language permutations. Since not all reference models are sequential, we start by comparing them to the baseline using their final performances (Eq. 4). The final performance is indicative of how well a single final model can encapsulate the knowledge across all languages at the end of training. 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**. In practical scenarios, however, we may not have access to all languages at the same time. Among non-continual learning reference models, *Inc Joint* is closest to *Multilingual* if all data may be preserved. However, this may also not be the case. In that case, *Inc Joint* is nearly as good. Training incrementally and sequentially (*Inc Joint*) is also more beneficial than fine-tuning on just the language of interest (*Lang-Spec FT*), as the former exploits cross-lingual transfer capabilities.

We focus, thereafter, on *Inc Joint*[7] and compare

[7]We do not use *Multilingual* since it is non-sequential. Metrics like forgetting are thus always zero, which makes this model not comparable with other continual learning approaches and sequential reference models.

Model	Intent Class (Acc)	Slot Filling (F1)
Naive Seq FT	91.06 ±1.08	69.37 ±1.06
Lang-Spec FT	93.40 ±0.08	73.90 ±0.83
Inc Joint	<u>94.16 ±0.18</u>	<u>74.88 ±0.38</u>
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 <u>underline</u> the second best across models.

its forgetting (Eq. 1) and transfer (Eq. 2) trends to the baseline *Naive Seq FT*, as shown in Table 3. *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, *Naive Seq FT* seems to slightly outperform *Inc Joint* in terms of transfer. This difference is not statistically significant.[8] 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.

Model	Intent Class (Acc)		Slot Filling(F1)	
	F ↓	T ↑	F ↓	T ↑
Naive Seq FT	2.93 ±1.24	**0.68** ±0.14	5.67 ±0.93	**1.37** ±0.53
Inc Joint	**0.11** ±0.10	0.52 ±0.19	**0.91** ±0.34	0.83 ±0.77

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** for each subtask and metric.

4.3 Is Continual Learning Effective in Boosting Knowledge Preservation, Accumulation, and Model Utility?

To study the effectiveness of continual learning approaches, we compare them to the baseline using the average over language permutations. We show, in Figures 4(a) and 4(c), the final performances (Eq. 4) and transfer (Eq. 2) of different approaches, respectively, versus their negative forgetting (Eq. 1). In general, we observe that continual learning approaches mitigate forgetting and improve final performance. They also improve transfer, to some degree, though gains are mostly not significant compared to *Naive Seq FT*(Appendix C.3) .

From Figure 4(a), we notice that model expansion approaches[9](*Lang-Spec Trans* and *Lang-Spec Enc[1-9]* described previously) are good at mitigating forgetting and improving the final performance while *Lang-Spec Task* is not. M-BERT, when trained in a language-specific manner, is responsible for encapsulating the cross-lingual representations necessary for enabling knowledge preservation, whereas 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)* is one way to do this more efficiently, but its performance still lags behind other model expansion approaches. *ER* achieves a performance close to *Lang-Spec Trans* and *Lang-Spec Enc[1-9]* and this suggests that **using a portion of the memory is beneficial**.[10]

In the baseline approach which suffers from the highest forgetting, we also notice the lowest final performance and transfer in Figures 4(a) and 4(c). 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 learn a stronger final model**. However, there is no direct correlation between forgetting and transfer. For example, *Lang-Spec Trans* is the best model in 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 of a model, which is the difference between the performance of that model and that of *Lang-Spec FT*, is almost null. On the other hand, although *Lang-Spec Ada(F)* has the highest transfer, it has the lowest final performance and close to average forgetting. Although the adapter will not be updated anymore after the model has been fine-tuned on, we think that the forgetting could be due to the shared task specific-layer leading to a forgetting closer to Lang-Spec Trans more than Lang-Spec Ada(T) which also shares M-BERT and tunes it. We show in Figure 4(b) that there is no direct correlation between final performance and transfer. This posits that all three metrics need

[8]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 C.3.

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

[10]An ablation study using different sizes of the memory is shown in Appendix D.6. It shows that even smaller sizes up to 5% are still beneficial. We report here the highest memory size as it leads to the best results.

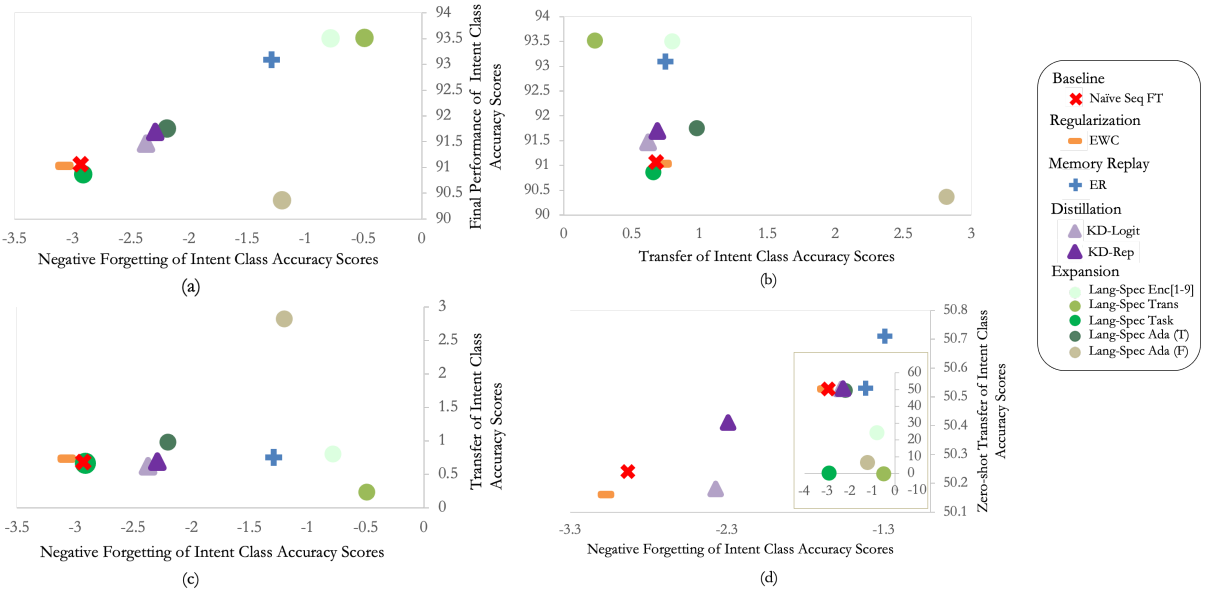


Figure 4: Correlations between different pairs of metrics: (a) Final performance versus negative forgetting for the task of intent classification. The lower the forgetting the higher the final performance. (b) Final performance versus transfer for the task of intent classification. As hypothesized, there is no direct correlation between final performance and transfer. (c) Transfer versus negative forgetting for intent classification task. In general, there is no direct correlation between transfer and forgetting. (d) Zero-shot generalization versus negative forgetting for intent classification. Model expansion approaches are highlighted in shades of green. We zoom over the rest of the models in the main graph and show an overview of all approaches in the lower right corner subplot. Mitigating forgetting leads to higher generalization with the exception of multi-headed models highlighted in green.

to be studied independently for a more insightful analysis.

4.4 Which Permutations Impose More Challenges on Knowledge Preservation, Accumulation, and Model Utility?

So far our analysis has focused on the average over language permutations, but are the same patterns observed for different language permutations? To shed light on this, we analyze the performance of different continual learning algorithms and the baseline in terms of their forgetting (Eq. 1), transfer (Eq. 2), and final performance (Eq. 4) over *H2L* and *L2H* permutations, in Table 4.[11] In general, we observe that it is **more challenging to learn from low to high resource languages**. However, model expansion and memory replay approaches reduce forgetting and final performance gaps between language permutations. We hypothesize that *L2H* being more challenging than *H2L* could be due to the fine-tuning data size that is different between languages.

To verify this hypothesis, we dig deeper to check if the differences among fine-tuning data sizes between languages is the main factor by performing an ablation study on that. Therefore, we use the same amount of fine-tuning and evaluation resources for each language (9,219 for train, 1,285 for dev, and 2,299 for test splits) and report the results on *Naive Seq FT* in Table 5. We notice that there is still a gap between these two language

Model	F ↓		T ↑		FP ↑	
	H2L	*L2H*	*H2L*	*L2H*	*H2L*	*L2H*
Naive Seq FT	**1.52**	5.52	**0.93**	0.57	**92.06**	88.80
Lang-Spec Trans	**0.40**	0.62	**0.59**	0.03	**93.86**	93.37
Lang-Spec Enc[1-9]	***0.60***	*1.05*	**1.00**	0.63	***93.75***	*93.15*
Lang-Spec Task	**1.53**	5.53	**0.84**	0.38	**91.93**	87.68
Lang-Spec Ada(T)	**1.18**	4.43	***1.29***	*0.79*	**92.36**	88.66
Lang-Spec Ada(F)	**0.84**	1.87	**3.41**	2.43	**91.08**	89.92
EWC	**1.82**	5.90	**0.74**	0.48	**91.16**	88.28
ER	**0.71**	2.35	**0.95**	0.78	**93.51**	92.58
KD-Logit	**1.42**	4.07	**0.77**	0.51	**91.60**	89.65
KD-Rep	**1.49**	4.00	**0.96**	0.53	**91.64**	90.17

Table 4: Comparison of intent classification for two language permutations. We highlight in **bold** the best forgetting (F), highest transfer (T), and final performance (FP) of accuracy scores among *H2L* and *L2H*, whereas the best and second best scores across approaches for *H2L* and *L2H* separately are underlined and *italicized*, respectively. We report mean performances for each metric and language order. All 95% confidence intervals range from ± 0.01 to ± 0.04.

[11] Full results for slot filling, more language permutations, and a balanced version of data can be found in Appendix D.3.

permutations for forgetting and final performance. This suggests that the difference in fine-tuning 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.

Model	F ↓		T ↑		FP ↑	
	H2L	*L2H*	*H2L*	*L2H*	*H2L*	*L2H*
Original Data	**1.52**	5.52	**0.93**	0.57	**92.06**	88.80
Balanced Data	**1.25**	5.81	**0.89**	0.75	**89.33**	85.81

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 **bold** the best among *H2L* and *L2H* for each metric.

4.5 How do Continual Learning Models Generalize to Unseen Languages?

To analyze the zero-shot transfer to languages unseen during fine-tuning, we plot the performance of zero-shot transfer (Eq. 3) as a function of negative forgetting over the average of different language permutations, to investigate any relationships between generalization and preservation. In Figure 4(d), we infer that **most continual learning approaches do not substantially improve generalization compared to *Naive Seq FT***. In particular, model expansion approaches (in red) hurt generalization 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 forgetting previous tasks. Except for model expansion approaches, 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.

5 Related Work

Continual learning for cross-lingual NLP work is under-explored, either focusing on proposing cross-lingual approaches that indirectly support continual learning, such as Artetxe et al. (2020), of 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); Pfeiffer et al. (2021); Minixhofer et al. (2022), who propose different lexical and semantic approaches to adapt to new low-resource languages for different downstream tasks. Similarly, 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. Muller et al. (2021) analyze the adaptability and usability of large language models to unseen and under-studied low-resource languages. However, they all focus on a one-hop analysis from high to low-resource language pairs or pre-training to fine-tuning tasks, unlike our work, which analyzes across multiple hops. More recently, Pfeiffer et al. (2022) propose a new methodology based on adapters and show that their approach mitigates negative interference between languages while enabling positive transfer. They use a multi-hop evaluation paradigm closer to our setup, but they only evaluate with respect to adapters using interference and transfer and do not analyze other aspects of cross-lingual continual learning capabilities.

6 Conclusion

We formulate the cross-lingual continual learning problem setup. We show that naive sequential fine-tuning is prone to catastrophic forgetting and has poor accumulation capabilities sensitive to different language permutations. 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 improving final model performance and to a lesser degree accumulation and 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 cover more benchmarks and datastream setups to gain more insights that go beyond conventional cross-lingual transfer learning.

Limitations

Application to Other Benchmarks A central limitation of our work is that the main experiments are based on a single task-oriented dialogue benchmark. While there are multiple other natural language understanding benchmarks like XNLI, XQUAD, MLQA, and PAWS-X (Conneau et al., 2018; Artetxe et al., 2020; Lewis et al., 2020; Yang et al., 2019) that can also be used to back up our claims, we argue that this is outside the scope of this paper. The main objectives of this paper are

to first come up with a new definition of a cross-lingual continual learning challenge and then to give an example using a comprehensive and realistic benchmark like task-oriented dialogue to catalyze more research in that direction.

Choice of Realistic Permutations For more realistic setups of continual learning, we need to come up with an approach to define continual learning annotation scenarios of languages. Rather than using brute force with all possible ways the languages could be annotated at different stages, a principled way would be more desired. Since it is hard to tell if there is any logic or pattern in the annotation process itself and given the sheer amount of realistic scenarios, we chose one scenario experienced by some of the users: a model is built for a user, then the user reveals that more languages are desired. We test in our work the plausibility of continual learning approaches where the sequence moves from one language to another without repetition of the same language. Working on scenarios where the data from different languages are integrated as soon as they are annotated, implying different languages for different hops, is out of the scope of this paper.

Data and Model Size Analysis In this paper, we pick certain model expansion approach variations to analyze the effect of model components (one aspect of model size) and two data distribution scenarios. However, analyzing extensively the effect of the scale of data and model size is beyond the scope of our work. We agree that different data sizes can be used and it is interesting to analyze different supervision levels such as using different proportions of the data for each language and simulating few-shot scenarios. We believe that for low-resource scenarios we need to investigate specific approaches to continual learning like meta-learning. We plan to investigate that in future work.

Application to Other Transformers Another possible limitation of our work is the restriction of the evaluation to a base model on top of M-BERT Transformers. With the advent of Transformer-based encoders as strong pillars for transfer-learning, several Transformers such as XLM-R have been proposed more recently. Although those models have been shown to outperform M-BERT on numerous downstream applications especially on low-resource languages (Conneau et al., 2020), M-BERT is still largely used due to its reduced number of parameters. In our specific continual learning challenge, efficiency is a top concern as we are training in multiple hops and benchmarking on different models. So, M-BERT has been feasible in our use case. We leave experimenting with other Transformer-based encoders to future work.

Acknowledgements

This material is partially based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19051600007. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. We also would like to extend our thanks to Xisen Yin for insightful discussions on continual learning and all anonymous reviewers and meta-reviewers for their valuable feedback.

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*, pages 4623–4637, Online. 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*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multi-lingual intent detection and slot filling in a joint bert-based model. *ArXiv preprint*, 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. *ArXiv preprint*, 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.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. 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 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13122–13131.

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*, pages 745–758, Suzhou, China. Association for Computational Linguistics.

Avijit Hazra. 2017. Using the confidence interval confidently. *Journal of Thoracic Disease*, 9(10):4124–4129.

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

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4764–4780, Seattle, United States. Association for Computational Linguistics.

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.

Mahendra Kariya. 2018. Dark knowledge in neural networks. Accessed on March 7th, 2023.

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*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7315–7330, Online. Association for Computational Linguistics.

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*, pages 2950–2962, Online. 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 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*, pages 7452–7467, Online and Punta Cana, Dominican Republic. 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.

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

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. 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: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

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

Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online structured laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3742–3752.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. 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 2990–2999.

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, Tinne Tuytelaars, and Andreas S. Tolias. 2022. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197.

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.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

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 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) and memory-based approaches (Shin et al., 2017; Chaudhry et al., 2019b,a). Only recently, continual learning has started gaining more interest in the NLP community. Most efforts on continual learning for NLP have focused on classification tasks and fall into the category of domain or class incremental continual learning (Han et al., 2020). Current approaches often fail to effectively retain previous knowledge and adapt to new information simultaneously (Biesialska et al., 2020; de Masson d'Autume et al., 2019). New challenges are formulated to study the problem of continual learning from different perspectives. Jin et al. (2022) formulate the lifelong learning pretraining challenge, where pertaining language models continually adapt to emerging data from new corpora.

Continual learning for cross-lingual NLP is underexplored, either focusing on proposing cross-lingual approaches that indirectly support continual 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), who propose a lexical approach to adapt to new low-resource languages for machine translation. Similarly, Pfeiffer et al. (2021) propose lexical-level adaptation schemes that can be applied to models relying on subword-based tokenization to adapt them to low-resource languages not covered or whose scripts are unseen during pre-training. Minixhofer et al. (2022) also propose adaptations that go beyond the lexical level. Their approach facilitates the creation of monolingual language models that are transferable for new languages. 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 all focus on a one-hop analysis from high to low-resource language pairs or pre-training to fine-tuning tasks, unlike our work, which analyzes across multiple hops. Muller et al. (2021) analyze the adaptability and usability of large language models to unseen and under-studied low-resource languages. Based on that and depending on the degree of additional pre-training and fine-tuning required, they categorize the low-resource languages into easy, intermediate, and hard. Although this work paves the way for a better understanding of the mechanics to transferability to low-resource scenarios, they do not study the scenario where the transferability needs to be performed in multiple hops following a sequential stream of data. More recently, Pfeiffer et al. (2022) propose a new methodology for language-specific modules to add additional capacity and deal with the curse of multilinguality and show that their approach mitigates both negative interference between languages while enabling positive transfer. They use a continual learning multi-hop evaluation paradigm which is closer to our setup but they

only evaluate using interference and transfer and only using one approach based adapters and do not analyze other aspects of cross-lingual continual learning capabilities using a holistic approach like our work.

B More Details about Approaches

B.1 Base Model Architecture

We use the same architecture as in Castellucci et al. (2019) and M'hamdi et al. (2021) to jointly learn intent classification and slot filling subtasks. As shown in Figure 5, 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.

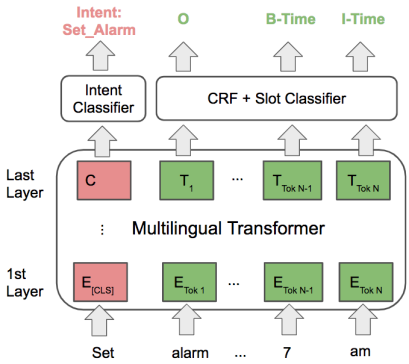


Figure 5: Architecture of task-oriented dialogue base model.

B.2 Model Expansion

Model expansion methods, such as *Lang-Spec Trans* and *Lang-Spec Enc[1-9]*, are fine-tuned for each language with either an entirely or partially language-specific M-BERT (whole 12 layers in addition to the embeddings or just the top 8 layers in the case of *Lang-Spec Trans* and *Lang-Spec Enc[1-9]* respectively). When fine-tuning them on a new language, the previously tuned parameters on the old languages are retained unchanged while the rest of the parameters that are not language-specific are fine-tuned. During the evaluation on a particular language, the tuned parameters for that language are restored and used if the language has been seen in training. Otherwise, the parameters initialized from M-BERT (before fine-tuning on any language) are used for zero-shot evaluation.

Adapters consist of downsampling layers followed by upsampling layers inserted between layers of our Transformer encoder in addition to their invertible components. We do not add task-specific adapters, as according to our ablation studies they didn't prove beneficial. We add adapter components to every encoder layer following MAD-X configuration and using their pre-trained weights.[12] We either fine-tune the weights for the languages available in AdapterHub or train from scratch for languages for which there are no pre-training adapter weights. At inference time, we use adapter layers fine-tuned independently for each language in the datastream.

B.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:

$$\mathcal{L}^i_{total} = \mathcal{L}^i_{cur} + \lambda \mathcal{L}^i_{reg}, \tag{5}$$

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 (which is fixed to 20). For efficiency purposes, we use the online version of EWC (*EWC-Online*). Following that, our regularization term is computed as, based on the formulation in van de Ven et al. (2022):

$$\mathcal{L}^i_{reg} = \sum_{j=1}^{N_p} \tilde{F}^{(i-1)}_{jj} (\theta_j - \theta^k_j)^2, \tag{6}$$

where θ are the parameters of the Transformer 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

[12] obtained from AdapterHub (Pfeiffer et al., 2020a) https://adapterhub.ml/explore/text_lang/

all $j \in 1 \ldots (i-1)$. $\tilde{F}^{(i)}_{jj} = \gamma \tilde{F}^{(i-1)}_{jj} + F^{i}_{jj}$ and $\tilde{F}^{1}_{jj} = F^{1}_{jj}$. In practice, F^i is simply the gradients all parameters flattened into one single matrix.

B.4 Experience Replay (ER)

After training for each $\mathcal{D}_i$ for all $i \in 1 \ldots N$, we populate the memory with randomly sampled examples from $\mathcal{D}_i$. For each $\mathcal{D}_i$ for all $i \in 2 \ldots N$, 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 $j \in 1 \ldots (i-1)$ and replays them using the current model checkpoint used for training on $\mathcal{D}_i$. We retrieve an equal amount of memory items 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.

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

We use the same strategy explained in §B.4 to select the memory to be replayed using a knowledge distillation loss. For each $\mathcal{D}_i$ for all $i \in 2 \ldots N$, after training for every $k = 100$ mini-batches, we randomly sample an equal batch from the memory for each $\mathcal{D}_j$ such that $j \in 1 \ldots (i-1)$. 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*.

C Experimental Setup Details

C.1 Datastreams

We use the following datastreams 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.

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.

C.2 Implementation Details

For all experiments, we use M-BERT(bert-base-multilingual-cased)[13] 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 manual 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 based on the dev performance we 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 bootstrap experiments. For additional experiments using multiple seeds, we fix three seeds. 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 and the mode of training used as shown in Table 7. With the exception of model expansion and language-specific 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 $\frac{1}{6}$ 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 is around 24% and takes 2 times less than

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

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. What impacts the runtime of ER is much more than just iterating over a small sampled memory. Its runtime does not only depend on the size of the memory as much as it depends on the frequency of interleaving happening at the fine-tuning schedule. After each k minibatch steps, we sample a minibatch from the memory and fine-tune on it interleaved with the fine-tuning on the main minibatch. So, that makes the runtime depend on k and not only the size of the memory. This make its training more time consuming than if we had to sample only after each epoch with the same memory size.

Model	Runtime	# Param
Naive Seq FT	3h16min	178,081,402
Lang-Spec FT	5h02min	1,068,488,412
Inc Joint	1d22h51min	178,081,402
Multilingual	16h45min	178,081,402
Lang-Spec Embed	7h46min	639,123,322
Lang-Spec Enc[1-3]	7h52min	284,399,482
Lang-Spec Enc[4-6]	7h12min	284,399,482
Lang-Spec Enc[7-9]	7h8min	284,399,482
Lang-Spec Enc[10-12]	7h20min	284,399,482
Lang-Spec Enc[1-9]	8h1min	497,035,642
Lang-Spec Trans	7h15min	1,067,348,602
Lang-Spec Enc[1-12]	7h53min	603,353,722
Lang-Spec Enc[1-6]	7h16min	390,717,562
Lang-Spec Enc[7-12]	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.

C.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 §2.5 over a single approach. For each language permutation, and for each $R_{i,\leq j}$, representing some performance metric on language $\boldsymbol{\ell}_i$ after training on $\mathcal{D}_{\leq j}$, 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 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:

$$CI = \frac{1.9639 \times std(x)}{\sqrt{600}}, \qquad std(x) = \sqrt{\frac{\sum (x - \bar{x})^2}{600}}. \tag{7}$$

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[14] over different language permutations for each metric.

D More Results & Analysis using Boostrap Sampling

D.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 applies to slot filling. This could be due to the transfer between intent classification and slot filling that is maximized when training them jointly. The best model for transfer is *Lang-Spec Ada(F)*, which we hypothesize is due to its lightweight adaptation to the current language which makes it overfit on that at the cost of a lower average and final performance overall.

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

Model	F ↓		T ↑		T^0 ↑		FP ↑	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Shared {Trans, Task} Baselines								
Naive Seq FT	2.93 ±1.24	5.67 ±0.93	0.68 ±0.14	1.37 ±0.53	50.24 ±3.43	36.32 ±1.91	91.06 ±1.08	69.37 ±1.06
Lang-Spec FT							93.40 ±0.08	73.90 ±0.83
Lang-Spec FT + Ada(T)							93.04 ±0.09	72.90 ±0.80
Lang-Spec FT + Ada(F)							88.79 ±0.13	67.46 ±0.89
Inc Joint	**0.11** ±0.10	**0.91** ±0.34	0.52 ±0.19	0.83 ±0.77	50.07 ±2.48	36.39 ±2.60	94.16 ±0.18	74.88 ±0.38
Multilingual							**94.25** ±0.07	**76.34** ±0.82
Model Expansion Baselines								
Lang-Spec Trans	0.49 ±0.08	1.32 ±0.23	0.23 ±0.21	0.95 ±0.21	-0.43 ±0.16	0.42 ±0.06	93.51 ±0.18	74.74 ±0.20
Lang-Spec Enc[1-9]	0.78 ±0.15	1.95 ±0.51	0.80 ±0.19	1.44 ±0.71	24.23 ±1.73	12.32 ±1.24	93.50 ±0.21	74.19 ±0.92
Lang-Spec Task	2.91 ±1.26	5.26 ±1.01	0.66 ±0.18	1.15 ±1.15	0.10 ±0.25	0.07 ±0.02	90.86 ±1.46	69.41 ±1.57
Lang-Spec Ada(T)	2.19 ±1.12	4.23 ±1.26	0.98 ±0.18	2.04 ±0.92	49.35 ±3.64	33.60 ±2.98	91.75 ±1.39	71.13 ±1.68
Lang-Spec Ada(F)	1.20 ±0.35	3.35 ±0.85	**2.82** ±0.33	**3.93** ±0.68	6.52 ±2.16	2.80 ±0.59	90.36 ±0.37	68.55 ±1.10
Other Continuous Learning Algorithms								
EWC	3.07 ±1.32	5.78 ±1.00	0.73 ±0.12	1.46 ±0.65	50.16 ±3.48	36.31 ±1.94	91.03 ±1.26	69.63 ±1.52
ER	1.29 ±0.51	3.06 ±0.59	0.75 ±0.17	1.47 ±0.85	**50.71** ±3.55	**36.91** ±2.14	93.09 ±0.29	73.00 ±0.52
KD-Logit	2.37 ±0.83	5.53 ±0.96	0.62 ±0.15	1.40 ±0.68	50.18 ±3.14	36.25 ±1.91	91.46 ±0.87	69.64 ±1.58
KD-Rep	2.29 ±0.80	5.35 ±0.69	0.69 ±0.20	1.43 ±0.59	50.41 ±2.92	36.26 ±1.96	91.69 ±0.71	70.03 ±1.09

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 ↓		T ↑		T^0 ↑		FP ↑	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Naive Seq FT	2.93 ±1.24	5.67 ±0.93	0.68 ±0.14	1.37 ±0.53	*50.24* ±3.43	*36.32* ±1.91	91.06 ±1.08	69.37 ±1.06
Lang-Spec FT							93.40 ±0.08	73.90 ±0.83
Lang-Spec Trans	**0.49** ±0.08	1.32 ±0.23	0.23 ±0.21	0.95 ±0.21	-0.43 ±0.16	0.42 ±0.06	93.51 ±0.18	**74.74** ±0.20
Lang-Spec Enc[1-12]	**0.49** ±0.08	**1.30** ±0.16	0.23 ±0.21	0.77 ±0.31	-0.31 ±0.18	0.57 ±0.09	**93.52** ±0.12	74.51 ±0.25
Lang-Spec Embed	3.13 ±1.35	5.88 ±0.95	0.74 ±0.20	1.24 ±0.79	50.67 ±2.98	36.62 ±1.89	90.69 ±1.28	69.59 ±1.23
Lang-Spec Enc[1-3]	1.88 ±0.77	4.32 ±0.69	0.77 ±0.19	1.37 ±0.64	**52.20** ±3.23	**37.42** ±1.99	92.25 ±0.76	71.59 ±1.52
Lang-Spec Enc[4-6]	1.47 ±0.65	2.87 ±0.36	*0.78* ±0.23	1.61 ±0.45	47.83 ±3.00	34.66 ±1.79	92.71 ±0.65	73.06 ±0.97
Lang-Spec Enc[7-9]	1.45 ±0.56	3.02 ±0.52	0.70 ±0.16	1.32 ±0.52	38.33 ±3.00	23.68 ±2.36	92.43 ±0.78	72.28 ±1.05
Lang-Spec Enc[10-12]	2.21 ±0.86	4.14 ±0.84	0.47 ±0.24	1.35 ±0.56	41.38 ±2.13	20.04 ±1.89	91.41 ±1.08	71.14 ±1.13
Lang-Spec Enc[1-6]	1.27 ±0.67	2.99 ±0.62	**0.87** ±0.17	**1.64** ±0.65	45.23 ±2.56	31.21 ±2.17	92.92 ±0.52	73.33 ±1.09
Lang-Spec Enc[7-12]	1.66 ±0.36	3.37 ±0.69	0.31 ±0.33	0.65 ±0.73	6.04 ±1.13	4.53 ±0.96	91.97 ±0.38	71.63 ±1.15
Lang-Spec Enc[1-9]	*0.78* ±0.15	*1.95* ±0.51	0.80 ±0.19	*1.44* ±0.71	24.23 ±1.73	12.32 ±1.24	*93.50* ±0.21	*74.19* ±0.92
Lang-Spec Enc[10-12]	2.21 ±0.86	4.14 ±0.84	0.47 ±0.24	1.35 ±0.56	41.38 ±2.13	20.04 ±1.89	91.41 ±1.08	71.14 ±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 in **bold**, underlined, and *italicized* respectively.

D.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* and *Lang-Spec Enc[1-12]* have the best in terms of forgetting and final performance. Training only the first 8 encoder layers *Lang-Spec Enc[1-9]*, excluding embeddings, in a language-specific manner comes next in terms of a low forgetting and a comparable final performance, with a relatively better transfer and zero-shot transfer performance. Other good model reaching a good compromise between transfer, zero-shot transfer and forgetting with less language-specific layers are *Lang-Spec Enc[1-3]* and *Lang-Spec Enc[1-6]*. *Naive Seq FT* is comparable to those model-expansion approaches in terms of zero-shot performance, but has a lower final performance and significantly 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 §4.5).

D.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, *L2H* (Thai → Spanish → Hindi → French → German → English) is still the most challenging one in terms of knowledge preservation, accumulation, generalization, and model utility. *H2L* (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.

D.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 and higher zero-shot transfer 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 between different languages. Approaches like *Lang-Spec Trans*, *Lang-Spec Enc[1-9]*, *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 language do so uniformly for all languages. The performance in terms of zero-shot transfer is significantly lower in the case of Thai.

D.5 More Analysis

Figure 6 plots final performance versus negative forgetting, final performance versus transfer, transfer versus negative forgetting, 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 7a and 7b 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 the 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 8a and 8b 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 decreases after seeing n languages, where $n \in [1 \dots 5]$. In this case, after seeing one language, 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 8c and 8d 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 in terms of both intent accuracy and slot filling F1 scores 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 9 shows more results for multi-hop versus one-hop analysis for more metrics and tasks. In general, we can observe the same trend, whereby multi-hop dotted boxplots analysis has smaller confidence intervals than one-hop crossed boxplots.

D.6 Experience Replay Ablation Studies

Table 17 shows a comparison between the performance of experience replay variants with different memory sizes ranging from 750 to 6000 instances which accounts for 5% to 60% of the training data

Model	*H2L*				*L2H*			
	Test Intent Accuracy On							
	F ↓	T ↑	T^0 ↑	FP ↑	F ↓	T ↑	T^0 ↑	FP ↑
Shared {Trans, Task} Baselines								
Naive Seq FT	**1.52** ±0.02	**0.93** ±0.02	**50.68** ±0.03	**92.06** ±0.02	5.52 ±0.04	0.57 ±0.01	44.66 ±0.02	88.80 ±0.02
Lang-Spec FT				93.40 ±0.08				93.40 ±0.08
Lang-Spec FT + Ada(T)				93.04 ±0.09				93.04 ±0.09
Lang-Spec FT + Ada(F)				88.79 ±0.13				88.79 ±0.13
Inc Joint	**-0.01** ±0.01	0.15 ±0.02	**50.32** ±0.03	93.91 ±0.01	0.12 ±0.01	**0.63** ±0.01	45.87 ±0.03	**94.30** ±0.01
Multilingual				94.25 ±0.07				94.25 ±0.07
Model Expansion Baselines								
Lang-Spec Trans	**0.40** ±0.01	**0.59** ±0.02	**-0.48** ±0.00	**93.86** ±0.01	0.62 ±0.02	0.03 ±0.01	-0.54 ±0.00	93.37 ±0.01
Lang-Spec Enc[1-9]	**0.60** ±0.01	**1.00** ±0.01	22.02 ±0.02	**93.75** ±0.02	1.05 ±0.02	0.63 ±0.01	**22.50** ±0.01	93.15 ±0.01
Lang-Spec Task	**1.53** ±0.02	**0.84** ±0.01	**0.17** ±0.00	**91.93** ±0.01	5.53 ±0.04	0.38 ±0.02	-0.11 ±0.00	87.68 ±0.02
Lang-Spec Ada(T)	**1.18** ±0.01	**1.29** ±0.01	**50.25** ±0.03	**92.36** ±0.02	4.43 ±0.04	0.79 ±0.02	42.35 ±0.02	88.66 ±0.02
Lang-Spec Ada(F)	**0.84** ±0.02	**3.41** ±0.02	3.80 ±0.00	**91.08** ±0.02	1.87 ±0.05	2.43 ±0.02	**9.68** ±0.01	89.92 ±0.02
Other Continuous Learning Algorithms								
EWC	**1.82** ±0.02	**0.74** ±0.01	**51.13** ±0.03	**91.16** ±0.02	5.9 ±0.04	0.48 ±0.02	44.73 ±0.03	88.28 ±0.02
ER	**0.71** ±0.01	**0.95** ±0.02	**49.59** ±0.03	**93.51** ±0.01	2.35 ±0.03	0.78 ±0.01	44.87 ±0.03	92.58 ±0.02
KD-Logit	**1.42** ±0.01	**0.77** ±0.02	**50.79** ±0.03	**91.60** ±0.02	4.07 ±0.04	0.51 ±0.01	44.38 ±0.03	89.65 ±0.02
KD-Rep	**1.49** ±0.01	**0.96** ±0.01	**51.17** ±0.03	**91.64** ±0.02	4.00 ±0.04	0.53 ±0.01	45.11 ±0.02	90.17 ±0.02
	Test Slot Filling On							
Shared {Trans, Task} Baselines								
Naive Seq FT	**4.15** ±0.18	**0.77** ±0.20	**37.03** ±0.05	67.80 ±0.13	7.06 ±0.23	0.77 ±0.17	33.29 ±0.03	**68.37** ±0.13
Lang-Spec FT				73.90 ±0.83				73.90 ±0.83
Lang-Spec FT + Ada(T)				72.90 ±0.80				72.90 ±0.80
Lang-Spec FT + Ada(F)				67.46 ±0.89				67.46 ±0.89
Inc Joint	0.78 ±0.11	**0.69** ±0.16	**37.92** ±0.05	**75.14** ±0.13	**0.37** ±0.14	-0.47 ±0.19	32.75 ±0.03	75.14 ±0.14
Multilingual				76.34 ±0.82				76.34 ±0.82
Model Expansion Baselines								
Lang-Spec Trans	**0.99** ±0.11	**0.92** ±0.18	0.33 ±0.00	**74.88** ±0.13	1.23 ±0.14	0.89 ±0.17	**0.39** ±0.00	74.85 ±0.14
Lang-Spec Enc[1-9]	2.35 ±0.15	**1.79** ±0.18	10.57 ±0.01	72.51 ±0.13	**2.03** ±0.15	0.74 ±0.19	**12.63** ±0.01	**74.01** ±0.14
Lang-Spec Task	**4.08** ±0.17	**1.91** ±0.16	**0.06** ±0.00	**68.88** ±0.15	7.23 ±0.24	-0.67 ±0.19	**0.06** ±0.00	66.28 ±0.13
Lang-Spec Ada(T)	**2.46** ±0.14	**2.75** ±0.16	**35.05** ±0.05	**71.79** ±0.15	6.42 ±0.23	0.40 ±0.17	29.89 ±0.03	67.70 ±0.12
Lang-Spec Ada(F)	**2.57** ±0.20	**4.77** ±0.17	**3.34** ±0.00	**70.33** ±0.15	5.01 ±0.24	2.70 ±0.20	2.59 ±0.00	67.07 ±0.12
Other Continuous Learning Algorithms								
EWC	**4.22** ±0.20	**1.19** ±0.17	**37.39** ±0.05	**68.33** ±0.13	7.53 ±0.25	0.52 ±0.16	33.25 ±0.03	66.91 ±0.14
ER	**2.32** ±0.15	**1.83** ±0.16	**37.50** ±0.05	**73.31** ±0.14	3.48 ±0.20	0.44 ±0.19	32.97 ±0.04	72.00 ±0.15
KD-Logit	**4.42** ±0.18	**1.79** ±0.15	**37.50** ±0.05	**68.13** ±0.14	7.36 ±0.27	0.13 ±0.19	32.86 ±0.04	67.13 ±0.14
KD-Rep	**4.56** ±0.18	**1.61** ±0.15	**37.42** ±0.05	68.28 ±0.13	6.65 ±0.28	1.03 ±0.17	32.57 ±0.03	**69.03** ±0.13

Table 10: Per language permutation view: a pairwise comparison between *H2L* (English → German → French → Hindi → Spanish → Thai) and *L2H* (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.

for each language. Although we notice that forgetting is the lowest and the final performance is the highest when a memory of 6000 instances is used, the gap is not that significant as the memory is scaled down. Moreover, differences in transfer are not correlated with the size of the memory. We notice that ER achieves a performance that surpasses *Naive Seq FT* even when using the lowest memory size. This suggests that even tiny bits of memory are helpful.

E More Results using Multiple Seeds

In this section, we show the results using different seeds for key experiments in the main paper. We show in Table 18 and 19 the average final performance, forgetting, and transfer averaged across different language permutations for the baseline model compared to reference models. We also show in Table 20 the performance on intent classification comparison between the baseline and different continual learning algorithm across *H2L* and *L2H*. Overall, we notice the same trends and findings observed earlier in Tables 2, 3, and 4.

F Statistical Significance

We show in Figures 10 and 11 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 10a, 11a, 10c, 10b, 11a, 10d, 10e,

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	**2.62** ±0.03	**0.59** ±0.01	52.07 ±0.03	**91.49** ±0.02	2.63 ±0.03	0.52 ±0.02	**55.0** ±0.02	90.74 ±0.02
Lang-Spec FT				93.40 ±0.08				93.40 ±0.08
Lang-Spec FT + Ada(T)				93.04 ±0.09				93.04 ±0.09
Lang-Spec FT + Ada(F)				88.79 ±0.13				88.79 ±0.13
Inc Joint	**0.11** ±0.01	0.47 ±0.01	**53.86** ±0.02	94.01 ±0.01	0.25 ±0.01	**0.61** ±0.01	50.51 ±0.02	**94.09** ±0.01
Multilingual				94.25 ±0.07				94.25 ±0.07
Model Expansion Baselines								
Lang-Spec Trans	**0.45** ±0.01	0.05 ±0.02	**-0.37** ±0.00	93.43 ±0.01	0.51 ±0.01	**0.39** ±0.02	-0.5 ±0.0	**93.63** ±0.01
Lang-Spec Enc[1-9]	**0.64** ±0.02	0.54 ±0.01	**26.32** ±0.02	**93.68** ±0.01	**0.81** ±0.02	0.82 ±0.02	25.26 ±0.02	93.59 ±0.01
Lang-Spec Task	**2.23** ±0.03	0.46 ±0.02	**0.47** ±0.00	**91.73** ±0.02	3.02 ±0.03	**0.85** ±0.02	-0.07 ±0.0	90.91 ±0.02
Lang-Spec Ada(T)	**1.36** ±0.02	**1.07** ±0.01	50.06 ±0.02	**92.70** ±0.02	2.33 ±0.03	0.78 ±0.02	**51.96** ±0.02	92.15 ±0.02
Lang-Spec Ada(F)	**0.82** ±0.02	2.61 ±0.02	5.68 ±0.01	**90.34** ±0.02	1.21 ±0.03	**2.75** ±0.02	**8.84** ±0.01	90.17 ±0.02
Other Continuous Learning Algorithms								
EWC	**2.55** ±0.02	**0.87** ±0.01	52.29 ±0.03	**92.04** ±0.02	2.57 ±0.03	0.71 ±0.02	**54.84** ±0.02	91.67 ±0.02
ER	**1.27** ±0.02	**0.70** ±0.02	54.29 ±0.02	**93.08** ±0.01	1.33 ±0.02	0.44 ±0.02	**55.05** ±0.03	93.05 ±0.02
KD-Logit	**2.16** ±0.02	**0.54** ±0.02	52.32 ±0.02	**92.23** ±0.02	2.18 ±0.03	0.45 ±0.02	**53.73** ±0.03	91.84 ±0.02
KD-Rep	**2.04** ±0.03	0.36 ±0.02	52.06 ±0.03	**92.25** ±0.02	2.13 ±0.03	**0.65** ±0.01	**53.55** ±0.03	92.06 ±0.02
	Test Slot Filling On							
Shared {Trans, Task} Baselines								
Naive Seq FT	**5.40** ±0.25	**1.95** ±0.17	36.2 ±0.04	**70.61** ±0.14	5.5 ±0.19	1.81 ±0.16	**38.41** ±0.04	70.30 ±0.15
Lang-Spec FT				73.90 ±0.83				73.90 ±0.83
Lang-Spec FT + Ada(T)				72.90 ±0.80				72.90 ±0.80
Lang-Spec FT + Ada(F)				67.46 ±0.89				67.46 ±0.89
Inc Joint	**0.81** ±0.14	1.57 ±0.16	37.46 ±0.05	74.9 ±0.16	1.03 ±0.15	**1.72** ±0.17	**37.54** ±0.04	**75.34** ±0.15
Multilingual				76.34 ±0.82				76.34 ±0.82
Model Expansion Baselines								
Lang-Spec Trans	1.57 ±0.18	**1.29** ±0.15	**0.49** ±0.00	74.56 ±0.13	**1.29** ±0.13	0.60 ±0.17	0.47 ±0.0	**74.57** ±0.15
Lang-Spec Enc[1-9]	1.80 ±0.19	**2.05** ±0.17	13.24 ±0.01	**75.2** ±0.16	**1.25** ±0.17	0.23 ±0.17	**13.57** ±0.01	74.67 ±0.14
Lang-Spec Task	4.94 ±0.24	**2.20** ±0.16	**0.11** ±0.00	**71.06** ±0.14	**4.77** ±0.22	1.14 ±0.18	0.05 ±0.0	70.63 ±0.14
Lang-Spec Ada(T)	**3.25** ±0.18	**3.26** ±0.16	34.88 ±0.04	**72.38** ±0.16	4.31 ±0.21	1.75 ±0.14	**35.48** ±0.03	70.39 ±0.13
Lang-Spec Ada(F)	**2.52** ±0.2	**4.03** ±0.18	3.10 ±0.0	68.22 ±0.14	3.06 ±0.2	**4.03** ±0.19	**3.57** ±0.0	**68.67** ±0.14
Other Continuous Learning Algorithms								
EWC	5.54 ±0.24	**1.99** ±0.16	36.34 ±0.04	**70.69** ±0.13	**5.46** ±0.23	1.07 ±0.18	**38.14** ±0.04	70.05 ±0.15
ER	3.01 ±0.18	**1.98** ±0.16	37.54 ±0.04	**72.92** ±0.13	**2.77** ±0.18	0.81 ±0.17	**38.66** ±0.04	72.82 ±0.14
KD-Logit	**5.00** ±0.25	**2.00** ±0.17	35.67 ±0.04	**71.82** ±0.14	5.46 ±0.23	1.52 ±0.17	**37.78** ±0.04	69.76 ±0.14
KD-Rep	**4.84** ±0.22	**1.46** ±0.17	35.96 ±0.04	**70.71** ±0.16	5.01 ±0.22	0.9 ±0.16	**37.25** ±0.04	70.37 ±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.

and 10f show confusion plots of statistical significance 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 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. Figures 12 and 13 show the corresponding statistical significance p-value confusion plots using multiple seeds. With a few exceptions like *Lang-Spec FT* + Ada(T) and *Lang-Spec FT* + Ada(F), most pairwise p-values which indicate statistical significance between two models using bootstrap sampling analysis are compliant with statistical significance computed using

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	2.97 ±0.03	**0.75** ±0.01	47.04 ±0.03	**91.63** ±0.02	**2.32** ±0.02	0.71 ±0.02	**51.97** ±0.03	**91.63** ±0.02
Lang-Spec FT				93.4 ±0.08				93.4 ±0.08
Lang-Spec FT + Ada(T)				93.04 ±0.09				93.04 ±0.09
Lang-Spec FT + Ada(F)				88.79 ±0.13				88.79 ±0.13
Inc Joint	0.21 ±0.01	**0.74** ±0.01	48.41 ±0.03	**94.44** ±0.01	**-0.02** ±0.01	0.54 ±0.02	**51.49** ±0.03	94.23 ±0.01
Multilingual				94.25 ±0.07				94.25 ±0.07
Model Expansion Baselines								
Lang-Spec Trans	**0.41** ±0.02	0.03 ±0.02	-0.57 ±0.0	**93.39** ±0.01	0.52 ±0.02	**0.29** ±0.02	**-0.11** ±0.0	93.38 ±0.01
Lang-Spec Enc[1-9]	0.80 ±0.02	0.74 ±0.01	23.18 ±0.02	93.35 ±0.01	**0.76** ±0.01	**1.05** ±0.01	**26.12** ±0.02	**93.46** ±0.01
Lang-Spec Task	2.84 ±0.03	0.67 ±0.01	-0.2 ±0.0	91.17 ±0.02	**2.32** ±0.02	**0.76** ±0.01	**0.36** ±0.0	**91.7** ±0.02
Lang-Spec Ada(T)	2.49 ±0.03	**1.05** ±0.01	47.67 ±0.03	**92.34** ±0.01	**1.35** ±0.02	0.89 ±0.01	**53.77** ±0.02	92.30 ±0.02
Lang-Spec Ada(F)	**1.13** ±0.03	**3.09** ±0.02	4.40 ±0.01	**90.50** ±0.02	1.32 ±0.02	2.64 ±0.02	**6.73** ±0.01	90.15 ±0.02
Other Continuous Learning Algorithms								
EWC	3.07 ±0.03	0.79 ±0.01	46.44 ±0.03	91.45 ±0.02	**2.51** ±0.02	**0.81** ±0.01	**51.54** ±0.02	**91.54** ±0.02
ER	1.11 ±0.02	0.72 ±0.01	48.23 ±0.03	93.00 ±0.02	**0.98** ±0.02	**0.92** ±0.01	**52.23** ±0.03	**93.32** ±0.01
KD-Logit	2.50 ±0.03	**0.86** ±0.01	47.96 ±0.03	91.27 ±0.02	**1.89** ±0.02	0.59 ±0.02	**51.88** ±0.03	**92.16** ±0.02
KD-Rep	2.24 ±0.03	0.81 ±0.01	48.08 ±0.03	91.89 ±0.02	**1.86** ±0.02	**0.83** ±0.02	**52.51** ±0.03	**92.16** ±0.02
	Test Slot Filling On							
Shared {Trans, Task} Baselines								
Naive Seq FT	6.51 ±0.22	**1.90** ±0.15	34.53 ±0.04	68.93 ±0.13	**5.38** ±0.25	1.00 ±0.18	**38.47** ±0.05	**70.22** ±0.14
Lang-Spec FT				73.9 ±0.83				73.9 ±0.83
Lang-Spec FT + Ada(T)				72.9 ±0.8				72.9 ±0.8
Lang-Spec FT + Ada(F)				67.46 ±0.89				67.46 ±0.89
Inc Joint	**0.99** ±0.15	**1.21** ±0.15	32.99 ±0.03	**74.45** ±0.16	1.52 ±0.15	0.27 ±0.18	**39.69** ±0.05	74.31 ±0.14
Multilingual				76.34 ±0.82				76.34 ±0.82
Model Expansion Baselines								
Lang-Spec Trans	1.65 ±0.17	**1.04** ±0.17	0.37 ±0.00	74.51 ±0.14	**1.17** ±0.14	0.97 ±0.16	**0.47** ±0.00	**75.04** ±0.14
Lang-Spec Enc[1-9]	**1.48** ±0.13	**2.18** ±0.17	10.66 ±0.01	**75.03** ±0.14	2.77 ±0.18	1.67 ±0.18	**13.25** ±0.01	73.73 ±0.14
Lang-Spec Task	5.72 ±0.21	**2.4** ±0.17	**0.06** ±0.00	**70.08** ±0.13	**4.80** ±0.24	-0.04 ±0.18	**0.06** ±0.00	69.54 ±0.13
Lang-Spec Ada(T)	4.96 ±0.25	**2.39** ±0.15	29.17 ±0.03	**72.28** ±0.13	**3.98** ±0.21	1.69 ±0.16	**37.14** ±0.05	72.27 ±0.13
Lang-Spec Ada(F)	**3.15** ±0.21	**4.51** ±0.18	1.90 ±0.00	**69.47** ±0.14	3.77 ±0.22	3.54 ±0.16	**2.31** ±0.00	67.57 ±0.14
Other Continuous Learning Algorithms								
EWC	6.38 ±0.23	**2.54** ±0.17	34.29 ±0.04	**71.25** ±0.14	**5.56** ±0.27	1.46 ±0.17	**38.44** ±0.05	70.57 ±0.16
ER	4.12 ±0.22	**2.90** ±0.16	35.45 ±0.04	73.39 ±0.14	**2.65** ±0.18	0.83 ±0.17	**39.34** ±0.05	**73.56** ±0.15
KD-Logit	6.03 ±0.27	**2.02** ±0.16	35.2 ±0.04	**70.70** ±0.14	**4.91** ±0.21	0.92 ±0.17	**38.49** ±0.05	70.31 ±0.14
KD-Rep	5.72 ±0.27	**2.6** ±0.15	35.54 ±0.04	**71.61** ±0.15	**5.35** ±0.21	0.97 ±0.15	**38.8** ±0.05	70.15 ±0.13

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.

multiple seeds.

Model	German	English	French	Spanish	Hindi	Thai
			Test Intent Accuracy On			
Shared {Trans, Task} Baselines						
Naive Seq FT	1.52 ±0.12	1.06 ±0.08	1.30 ±0.14	1.49 ±0.13	2.90 ±0.38	5.51 ±1.35
Inc Joint	**0.31** ±0.05	**0.12** ±0.04	**0.19** ±0.05	**0.15** ±0.04	**0.04** ±0.07	**0.28** ±0.08
Model Expansion Baselines						
Lang-Spec Trans	0.36 ±0.06	0.33 ±0.04	0.44 ±0.07	0.34 ±0.06	0.42 ±0.08	0.46 ±0.08
Lang-Spec Enc[1-9]	0.54 ±0.07	0.45 ±0.05	0.51 ±0.08	0.59 ±0.06	0.66 ±0.10	0.90 ±0.15
Lang-Spec Task	1.22 ±0.12	0.95 ±0.09	1.49 ±0.14	1.37 ±0.12	3.20 ±0.40	5.44 ±1.67
Lang-Spec Ada(T)	0.88 ±0.08	0.81 ±0.08	1.16 ±0.12	1.00 ±0.09	1.85 ±0.24	4.23 ±1.15
Lang-Spec Ada(F)	0.58 ±0.08	0.61 ±0.08	0.81 ±0.11	0.54 ±0.10	0.86 ±0.11	1.88 ±0.33
Other Continuous Learning Algorithms						
EWC	1.40 ±0.15	1.00 ±0.08	1.74 ±0.15	1.56 ±0.13	3.26 ±0.37	5.62 ±1.75
ER	0.76 ±0.07	0.53 ±0.05	0.87 ±0.08	0.71 ±0.08	1.13 ±0.12	2.19 ±0.22
KD-Logit	1.23 ±0.12	0.97 ±0.08	1.47 ±0.12	1.27 ±0.12	2.19 ±0.27	4.41 ±0.75
KD-Rep	1.20 ±0.11	0.80 ±0.07	1.45 ±0.11	1.42 ±0.12	2.29 ±0.27	4.02 ±0.63
			Test Slot Filling On			
	German	English	French	Spanish	Hindi	Thai
Shared {Trans, Task} Baselines						
Naive Seq FT	3.64 ±1.31	3.91 ±1.14	2.80 ±0.94	2.94 ±0.94	6.48 ±1.85	8.85 ±3.19
Inc Joint	1.21 ±0.85	1.12 ±0.70	**0.64** ±0.71	**0.96** ±0.62	**1.13** ±0.70	**0.77** ±0.57
Model Expansion Baselines						
Lang-Spec Trans	**0.90** ±0.71	**1.02** ±0.62	1.03 ±0.65	1.21 ±0.74	1.28 ±0.75	1.06 ±0.64
Lang-Spec Enc[1-9]	2.03 ±0.93	1.83 ±0.81	1.03 ±0.77	1.31 ±0.69	1.76 ±0.81	2.00 ±0.76
Lang-Spec Task	3.32 ±1.29	2.96 ±0.97	2.74 ±0.93	2.76 ±0.89	6.89 ±2.01	8.17 ±3.05
Lang-Spec Ada(T)	2.96 ±1.12	3.05 ±0.88	1.49 ±0.76	1.52 ±0.82	4.34 ±1.17	6.84 ±2.26
Lang-Spec Ada(F)	1.82 ±0.97	1.85 ±0.88	1.33 ±0.83	1.89 ±0.96	2.72 ±0.99	5.81 ±1.98
Other Continuous Learning Algorithms						
EWC	3.41 ±1.25	3.90 ±1.24	3.08 ±0.95	3.32 ±0.96	6.29 ±1.86	8.74 ±3.22
ER	1.94 ±0.82	2.01 ±0.96	1.60 ±0.76	1.82 ±0.80	3.65 ±1.04	4.73 ±1.18
KD-Logit	3.69 ±1.31	3.70 ±1.03	3.10 ±1.01	3.55 ±1.11	5.66 ±1.68	8.05 ±2.68
KD-Rep	3.49 ±1.18	3.85 ±1.09	3.13 ±0.95	2.99 ±0.92	5.81 ±1.66	7.93 ±2.18

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

Model	German	English	French	Hindi	Spanish	Thai
			Test Intent Accuracy On			
Shared {Trans, Task} Baselines						
Naive Seq FT	0.37 ±0.07	0.30 ±0.06	0.77 ±0.08	1.14 ±0.07	0.64 ±0.09	0.85 ±0.11
Inc Joint	0.25 ±0.07	0.04 ±0.06	0.74 ±0.09	1.25 ±0.06	0.27 ±0.12	0.57 ±0.11
Model Expansion Baselines						
Lang-Spec Trans	-0.36 ±0.08	-0.07 ±0.06	0.29 ±0.10	0.93 ±0.08	0.12 ±0.10	0.47 ±0.11
Lang-Spec Enc[1-9]	0.39 ±0.07	0.28 ±0.05	0.96 ±0.08	1.09 ±0.07	0.80 ±0.11	1.25 ±0.10
Lang-Spec Task	0.22 ±0.07	0.12 ±0.06	0.99 ±0.08	1.11 ±0.07	0.69 ±0.10	0.84 ±0.09
Lang-Spec Ada(T)	1.38 ±0.07	0.41 ±0.06	1.30 ±0.09	0.93 ±0.11	1.20 ±0.09	0.65 ±0.10
Lang-Spec Ada(F)	**2.47** ±0.10	**1.43** ±0.08	**3.03** ±0.11	**3.17** ±0.11	**2.00** ±0.15	**4.84** ±0.33
Other Continuous Learning Algorithms						
EWC	0.26 ±0.08	0.12 ±0.05	1.13 ±0.07	1.10 ±0.07	0.85 ±0.09	0.92 ±0.11
ER	0.27 ±0.08	0.07 ±0.06	1.01 ±0.08	1.16 ±0.07	0.96 ±0.10	1.04 ±0.11
KD-Logit	0.16 ±0.08	0.13 ±0.06	0.96 ±0.09	0.96 ±0.07	0.68 ±0.10	0.82 ±0.11
KD-Rep	0.12 ±0.08	0.09 ±0.06	0.82 ±0.09	1.30 ±0.07	0.74 ±0.10	1.06 ±0.10
	German	English	French	Spanish	Hindi	Thai
			Test Slot Filling On			
Shared {Trans, Task} Baselines						
Naive Seq FT	1.71 ±0.98	1.24 ±0.71	2.01 ±0.90	0.54 ±0.97	0.20 ±0.91	2.50 ±0.75
Inc Joint	1.59 ±0.90	-0.17 ±0.89	1.22 ±0.84	1.08 ±0.94	-1.10 ±1.04	2.36 ±0.79
Model Expansion Baselines						
Lang-Spec Trans	1.75 ±0.95	1.37 ±0.80	1.85 ±0.83	-0.25 ±0.91	-0.67 ±0.93	1.67 ±0.74
Lang-Spec Enc[1-9]	1.80 ±0.92	0.45 ±1.05	2.11 ±0.86	0.67 ±0.98	0.51 ±0.88	3.12 ±0.88
Lang-Spec Task	2.28 ±1.07	-0.27 ±0.86	1.55 ±1.07	0.56 ±1.26	0.44 ±0.94	2.36 ±0.86
Lang-Spec Ada(T)	3.24 ±0.94	-0.54 ±0.72	1.04 ±0.95	1.59 ±0.94	**3.37** ±0.98	3.53 ±0.82
Lang-Spec Ada(F)	**3.48** ±1.00	**3.38** ±0.87	1.46 ±1.00	**4.68** ±1.04	2.11 ±1.06	**8.48** ±1.27
Other Continuous Learning Algorithms						
EWC	1.58 ±1.02	0.39 ±0.82	2.11 ±0.87	1.58 ±1.05	-0.09 ±0.93	3.19 ±0.73
ER	1.97 ±0.93	0.29 ±0.89	2.05 ±0.94	1.38 ±1.04	0.23 ±0.87	2.87 ±0.93
KD-Logit	2.20 ±0.98	0.50 ±0.83	2.00 ±0.84	1.35 ±1.00	-0.64 ±0.94	2.97 ±0.76
KD-Rep	1.90 ±0.88	0.90 ±0.75	**2.54** ±0.88	1.01 ±0.91	-0.23 ±0.96	2.45 ±0.75

Table 15: CCL 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	58.53 ±1.49	69.09 ±12.56	60.83 ±3.24	59.42 ±24.92	33.38 ±1.35	20.17 ±1.10
Inc Joint	58.48 ±2.13	70.13 ±12.56	61.17 ±2.62	61.18 ±19.86	32.28 ±2.56	17.20 ±0.19
Model Expansion Baselines						
Lang-Spec Trans	-1.42 ±0.00	0.44 ±0.01	-0.01 ±0.01	-0.95 ±0.01	-0.15 ±0.00	-0.47 ±0.00
Lang-Spec Enc[1-9]	26.17 ±7.44	33.16 ±10.88	25.56 ±7.00	27.21 ±18.32	21.79 ±2.33	11.51 ±0.77
Lang-Spec Task	-0.25 ±0.12	0.38 ±0.01	0.63 ±0.06	-0.66 ±0.02	0.60 ±0.03	-0.09 ±0.01
Lang-Spec Ada(T)	55.95 ±0.91	67.93 ±14.89	60.21 ±4.16	58.14 ±33.89	**36.44** ±4.20	17.40 ±1.10
Lang-Spec Ada(F)	5.08 ±0.51	14.37 ±1.06	7.61 ±0.49	6.87 ±1.00	5.50 ±0.90	-0.30 ±0.04
Other Continuous Learning Algorithms						
EWC	58.57 ±1.77	69.39 ±12.59	60.71 ±3.48	58.99 ±24.22	33.59 ±1.40	19.71 ±1.30
ER	**59.70** ±1.68	**70.20** ±13.83	**61.32** ±4.05	**60.09** ±24.40	33.38 ±1.24	19.57 ±1.48
KD-Logit	58.12 ±1.32	68.87 ±12.38	60.85 ±3.45	59.69 ±24.27	33.55 ±1.46	19.99 ±1.19
KD-Rep	58.47 ±1.20	68.64 ±12.23	60.96 ±3.56	59.69 ±24.54	34.22 ±1.07	**20.49** ±1.00
	Test Slot Filling On German	English	French	Spanish	Hindi	Thai
Shared {Trans, Task} Baselines						
Naive Seq FT	44.25 ±1.16	48.42 ±8.10	47.58 ±1.63	46.60 ±15.31	18.97 ±0.44	12.09 ±0.33
Inc Joint	**44.73** ±1.68	48.74 ±10.90	47.67 ±2.19	46.98 ±18.10	18.05 ±0.31	12.20 ±0.22
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[1-9]	14.81 ±3.81	15.50 ±6.12	16.09 ±4.03	16.11 ±8.84	6.62 ±1.29	4.80 ±0.35
Lang-Spec Task	0.07 ±0.00	0.15 ±0.00	0.08 ±0.00	0.04 ±0.00	-0.02 ±0.00	0.09 ±0.00
Lang-Spec Ada(T)	41.08 ±1.24	44.36 ±18.19	45.26 ±2.44	42.56 ±21.09	17.62 ±1.27	10.72 ±0.13
Lang-Spec Ada(F)	4.42 ±0.10	1.12 ±0.04	4.51 ±0.32	4.86 ±0.93	1.80 ±0.03	0.09 ±0.00
Other Continuous Learning Algorithms						
EWC	44.17 ±1.16	48.52 ±8.21	47.51 ±1.62	46.38 ±15.32	18.94 ±0.42	12.32 ±0.30
ER	**44.73** ±1.45	**49.60** ±9.35	**48.17** ±2.22	**47.26** ±15.85	**19.06** ±0.44	12.62 ±0.24
KD-Logit	43.79 ±1.04	48.30 ±8.21	47.31 ±2.05	46.77 ±15.51	18.85 ±0.37	12.49 ±0.22
KD-Rep	43.81 ±1.35	48.10 ±7.99	47.38 ±1.85	46.60 ±15.21	18.83 ±0.45	**12.82** ±0.26

Table 16: CCL 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.

Model	F ↓ Acc	F1	T ↑ Acc	F1	T^0 ↑ Acc	F1	FP ↑ Acc	F1
Naive Seq FT	2.93 ±1.24	5.67 ±0.93	0.68 ±0.14	1.37 ±0.53	50.24 ±3.43	36.32 ±1.91	91.06 ±1.08	69.37 ±1.06
ER-750	1.97 ±0.73	4.28 ±0.63	0.65 ±0.19	1.46 ±0.59	50.41 ±3.19	36.53 ±1.91	92.10 ±0.68	71.65 ±1.02
ER-1500	1.55 ±0.44	3.88 ±0.42	0.68 ±0.26	1.55 ±0.69	50.83 ±3.38	36.59 ±1.93	92.65 ±0.35	71.68 ±0.71
ER-3000	1.40 ±0.44	3.36 ±0.47	0.70 ±0.25	**1.48** ±0.71	**51.03** ±3.60	36.77 ±2.06	92.93 ±0.37	72.71 ±0.56
ER-4500	1.43 ±0.58	3.39 ±0.75	0.59 ±0.11	1.44 ±0.38	50.46 ±3.68	36.91 ±2.19	92.73 ±0.72	72.46 ±1.05
ER-6000	**1.29** ±0.51	**3.06** ±0.59	**0.75** ±0.17	1.47 ±0.85	50.71 ±3.55	**36.91** ±2.14	**93.09** ±0.29	**73.00** ±0.52

Table 17: Ablation Studies of Experience Replay where we experiment with different memory sizes per language. For each metric and score, we highlight the best score in **bold** and underline the second best score.

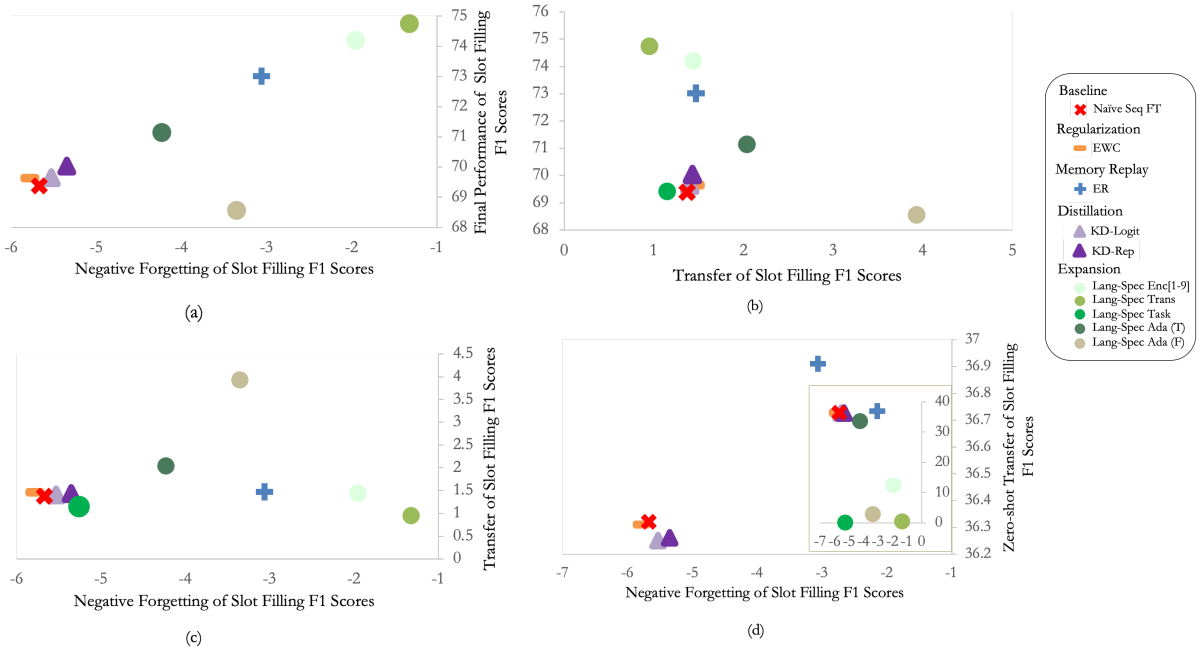
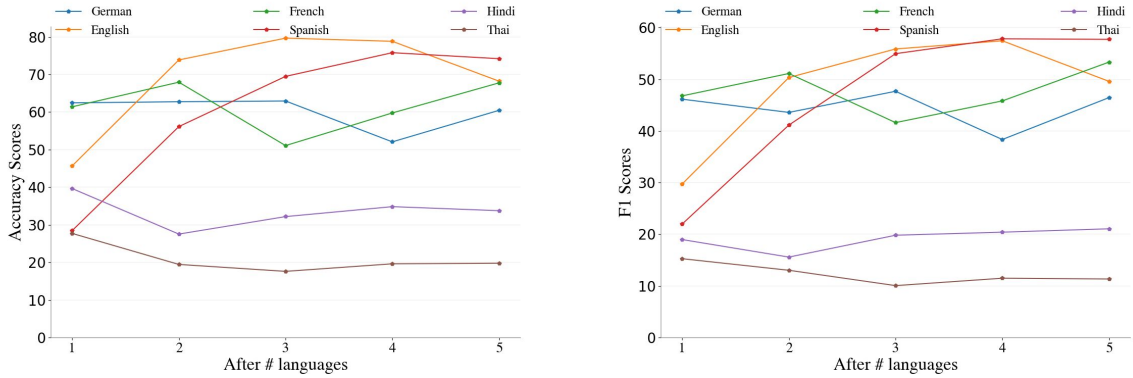


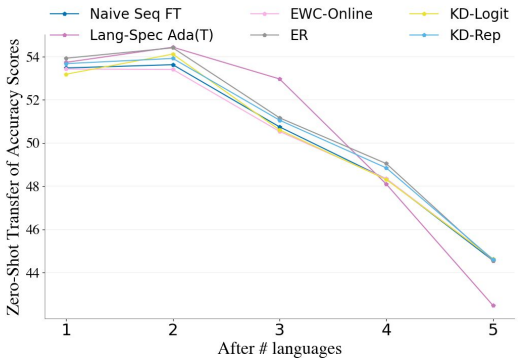
Figure 6: Correlations between different pairs of metrics: (a) Final performance versus negative forgetting for the task of slot filling. The lower the forgetting the higher the final performance. (b) Final performance versus transfer for the task of slot filling. (c) Transfer versus negative forgetting for slot filling task. (d) Zero-shot generalization versus negative forgetting for slot filling. Model expansion approaches are highlighted in shades of green. We zoom over the rest of the models in the main graph and show an overview of all approaches in the lower right corner subplot. The same trends observed for intent classification in Figure 4 can be observed here.



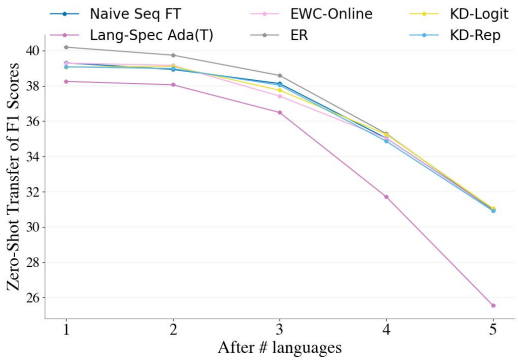
(a) Accuracy for intent classification.

(b) F1 score for slot filling.

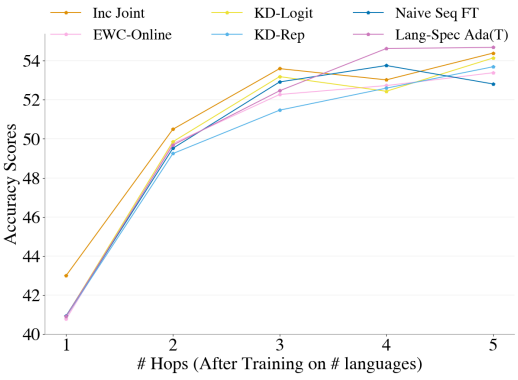
Figure 7: Comparing cross-lingual generalization of *Naive Seq FT* across many hops and different languages for intent classification and slot filling.



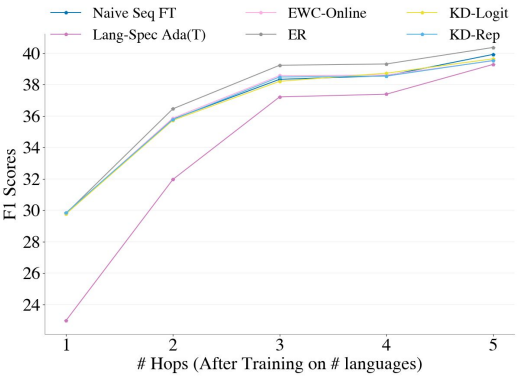
(a) Zero-shot transfer of accuracy for intent classification.



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

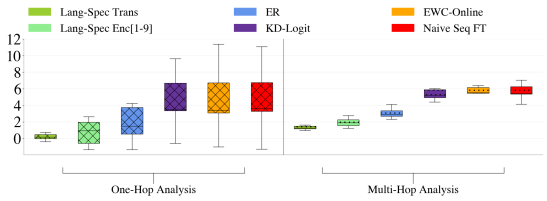


(c) Accuracy for intent classification.

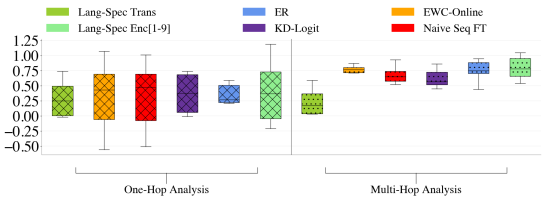


(d) F1 score for slot filling.

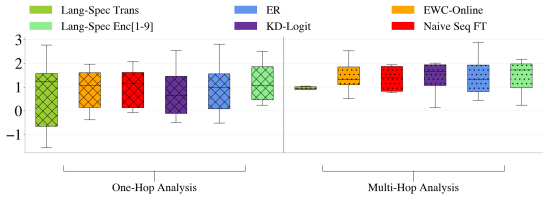
Figure 8: Measuring cross-lingual generalization to new languages across many hops for intent classification and slot filling. This is both in terms of zero-shot transfer metric and plain accuracy and f1 scores.



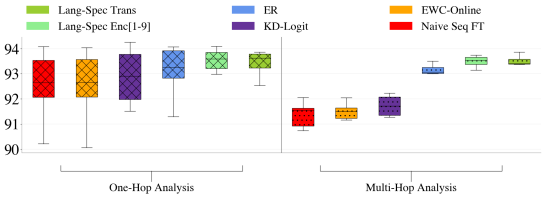
(a) Forgetting for slot filling.



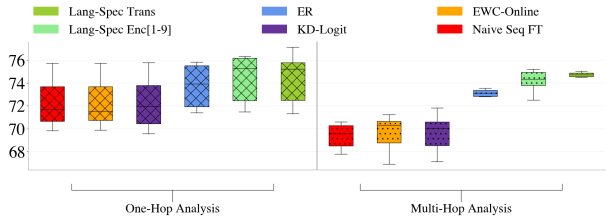
(b) Transfer for intent classification.



(c) Transfer for slot filling.

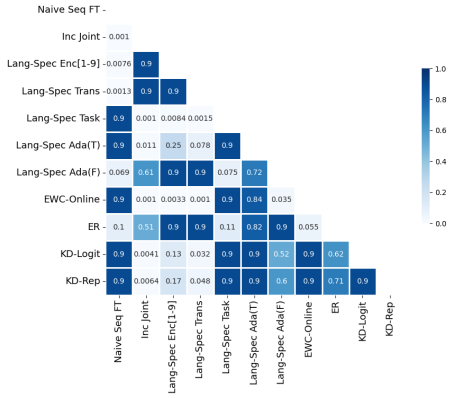


(d) Final performance for intent classification.

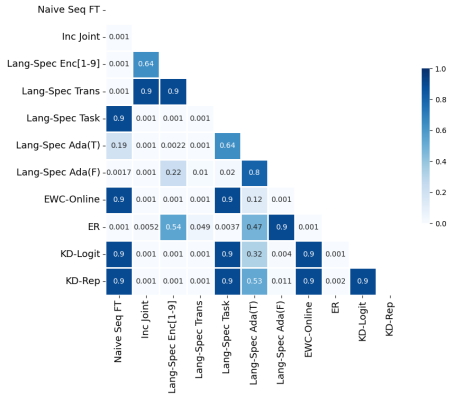


(e) Final performance for slot filling.

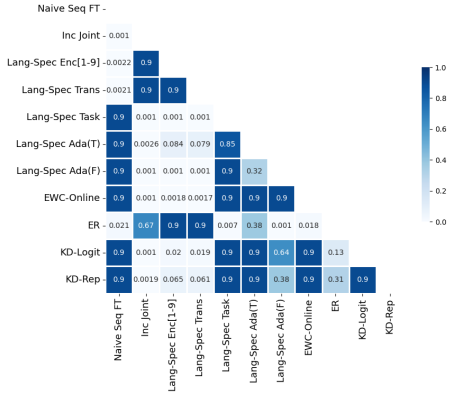
Figure 9: Comparison between different metrics using one-hop (crossed boxplots) and multi-hop analysis (dotted boxplots), on the left and right respectively for each approach.



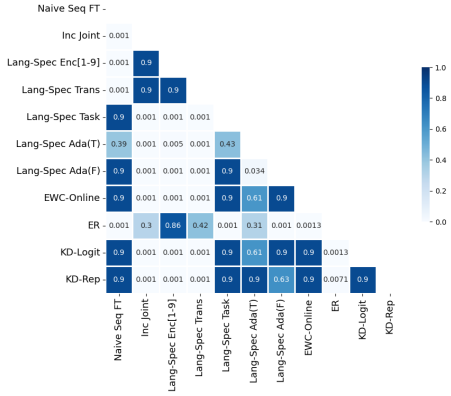
(a) Forgetting of intent accuracy.



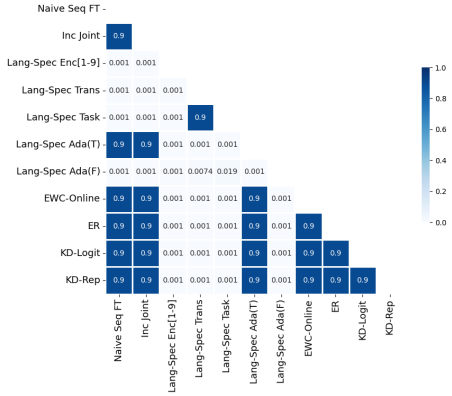
(b) Forgetting of slot filling.



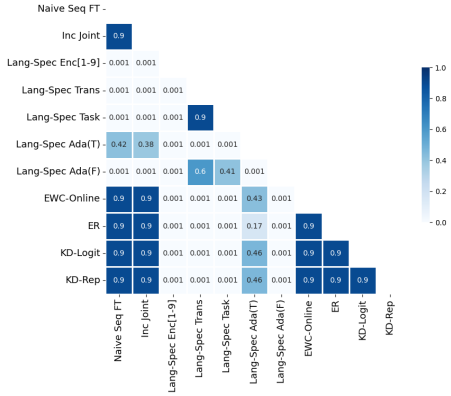
(c) Final performance of intent accuracy.



(d) Final performance of slot filling.

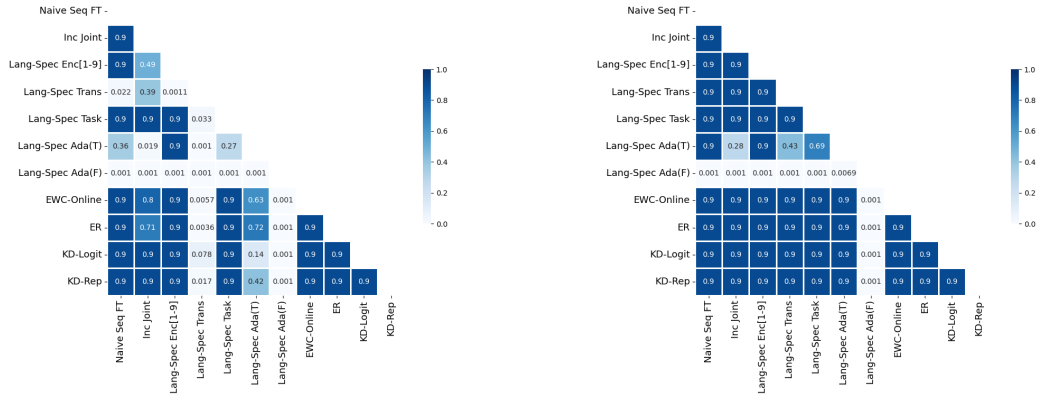


(e) Zero-shot transfer of intent accuracy.



(f) Zero-shot transfer of slot filling.

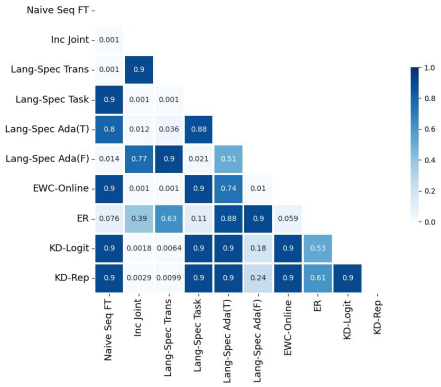
Figure 10: P-values for different pairwise comparison of different continual learning approaches using Tukey's honestly significant difference (HSD) test using bootstrap sampling.



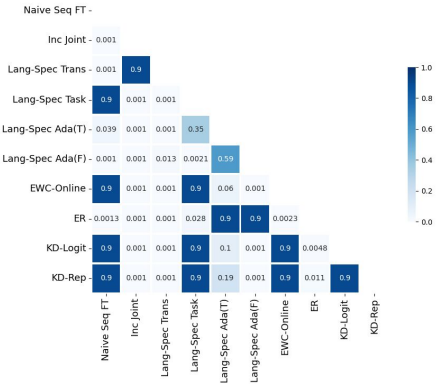
(a) Transfer of intent accuracy.

(b) Transfer of slot filling.

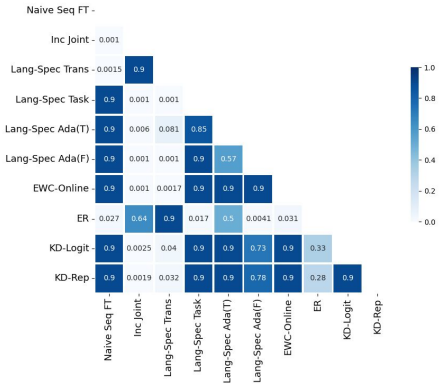
Figure 11: P-values for different pairwise comparison of different continual learning approaches using Tukey's honestly significant difference (HSD) test using bootstrap sampling (Cont.).



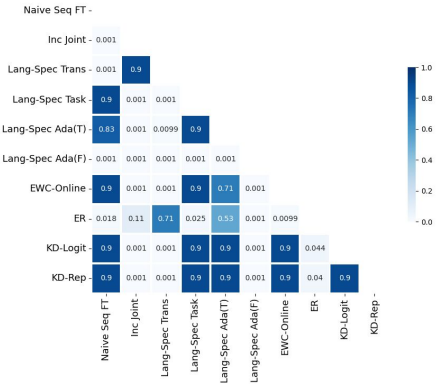
(a) Forgetting of intent accuracy.



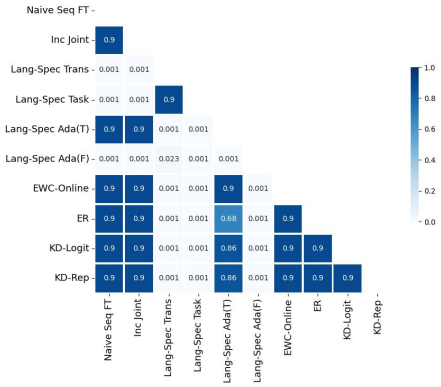
(b) Forgetting of slot filling.



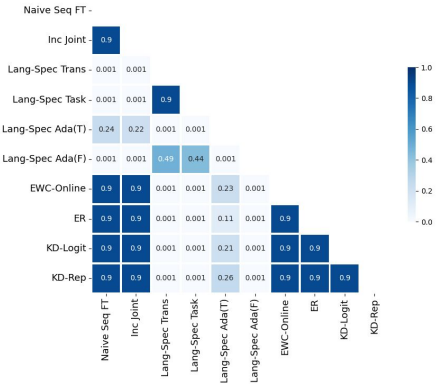
(c) Final performance of intent accuracy.



(d) Final performance of slot filling.

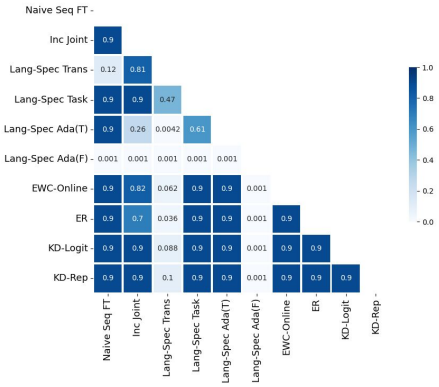


(e) Zero-shot transfer of intent accuracy.

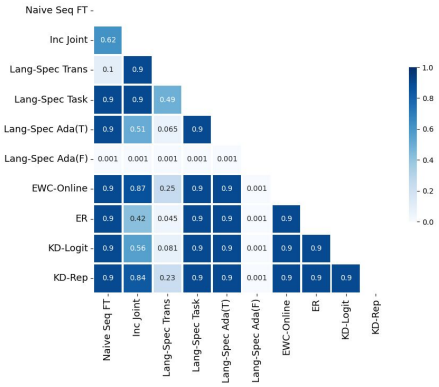


(f) Zero-shot transfer of slot filling.

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



(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 using different seeds (Cont.).

Model	Acc	F1
Naive Seq FT	90.40 ±1.53	65.01 ±1.25
Lang-Spec FT	93.28 ±0.31	68.93 ±1.17
Inc Joint	<u>94.14 ±0.08</u>	<u>71.70 ±0.43</u>
Multilingual	**94.20** ±0.21	**72.23** ±0.99

Table 18: The average final performance across different language permutations for the baseline compared to reference models using multiple seeds. We highlight the best scores in **bold** and <u>underline</u> the second best across models. We notice the same findings as when using bootstrap sampling but with tighter confidence intervals as shown in Table 2.

Model	F ↓		T ↑	
	Acc	F1	Acc	F1
Naive Seq FT	3.2 ±1.66	5.47 ±0.87	**0.73** ±0.16	**2.75** ±0.63
Inc Joint	**-0.1** ±0.01	**-0.38** ±0.45	0.57 ±0.14	1.73 ±1.05

Table 19: Forgetting (F) and transfer (T) performance averaged across different language permutations for *sequential baseline and reference models* using different seeds. We highlight the best models in **bold**. We notice exactly the same trends as when using bootstrap sampling for our analysis in Table 3.

Model	F ↓		T ↑		FP ↑	
	H2L	*L2H*	*H2L*	*L2H*	*H2L*	*L2H*
Naive Seq FT	**1.37** ±0.14	5.38 ±0.34	**0.95** ±0.03	0.56 ±0.07	**91.83** ±0.55	88.28 ±0.55
Lang-Spec Trans	**0.01** ±0.01	0.17 ±0.08	**0.57** ±0.06	0.09 ±0.01	**93.81** ±0.06	93.27 ±0.10
Lang-Spec Task	**1.29** ±0.08	5.52 ±0.87	**0.88** ±0.12	0.43 ±0.19	**92.12** ±0.18	87.20 ±1.76
Lang-Spec Ada(T)	**0.81** ±0.08	4.17 ±0.30	**1.16** ±0.09	0.65 ±0.06	**92.53** ±0.22	88.61 ±0.44
Lang-Spec Ada(F)	**0.38** ±0.09	1.04 ±0.61	**3.54** ±0.15	2.34 ±0.11	**91.15** ±0.04	90.0 ±0.39
EWC	**1.35** ±0.24	5.42 ±0.60	**0.87** ±0.11	0.71 ±0.12	**91.86** ±0.52	88.09 ±0.20
ER-6000	**0.69** ±0.14	1.93 ±0.28	**0.93** ±0.07	0.72 ±0.14	**93.43** ±0.08	92.50 ±0.25
KD-Logit	**1.33** ±0.11	3.82 ±0.23	**0.81** ±0.11	0.54 ±0.07	**91.86** ±0.31	89.85 ±0.4
KD-Rep	**1.37** ±0.1	3.7 ±0.25	**0.85** ±0.23	0.52 ±0.13	**91.64** ±0.49	89.73 ±0.8

Table 20: Performance on intent classification comparison between the baseline and continual learning algorithms across two language permutations using multiple seeds. We highlight in **bold** the lowest forgetting (F), highest transfer (T), and final performance (FP) of accuracy scores among *H2L* and *L2H*, whereas the best scores across approaches for *H2L* and *L2H* separately are underlined. We notice the same trends and findings as Table 4 where only bootstrap sampling is used to compute the confidence intervals.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
We discuss the limitations in the limitations section after the conclusion. There are no strong assumptions, claims or biases we are aware of that are not stated in the paper. We state only claims that are supported by evidence and experimental setup that we design and describe clearly in the main paper and in the supplemental material (appendix and code) (more details on the experimental setup can be found in Appendix C).

☐ A2. Did you discuss any potential risks of your work?
Not applicable. To the best of our knowledge, there is no potential risk or harm of any kind that could result from this work. Our research is just an analysis paper about cross-lingual continuous learning where we share our experiments on pre-existing approaches.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
The main claims and contributions are stated clearly in the abstract, introduction (section 1) especially the contribution paragraph, emphasized in the results and analysis section (section 4) and summarized briefly in the conclusion (section 6). The distinction between the main claims and future work is also clear, especially in section 4 where we discuss our claims based on the current experiments and provide speculations and hypotheses that can be verified in future work.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

We provide the github repository for the code we create in footnote 1. We provide citations to the dataset and pretrained models used in Sections 2.4 and Section 3.

☑ B1. Did you cite the creators of artifacts you used?
The dataset and pre-trained model used are all cited and the approaches used with their algorithms are also cited in Section 2.4 and Section 3. We write our own code for the experiments. Each time we use a specific tool we also cite or include its link in main text, footnotes, and appendices.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The license for the usage of the dataset is in section C.1 in the appendix. The dataset was released as open source from Facebook. The dataset is only used in our case, there is no distribution, repackaging etc.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. To the best of our knowledge, there is no author information or offensive content in the open source data that we used (not our own data). This is a dataset created using manual translation from English data where the creators of the data asked crowd-sourcers to generate natural language sentences for task-oriented dialogue with neutral domains about alarm, weather, etc.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Since this is a multilingual dataset, we provide the coverage of the datasets per language(the number of sentences for each language) in the original data in Table 1 in section 2.4.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Yes, we report the exact number of train/test/dev split per language in Table 1 in section 2.4.

C ☑ Did you run computational experiments?

Section 4 and appendices C-F.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We report all the information asked here in Section C.2 in the appendix. We report the number of parameters used in multilingual BERT and all their language-specific versions we create in Table 7 in the appendix. We also describe the total computational budget and the characteristics of the GPU used.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
We report all the hyperparameters used and the procedure used to pick them using the dev data in Section C.2.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
We report error bars and confidence intervals by using bootstrap sampling which we detail in section C.3 in the appendix. For boostrap sampling, we state clearly the number of runs used which is 1 run for each experiment given that we use bootstrap sampling which allows the computation of confidence intervals reliably. For multiple seeds, we use 3 seeds (Appendix E).

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
We use the released data which already comes preprocessed. We also cite any additional libraries used more details in the appendix sections B and C.2.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.