Leitner-Guided Memory Replay for Cross-lingual Continual Learning

Anonymous ACL submission

Abstract

Cross-lingual continual learning aims to continuously fine-tune a downstream model on emerging data from new languages. One major challenge in cross-lingual continual learning is catastrophic forgetting: a stability-plasticity dilemma, where performance on previously seen languages decreases as the model learns to transfer to new languages. Experience replay, which revisits data from a fixed-size memory of old languages while training on new ones, is among the most successful approaches for solving this dilemma. Faced by the challenge of dynamically storing the memory with high-quality examples while complying with its fixed size limitations, we consider Leitner queuing, a human-inspired spaced-repetition technique, to determine what should be replayed at each phase of learning. Via a controlled set of quantitative and qualitative analyses across different memory strategies, we show that, just like humans, carefully picking informative examples to be prioritized in cross-lingual memory replay helps tame the plasticity-stability dilemma. Compared to vanilla and strong memory replay baselines, our Leitner-guided approach significantly and consistently decreases forgetting while maintaining accuracy across natural language understanding tasks, language orders, and languages.

1 Introduction

Cross-lingual continual learning is a machine learning paradigm aimed at continually adapting a downstream model to datastreams drawn from different languages (M'hamdi et al., 2023). Naive approaches to cross-lingual continual learning involve training a new model from scratch each time a new language is available or training jointly over all languages which can be inefficient and even inaccessible. Faced with an overwhelming stream of languages, modelers turn to continual learning techniques to adapt models such that maximal learning from data is achieved when available data is temporally limited. The consequence of a finite data buffer limitation on a potentially infinite data source is *catastrophic forgetting* (McCloskey and Cohen, 1989). Catastrophic forgetting exemplifies the stability-plasticity dilemma (Carpenter and Grossberg, 1988; Hadsell et al., 2020; Wolczyk et al., 2021): It is inherently hard to preserve the previously acquired knowledge (stability) while learning novel information (plasticity).

Various continual learning approaches have proposed to mitigate catastrophic forgetting by either restricting entire sets of parameters from changing (Kirkpatrick et al., 2017; Zenke et al., 2017; Ritter et al., 2018), designing language-specific model components (Pfeiffer et al., 2020; M'hamdi et al., 2023), or replaying a fixed buffer memory from previously seen languages (Shin et al., 2017; Chaudhry et al., 2019a,b). M'hamdi et al. (2023) show that memory-based approaches are more robust than other approaches in taming the plasticity stability dilemma. Moreover, they are more scalable than other approaches such as model expansion, which grows in complexity as a function of the underlying downstream architecture.

Experience replay (ER) (Chaudhry et al., 2019b) is a cognitively inspired memory-based approach that reinforces previously seen experiences similar to the process of memory consolidation in biological systems (Isele and Cosgun, 2018). As more languages are incorporated into the datastream, fitting examples from new languages into a fixed-size memory buffer becomes more challenging. This invites a critical question: how to dynamically come up with informative memory examples to keep for each language?

In this paper, we propose a human-inspired approach for learning what to replay at each phase of cross-lingual continual learning. We hypothesize that in such a setup at the beginning, most data is difficult but as training progresses some data becomes well-learned and informative. We surmise

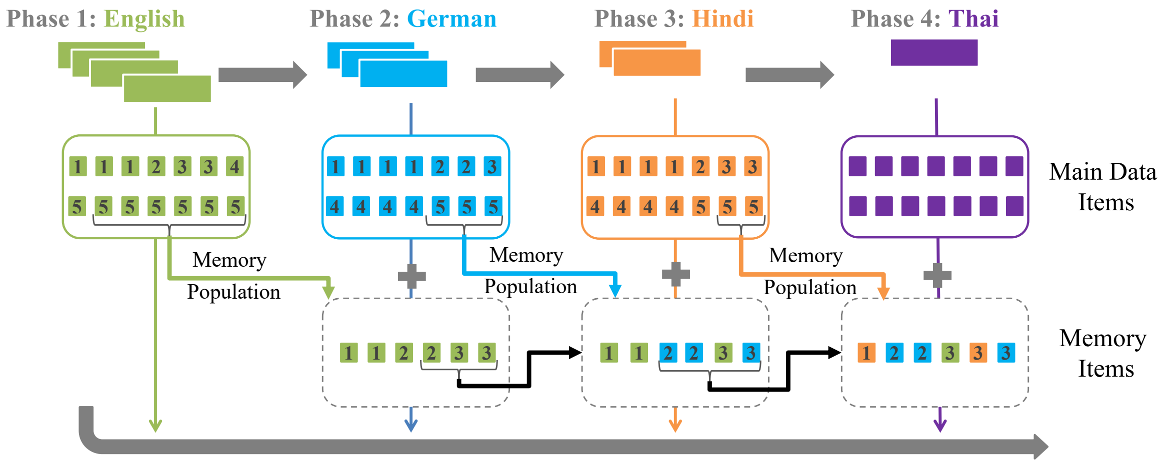


Figure 1: An overview of Leitner-guided memory replay for multi-phase cross-lingual continual learning: On top of a cross-lingual datastream, we build a skill rating system to continually guide the memory population and update. Skill ratings are scores from 1 to 5 obtained from Leitner queues; a higher score reflects greater learnability. At the end of each phase, the skill ratings on the main data items from the phase language are used to choose what goes in the memory, and the skill ratings of data items already in the memory are re-evaluated to determine if they can remain.

that reducing the forgetting of previously learned examples requires using a strategy of alternately learning new difficult examples along with reinforcement of well-learned examples. To design cross-lingual memory, we leverage Leitner queues, a cognitive technique that has been used for strategically planning what to review in humans (Leitner, 1974; Reddy et al., 2016) and for determining informative and spurious data in self-training non-continual learning applications (Amiri et al., 2018; Amiri, 2019). Our Leitner-guided memory sampling policy is a dynamic language-agnostic skill rating system which selects candidates for inclusion into memory according to how well they are learned (Figure 1). We analyze memory design attributes that contribute to reducing cross-lingual continual learning forgetting and evaluate on typologically diverse benchmarks ranging in difficulty.[1]

We summarize our contributions as follows:

(1) We are the first to formalize a human-inspired solution based on Leitner queues to guide cross-lingual memory replay (§2.3).

(2) We show that our Leitner-inspired approach for selecting memory replay items reduces forgetting without sacrificing transfer learning gains (§4.1).

(3) We provide a fine-grained analysis over different language orders and languages showing that our approach is consistently and robustly beneficial (§4.2).

(4) We provide a qualitative analysis that investigates the usefulness of data as a function of its learnability (§4.3).

2 Methodology

In this section, we start by describing our ER approach adapted to cross-lingual continual learning (§2.1). Then, we explain the mechanism for determining skill ratings based on Leitner queues (§2.2). After that, we explain how we use this Leitner-based skill rating system to guide memory storage and update in cross-lingual ER (§2.3).

2.1 Cross-lingual Experience Replay

We follow the same setup for cross-lingual continual learning and ER defined by M'hamdi et al. (2023). The learning process consists of sequentially fine-tuning a model on a cross-lingual datastream in multiple phases. A cross-lingual datastream $\mathcal{D}_{1 \cdots N}$ is a set of N distinct labeled datasets sampled from different languages one at a time. Each dataset $\mathcal{D}_i$ is drawn from a single distinct language $\ell_i \in \mathscr{L} = \{\ell_1, \ell_2 \cdots \ell_N\}$. Each phase $\mathcal{P}_i \in \mathcal{P}_{1 \cdots N}$ is a stage in cross-lingual continual learning where the model gets fine-tuned on a dataset $\mathcal{D}_i$ for a number of epochs. The ER approach is implemented as follows: At the end of each phase (except the last one) $\mathcal{P}_i \in \mathcal{P}_{1 \cdots N-1}$, we choose some data from $\mathcal{D}_i$ to add to a memory

[1] We will release our code in the camera-ready version.

buffer $\mathcal{M}$ of fixed size $|\mathcal{M}|$. In later phases $\mathcal{P}_j$ after $\mathcal{P}_i$, we replay from $\mathcal{M}$, which contains memory data drawn from $\mathcal{D}_{<j}$ interleaved with the main loss on data drawn from $\mathcal{P}_j$.

2.2 Leitner-based Skill Rating System

We draw inspiration from Leitner queues (Leitner, 1974), a method of prioritization originally conceived of as a strategy for human memorization and later used in machine learning applications (Amiri et al., 2018). The key prioritization insight we leverage is that of *demonstrated mastery*. That is, items in a (training) data set may be rated by the degree to which they have been mastered by the learner. We instantiate this by associating a rating r to each training data item d, and changing $r(d)$ based on a model m's ability to correctly classify d during training. Let $[s, e]$ be the acceptable rating range, let $r_{m'}(d)$ be the rating for d according to some previous model m', and let $\phi_m(d) \in \{-1, 1\}$ indicate that model m classified d {incorrectly, correctly}, respectively. Then

$$r_m(d) = \max(\min(r_{m'}(d) + \phi_m(d), e), s)$$

Thus, r is raised when d is correctly classified and lowered when it is misclassified, subject to the acceptable range. In this work, we set $[s, e] = [1, 5]$, following established practice (Reddy et al., 2016; Amiri et al., 2018).

2.3 Leitner-Guided Cross-lingual Experience Replay (*LER*)

We explore the use of $r(d)$ to determine whether or not to include d in $\mathcal{M}$. At the start of phase $\mathcal{P}i$, by convention, for all $d \in \mathcal{D}_i \cup \mathcal{M}$, we set $r_\emptyset(d)$, the initial rating, to s. At the end of each epoch within the phase, we update r for each data item in $\mathcal{D}_i$ and $\mathcal{M}$ according to the model m at that point in training. At the end of $\mathcal{P}i$, we use r values to form the new $\mathcal{M}$, selecting $\frac{|\mathcal{M}|}{i}$ items from $\mathcal{D}_i$ and $|\mathcal{M}| - \frac{|\mathcal{M}|}{i}$ items from the current $\mathcal{M}$ according to one of two strategies:

- *LER (Easy)*: Highest-rated items are prioritized.
- *LER (Hard)*: Lowest-rated items are prioritized.

Our approach for selecting data from $\mathcal{D}_i$ inversely proportional to i enables the fixed and limited $\mathcal{M}$ to contain an even distribution of samples from all $\mathcal{D}_{<i}$ thus seen, militated by the relative learning difficulty of different phase datasets.

3 Experimental Setup

We start by presenting the different baselines and model variants used to compare between different experimental scenarios (§3.1). We then describe the benchmark datasets and their base models (§3.2) along with the multilingual datastreams (§3.3) that we focus on in this evaluation. More implementation details such as the hyperparameters, number of parameters used, and runtime for different models can be found in Appendix A.

3.1 Baselines & Model Variants

Baselines Before delving into different variants of Leitner-guided memory replay, we consider the following baselines:

- *No ER*. This is our lower-bound naive sequential fine-tuning baseline. This sequentially fine-tunes on datasets sampled from one language at a time $\mathcal{D}_i \in \mathcal{D}_{1 \dots N}$ without using any experience replay.
- *Balanced*. This is an experience replay approach adapted from Lopez-Paz and Ranzato (2017) which allocates equally sized buffers balanced across language. At the end of each phase $\mathcal{P}_i$, $|\mathcal{M}|/(N-1)$ examples are randomly picked from $\mathcal{D}_i$ and added to $\mathcal{M}$.
- *Random*. This is a more realistic experience replay approach, adapted from Riemer et al. (2019), which randomly samples and updates $|\mathcal{M}|$ from $\mathcal{D}_{<i}$ at the end of each phase $\mathcal{P}_i$.

Other techniques have been proposed to produce memory exemplars such as K-Means clustering (Chaudhry et al., 2019b), mean of features (Rebuffi et al., 2017), and prototypical networks (Ho et al., 2023). However, we don't explore those approaches since they don't lead to clear improvements against *Random* (reservoir sampling) (Chaudhry et al., 2019b).

Model Variants We design the following model variants on top of *LER*. The research question we analyze here is: does dynamically prioritizing easy elements help in mitigating forgetting more than hard elements or vice versa? Our analysis evaluates the aggregated effectiveness of different strategies used for memory construction. This consists of *LER (Easy)* and *LER (Hard)* which use easy and hard examples to fill and update the memory, respectively.

3.2 Benchmarks & Base Models

We conduct experiments on two datasets commonly used in natural language understanding literature, covering different typologically diverse languages and requiring different levels of reasoning: multilingual task-oriented dialog (*MTOD*) and multilingual question answering (*MQA*).

MTOD This is a multilingual goal-oriented system focusing on the natural language understanding module. This module consists of two subtasks namely intent detection and slot filling. For *MTOD* evaluation, we use two multilingual task-oriented dialog datasets: *MTOP* (Li et al., 2021) and *MultiATIS++* (Xu et al., 2020). While *MultiATIS++* covers 18 intents and 84 slots on average per language from one domain, *MTOP* covers 117 intents and 78 slots from 11 domains. We choose *MTOP* and *MultiATIS++* since they are among the large-scale datasets available for task-oriented dialog covering typologically diverse languages. We use the same architecture as in Castellucci et al. (2019) to jointly learn intent classification and slot-filling subtasks. M-BERT (Devlin et al., 2019) is used to encode each input sentence. On top of the $[CLS]$ representation of the sentence, we use a linear layer plus Softmax to predict its intent class. We use a sequence labeling layer in the form of a linear layer plus CRF (Lafferty et al., 2001) to predict slot labels in BIO annotation. We optimize jointly over the sum of intent and slot losses. For evaluation, we use accuracy and F1 scores to evaluate intent classification and slot filling, respectively.

MQA This is a multilingual span-based question-answering task that extracts the answer token span to a question given a defined context. To ensure a challenging and trustworthy evaluation for *MQA*, we choose TyDiQA (Clark et al., 2020), which is a translation-free realistic information-seeking benchmark. We follow the same pre-processing and architecture as in Hu et al. (2020). Specifically, we concatenate the input question (after pre-pending it with a $[CLS]$ token) and the context as a single packed sequence separated by a $[SEP]$ token and feed that to M-BERT. Then, the embeddings of the context are fed to a linear layer plus Softmax to compute the probability that each token in the context is the start or end token of the answer span. We optimize for the joint loss over the start and end tokens predictions. Complying with Hu et al. (2020) evaluation, we use F1-score macro-averaged over examples.

Table 1 shows the statistics per language and split for *MTOP*, *MultiATIS++*, and *TyDiQA* datasets.

Dataset	Language	Train	Dev	Test
MTOP	English	15,667	2235	4386
	German	13,424	1815	3549
	Hindi	11,330	2012	2789
	Thai	10,759	1671	2765
MultiATIS++	English	4488	490	893
	French	4488	490	893
	Chinese	4488	490	893
	Turkish	578	60	715
TyDiQA	Indonesian	5131	571	565
	Russian	5841	649	812
	Swahili	2479	276	499
	Telugu	5006	557	669

Table 1: Statistics of *MTOP*, *MultiATIS++*, and *TyDiQA* per language and split.

3.3 Datastreams

We design a balanced set of distinct language permutations, following the cross-lingual continual learning evaluation paradigm established by M'hamdi et al. (2023). Formally, for a given set of $N = 4$ languages, we sample a subset of N language permutations $P \subset \mathfrak{S}(\mathscr{L})$ where each language appears exactly once in each permutation. Table 2 shows the language permutations we consider for different downstream benchmarks.

Dataset	#	Order
MTOP	1	English→German→Hindi→Thai
	2	German →English→Thai→Hindi
	3	Hindi→Thai→English→German
	4	Thai→Hindi→German→English
MultiATIS++	1	English→French→Turkish→Chinese
	2	French→English→Chinese→Turkish
	3	Turkish→Chinese→English→French
	4	Chinese→Turkish→French→English
TyDiQA	1	Russian→Indonesian→Telugu→Swahili
	2	Indonesian→Russian→Swahili→Telugu
	3	Telugu→Swahili→Russian→Indonesian
	4	Swahili→Telugu→Indonesian→Russian

Table 2: Language permutations for *MTOP*, *MultiATIS++*, and *TyDiQA*.

4 Results & Analysis

In this section, we provide an extensive analysis to demonstrate the effectiveness of our Leitner-guided cross-lingual experience replay approach. Our primary analytical tool is *forgetting*, which measures the degree to which a learned skill is lost when a model is trained on out-of-language data.

Lower forgetting is better while negative forgetting indicates the model has improved as a result of out-of-language training. We also show *final performance*, which is simply a metric's value after all phases of continual learning.[2] We present both a summary of the test performance based on the best epoch given Dev data split performance and over each epoch throughout different training stages (§4.1). Then, we present a more fine-grained analysis, shedding light on which language orders and languages our Leitner-based skill rating system is particularly helpful (§4.2). Last but not least, we present a qualitative analysis of different categories of skill ratings and what makes ruling out hard examples useful (§4.3).

Approach	*MTOP* Intent Accuracy ↓	*MTOP* Slot F1 ↓	*MultiATIS++* Slot F1 ↓	*TyDiQA* F1 ↓
No ER	5.84	7.56	2.62	1.52
Balanced[†]	0.92	1.15	−0.63	0.92
Random[‡]	0.68	0.97	−0.56	**0.73**
LER (Easy)	**0.49**	**0.51**	**−0.73**	0.83
LER (Hard)	0.82	2.27	1.10	1.14

Table 3: Average Test forgetting scores based on the Dev data split performance of different models and baselines. We compare two Leitner-guided memory replay variants *LER (Easy)* and *LER (Hard)* to the baselines. Since no previous work on experience replay in the cross-lingual setup reports any forgetting results, we implement in addition to *No ER* our internal baselines: *Balanced* and *Random* adapted from [†](Lopez-Paz and Ranzato, 2017) and [‡](Riemer et al., 2019), respectively. Best (lowest ↓) forgetting scores are highlighted in **bold** for each task and subtask.

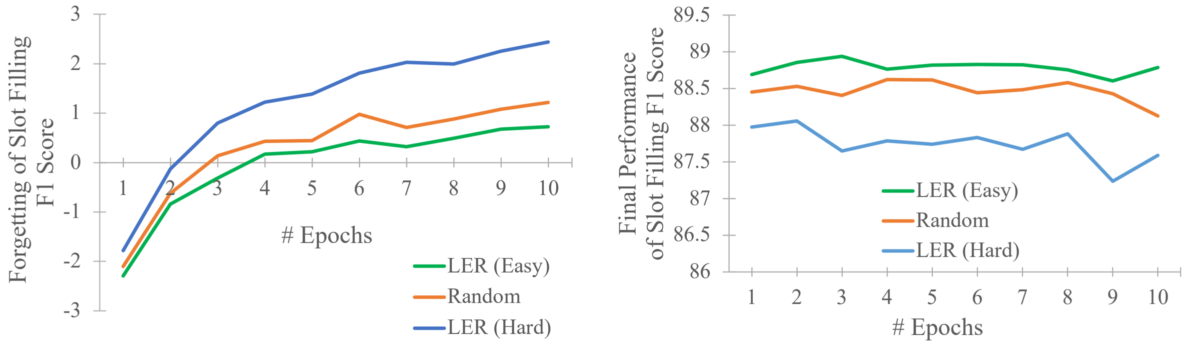
4.1 Average Performance

In Table 3, we compare between different Leitner-guided memory selection strategies and baselines for *MTOP*, *MultiATIS++*, and *TyDiQA* benchmarks in terms of their forgetting. We start by showing their forgetting on the test data averaged over different language orders based on the best-performing model on the Dev data split. Compared to *No ER* baseline, all *ER* approaches: *Balanced*, *Random*, and *LER* variants are beneficial in reducing forgetting, irrespective of the strategy followed in memory storage and update. It is clear that the forgetting gap between *No ER* and *ER* approaches is more pronounced for *MTOP* and *MultiATIS++* tasks than it is for *TyDiQA*. We conjecture that this is due to the formulation of *TyDiQA* as a span-based question-answering task. The latter employs a simple token classification model which is less challenging than joint optimization over classification and sequence modeling objectives in *MTOD*. The gains are even more pronounced for *MTOP*, whose ontology covers more domains, intents, and slots than that of single-domain *MultiATIS++*. Among *MTOP* subtasks, slot filling has a higher overall forgetting than intent detection. The implication of all of these findings is that forgetting is more pronounced, and our technique more crucial, when tasks are more difficult.

By keeping a balanced memory across languages, *Balanced* could have the benefit of making sure to revisit all languages assuming knowledge of the total number of languages involved in the continual learning. However, using a balanced memory across languages *Balanced* doesn't lead to lower forgetting than picking a random memory across languages *Random*. This could be because *Balanced* picks a balanced amount of examples per language, exposing the model to less diversity compared to *Random*. This could also show the need to continuously update the diversity of memory to make room for higher-quality examples in continual learning. *LER (Easy)* stands out as one of the most successful strategies in reducing forgetting beating both experience replay baselines. *LER (Easy)* reaches the lowest forgetting with reductions of 1.76, 0.64, and 0.46 in forgetting of F1 score for slot filling compared to *LER (Hard)*, *Balanced*, and *Random* respectively. We perform a two-tail paired two-sample for means t-Test on forgetting scores across different language orders and training epochs. We find that the gains from using *LER (Easy)* are statistically significant compared to *Random* with a p-value of 2.6×10^{-6} on slot filling. Results on *MultiATIS++* and to some degree *TyDiQA* confirm the consistent superiority of *LER (Easy)* and the inferiority of *LER (Hard)* approach.

For the remaining analysis, we focus on *MTOP*, shedding more light on the added value of *LER* approaches compared to the best-performing *ER* baseline *Random*. Figures 2a and 2b show the learning curves of different models on slot filling in terms of forgetting and final performance, respectively. Throughout training, *LER (Easy)* is consistently more effective than *Random* and *LER (Hard)* in minimizing forgetting while improving final performance, thus taming the stability plasticity dilemma. *LER (Easy)* can converge and stabilize at a low

[2] For forgetting and final performance, we use the same formulation of evaluation protocols as M'hamdi et al. (2023). A refresher is provided in Appendix B.



(a) Average forgetting ↓. (b) Average final performance ↑.

Figure 2: Average forgetting and final performance of slot filling for different model variants compared to the *Random* baseline averaged over different language orders. The lower the forgetting and the higher the final performance the better.

forgetting score earlier in training. On the other hand, the *LER (Hard)* strategy exacerbates the forgetting problem as training proceeds. This shows that replaying easy examples is consistently more effective than revisiting hard ones that the model is struggling with. Our Leitner-based skill rating system provides a dynamic measure that keeps selecting pertinent instances as language exemplars in constructing the memory replay.

4.2 Fine-grained Language Analysis

Figures 3 and 4 show a fine-grained analysis of forgetting between different models across different language orders and languages, respectively.[3] For each language order and language, we report Test results for the best-performing model based on Dev data split. Overall, we observe that *LER (Easy)* consistently outperforms *LER (Hard)* and *Random* across different language orders and languages. Certain language orders such as Thai→Hindi→German→English (4) and Hindi→Thai→English→German (3) have more forgetting than others. The languages that benefit the most compared to *Random* are Hindi and German whereas the gains for Thai and English are more minimal. *LER (Easy)* manages to bridge that gap in forgetting, keeping it within a low range.

4.3 Discussion

In this part, we conduct a qualitative analysis to complement our conclusion from our quantitative analysis that choosing training data for the memory that is easy to learn is more beneficial than choosing data that is not easily learned. To dig deeper into why ruling out harder examples from the memory is beneficial, we look more closely at the characteristics of those hard cases among training data. We define an *intractable* example as an example whose skill rating never gets promoted and stays 1 throughout training. At the other end of the spectrum, is a *confident* example whose skill rating converges to 5 and never gets demoted after

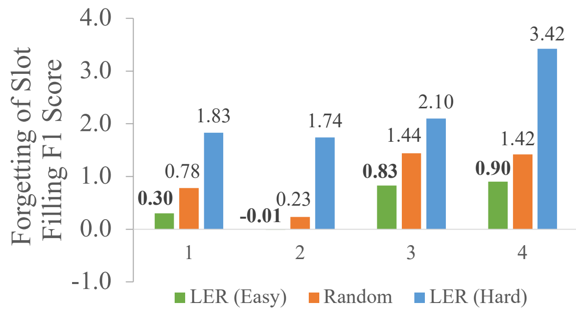


Figure 3: Fine-grained analysis of forgetting of slot filling over different language orders as defined in Table 2. Best (lowest) results for each language order are highlighted in **bold**.

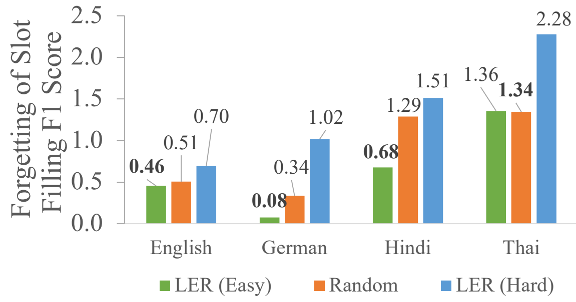


Figure 4: Fine-grained analysis of forgetting of slot filling over different languages. Best (lowest) results for each language are highlighted in **bold**.

[3] More results for other subtasks can be found in the Appendix C.

that.

In Figure 5, we report percentages of intractable and confident training data in *MTOP* for each language, averaged over all phases and language orders. We notice that for each language, 70% or more examples are confident. Thus, the *Random*approach to memory selection is unlikely to differ all that much from the *LER (Easy)* approach, at least in intent detection for *MTOP*. For other tasks with lower rates of easy examples, it might not be straightforward to pick easy examples with a random approach. We also observe a trend where the more high-resource the language is the less likely its examples are intractable and the more likely its examples are confident. Thai, which has the highest percentage of intractable examples, is the most low-resource in *MTOP*. This explains why *LER (Easy)* is much more beneficial than *Random* for Thai and Hindi compared to English and German for intent classification in Figure 9 (Appendix C).

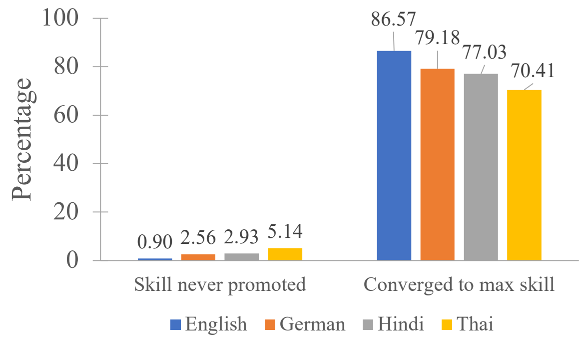


Figure 5: Percentages of examples that never get promoted past skill rating 1 (Skill never promoted) and those that converge to the maximum skill rating 5 (Converged to max skill) per language averaged over different language orders.

As an exemplar, we focus now on English data, specifically concentrating on training data analysis from the end of the first phase in language order English→German→Hindi→Thai. To understand what makes an example particularly intractable, we design the following categories:

- **Low-resource (LR)**: A training instance is considered low-resource if the number of training instances per its intent label is below 10. For English, there are 137 training instances per intent on average. This makes low-resource labels fall within the 25% percentile.
- **Difficult to disambiguate (DD)**: This is the case if the true class is among the most similar to the predicted class. We determine that by computing the $[CLS]$ token representations of all training sentences. We then compute the centroid of the sentence representations per class label. For each label, we determine its most similar predicted classes based on the 5 nearest neighbors.
- **Poorly-defined (PD)**: Unlike low-resourced and difficult to disambiguate examples which are automatically determined by their labels, we inspect here case by case for poorly-defined sentences. We define a poorly-defined example as any sentence that doesn't make sense to be attributed to a certain label. This could be due to a mismatch or lack of commonsense in the way the ontology was defined for certain labels.

We show in Figure 6 some statistics of different categories of intractable examples. Most intractable examples are either DD, LR, or both. Inspecting confident examples reveals that no LR or DD examples are encountered among them. This demonstrates that our Leitner-guided approach *LER (Easy)* can detect such hard categories and rule them out. By imposing a more fine-grained skill rating system, our Leitner-guided memory replay approach provides a more confident approach to determine which labels the model is struggling with more than relying on prediction loss (Amiri et al., 2018). The skill rating system adds information that prediction loss alone does not. In fact, only 27% of the English examples that have skill ratings between 2 and 4 (neither intractable nor highly confident) are wrongly predicted at the end of the first phase. Those unstable examples are part of the selection of *LER (Hard)* so not prioritizing such examples is beneficial.

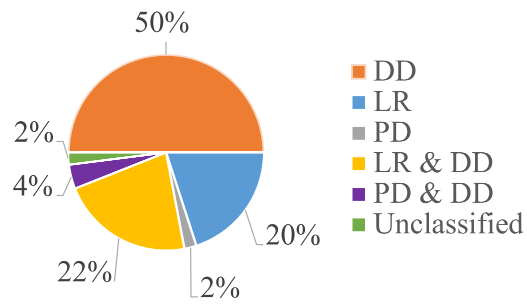


Figure 6: Distribution of different categories of intractable examples in the English data.

In Table 4, we provide some examples of different categories of wrongly predicted labels. We observe inconsistencies in those examples. Those that are specifically DD are so close to being picked

Type	Utterance	True Class	Prediction Classes
LR	Put this song on repeat. What is Tyler studying in school? Merge another call with this one.	music:LOOP_MUSIC people:GET_MAJOR calling:MERGE_CALL	music:REPLAY_MUSIC people:GET_UNDERGRAD calling:END_CALL
DD	Did Jack get sentenced today? How to make Arab tahini sauce? What time does the sun come up tomorrow?	news:GET_DETAILS_NEWS recipes:GET_INFO_RECIPES weather:GET_SUNRISE	news:QUESTION_NEWS recipes:GET_RECIPES weather:GET_SUNSET
PD	Where does Kade work? Pause the current timer and delete. Increase my timer to 30 minutes.	people:GET_LOCATION timer:PAUSE_TIMER timer:CREATE_TIMER	people:GET_EMPLOYER timer:DELETE_TIMER timer:RESTART_TIMER

Table 4: Examples of intractable examples and their golden truth and prediction intent labels from each category.

as representative examples of certain classes which can only confuse the learner. For example, while "Pause the current timer and delete." is supposed to be classified as *timer:PAUSE_TIMER*, this label is far from being comprehensively descriptive of the sentence intent. Its predicted label *timer:DELETE_TIMER* is not wrong either as it detects the intent to delete which is the second part of the example. We suspect that reinforcing the learning using difficult cases can only mislead the learner.

In Figure 7, we show a t-SNE projection of the centroids of different intent label representations. We highlight in that figure the most common DD labels whose representations are indistinguishable in the vector space. Some of those labels like GET_STORIES_NEWS and GET_DETAILS_NEWS are to the human eye also DD, which could be an artifact of how the intent ontology was defined. Our Leitner-guided strategy *LER (Easy)* rules them out, favoring examples the learner is more confident about with class labels that correspond to more clearly separable representations.

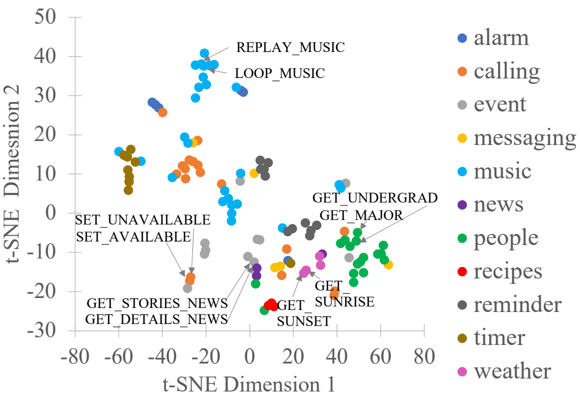


Figure 7: t-SNE visualization of centroids of different intent labels highlighting some ambiguous labels indistinguishable in the embeddings space.

5 Related Work

Continual learning work inspired by human-like learning can be divided into the following categories: spaced-repetition (Smolen et al., 2016; Amiri et al., 2017; Amiri, 2019; Feng et al., 2019; Klasson et al., 2023), mechanisms of sleep (Ball et al., 2020; Mallya and Lazebnik, 2018; Schwarz et al., 2018), reactivation of memories (Hayes et al., 2020; van de Ven et al., 2020), etc. Leitner queues, one of the most famous spaced repetition techniques, started garnering attention for machine learning recently. However, most of the work is focused on scheduling when to review data in non-continual learning setups. Amiri et al. (2017) show the sample efficiency of a human-inspired memory model to determine when to review each item as a function of the difficulty of the item and the strength of the network. Klasson et al. (2023) propose a Monte Carlo tree search approach for memory replay. More work (Amiri et al., 2018; Amiri, 2019) demonstrates the effectiveness of Leitner queues at determining spurious data and confident labels for self-training applications. In our work, we are the first to test for the effectiveness of Leitner queue-based skill ratings in mitigating forgetting in cross-lingual continual learning.

6 Conclusion

In this paper, we formulate a human-inspired experience replay approach specifically for cross-lingual continual learning. We propose a Leitner-based skill rating system to dynamically populate and update the memory with high-quality items. Our approach can deal with the plasticity stability dilemma better than random selection, especially for complex tasks and consistently over languages and language orders. The implications of this analysis include a recipe for how to incorporate aspects of human learning in the design of memory replay in cross-lingual continual learning.

Limitations

In this paper, we have focused on Leitner queues as an approach to guide the process of memory storage and update. In future work, other variants of Leitner queues or other approaches based on human learning theories could be explored. For example, more fine-grained approaches based on theories of how languages get forgotten to model the retention curve as a function of the task difficulty, review periods, and strength of the model could be investigated. This could help us understand how the process of forgetting works and when to schedule revisions accordingly to circumvent that.

In this paper, we evaluate on a representative set of natural language understanding tasks. We prove that our approach benefits challenging tasks more consistently. However, we don't closely investigate if there is a correlation between the difficulty of a task and the effectiveness of our Leitner-based skill rating approach on it. For such an analysis to be possible, we need a principled way to define what makes a task more difficult naturally or to simulate that synthetically. We leave a systematic fine-grained analysis over more downstream tasks ranging in difficulty for future work.

References

Hadi Amiri. 2019. Neural self-training through spaced repetition. 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 21–31, Minneapolis, Minnesota. Association for Computational Linguistics.

Hadi Amiri, Timothy Miller, and Guergana Savova. 2017. Repeat before forgetting: Spaced repetition for efficient and effective training of neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2401–2410, Copenhagen, Denmark. Association for Computational Linguistics.

Hadi Amiri, Timothy Miller, and Guergana Savova. 2018. Spotting spurious data with neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2006–2016, New Orleans, Louisiana. Association for Computational Linguistics.

Philip J. Ball, Yingzhen Li, Angus Lamb, and Cheng Zhang. 2020. A study on efficiency in continual learning inspired by human learning. *CoRR*, abs/2010.15187.

Gail A. Carpenter and Stephen Grossberg. 1988. Art 2: Self-organization of stable category recognition codes for analog input patterns. In *Other Conferences*.

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

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

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

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

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.

Kanyin Feng, Xiao Zhao, Jing Liu, Ying Cai, Zhifang Ye, Chuansheng Chen, and Gui Xue. 2019. Spaced learning enhances episodic memory by increasing neural pattern similarity across repetitions. *The Journal of Neuroscience*, 39:2741–18.

Raia Hadsell, Dushyant Rao, Andrei Rusu, and Razvan Pascanu. 2020. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 24:1028–1040.

Tyler L. Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. 2020. REMIND your neural network to prevent catastrophic forgetting. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VIII*, volume 12353 of *Lecture Notes in Computer Science*, pages 466–483. Springer.

Stella Ho, Ming Liu, Lan Du, Longxiang Gao, and Yong Xiang. 2023. Prototype-guided memory replay for continual learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–11.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson.

2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

David Isele and Akansel Cosgun. 2018. Selective experience replay for lifelong learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3302–3309. AAAI Press.

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.

Marcus Klasson, Hedvig Kjellström, and Cheng Zhang. 2023. Learn the time to learn : Replay scheduling in continual learning. *Transactions on Machine Learning Research*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

S. Leitner. 1974. *So lernt man lernen*. Herder.

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.

David Lopez-Paz and Marc' Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Arun Mallya and Svetlana Lazebnik. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7765–7773. Computer Vision Foundation / IEEE Computer Society.

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

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

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. 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.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5533–5542. IEEE Computer Society.

Siddharth Reddy, Igor Labutov, Siddhartha Banerjee, and Thorsten Joachims. 2016. Unbounded human learning: Optimal scheduling for spaced repetition. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1815–1824. ACM.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2019. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4535–4544. PMLR.

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.

Paul Smolen, Yili Zhang, and John Byrne. 2016. The right time to learn: Mechanisms and optimization of spaced learning. *Nature Reviews Neuroscience*, 17:77–88.

Gido van de Ven, Hava Siegelmann, and Andreas Tolias. 2020. Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, 11:4069.

Maciej Wolczyk, Michal Zajac, Razvan Pascanu, Lukasz Kucinski, and Piotr Milos. 2021. Continual world: A robotic benchmark for continual reinforcement learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28496–28510.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5052–5063, Online. 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 Implementation Details

We specify below more implementation details such as hyperparameters and datasets in addition to the runtime and number of parameters of different models.

A.1 Hyperparameters

For all experiments, we use M-BERT (bert-base-multilingual-cased)[4] with 12 layers as our pre-trained multilingual Transformer-based encoder model. Consistent with M'hamdi et al. (2023) and Hu et al. (2020) for *MTOP* and *TyDiQA*, respectively, we use the Adam optimizer (Kingma and Ba, 2015), fixing the learning rate to $3e-5$ for all experiments for a fair comparison. M'hamdi et al. (2023) perform a manual hyperparameter search over the range $[1\times10^{-4}, 3\times10^{-4}, 1\times10^{-5}, 3\times10^{-5}]$ to choose the most optimal learning rate based on Dev data split performance. For *TyDiQA*, those hyperparameters are chosen based on Hu et al. (2020). For *MultiATIS++*, we perform a manual search over the same learning rates range and find that 3×10^{-5} performs comparably to other learning rates. So, we fix a learning rate of 3×10^{-5}, $\epsilon = 1\times10^{-8}$, $\beta_1 = 0.9$, $\beta_2 = 0.99$ in the optimizer for a fair comparison for all experiments. For *TyDiQA* experiments, we find it helpful when a scheduler with linear decaying learning rates is used. We use batch sizes of 4, 16, and 4 for *MTOP*, *MultiATIS++*, and *TyDiQA*, respectively. In all baseline models *Balanced* and *Random* and Leitner-guided *ER* (*LER*) model variants, we choose a fixed memory proportion to 20% of the training data from each benchmark. Based on that, we fix $|\mathcal{M}|$ memory size to 10,105, 500, and 500 for all *MTOP*, *MultiATIS++*, and *TyDiQA* experiments, respectively. We also fix the sampling frequency from the memory to every 10 minibatches. For all experiments, we run for 10 epochs maximum and pick the best model based on Dev data split. We use the same seed across all experiments to report the mean results. We also fix a seed of 42 for the random initialization of Numpy, Random, and Torch libraries over all experiments. All experiments are run on the same computing infrastructure using 1 NVIDIA A40 GPU of 46,068 MiB of memory CUDA version 11.6 and Pytorch version 1.13.1.

A.2 Dataset License

MTOP dataset has been released by Facebook under Creative Commons Attribution-ShareAlike 4.0 International Public License. *MultiATIS++* and *TyDiQA* datasets have been released under the Apache License which allows the use, modification, and distribution of the dataset.

A.3 Runtime

We show in Table 5 the runtime of different approaches and baselines for one single language order on *MTOP*. This runtime includes both the costs of training and evaluation. Our *LER* only incurs 3 hours more than *No ER* approach, with most of it spent calculating the skill rating at the end of each epoch. Table 6 compares between the number of parameters of models used for different downstream benchmarks. Task-oriented dialog benchmarks (*MTOP* and *MultiATIS++*) require more parameters and thus are more challenging compared

[4] `github.com/huggingface/transformers` version 3.4.0 pre-trained on 104 languages, including all languages covered in our evaluation.

to span-based question answering (*TyDiQA*).

Model	Total Runtime
No ER	6 hrs 23 min 20 sec
Balanced	6 hrs 45 min 22 sec
Random	6 hrs 25 min 25 sec
LER(easy/hard)	9 hrs 23 min 13 sec

Table 5: Fine-grained runtime analysis per model for one single language order on *MTOP*.

Model	# Parameters
MTOP	178,081,402
MultiATIS++	178,036,139
TyDiQA	177,264,386

Table 6: Fine-grained parameter analysis per benchmark.

B Evaluation Metrics

We follow M'hamdi et al. (2023) cross-lingual continual learning evaluation protocols. 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 ℓ_i fine-tuning K on $\mathcal{D}_{\leq j}$. For a more succinct analysis that sheds light on the stability-plasticity dilemma, we focus on the following two metrics:

- **Forgetting (F ↓).** We compute forgetting averaged *over* $\mathcal{D}_{2 \cdots \leq N}$ as follows:

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

  where $F_{\leq j}$ is the average forgetting that resulted at the point of training on $\mathcal{D}_j$. $F_{i,\leq j} = \max_{k \in [1,j-1]} R_{i,\leq k} - R_{i,\leq j}$. $F_{i,\leq j}$ is the degree to which performance on $\mathcal{D}_i$ has degraded by continuing to train on $\mathcal{D}_{\leq j}$ instead of stopping before including $\mathcal{D}_j$.

- **Final performance (FP ↑).** This is the final performance at the last phase $\mathcal{P}N$ averaged over all datasets $\mathcal{D}_{\leq N}$:

$$FP = \frac{1}{N} \sum_{i=1}^{N} R_{i,\leq N}. \tag{2}$$

C More Results

Figures 8 and 9 show a fine-grained analysis of the forgetting of intent classification over different language orders and languages, respectively.

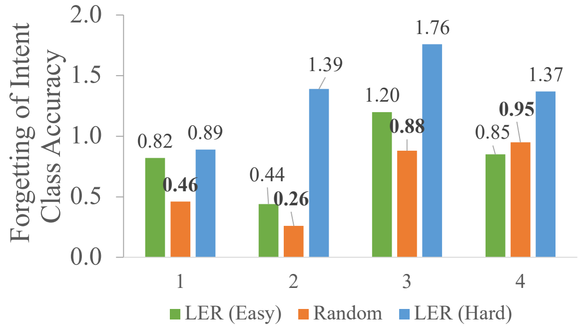


Figure 8: Fine-grained analysis of forgetting of intent classification over different language orders as defined in Table 2. Best (lowest) results for each language order are highlighted in **bold**.

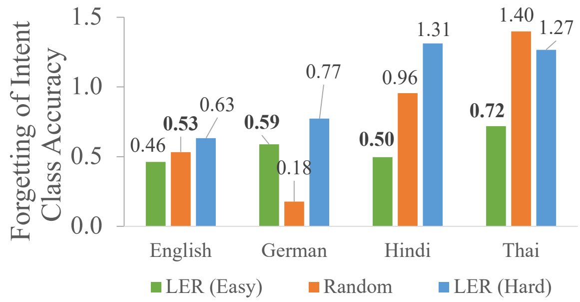


Figure 9: Fine-grained analysis of forgetting of intent classification over different languages. Best (lowest) results for each language are highlighted in **bold**.