A Study of Cross-lingual Transfer in Continual Slot Filling for Natural Language Understanding

Anonymous ACL submission

Abstract

Knowledge transfer between neural language models is a widely used technique that has proven to improve performance in a multitude of natural language tasks. In recent years, high cross-lingual transfer has been shown to occur in multilingual language models. Hence, it is of great importance to better understand this phenomenon as well as its limits. While most studies focus on training on independent and identically distributed (*i.e. i.i.d.*) samples, in this paper we study cross-lingual transfer in continual slot filling for natural language understanding. We investigate this by training multilingual BERT on one language at a time in sequence from the MultiATIS++ corpus, that contains a total of 9 languages. Our main findings are that forward transfer is retained although forgetting is still present, and that lost performance can be recovered with as little as a single training epoch. This may be explained by a progressive shift of model parameters towards a better multilingual initialization. We also find that commonly used metrics might be insufficient to describe continual learning performance.

1 Introduction

In recent years, task-oriented dialogue systems have been widely used in a variety of industries, often appearing in websites to help users navigate and find useful information. A key component of these systems is the task of natural (or spoken) language understanding (NLU) (Tur and De Mori, 2011). This task consists in determining the user's intent as well as the different concepts mentioned by the user to process the query. The former is defined as an utterance multi-class classification problem (intent detection) and the latter as a sequence labeling problem (slot filling).

State-of-the-art models for NLU usually leverage deep neural networks. In particular, pre-trained Transformer-based (Vaswani et al., 2017) language models like BERT (Devlin et al., 2019) have proven to perform very well on both intent detection and slot filling (Chen et al., 2019; Zhang et al., 2019). These models are pre-trained in a self-supervised way on large text corpora and rely on knowledge transfer to solve downstream tasks.

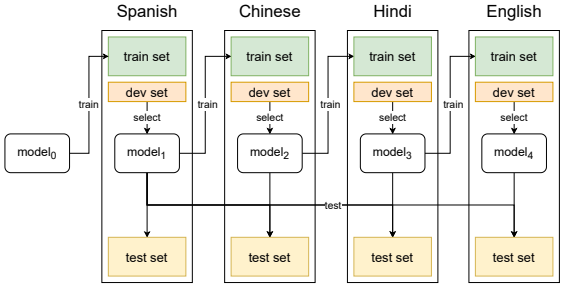


Figure 1: Depiction of a training sequence across 4 languages. For each language in the given order, we train the model on its training set, select the best epoch on the development set and then test on all test sets independently.

Oftentimes, collecting utterances and annotating them is expensive, which makes training data scarce or incomplete at the beginning of a project. Moreover, system requirements might evolve with time based on the needs of the users. Thus, a highly desirable feature of an NLU model is its ability to adapt based on new data (*e.g.* coming from interactions with real users). This means that model adaptation needs to happen sequentially as training data becomes available. However, many adaptation axes exist, like new slot labels, intents, domains or languages. Adapting a previously trained model is a costly endeavour, as it requires either re-training from scratch or maintaining many distinct models.

In this work, we choose to study cross-lingual transfer when progressively adapting a slot filling model to new languages. Although NLU typically consists of both slot filling and intent detection, we decide to focus solely on slot filling, as we believe it represents a more challenging scenario for a con-

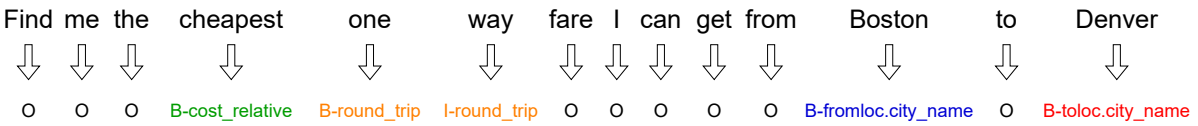


Figure 2: Example of slot filling IOB (Ramshaw and Marcus, 1995) labels for an utterance of MultiATIS++ (Xu et al., 2020) in English. Label "O" (from *outside*) denotes that no concept is mentioned, "B" (from *beginning*) denotes the first word of a concept and "I" (from *inside*) the continuation of a concept. Different slot types are shown in different colors.

tinually trained model. Additionally, we choose to work with multilingual BERT, that has shown high transfer capabilities between languages (K et al., 2020; Wang et al., 2020). While most cross-lingual transfer studies focus either on joint training or training on a source and a target language (Wang et al., 2020; Xu et al., 2020; Schuster et al., 2019), our main contribution is a study with special focus on *continual* cross-lingual transfer for the slot filling task.

Progressively training on multiple datasets that become available one by one is at the heart of continual learning (Hadsell et al., 2020), where the goal is for a model to improve itself both on past and new data. We refer to these datasets and the order in which they appear as a *training sequence* (f.i. see Figure 1). Traditional training schemes assume that training examples (in our case utterances) are independent and identically distributed (*i.i.d.*), which does not usually hold when data becomes available sequentially. Moreover, access to previous data is not allowed[1], as this represents a linear use of resources with respect to the length of the sequence, which can in theory be infinite. In this context, transfer is generally divided in two: forward and backward (Hadsell et al., 2020; Lopez-Paz and Ranzato, 2017; Arora et al., 2019), defined in our case as improvement on future and already acquired languages respectively. The biggest challenge of continual learning systems is catastrophic forgetting (Hadsell et al., 2020; French, 1999), which is defined as a strong performance loss in previously acquired knowledge (*i.e.* negative backward transfer). While previous studies of continual learning on dialogue tend to focus on the domain axis (Lee, 2017; Madotto et al., 2020), we concentrate on the axis of language adaptation. Similar work also investigates continual learning on multilingual language models (Liu et al., 2021). However, they try to preserve masked language modeling performance and cross-lingual ability after fine-tuning on a new task, while our work focuses on fine-tuning on a single task over a sequence of many languages.

[1]Access to previous data is sometimes allowed if limited (Robins, 1995)

Language	Utterances			Slots
	train	*dev*	*test*	
Hindi	1440	160	893	75
Turkish	578	60	715	71
Others	4488	490	893	84

Table 1: Number of utterances per subset and number of slots for each language in MultiATIS++ (Xu et al., 2020).

Our study is guided by the following research questions, presented through sections 5 to 7:
Q1: does cross-lingual transfer exist during continual training or does catastrophic forgetting prevent it? How much transfer can we expect relative to *i.i.d.* training?
Q2: how is transfer affected by the training sequence?
Q3: can lost performance due to forgetting be recovered? What is the associated cost?

2 Task

We study the task of slot filling for NLU in task-oriented dialogue, where certain words are associated to concepts that a system might need to execute a user's query (see Figure 2). The slot filling task is evaluated using the slot micro F1 score (Tjong Kim Sang and Buchholz, 2000).

We choose to work with the Air Travel Information System (ATIS) corpus (Hemphill et al., 1990). Since our goal is to study cross-lingual transfer and forgetting, we use the recent MultiATIS++ extension (Xu et al., 2020), as it is the only existing NLU corpus we could find that includes such a wide variety of languages and language families. This multilingual dataset consists of the manual translation of the original English (EN) ATIS sen-

tences into 6 different languages: Spanish (ES), Portuguese (PT), German (DE), French (FR), Chinese (ZH) and Japanese (JA). It also includes two additional languages: Hindi (HI) and Turkish (TR), that were added as part of MultiATIS in (Upadhyay et al., 2018).

MultiATIS++ utterances are labeled using the IOB format (Ramshaw and Marcus, 1995), where labels consist of a prefix (B,I or O) and an optional slot type that categorizes the identified concept. While O indicates that the word is not part of a concept, B and I indicate that it is the beginning or continuation of a concept. An example of this labeling scheme is shown in Figure 2.

Contrary to the translations added in MultiATIS++, the number of utterances of Hindi and Turkish translations are not as many as for the other languages. More details on the composition of MultiATIS++ are shown in Table 1. In the rest of the paper, we denote the *train*, *dev* and *test* sets of a given language i with a subscript (e.g. $train_i$).

3 Model

We use the multilingual BERT (Devlin et al., 2019) base model, consisting of 12 multi-head attention layers with 12 heads and hidden size of 768 (177M parameters). This model was trained on large Wikipedia dumps from 104 different languages using masked language modelling and next sentence prediction objectives.

As we use the model exclusively for the slot filling task, we append a two-layer feed-forward classifier with hidden size 768 and ReLU (rectified linear unit) activation (Nair and Hinton, 2010). The input of the classifier are the last layer word hidden states after applying dropout with $p = 0.1$.

Following (Xu et al., 2020), we train using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-5} and a batch size of 32 utterances for 50 epochs (unless stated otherwise), selecting the model with the highest slot F1 on the corresponding *dev* set. We evaluate the model on $test_i$ sets for every language i using the slot F1 calculated with the `seqeval` library (Nakayama, 2018).

4 Metrics

Cross-lingual transfer can be defined as the performance improvement of a model on a particular language based on knowledge of other languages. This can take several forms depending on the training structure. In an *i.i.d.* context, we think of transfer in terms of joint training. If training on language i and j jointly (multilingual) yields better performance on j than training only on j (monolingual), then there is transfer from i to j.

However, continual learning adds a different dimension. Indeed, when training on a language sequence we can identify two types of transfer: forwards and backwards (Hadsell et al., 2020; Lopez-Paz and Ranzato, 2017). Forward transfer denotes the performance and learning efficiency improvement on a given language thanks to previously acquired knowledge of other languages. Conversely, backward transfer denotes the performance improvement on a previously acquired language when learning a new one. More formally, and similarly to Lopez-Paz and Ranzato (2017), given a sequence of L languages, we define the performance matrix $P \in \mathbb{R}^{L \times L}$, where P_{ij} is the performance of language i after learning language j. In this context, backward transfer of i is defined as:

$$\mathrm{BT}_i = P_{iL} - P_{ii} \tag{1}$$

Negative backward transfer is also called forgetting, as it denotes performance loss on previous languages. Since P_{11} is equivalent to monolingual performance mono_1, we can define backward transfer of the first language after learning language j:

$$\mathrm{BT}_{1j} = P_{1j} - \mathrm{mono}_1 \tag{2}$$

Conversely, we define forward transfer as:

$$\mathrm{FT}_i^{\mathrm{mono}} = P_{ii} - \mathrm{mono}_i \tag{3}$$

where mono_i denotes monolingual performance on language i. By comparing performance with a different baseline like multilingual, we can measure how close forward transfer is to joint transfer:

$$\mathrm{FT}_i^{\mathrm{multi}} = P_{ii} - \mathrm{multi}_i \tag{4}$$

where multi_i denotes the multilingual performance on language i. These definitions will be useful for the analysis in Section 6.

5 Cross-lingual Transfer

Does transfer exist during continual training or does catastrophic forgetting prevent it?

Before studying the continual learning scenario, we measure different types of cross-lingual transfer to serve as a point of comparison.

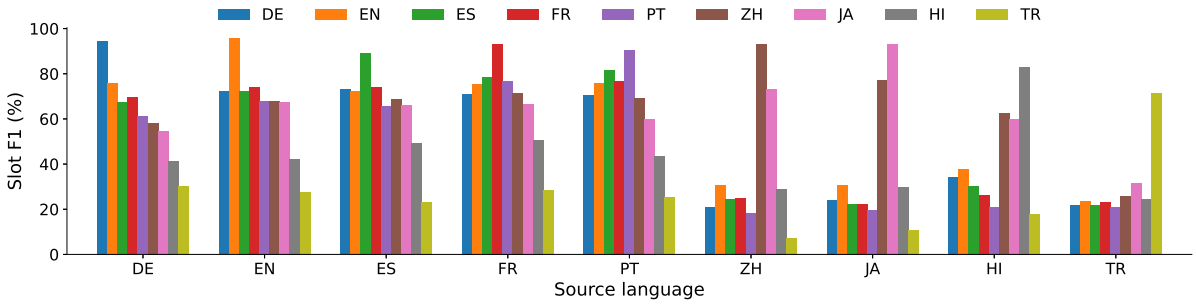


Figure 3: Performance obtained on $test_i$ for every target language i after monolingual training on each source language (x axis) averaged across 5 runs.

First, we look at how much transfer a model trained on a single language can achieve on other languages, (*i.e. zero-shot* transfer). Second, we measure transfer when training the model on all languages at once (*i.e. joint* transfer). Lastly, having this frame of reference, we investigate transfer when training the model on each language sequentially (*i.e. continual* transfer).

5.1 Zero-shot Transfer

In this section we look at zero-shot performance between language pairs. We train the model on a single language (monolingual) and then look at the performance on each of the other languages without further training.

In Figure 3, we observe two distinct phenomena that are consistent with previous related studies (Rahimi et al., 2019). First, zero-shot transfer seems to be maximal within languages with similar scripts. For example, Chinese achieves its highest performance when training on Japanese (and vice-versa), but performance of other languages after training on Japanese is poor. The only exception we find is Turkish, that uses a Latin script but obtains poor zero-shot transfer to other languages with the same script. This may be explained by its reduced set of training utterances. Second, language pairs with highest transfer are not always symmetric: the best source language for Spanish is Portuguese, but for Portuguese it is French.

Finally, we note that overall zero-shot performance is the highest when the source language is European, even to other language families and scripts.

5.2 Joint Transfer

In order to measure transfer in unstructured *i.i.d.* training, we train the model on all languages together (multilingual) and compare the performance that we obtain with monolingual training. Note that multilingual training corresponds to concatenating all $train_i$ for training and all dev_i for validation. We report the mean and standard deviation of *test* slot F1 per language across 5 runs to reduce the effect of randomness.

In Table 2, we observe that multilingual is always stronger than monolingual (except for Chinese and Japanese), which confirms the existence of joint cross-lingual transfer. European languages (German, English, Spanish, French and Portuguese) show modest but visible gains from transfer, whereas Asian languages (Chinese and Japanese) do not seem to benefit from it. However, transfer for the two low resource languages (Hindi and Turkish) is outstanding, with an absolute 4.8% and 13.9% improvement. As noted in (Do et al., 2020), MultiATIS++ translations keep the same (unrealistic) slot values for particular labels (e.g. American *departure city* and *destination city* in Turkish utterances). We suspect this may be the reason why transfer is particularly high in this corpus.

On the other hand, multilingual assumes that all languages are available at once. As mentioned before, this is not always true in practice, since utterances may be scarce and annotations expensive. Moreover, given N the maximum number of utterances per language and L the number of languages, training on a new language has time cost $O(LN)$, as the whole model needs to be trained from scratch. A naive solution is to use multiple monolingual models, raising however the space cost to $O(LN)$. Reducing both costs to $O(N)$ motivates our decision to structure training as a sequence.

5.3 Continual Transfer

Given a training sequence (a list of languages in a given order), continual learning consists in training

Training	DE	EN	ES	FR	PT	ZH	JA	HI	TR	Model Cost Time	Model Cost Space	Data Cost Space
Monolingual	94.4 (0.2)	95.6 (0.1)	88.9 (0.4)	93.2 (0.1)	90.3 (0.6)	93.3 (0.4)	93.1 (0.4)	82.4 (0.5)	71.3 (0.9)	≤224K	1.6B	≤4K
Multilingual	95.0 (0.2)	96.0 (0.2)	90.4 (0.4)	94.0 (0.3)	91.4 (0.2)	93.6 (0.2)	93.0 (0.1)	87.2 (0.3)	85.2 (0.6)	1.7M	178M	33K
Continual (P_{LL})	94.9 (0.2)	95.9 (0.1)	89.9 (0.5)	93.9 (0.3)	91.3 (0.3)	93.9 (0.3)	93.1 (0.3)	85.6 (0.7)	84.0 (0.6)	≤224K	178M	≤4K
Continual (P_{1L})	94.0 (0.7)	95.5 (0.2)	89.2 (0.5)	91.4 (1.7)	88.4 (4.9)	92.0 (1.0)	91.7 (0.7)	80.5 (1.8)	68.1 (3.5)			

Table 2: Slot F1 performance on $test_i$ sets for monolingual, multilingual and continual experiments. The latter are calculated as the average of the first (P_{1L}) or last (P_{LL}) language (indicated by the column) at the end of the sequence. Reported values are the average of 5 runs with standard deviation shown in parenthesis. Model time cost denotes the cost of adding a new language to the model measured in iterations. Model space cost is the size of the model measured in number of parameters. Data space cost represents the number of utterances stored in memory at the same time.

the model on $train_i$ (and validating on dev_i) for each language i in the given order, as depicted in Figure 1. Although having all languages at once is not required and the language addition cost is the lowest, this approach is prone to forgetting previously learned languages.

In the experiments of this section, we report for both forward and backward transfer the average performance per language. The experiments consist of 3 sequences per language and per transfer type repeated 5 times to reduce the effect of randomness, making a total of 54 sequences and 270 experiments. These 3 sequences per language are chosen randomly and maximizing the Kendall rank correlation coefficient (Abdi, 2007) as a distance criterion to make sure they are as dissimilar as possible.

We first investigate whether forward transfer exists in continual training by looking at the average P_{LL} performance (*e.g.* model_4 evaluated on English in Figure 1) against monolingual and multilingual. Notice that we look at the performance of the last language, as this allows us to measure whether the model leverages past knowledge to learn a new language. This has the advantage of isolating the effect of forward transfer from that of backward transfer. We also make sure that each language appears at the *end* of the sequence the same number of times.

Similarly, we look at backward transfer by comparing the average P_{1L} performance (*e.g.* model_4 evaluated on Spanish in Figure 1) against monolingual, making sure that each language appears at the *beginning* the same number of times. This way we can determine whether the initial performance (equal to monolingual) improves with the introduction of new languages to the model. We also look at the performance of the first language, so that the effect of backward transfer is isolated from that of forward transfer.

Notice that whether we focus on the first or the last language, we always look at the performance at the end of the training sequence so that the comparison to multilingual is fair.

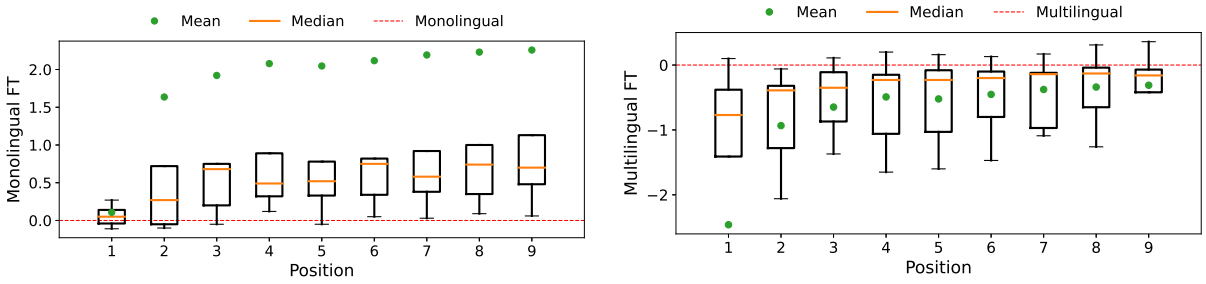
In Table 2, we observe that continual training benefits from cross-lingual forward transfer. Indeed, P_{LL} is on average closer to multilingual than to monolingual performance. However, although transfer is high for the last language, P_{1L} suffers from the opposite effect, even falling under monolingual performance. Our results show that contrary to what we expected from the identical slot values of MultiATIS++ (Xu et al., 2020) (e.g. American *departure city* and *destination city* in Turkish utterances), the naturally occurring cross-lingual transfer completely vanishes in previous languages.

6 Training Sequence

How is transfer affected by the training sequence?

In order to better understand the effect of the training sequence, we first look at measures of forward transfer at each position relative to monolingual and multilingual. Secondly, we study the impact of the training sequence length on backward transfer measured on the first language. Note that in the figures of this section the mean, median and percentiles do take into account eventual outlier languages, while the minimum and maximum do not.

When considering forward transfer, Figure 4a shows that apart from the first position (equal to monolingual), the model consistently benefits from transfer at any point in the sequence, as performance is higher than monolingual. Interestingly, due to some outlier languages (generally Hindi and Turkish), we observe that the means are poor estimates of the distribution when measuring $\text{FT}_i^{\text{mono}}$.



(a) $\mathrm{FT}_i^{\mathrm{mono}} = P_{ii} - \mathrm{mono}_i$ (higher is better)

(b) $\mathrm{FT}_i^{\mathrm{multi}} = P_{ii} - \mathrm{multi}_i$ (higher is better)

Figure 4: Distributions of forward transfer on $test_i$ relative to monolingual and multilingual for different positions i in the sequence. We average over 54 sequences and 5 runs. Note that forward transfer is 0 when performance is equal to (a) monolingual and (b) multilingual. Outliers not shown for readability.

This is an indicator that commonly used continual transfer metrics might over- or underestimate real performance when transfer is not uniformly distributed among languages. Indeed, these metrics usually consist of averages across the adaptation axis (Lopez-Paz and Ranzato, 2017). In Figure 4b, we also observe that performance gets closer to multilingual as the sequence advances, although it rarely outperforms it.

As per backward transfer, Figure 5 shows that performance of the first language is in general worse than monolingual for any given sequence length. In particular, we observe that performance loss is not strictly monotonic, which means that measuring forgetting between the beginning and the end of the sequence may not be sufficient to explain how the model forgets. Note that a sequence of $L = 7$ would have shown less forgetting than a sequence of $L = 5$.

Furthermore, as hinted by continual experiments from Table 2, we observe that backward transfer deteriorates as forward transfer improves with the length of the sequence. Since negative backward transfer (*i.e.* forgetting) tends to be linked to a loss of previously acquired knowledge, it is surprising that new language performance keeps increasing while performance of known languages decreases. Our results indicate that the preserved knowledge that facilitates the acquisition of a new language in multilingual BERT for slot filling is not the same knowledge that preserves previous language performance. This might be explained by a progressive shift of model parameters towards a better multilingual initialization for the ATIS task that might however fail to retain the specificities of previous languages. This hypothesis motivates our next research question.

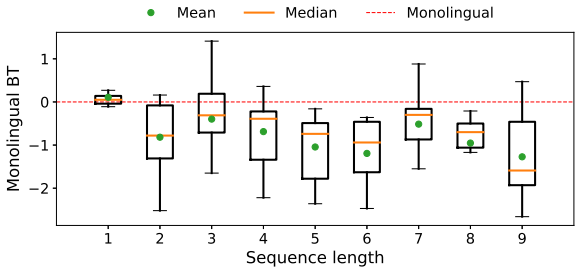


Figure 5: Distributions of first language backward transfer $\mathrm{BT}_{1j} = P_{1j} - \mathrm{mono}_1$ (higher is better) on $test_1$ for different sequence lengths j. We average across 54 sequences and 5 runs. Note that $\mathrm{BT}_{1j} = 0$ if performance is equal to monolingual. Outliers not shown for readability.

7 Fast Recovery

Can lost performance due to forgetting be recovered?

Given that forward transfer does not seem to be affected by forgetting, we investigate in this section whether performance lost as a result of forgetting can be recovered quickly after continual training. In order to understand if this is possible, we first set out to discover whether the model shifts towards a better multilingual initialization. Hence we compare the multilingual performance of the initial model_0 (consisting of BERT and a random classifier) against model_L, the model at the end of training sequence (*e.g.* model_4 in Figure 1). In particular, we train both models on all available languages jointly for different numbers of epochs. Notice that model_L comes from our continual P_{1L} experiments over 27 sequences (see Table 2). The results are presented in Table 3.

The comparison between model_0 multilingual

Model	Epochs	DE	EN	ES	FR	PT	ZH	JA	HI	TR
model_0 multilingual (*i.i.d.*)	1	82.7 (1.2)	83.6 (0.7)	78.2 (0.3)	80.7 (0.7)	79.4 (0.5)	83.5 (0.7)	82.7 (1.0)	79.6 (0.7)	69.8 (1.5)
	5	94.7 (0.2)	95.3 (0.2)	89.9 (0.2)	93.2 (0.2)	90.7 (0.2)	94.0 (0.2)	93.2 (0.5)	85.9 (0.3)	83.6 (0.7)
	10	94.8 (0.2)	95.7 (0.1)	90.0 (0.6)	93.8 (0.1)	91.0 (0.2)	93.9 (0.3)	93.4 (0.3)	86.0 (0.4)	84.9 (0.3)
	50	95.0 (0.2)	96.0 (0.2)	90.4 (0.4)	94.0 (0.3)	91.4 (0.2)	93.6 (0.2)	93.0 (0.1)	87.2 (0.3)	85.2 (0.6)
model_L multilingual	1	94.8 (0.3)	95.9 (0.2)	89.7 (0.6)	93.8 (0.3)	91.2 (0.4)	93.6 (0.5)	93.3 (0.3)	85.7 (0.9)	82.8 (1.3)
	5	94.9 (0.2)	95.9 (0.2)	90.0 (0.5)	93.9 (0.3)	91.3 (0.4)	93.7 (0.4)	93.3 (0.3)	86.0 (0.8)	83.4 (1.0)
	10	94.9 (0.2)	95.9 (0.2)	90.1 (0.5)	93.9 (0.3)	91.3 (0.4)	93.7 (0.4)	93.3 (0.3)	86.3 (0.7)	83.6 (0.9)
model_L + rnd classifier multilingual	1	93.1 (0.5)	93.7 (0.5)	87.9 (0.5)	91.1 (0.5)	88.5 (0.6)	92.6 (0.5)	92.3 (0.6)	83.4 (0.8)	80.8 (1.3)
	5	94.8 (0.2)	95.8 (0.2)	89.9 (0.5)	93.6 (0.3)	91.1 (0.4)	93.7 (0.4)	93.3 (0.3)	86.3 (0.6)	84.1 (0.8)
	10	94.9 (0.2)	95.9 (0.2)	90.0 (0.5)	93.9 (0.3)	91.2 (0.4)	93.8 (0.4)	93.3 (0.3)	86.5 (0.5)	84.2 (0.8)
model_0 monolingual (*i.i.d.*)	50	94.4 (0.2)	95.6 (0.1)	88.9 (0.4)	93.2 (0.1)	90.3 (0.6)	93.3 (0.4)	93.1 (0.4)	82.4 (0.5)	71.3 (0.9)
model_L monolingual	1	95.1 (0.2)	95.8 (0.2)	90.2 (0.4)	93.6 (0.4)	91.2 (0.4)	93.5 (0.5)	93.4 (0.2)	86.3 (0.6)	79.1 (1.5)
	5	95.0 (0.2)	95.8 (0.2)	90.0 (0.4)	94.0 (0.2)	91.3 (0.2)	93.8 (0.4)	93.4 (0.2)	86.7 (0.4)	81.6 (0.8)
	10	95.1 (0.2)	95.8 (0.2)	90.0 (0.5)	93.9 (0.3)	91.3 (0.4)	93.8 (0.4)	93.4 (0.2)	86.7 (0.4)	82.2 (0.9)

Table 3: Slot F1 performance on *test*$_i$ sets for fast recovery experiments. model_L monolingual performance is averaged over 3 sequences (the P_{1L} experiment ones starting with the language in question), while model_L multilingual is averaged over all 27 sequences from P_{1L} experiments. Both model_0 and model_L experiments are averaged over 5 runs (standard deviation in parenthesis).

and model_L multilingual shows two interesting results. On one hand, we observe that even one epoch of multilingual training for model_L achieves better performance than the monolingual baseline (model_0 monolingual) and is even close to the multilingual topline (model_0 multilingual), both of which are trained on 50 epochs. This means that model_L is capable of achieving good multilingual performance with very little training, hence canceling the effect of forgetting. On the other hand, we see that model_L multilingual performance is greatly superior to model_0 multilingual with a single training epoch. This is not surprising given that the classifier is initialized randomly in model_0, but it shows that the model is capable of retaining knowledge from previous languages, although it is not clear whether that knowledge is preserved in the classifier or in BERT.

We dive deeper into this question by training model_L with a random classifier in the same manner (see model_L + rnd classifier multilingual in Table 3). We observe that performance is still greatly superior to model_0 multilingual with a single epoch, although not as high as model_L multilingual, which keeps its continually trained classifier. This indicates that knowledge retained from previous languages is in fact shared between BERT and the classifier, although judging by the performance gap it would seem that BERT stores most of it.

Overall, these results lead us to think that for the ATIS slot filling task, continual training over the language sequence does indeed shift model parameters to a better multilingual initialization. As a result, we explore the possibility to leverage this phenomenon in order to quickly recover lost language specificities due to forgetting. To do this, we train model_L on the first language of the sequence a second time (*i.e.* as if it were an $(L+1)^{\text{th}}$ language). As shown in Table 3, when comparing model_L monolingual to model_0 monolingual (equal to first language performance P_{11}), we see that the performance of the first language can be recovered and improved upon with as little as a single training epoch, even achieving 50-epoch model_0 multilingual performance in most cases. Moreover, languages that do not achieve this topline performance still show a big improvement. In particular, Hindi and Turkish improve an absolute 3.9% and 7.8% from model_0 monolingual respectively.

Note that increasing the number of recovery epochs for the first language does not bring considerable improvements. The only exception to this observation is Turkish, which might be explained by the small size of its training set. Although the cost of adding a language remains $O(N)$, the ability to recover all languages raises costs to $O(LN)$, making it expensive to use in practice. The design of a strategy taking full advantage of these recovery capabilities to limit forgetting with lower cost is left for future work.

8 Discussion

To summarize, we observe a high level of cross-lingual transfer in the *i.i.d.* setting when learning the ATIS slot filling task on all languages jointly. In a real low resource scenario where data and annotations are scarce, it may be difficult or even impossible to implement either a monolingual or multilingual adaptive approach, as time/space complexity is high and not all languages might be available at once. In a continual learning setting where languages are learned in sequence, these costs are the lowest and cross-lingual transfer is retained in the form of forward transfer. However, although performance loss in previous languages is not catastrophic, it is sufficient to consistently drop below monolingual.

When looking at continual cross-lingual transfer across the entire sequence, we obtain two surprising results. First, commonly used continual transfer metrics may not be a reliable estimate of the performance distribution across languages when transfer is not evenly distributed. Since even in other adaptation axes a considerable variability across datasets is to be expected, we believe a statistic like the median might be a better choice, as we believe it better represents expected performance at any given point. Second, as the sequence progresses, forward transfer improves, while backward transfer diminishes. This might indicate that model parameters remain a good initialization for future languages but that previous language specificities might be lost.

Motivated by this hypothesis, we compare the model at the beginning and at the end of the training sequence. Our results suggest that the model may indeed shift towards a better multilingual initialization, which makes it suitable to quickly recover the performance lost as a result of forgetting. We then measure the recovery capabilities of the model with respect to the first language of the sequence. We empirically show that lost performance can not only be recovered, but greatly improved with as little as a single training epoch, most languages even achieving *i.i.d.* multilingual performance.

In light of the above, we believe that effective continual learning methods for this task would benefit from leveraging recovery capabilities (either for a single language or many languages jointly) to limit the effect of forgetting, while preserving or even boosting forward transfer.

9 Conclusion

In this paper, we presented an analysis of cross-lingual transfer in continual learning for the slot filling task using multilingual BERT (Devlin et al., 2019) and MultiATIS++ (Xu et al., 2020).

Our main finding suggests that although forgetting is present, cross-lingual transfer is retained in the form of forward transfer, which allows the model to have substantial recovery capabilities. Moreover, we empirically show that this may be caused by a progressive shift of model parameters towards a better multilingual initialization. Finally, we also find that current continual learning metrics may need to be adapted if we want to better estimate the distribution of transfer across the adaptation axis.

As future work, we would like to reduce training costs by leveraging fast recovery for continual learning across languages. Another interesting research direction would be a study on the continual acquisition of languages not already present in multilingual BERT.

Reproducible Research

In the spirit of reproducible research, we will release our code as open source upon publication.

References

Hervé Abdi. 2007. The Kendall rank correlation coefficient. *Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA*, pages 508–510.

Gaurav Arora, Afshin Rahimi, and Timothy Baldwin. 2019. Does an LSTM forget more than a CNN? an empirical study of catastrophic forgetting in NLP. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 77–86, Sydney, Australia. Australasian Language Technology Association.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

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.

Quynh Do, Judith Gaspers, Tobias Roeding, and Melanie Bradford. 2020. To what degree can language borders be blurred in BERT-based multilingual spoken language understanding? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2699–2709, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Robert M. French. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128 – 135.

Raia Hadsell, Dushyant Rao, Andrei A. Rusu, and Razvan Pascanu. 2020. Embracing Change: Continual Learning in Deep Neural Networks. *Trends in Cognitive Sciences*, 24:1028–1040.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-Lingual Ability of Multilingual BERT: An Empirical Study. In *International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *InternationalConference on Learning Representations (ICLR)*.

Sungjin Lee. 2017. Toward continual learning for conversational agents. *arXiv preprint arXiv:1712.09943*.

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*, volume 30. Curran Associates, Inc.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, and Zhiguang Wang. 2020. Continual learning in task-oriented dialogue systems. *arXiv preprint arXiv:2012.15504*.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively Multilingual Transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text Chunking using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*.

Anthony Robins. 1995. Catastrophic Forgetting, Rehearsal and Pseudorehearsal. *Connection Science*, 7(2):123–146.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. 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 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task Chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

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.

Zhichang Zhang, Zhenwen Zhang, Haoyuan Chen, and Zhiman Zhang. 2019. A joint learning framework with bert for spoken language understanding. *IEEE Access*, 7:168849–168858.