Limitations of Knowledge Distillation for Zero-shot Transfer Learning

Saleh Soltan
Amazon Alexa AI
ssoltan@amazon.com

Haidar Khan
Amazon Alexa AI
khhaida@amazon.com

Wael Hamza
Amazon Alexa AI
waelhamz@amazon.com

Abstract

Pretrained transformer-based encoders such as BERT have been demonstrated to achieve state-of-the-art performance on numerous NLP tasks. Despite their success, BERT style encoders are large in size and have high latency during inference (especially on CPU machines) which make them unappealing for many online applications. Recently introduced compression and distillation methods have provided effective ways to alleviate this shortcoming. However, the focus of these works has been mainly on monolingual encoders. Motivated by recent successes in zero-shot cross-lingual transfer learning using multilingual pretrained encoders such as mBERT, we evaluate the effectiveness of Knowledge Distillation (KD) both during pretraining stage and during fine-tuning stage on multilingual BERT models. We demonstrate that in contradiction to the previous observation in the case of monolingual distillation, in multilingual settings, distillation during pretraining is more effective than distillation during fine-tuning for zero-shot transfer learning. Moreover, we observe that distillation during fine-tuning may hurt zero-shot cross-lingual performance. Finally, we demonstrate that distilling a larger model (BERT Large) results in the strongest distilled model that performs best both on the source language as well as target languages in zero-shot settings.

1 Introduction

Since the introduction of pretrained word embeddings (Mikolov et al., 2013b; Pennington et al., 2014), unsupervised representation learning has become an essential part of Natural Language Processing (NLP). The idea of exploiting unlabeled corpuses to improve performance on downstream NLP tasks has been excelled in recent years by the introduction of contextualized representation learning techniques (Peters et al., 2018) using the Language Modeling (LM) objective. Follow-up works utilized transformer-based architecture (Vaswani et al., 2017) and subword tokenization techniques (Sennrich et al., 2016) to further improve these contextualized representations and achieve state-of-the-art performance on various NLP tasks when fine-tuned on labeled data (Radford et al., 2018; Devlin et al., 2019).

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) has also been successfully pretrained on corpuses from 100+ languages to learn a shared representation among different languages (a.k.a mBERT). mBERT has been demonstrated to be capable of performing cross-lingual zero-shot transfer learning given the labeled data in a single language (e.g., English) without ever seeing a parallel corpus during pretraining (Pires et al., 2019). Moreover, it has been demonstrated that by increasing the size of the encoder and the pretaining corpus, such a multilingual encoder can achieve state-of-the-art cross-lingual zero-shot performance across many tasks (Conneau et al., 2020; Mukherjee and Awadallah, 2020).

Despite their success in many NLP task, BERT encoders are not very attractive for practical applications due to their huge size (often a few Gbs) and their latency (especially on CPU machines). Hence, there has been a great effort in recent years to compress these encoders without a big drop in performance using knowledge distillation (Hinton et al., 2015), network pruning (Li et al., 2020b), and quantization (Zafrir et al., 2019). However, there has been little work on compressing multilingual BERT encoders and evaluating their effectiveness in maintaining their cross-lingual learning capabilities after compression, which is the main focus of our work.

In this work, we focus on Knowledge Distillation (KD) which is an effective approach for reducing both size and latency of a BERT style encoder. In particular, we first pretrain a multilingual BERT base and a multilingual BERT large encoders on

Wikipedia data of 8 languages (namely mBERT8 and mBERT8$_{\text{Large}}$) and follow (Jiao et al., 2019) to distill them into 4-layer encoders, namely mLeanBERT and mXLeanBERT, respectively (see Table 1). We then compare the zero-shot cross-lingual performance of the student to teacher models on 5 datasets including XNLI, PAWS-X, Wikiann, MultiATIS++, and mTOP (see Table 3) when the models are fine-tuned only on English training data. *We show that mLeanBERT performance degrades by 2-9% on English test sets but by 6-25% on non-English ones (zero-shot) compared to mBERT8. However, mLeanBERT is 0.25-10% better than a similar sized model pretrained with no KD (Vanilla mLeanBERT) on zero-shot performance. Moreover, our results demonstrate that mXLeanBERT (distilled from mBERT8$_{Large}$) performs better than Vanilla mLeanBERT by up to 18% in zero-shot.*

Finally, we study the effectiveness of distillation during fine-tuning and demonstrate that in contrast to previous observations in monolingual settings (Jiao et al., 2019), it can hurt zero-shot cross-lingual performance. *In particular, extra distillation during fine-tuning degrades zero-shot cross-lingual performance of mLeanBERT by up to 2.5% despite improving or not changing its performance on English test set.*

The main contributions of this work are:

- Evaluating state-of-the-art knowledge distillation method on pretrained multilingual BERT Base and Large models and demonstrating its effectiveness (when done during pretraining) in multilingual settings.

- Demonstrating that in contradiction to monolingual settings, knowledge distillation during fine-tuning mostly hurts zero-shot cross-lingual transfer. This argument holds even if knowledge distillation during fine-tuning is the only distillation step (not as a second stage distillation step).

- Pretraining a multilingual BERT large model and utilizing it as the teacher during knowledge distillation to demonstrate that a larger model as teacher results in a stronger distilled model performing better for zero-shot cross-lingual transfer.

2 Related Work

2.1 Cross-lingual Transfer Learning

Transferring a task specific knowledge between different languages has always been an intriguing subject in NLP. Such a transfer is demonstrated to be possible by aligning word embeddings using a limited dictionary of words (Mikolov et al., 2013a; Schuster et al., 2019) or even without a dictionary (Lample et al., 2018). Aligning sentence representations is also demonstrated to be effective for cross-lingual transfer for sentence level tasks such as Natural Langual Inference (NLI) (Conneau et al., 2018).

Despite the success of alignment methods, sharing an encoder for multiple languages is demonstrated to be a more effective way of cross-lingual transfer learning. In particular, (Artetxe and Schwenk, 2019) showed that by sharing an encoder to learn sentence representations for multiple languages for Machine Translation (MT) can outperform previous methods for sentence level tasks. However, BERT-style pretraining tasks (Devlin et al., 2019) on mixed corpus from different languages is shown to be as effective as MT task in cross-lingual transfer without access to any parallel data (Siddhant et al., 2020). Moreover, recent work by (Conneau et al., 2020) demonstrated that by using larger encoders and larger pretraining corpus, BERT-style pretrained encoders (i.e., XLM-R) can outperform all previous methods on various NLP tasks.

Another recently proposed approach for cross-lingual transfer mixes the word alignment approach with encoder sharing to expand a monolingual BERT model to a new language and allow transfer learning (Artetxe et al., 2020; Tran, 2020). The approach is shown to be as effective as joint pretraining on mixed corpus and mixed vocabulary.

Finally, when available, (Lauscher et al., 2020) demonstrated that using a few target language training examples can significantly improve cross-lingual performance of pretrained multilingual models.

2.2 Model Compression and Distillation

Knowledge distillation is widely used to transfer knowledge from large (teacher) models into small (student) models (Ba and Caruana, 2014; Hinton et al., 2015). This type of compression has a number of benefits, including reducing model size/latency, semi-supervised learning, and con-

Model	KD?	#lgs	tokenization	L	H_m	H_{ff}	A	V	#params*
mBERT8$_{\text{Large}}$	no	8	BPE	24	1024	4096	16	120k	302M
mBERT8	no	8	BPE	12	768	3072	12	120k	85M
mBERT	no	104	WordPiece	12	768	3072	12	110k	85M
mXLeanBERT	yes	8	BPE	4	768	1200	12	120k	17M
mLeanBERT	yes	8	BPE	4	768	1200	12	120k	17M
Vanilla mLeanBERT	no	8	BPE	4	768	1200	12	120k	17M

Table 1: List of models. L, H_m, H_{ff}, A and V denote the number of layers, transformer hidden dimension, feed-forward layers' hidden dimension, number of attention heads, and the vocabulary size, respectively. mLeanBERT and mXLeanBERT are distilled (1st stage distillation) from mBERT8 and mBERT8$_{\text{Large}}$, respectively. Vanilla mLeanBERT, however, is pretrained normally with no teachers. The same multilingual data is used to pretrain and distill all the models. *Number of parameters excluding the embeddings.

tinual learning. Previous work has demonstrated successful compression of teachers into students of the same type (Kim and Rush, 2016; Sanh et al., 2019) as well as into students with different model types and architectures (Tang et al., 2019). Depending on the approach, the limits of compression, and the ideal initialization procedure differ substantially (Sanh et al., 2019; Yang et al., 2019b; Turc et al., 2019). The resulting student models are shown to preserve many of the attractive qualities of the teacher model and perform better than training the same models without knowledge distillation. Some distillation approaches jointly train the teacher and student, such as in (Jin et al., 2019), however we consider the common case of a "fixed" teacher at distillation time in this work.

The most common approach to distillation is output matching between the teacher and the student, where the student is trained to minimize two losses during training; the standard cross-entropy loss and the cross-entropy loss between the teacher output and the student output (Hinton et al., 2015). There exists variations of this approach, such as adding loss functions leveraging cues from internal representations of the teacher (Jiao et al., 2019; Sun et al., 2019; Aguilar et al., 2020; Romero et al., 2014).

In the context of pretrained models, distillation can be executed directly after teacher pretraining as in (Sanh et al., 2019), after teacher fine-tuning, or both (Jiao et al., 2019). In an ablation study, (Jiao et al., 2019) found that distillation after teacher finetuning (2nd stage) is more important than distillation after teacher pretraining (1st stage). However, the relative importance of the two stages of distillation remains an open question for distilling multilingual models.

Distillation of multilingual models for NER is explored in (Mukherjee and Awadallah, 2020) in the setting of 2nd stage distillation, i.e. distillation after finetuning, while 1st stage distillation is not considered. In addition, (Mukherjee and Awadallah, 2020) did not investigate the zero-shot performance of the distilled models. In a similar vein, (Tan et al., 2019; Wang et al., 2020b; Reimers and Gurevych, 2020) distill language-specific models into a combined multilingual machine translation model. In a recent parallel work to ours, (Wang et al., 2020a) introduced a new distillation method and demonstrated its effectiveness in multilingual settings. However, they focused on distillation during pretraining and evaluate their model only on XNLI from XTREME benchmarks. Since they used common crawl data for pretraining and a different architecture, their models are not directly comparable to ours.

3 Method

3.1 Pretraining

We pretrain two multilingual BERT models, namely *mBERT8$_{Base}$* and *mBERT8$_{Large}$*, on 8 languages including English, French, German, Japanese, Hindi, Italian, Spanish and Portuguese. For pretraining, we use Wikipedia data of the corresponding languages. For English, we also add Google Billion Words Corpus (Chelba et al., 2014) to the mix. All data are tokenized using a tokenizer that transforms written format to spoken format (e.g., removes casing and punctuation).

We mix and upsample the data following (Conneau and Lample, 2019). In particular, we sample 500M sentences according to a multinomial distri-

Model	perplexity	#epochs
mBERT8$_{\text{Large}}$	5.89	2
mBERT8	6.78	4
mXLeanBERT	9.76	4
mLeanBERT	10.18	4
Vanilla mLeanBERT	9.94	4

Table 2: Models' perplexity on the validation set (comprised of all languages) after pretraining. Epochs denotes the total number of epochs used for pretraining each model.

bution with probabilities $(q_1, q_2, \ldots, q_N)$, where:

$$q_i = \frac{p_i^\alpha}{\sum_{j=1}^{N} p_j^\alpha},\ p_i = \frac{n_i}{\sum_{j=1}^{N} n_j} \quad (1)$$

in which N is the total number of languages and n_i is the total number of sentences in language i (we set $\alpha = 0.5$). Byte Per Encoding (BPE) codes (Sennrich et al., 2016) of size 120k are then learnt from the mixed training set for subword tokenization. For pretraining task, we only use Masked Language Modeling (MLM) (Devlin et al., 2019) without any sentence-level tasks, similar to RoBERTa (Liu et al., Jul. 2019). However, we limit the length of sentences to 256 characters to speedup the pretraining. We pretrain mBERT8$_{\text{Base}}$ and mBERT8$_{\text{Large}}$ for 4 and 2 epochs, respectively. The perplexities (MLM perpelxity) on the validation set after the pretrainings are provided in Table 2.

3.2 Knowledge Distillation (KD)

We distill the pretrained mBERT8 and mBERT8$_{\text{Large}}$ models into the *LeanBERT* architecture, namely a 4 layer, 768 unit, 1200 hidden unit, and 12 attention head transformer. We differentiate between distillation during pretraining of the student model (1st stage), where the learning task is the unsupervised MLM task using dataset described in Section 3.1, and distillation during fine-tuning (2nd stage distillation), where the task is sentence/token level classification or sequence-to-sequence generation.

For 1st stage and 2nd stage distillation, we train the student model with a combination of distillation losses in addition to the standard cross-entropy loss. We follow the distillation losses proposed in (Jiao et al., 2019) that comprise of output/logit based loss functions and intermediate representation loss functions. At the output level, we penalize the student model by the cross-entropy between the student output predictions and the teacher output predictions. Given the student output predictions $\mathbf{z}_s$ and the teacher output predictions $\mathbf{z}_t$, the output layer distillation loss is:

$$\mathcal{L}_{output} = -\mathbf{z}_t \cdot \log \mathbf{z}_s. \quad (2)$$

For intermediate representations, the student model is trained to minimize the L_2 distance to the teacher's corresponding intermediate representation. When the number of layers or units in the teacher and student model do not match, this requires establishing a mapping $g(m)$ between the teacher/student layers and a projection $f(\mathbf{x})$ to match number of units. Given the m-th teacher intermediate representation $\mathbf{x}_{t,m}$ and the corresponding student intermediate representation $\mathbf{x}_{s,g(m)}$, the intermediate layer distillation loss is:

$$\mathcal{L}_{inter} = ||\mathbf{x}_{t,m} - f(\mathbf{x}_{s,g(m)})||_2 \quad (3)$$

Note that this loss function is applied to three types of intermediate representations: embeddings, transformer outputs, and self-attention scores. In our experiments, we adopt the same mapping $g(m)$ as (Jiao et al., 2019), namely that the student model learns from every three layers of the teacher model.

The combinations of these loss functions can be applied to distill the pretrained teacher for 1st stage distillation and to distill the pretrained+fine-tuned teacher for 2nd stage distillation (we refer to this as **ft-distill**).

We refer to the models distilled from mBERT8 and mBERT8$_{\text{Large}}$ during pretraining (1st stage distillation) as *mLeanBERT* and *mXLeanBERT*, respectively (see Table 1). We used all the data as described in Section 3.1 during distillation. For comparison purposes, we also pretrained a *Vanilla mLeanBERT* model with no distillation by following the steps described in the previous subsection. The perplexity of these models after pretraining can be seen in Table 2.

It can be seen that mXLeanBERT which has been distilled from the mBERT8$_{\text{Large}}$ results in the lowest perplexity between the 3 models. However, the Vanilla mLeanBERT results in a lower perplexity compared to mLeanBERT which is distilled from mBERT8 model. *As we demonstrate in the next section, this lower perplexity is not a good predictor for better cross-lingual capability of the Vanilla mLeanBERT model.*

Corpus	Task	|Train|	|Dev|	|Test|	Metric
XNLI	Classification	392,702	2,490	5,010	Acc.
PAWS-X	Classification	49,401	2,000	2,000	Acc.
Wikiann	NER	19,999	9,999	999-9,999	F1
MultiATIS++	IC/NER	4,488	490	∼900	Exact Match
mTOP	Semantic Parsing	15,667	1,527	∼3,000	Exact Match

Table 3: Characteristics of the datasets.

Model	XNLI	PAWS-X	Wikiann	MultiATIS++	mTOP
$\text{mBERT8}_{\text{Large}}$	82.49	92.73	84.21	87.53	82.24
mBERT8	81.67	91.68	83.46	87.98	82.09
mBERT	81.36	93.4	84.85	88.12	81.8
Vanilla mLeanBERT	77.47	85.22	80.47	87.34	80.26
KD during pretraining (1st stage)					
mLeanBERT	77.23	84.21	79.92	86.59	79.85
mXLeanBERT	**78.91**	**88.58**	**80.76**	**87.42**	**81.24**
KD during fine-tuning (2nd stage)					
mLeanBERT (ft-distill)	77.13	87.93	80.28	86.17	79.77
Vanilla mLeanBERT (ft-distill)	77.68	85.73	80.63	86.75	80.44

Table 4: Results on English test set (averaged over 4 seed runs). All models are fine-tuned only on English training data. Bold numbers denote the best score across all LeanBERT sized models.

4 Experimental Results

4.1 Data

We evaluate our models on multiple tasks (summarized in Table 3). We limited our experimentation to these datasets due to the sheer number of required experimentation (9 models, 5 task, and 4 seeds).

XNLI is a Cross-Lingual Natural Language Inference task that asks whether a premise sentence entails, contradicts, or is neutral toward a hypothesis (Conneau et al., 2018).

PAWS-X requires the model to determine whether two sentences are paraphrase of each other (Zhang et al., 2019; Yang et al., 2019a). We obtained the data using XTREME repository code (Hu et al., 2020). However, we downloaded the test sets directly.[1]

Wikiann is a Name Entity Recognition dataset (NER). The data is extracted from Wikipedia in which name entities are tagged as `LOC`, `PER`, and `ORG` (Pan et al., 2017). As in PAWS-X, we obtained the data through XTREME repository but downloaded test sets directly (Pan et al., 2017).

MultiATIS++ is a Natural Language Understanding (NLU) dataset that requires the model to detect the *intent* of an utterance as well as the related *slots*. It is an extension to MultiATIS dataset (Upadhyay et al., 2018) to Spanish, German, Chinese, Japanese, Portuguese, and French languages in addition to English, Hindi, and Turkish (Xu et al., 2020).[2]

mTOP is a multilingual Task Oriented Parsing dataset that covers English, German, French, Spanish, Hindi, and Thai languages (Li et al., 2020a).

4.2 Models

We consider models in Table 1 as introduced in Section 3 and fine-tune them only on the English training data for the tasks described in Section 4.1. Besides these models, we also consider distillation during fine-tuning for mLeanBERT and Vanilla mLeanBERT. In all distillations during fine-tuning,

[1] XTREME code does not download test set on purpose to keep them private for its leaderboard. Although, the test sets are available directly from their original source (Zhang et al., 2019; Yang et al., 2019a).

[2] We noticed that in the test sets of MultiATIS++ there are a few utterances for which number of tags do not match the number of tokens. We removed those instances from the test sets.

Model	XNLI	PAWS-X	Wikiann	MultiATIS++	mTOP
$\text{mBERT8}_{\text{Large}}$	72.61	85.69	77.88	46.25	37.41
mBERT8	70.31	83.47	76.73	45.42	34.60
mBERT	67.54	85.18	77.33	35.24	25.80
Vanilla mLeanBERT	64.53	74.06	71.30	34.06	23.74
KD during pretraining (1st stage)					
mLeanBERT	66.09	75.07	71.48	**37.58**	25.96
mXLeanBERT	**67.69**	**80.26**	71.17	37.05	**28.01**
KD during fine-tuning (2nd stage)					
mLeanBERT (ft-distill)	65.52	77.98	**71.62**	36.79	25.33
Vanilla mLeanBERT (ft-distill)	64.97	74.76	71.57	34.01	24.15

Table 5: Average zero-shot results across selected languages (averaged over 4 seed runs). All models are fine-tuned only on English training data. Bold numbers denote the best score across all LeanBERT sized models.

we use mBERT8 as the teacher. For these experiments, the teacher is fine-tuned on the English training data for the task and the same data is used for distillation.

4.3 Experimentations Setup

For all of the tasks except mTOP, we followed the original BERT paper (Devlin et al., 2019) and fine-tuned the encoder on the downstream task with additional output blocks. We also *kept the embedding layers frozen* to prevent the change only in subset of subwords that appear in the training data.

For mTOP, we followed (Rongali et al., 2020) and fine-tuned a seq2seq model with pointers which has been shown to achieve state-of-the-art results on TOP dataset (Gupta et al., 2018). In all experiments, we used the pretrained encoders but randomly initialized a 3-layer decoder with the same hidden dimension and number of heads as the encoder. The feedforward layers in the decoder have the dimension of 512. As in all of the other experiments, we kept the encoder embedding layers frozen during fine-tuning.

4.4 Key Findings

The results on the English test sets and the average zero-shot performance across different languages can be seen in Tables 4 and 5, respectively. Per language performance of different models on XNLI, PAWS-X, Wikiann, MutiATIS++, and mTOP can be seen in Tables 6, 7, 8, 9,10, respectively.

The following subsections describe the key findings from these results.

4.4.1 Stage 1 Distillation is Effective for Multilingual Models

It has been previously demonstrated by (Jiao et al., 2019) that stage 1 distillation is effective in improving performance in monolingual settings. We confirm this finding in our experiments by comparing mBERT8, mLeanBERT, and Vanilla mLeanBERT performances on English test sets in Table 4. Our new finding, however, is that as can be seen in Table 5, mLeanBERT outperforms Vanilla mLeanBERT in all the tasks in zero-shot setting as well. **In particular mLeanBERT outperforms Vanilla mLeanBERT on all the tasks in zero-shot setting (by more that 9% on MultiATIS++ and mTOP).**

Moreover, by comparing mBERT8 and mLeanBERT performances, we can see that despite being 5 times smaller, mLeanBERT performance degrades only by 5.74%, 8.87%, 4.43%, 1.6%, and 2.8% on English test set and by 6.00%, 10.06%, 6.84%, 17.26%, and 24.97% in zero-shot performance on XNLI, PAWS-X, Wikiann, MultiATIS++, and mTOP tasks, respectively. We observe that performance degradation in zero-shot performance is greater than on English which emphasizes the challenges in multilingual distillation.

4.4.2 Stage 2 Distillation Can Hurt Zero-shot Cross-lingual Performance

Another important observation from our experimental results is that in contrast to previous observations by (Jiao et al., 2019) in the monolingual case, 2nd stage distillation may hurt zero-shot cross-lingual performance. In particular, although as can be seen in Tables 4 and 5 ft-distill im-

Model	en	fr	de	es	hi
mBERT8$_{\text{Large}}$	82.49	75.71	73.48	76.19	65.07
mBERT8	81.67	73.32	70.70	74.94	62.28
mBERT	81.36	70.32	67.94	73.07	58.82
Vanilla mLeanBERT	77.47	67.80	65.01	68.91	56.4
KD during pretraining (1st stage)					
mLeanBERT	77.23	68.79	66.41	70.25	58.91
mXLeanBERT	**78.90**	**70.37**	**67.85**	**71.51**	**61.04**
KD during fine-tuning (2nd stage)					
mLeanBERT (ft-distill)	77.13	68.85	66.15	69.37	57.71
Vanilla mLeanBERT (ft-distill)	77.68	68.15	65.60	68.92	57.23

Table 6: XNLI results (accuracy) per language (averaged over 4 seed runs). All models are fine-tuned only on English training data. Bold numbers denote the best score across all LeanBERT sized models.

Model	en	fr	de	es	ja*
mBERT8$_{\text{Large}}$	92.95	86.15	85.65	85.85	66.8
mBERT8	91.55	84.8	83.65	85.05	62.1
mBERT	93.4	85.35	84.35	85.85	71.8
Vanilla mLeanBERT	82.9	74.43	73.21	74.55	59.18
KD during pretraining (1st stage)					
mLeanBERT	84.21	75.53	73.9	75.77	60.77
mXLeanBERT	**88.58**	**80.52**	**78.67**	**81.6**	**63**
KD during fine-tuning (2nd stage)					
mLeanBERT (ft-distill)	87.93	78.55	76.66	78.75	61.22
Vanilla mLeanBERT (ft-distill)	85.73	75.22	73.78	75.28	61.02

Table 7: PAWS-X results (accuracy) per language (averaged over 4 seed runs). All models are fine-tuned only on English training data. Bold numbers denote the best score across all LeanBERT sized models.*Results on Japanese test set are not included in the average zero-shot performance presented in Table 5. The main reason for that is that the tokenizer we used for Japanese is incompatible with PAWS-X data describing the degradation in accuracy compared to the public mBERT.

Model	en	fr	de	es	pt	it	hi
mBERT8$_{\text{Large}}$	84.21	79.65	79.06	75.20	80.01	79.82	73.56
mBERT8	83.47	78.06	78.48	76.90	79.37	78.79	68.77
mBERT	84.85	80	78.18	74.17	81.42	80.89	61.81
Vanilla mLeanBERT	80.47	**73.46**	73.47	73.81	73.82	74.38	58.9
KD during pretraining (1st stage)							
mLeanBERT	79.92	73.33	72.13	70.58	75.02	72.98	64.89
mXLeanBERT	**80.76**	72.24	73.00	69.23	**74.54**	74.17	63.87
KD during fine-tuning (2nd stage)							
mLeanBERT (ft-distill)	80.28	73.34	72.15	70.64	75.00	72.93	**65.67**
Vanilla mLeanBERT (ft-distill)	80.63	73.25	**73.90**	**73.98**	74.29	**74.81**	59.2

Table 8: Wikiann results (F1 scores) per language (averaged over 4 seed runs). All models are fine-tuned only on English training data. Bold numbers denote the best score across all LeanBERT sized models.

Model	en	fr	de	es	pt	hi
$\text{mBERT8}_{\text{Large}}$	87.54	62.04	61.97	53.29	45.59	8.42
mBERT8	87.99	59.98	59.77	51.32	41.84	14.18
mBERT	88.12	44.37	35.87	43.78	35.76	16.46
Vanilla mLeanBERT	87.35	45.23	45.10	41.79	37.5	3.39
KD during pretraining (1st stage)						
mLeanBERT	86.59	48.13	**53**	39.73	**41.36**	5.71
mXLeanBERT	**87.43**	**52.81**	41.14	**43.53**	41.20	**6.58**
KD during fine-tuning (2nd stage)						
mLeanBERT (ft-distill)	86.17	47.09	51.97	38.75	40.5	5.62
Vanilla mLeanBERT (ft-distill)	86.76	44.16	44.70	40.56	37.70	2.97

Table 9: MultiATIS++ results (Exact Match) per language (averaged over 4 seed runs). All models are fine-tuned only on English training data. Bold numbers denote the best score across all LeanBERT sized models.

Model	en	fr	de	es	hi
$\text{mBERT8}_{\text{Large}}$	82.25	47.34	40.80	43.12	18.42
mBERT8	82.10	46.98	36.07	39.72	15.65
mBERT	81.8	34.85	28.43	34.75	5.2
Vanilla mLeanBERT	80.26	33.34	24.32	31.98	5.36
KD during pretraining (1st stage)					
mLeanBERT	79.85	34.95	26.59	**33.48**	8.86
mXLeanBERT	**81.24**	**37.85**	**30.43**	33.15	**10.61**
KD during fine-tuning (2nd stage))					
mLeanBERT (ft-distill)	79.77	33.71	26.99	32.4	8.25
Vanilla mLeanBERT (ft-distill)	80.45	33.50	24.79	33.09	5.24

Table 10: mTOP results (Exact Match) per language (averaged over 4 seed runs). All models are fine-tuned only on English training data. Bold numbers denote the best score across all LeanBERT sized models.

proves Vanilla LeanBERT performance (except on MultiATIS++), **ft-distill *degrades* zero-shot performance of mLeanBERT by** 0.86%**,** 2.1%**, and** 2.43% **on XNLI, MultiATIS++, and mTOP, respectively (see Table 5).**

These mixed results demonstrate that 2nd stage fine-tuning cannot be reliably used to improve cross-lingual performance of multilingual models. We hypothesize that 2nd stage fine-tuning makes multilingual models to overfit to the source language resulting in degradation in cross-lingual performance (i.e., catastrophic forgetting).

4.4.3 Distilling a Larger Model Results in a Stronger Distilled Model

Finally, as has been previously observed by (Li et al., 2020b) in the network pruning context, compressing larger models to the same size results in stronger compressed models. Our results demonstrate the same performance gain both on the English test set and in the zero-shot setting when distilling from $\text{BERT8}_{\text{Large}}$.

As can be seen in Table 4, **mXLeanBERT performs better than mLeanBERT on all the tasks on English test sets (outperforming it by** 5.2% **on PAWS-X). For zero-shot cross-lingual performance, we also observe the same trend in Table 5. This time mXLeanBERT outperforms mLeanBERT by** 6.46% **on zero-shot cross-lingual performance on PAWS-X and by** 7.31% **on mTOP.** On Wikiann and MultiATIS++ the two models provide a very similar performance (which can be due to instability of exact match metric for NER).

Based on the three observations, we observe that when it comes to distilling multilingual models, pretraining a larger model and then distilling it results in the best distilled multilingual model.

5 Conclusion

In this work, we applied state-of-the-art knowledge distillation methods on multilingual BERT encoders and demonstrated that distillation during pretraining is as effective in multilingual setting. However, in contradiction to previous observations in monolingual settings, we demonstrated that distillation during fine-tuning may degrade zero-shot cross-lingual performance. When it comes to multilingual distillation, we showed that pretraining a larger model and then distilling it results in the best performing distilled model.

References

Gustavo Aguilar, Y. Ling, Y. Zhang, B. Yao, Xing Fan, and Edward Guo. 2020. Knowledge distillation from internal representations. In *Proc. AAAI'20*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proc. ACL'20*.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *TACL*, 7:597–610.

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *Proc. NeurIPS'14*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *Proc. Interspeech'14*.

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 *Proc. ACL'20*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Proc. NeurIPS'19*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proc. EMNLP'18*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL'19*.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proc. EMNLP'18*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv:1503.02531*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv:2003.11080*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv:1909.10351*.

Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. 2019. Knowledge distillation via route constrained optimization. In *Proc. IEEE ICCV'19*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proc. EMNLP'16*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *Proc. ICLR'18*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proc. EMNLP'20*.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2020a. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. *arXiv:2008.09335*.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joseph E Gonzalez. 2020b. Train large, then compress: Rethinking model size for efficient training and inference of transformers. *arXiv:2002.11794*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Jul. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv:1309.4168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proc. NeurIPS'13*.

Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. Xtremedistil: Multi-stage distillation for massive multilingual models. In *Proc. ACL'20*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proc. ACL'17*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proc. EMNLP'14*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. NAACL-HLT'18*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proc. ACL'19*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv:2004.09813*.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv:1412.6550*.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. In *Proc. WWW'20*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv:1910.01108*.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proc. NACL'19*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. ACL'16*.

Aditya Siddhant, Melvin Johnson, Henry Tsai, Naveen Ari, Jason Riesa, Ankur Bapna, Orhan Firat, and Karthik Raman. 2020. Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. In *Proc. AAAI'20*.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. *arXiv:1908.09355*.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *Proc. ICLR'19*.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv:1903.12136*.

Ke Tran. 2020. From english to foreign languages: Transferring pre-trained language models. *arXiv:2002.07306*.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv:1908.08962*.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. 2018. (Almost) zero-shot cross-lingual spoken language understanding. In *Proc. IEEE ICASSP'18*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS'19*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020a. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proc. NeurIPS'20*.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Fei Huang, and Kewei Tu. 2020b. Structure-level knowledge distillation for multilingual sequence labeling. In *Proc. ACL'20*.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. *arXiv:2004.14353*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019a. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proc. EMNLP-IJCNLP'19*.

Ze Yang, Linjun Shou, Ming Gong, Wutao Lin, and Daxin Jiang. 2019b. Model compression with multi-task knowledge distillation for web-scale question answering system. *arXiv:1904.09636*.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. *arXiv:1910.06188*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proc. NAACL'19*.