applied sciences

MDPI

Article

Exploring the Data Efficiency of Cross-Lingual Post-Training in Pretrained Language Models

Chanhee Lee [1], Kisu Yang [1], Taesun Whang [2], Chanjun Park [1], Andrew Matteson [3] and Heuiseok Lim [1,*]

[1] Department of Computer Science and Engineering, Korea University, 145, Anam-ro, Seongbuk-gu, Seoul 02841, Korea; chanhee0222@korea.ac.kr (C.L.); willow4@korea.ac.kr (K.Y.); bcj1210@korea.ac.kr (C.P.)
[2] Wisenut Inc., 49, Daewangpangyo-ro 644beon-gil, Bundang-gu, Seongnam-si, Gyeonggi-do 13493, Korea; taesunwhang@wisenut.co.kr
[3] Independent Researcher, Sujeong-gu, Seongnam-si, Gyeonggi-do 13106, Korea; xt.knight@gmail.com
* Correspondence: limhseok@korea.ac.kr

Abstract: Language model pretraining is an effective method for improving the performance of downstream natural language processing tasks. Even though language modeling is unsupervised and thus collecting data for it is relatively less expensive, it is still a challenging process for languages with limited resources. This results in great technological disparity between high- and low-resource languages for numerous downstream natural language processing tasks. In this paper, we aim to make this technology more accessible by enabling data efficient training of pretrained language models. It is achieved by formulating language modeling of low-resource languages as a domain adaptation task using transformer-based language models pretrained on corpora of high-resource languages. Our novel cross-lingual post-training approach selectively reuses parameters of the language model trained on a high-resource language and post-trains them while learning language-specific parameters in the low-resource language. We also propose implicit translation layers that can learn linguistic differences between languages at a sequence level. To evaluate our method, we post-train a RoBERTa model pretrained in English and conduct a case study for the Korean language. Quantitative results from intrinsic and extrinsic evaluations show that our method outperforms several massively multilingual and monolingual pretrained language models in most settings and improves the data efficiency by a factor of up to 32 compared to monolingual training.

Keywords: cross-lingual; pretraining; language model; transfer learning; deep learning; RoBERTa

Citation: Lee, C.; Yang, K.; Whang, T.; Park, C.; Matteson, A.; Lim, H. Exploring the Data Efficiency of Cross-Lingual Post-Training in Pretrained Language Models. *Appl. Sci.* **2021**, *11*, 1974. https://doi.org/10.3390/app11051974

Academic Editor: Rafael Valencia-Garcia

Received: 7 February 2021
Accepted: 19 February 2021
Published: 24 February 2021

1. Introduction

Bidirectional Encoder Representations from Transformers (BERT) [1] is a Transformer network [2] pretrained with a language modeling objective and a vast amount of raw text. BERT was able to obtain state-of-the-art performance in many challenging natural language understanding tasks by a sizable margin. Thus, BERTology has become one of the most influential and active research areas in Natural Language Processing (NLP). This led to the development of many improved architectures and training methodologies for Pretrained Language Models (PLMs), such as RoBERTa [3], ALBERT [4], BART [5], ELECTRA [6], and GPT [7,8], improving various NLP systems and even achieving superhuman performance [1,9,10].

The language modeling objective can be optimized via unsupervised training, requiring only a raw corpus without costly annotation. However, even among over 7000 languages spoken worldwide [11] , only a handful provide such raw corpora large enough for training. As Wikipedia is actively updated and uses mostly formal language, it serves as a reasonable resource for obtaining a raw corpus and is thus useful for measuring language resource availability. Figure 1 shows the number of documents per language in Wikipedia as of September 2020. These data were collected from 303 languages with at least 100 documents, which is only a fraction of all existing languages. Still, as can be seen

in Figure 1a, the number of documents generally follows the power law distribution where the majority of documents are from a handful of languages. Additionally, while there are more than 6 million documents in the most used language, 154 of the 303 languages have less than 10,000 documents. In Figure 1b, the percentage of the top 10 languages based on the number of documents is visualized. Here, more than the half of all documents are from these 10 languages, which also indicates that language resource availability remains highly imbalanced.

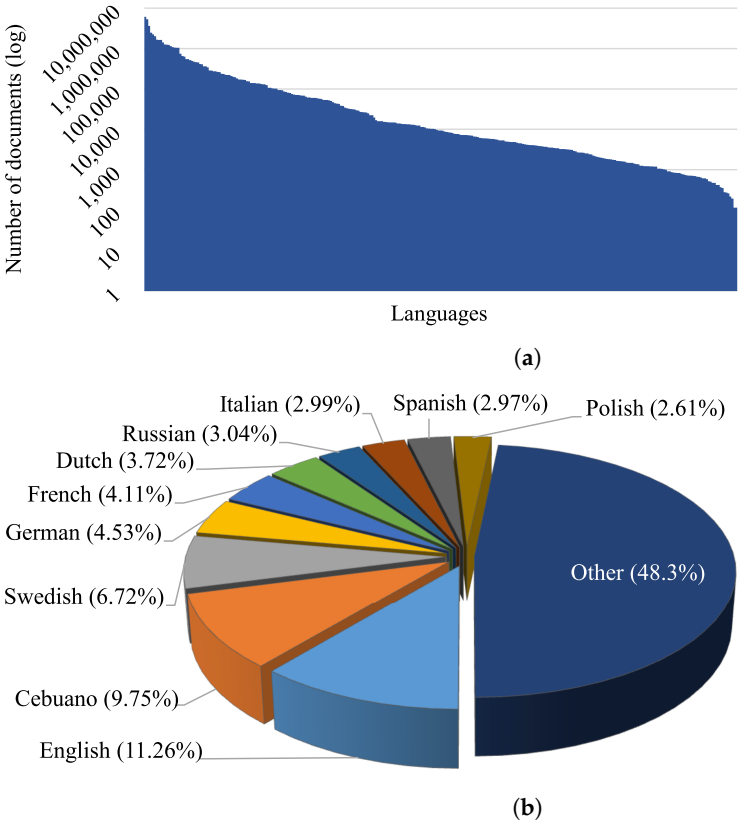


Figure 1. Multilingual statistics of Wikipedia. (**a**) Number of documents per language. The *y*-axis is in log scale and the *x*-axis includes 303 languages, sorted by the number of documents. (**b**) Top 10 languages with most documents and their respective percentages.

To tackle this problem, transfer learning via multilingual PLMs has been proposed [1,12,13]. In the multilingual PLM literature, transfer learning mostly focuses on zero-shot transfer, which assumes that there is no labeled data available in the target language. This is, however, unrealistic, as in real-world scenarios there are in fact labeled data available in many cases [14,15]. Furthermore, zero-shot scenarios force the model to maintain its performance in the source language, which might prevent the model from fully adapting to the target language. Multilingual PLMs are also much more costly to train due to increased data and model size [13]. The most practical solution would be to use a PLM trained in the source language and fully adapt it to the target language to perform supervised fine-tuning in that language. However, this approach is largely unexplored.

To overcome such limitations we propose cross-lingual post-training (XPT), which formulates language modeling as a pretraining and post-training problem. Starting from a monolingual PLM trained in a high-resource language, we fully adapt it to a low-resource language via unsupervised post-training, which is then fine-tuned in the target language. To aid in adaptation, we introduce Implicit Translation Layers (ITLs) which aim to learn linguistic differences between the two languages. To evaluate our proposed method, we conduct a case study for Korean, using English as the source language. We limit the target language to Korean for two reasons. First, Korean is a language isolate [16–18] that was shown to be challenging to transfer from English [19]. Second, by evaluating on a single

language we utilize linguistic characteristics as a control variable. This let us focus on data efficiency, the primary metric that we aim to measure.

Evaluating our method on four challenging Korean natural language understanding tasks, we find that cross-lingual post-training is extremely effective at increasing data efficiency. Compared to training a language model from scratch, data efficiency improvement of up to 32 times was observed. Further, XPT outperformed or matched the performance of publicly available massively multilingual and monolingual PLMs while utilizing only a fraction of the data used to train those models.

2. Related Work

Pretraining a neural network with some variant of the language modeling objective has become a popular trend in NLP due to its effectiveness. While not being the first, BERT [1] has arguably become the most successful in generating contextualized representations, leading to a new research field termed BERTology with hundreds of publications [20]. However, the success is largely centered around English and few other high-resource languages [21], limiting the use of this technology in most of the world's languages [14].

To overcome this limit, there has been focus on bringing these advancements to more languages by learning multilingual representations. In the case of token-level representations such as word2vec [22,23], this was achieved by aligning monolingual representations [24–26] or jointly training on multilingual data [27,28]. Aligning monolingual embeddings was also attempted in contextualized representations [29,30], but the most successful results were obtained from joint training. Initially, these joint models were trained using explicit supervision from sentence aligned data [31], but later it was discovered that merely training with a language modeling objective on a concatenation of raw corpora from multiple languages can yield multilingual representations [12,32]. This approach was later extended by incorporating more pretraining tasks [33,34] and even learning a hundred languages using a single model [13]. While these massively multilingual language models are effective at increasing the sample efficiency in low-resource languages, they are prohibitively expensive to train since the training cost increases linearly with the size of the data in use. Further, learning from many languages requires the model to have higher capacity [13]. This leads to difficulties when trying to adapt this method to more efficient and capable architectures or deploy to devices with limited computing resources.

The fact that mBERT [35] and XLM-R [13] learn multilingual representations without any explicit supervision has led to more research investigating their zero- and few-shot performance on various tasks. In [36], the authors concluded that the overlap in subword vocabulary between different languages plays an important role in acquiring multilinguality. On the other hand, it has also been reported that they even generalize to languages written in different scripts, thus having no such overlap [37], and when the overlap is intentionally removed [38]. Lauscher et al. [19] demonstrated that the performance of these models can be improved in a few-shot scenario with as low as 10 annotated sentences. UDapter [39] and MAD-X [40] improve data efficiency even further by limiting the parameter update to a small set of Adapter modules [41–43]. However, despite their strong zero- and few-shot performance, all approaches in this category inherit the same limitations of massively multilingual language models.

Training a bilingual language model by transferring from a monolingual one is a much cheaper alternative to multilingual pretraining, as most publications regarding pretrained Transformers publish a trained model checkpoint as well. Despite this advantage, it is a less explored approach. In [29], monolingual ELMo embeddings are aligned to a common space to perform zero-shot dependency parsing. MonoTrans [44] transfers English BERT to other languages by learning a new token embedding from scratch for each target language. Transformer encoder layers are frozen while learning the embeddings to prevent catastrophic forgetting. RAMEN [45] takes a similar approach, but initializes each target language embedding as a linear combination of English embeddings. These approaches

are the ones most close to ours, but limited in a sense that the model is forced to maintain its ability in the source language, restricting its adaptation to the target language.

3. Proposed Method

In this section, we describe our proposed XPT in detail. The overall process is illustrated in Figure 2a, alongside with the illustration of the multilingual pre-training and monolingual transfer learning approaches.

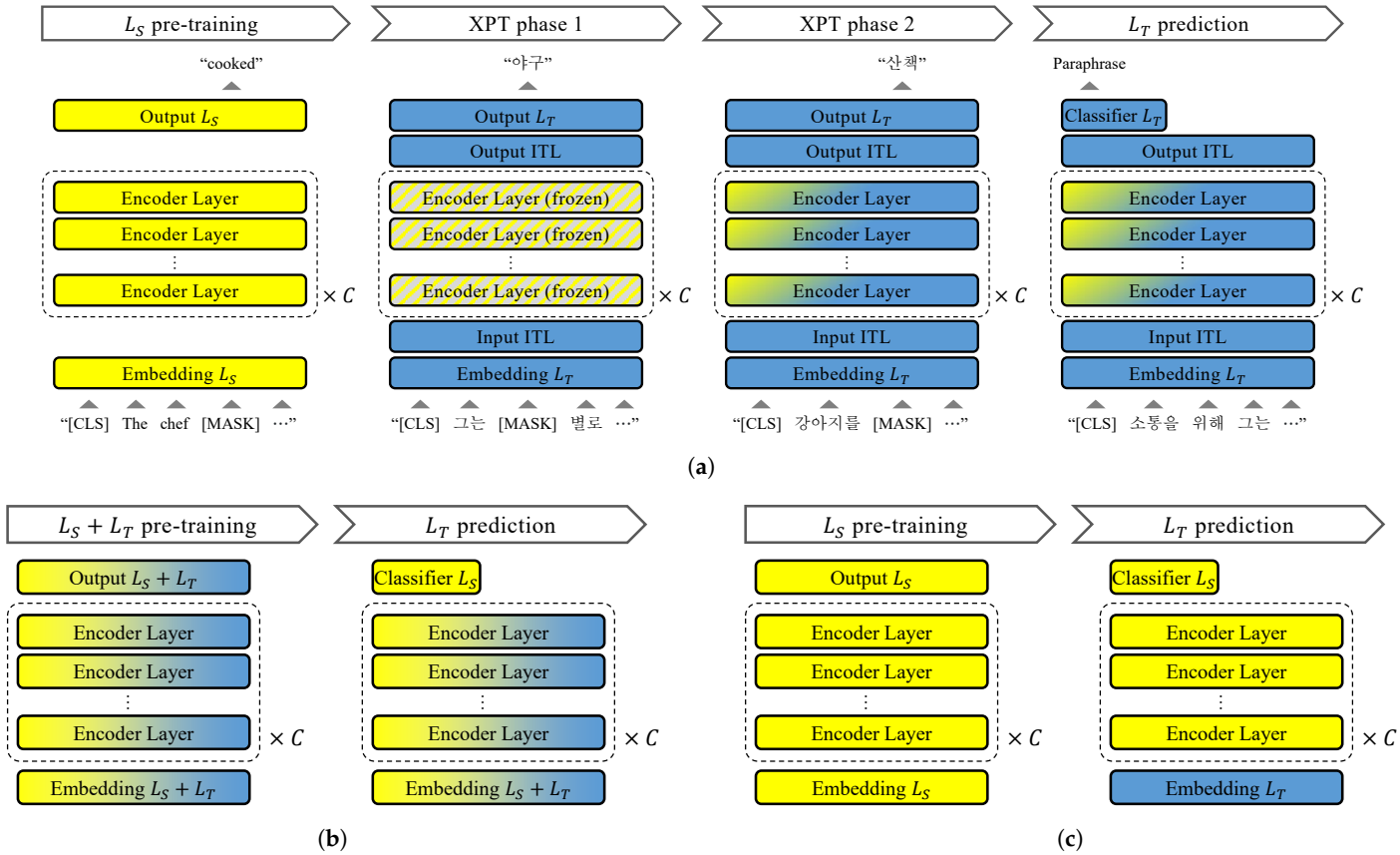


Figure 2. Illustration of the proposed approach and previous approaches. (**a**) Cross-lingual post-training (ours). (**b**) Multilingual language model pretraining. (**c**) Monolingual transfer learning.

3.1. Transfer Learning as Post-Training

Our proposed method aims at transferring the knowledge from a high-resource language L_S to a low-resource language L_T. Transfer learning in the context of multilingual PLM has mostly revolved around zero-shot learning, and the small number of existing few-shot and supervised learning approaches limit the performance in L_T by forcing the model to maintain its ability in L_S. This is based on the assumption that there are none or few labeled examples in L_T while unlabeled data is abundant. However, it has been suggested that this assumption is neither realistic nor practical [14,15].

Instead of this limited transfer learning approach, we assume that both unlabeled and labeled data are available in L_T and formulate this as a pretraining and post-training problem. Post-training refers to the process of performing additional unsupervised training to a PLM such as BERT using unlabeled domain-specific data, prior to fine-tuning. It has been shown that this leads to improved performance by helping the PLM to adapt to the target domain [46–49]. We start with a monolingual PLM in L_S and completely adapt it to L_T.

Another key advantage of this approach is that this makes it possible to completely skip the training in L_S. This is because most recent publications in PLM literature make the trained model checkpoint publicly available, and the model architecture and training objectives in L_S are inherited to L_T when post-training. Beside the cost saving, this also enables the use of this method when the pretraining data in L_S is not available in certain scenarios (e.g., privacy concerns, licensing, etc.).

3.2. Selecting Parameters to Transfer

A language model consumes a word sequence and emits a contextualized vector representation of it. Then these vectors can be used to assign some probability to a word or to perform some tasks such as sequence classification.

More formally, assume we have an input as a sequence of tokens $\mathcal{T} = [t_1, t_2, \cdots, t_n]$, $1 \leq t_i \leq V \ \in \mathbb{N}$, where V is the vocabulary size. Then, the output of a language model LM is given by

$$\mathbf{E}_{L_S} = [\mathbf{e}_1^S; \mathbf{e}_2^S; \cdots; \mathbf{e}_V^S], \mathbf{e}_i^S \in \mathbb{R}^d \tag{1}$$

$$\mathcal{H}_0 = Embedding(\mathcal{T}, \mathbf{E}) = [\mathbf{e}_{t_1}^S, \mathbf{e}_{t_2}^S, \cdots, \mathbf{e}_{t_n}^S] \tag{2}$$

$$\mathcal{H}_l = Encoder_l(\mathcal{H}_{l-1}, \theta_l) = [\mathbf{h}_1^l, \mathbf{h}_2^l, \cdots, \mathbf{h}_n^l] \tag{3}$$

$$LM(\mathcal{T}, \Theta) = \mathcal{H}_C = [\mathbf{h}_1^C, \mathbf{h}_2^C, \cdots, \mathbf{h}_n^C], \tag{4}$$

where $\mathbf{E}$ is the embedding matrix, $Embedding(\cdot)$ is a lookup function, θ_l is the parameters of the lth encoder layer, $\Theta = [\mathbf{E}_{L_S}, \theta_1, \theta_2, \cdots, \theta_C]$, and C is the number of encoder layers in LM. Then, the probability of a token is computed as

$$P(t_i = t_j|\mathcal{T}) = \frac{\exp\left(\mathbf{h}_{t_i}^C \cdot \mathbf{e}_{t_j}^S\right)}{\sum_{k=1}^{n} \exp\left(\mathbf{h}_{t_k}^C \cdot \mathbf{e}_{t_j}^S\right)}, t_i = [MASK] \tag{5}$$

for the masked language modeling task, and

$$P(t_{n+1} = t_j|\mathcal{T}) = \frac{\exp\left(\mathbf{h}_{t_n}^C \cdot \mathbf{e}_{t_j}^S\right)}{\sum_{k=1}^{n} \exp\left(\mathbf{h}_{t_k}^C \cdot \mathbf{e}_{t_j}^S\right)} \tag{6}$$

for the next word prediction task.

Among the parameters in Θ, some could be helpful in modeling L_T while some could be harmful. The most important part of the modeling process is the contextualization of embedding vectors, performed by the encoder layers. We reuse them in post-training as these layers are known to acquire mostly language-independent knowledge [34,36–38]. On the other hand, embedding vectors project the tokens in a language into the semantic space and thus cannot be directly transferred to another language. It is possible to indirectly use them using bilingual word embedding techniques [26,45], but we randomly initialize the word vectors of L_T for simplicity.

Modern transformer-based architectures have some additional parameters such as positional embeddings and language modeling head [1,3]. We also reuse them in L_T as they are not language-dependent and have shown to improve performance in the preliminary experiments.

3.3. Implicit Translation Layer

Reusing the encoder layer trained in the source language and only learning the word embeddings in the target language can be seen as finding a token-level mapping between the two languages. However, this is suboptimal for two reasons. First, such mapping is most likely to be impossible due to ambiguity such as homographs. Second, linguistic differences beyond token-level, such as word order, cannot be learned with this method.

To overcome these shortcomings, we propose the Implicit Translation Layer (ITL) to find this mapping at a sequence level. The ITL takes a sequence of vectors as input and outputs contextualized vectors of equal length. To maximize compatibility, we utilize the same architecture used as the encoder layer. Two ITLs are added to the language model, one before the first encoder layer (input-to-encoder) and another one after the last encoder layer (encoder-to-output). These ITLs can be seen as implicitly translating from L_T to L_S and L_S to L_T, respectively. With the addition of ITLs, the computation by a LM in L_T becomes

$$\mathbf{E}_{L_T} = [\mathbf{e}_1^T; \mathbf{e}_2^T; \cdots; \mathbf{e}_V^T], \mathbf{e}_i^T \in \mathbb{R}^d \tag{7}$$

$$\mathcal{H}_{L_T} = Embedding(\mathcal{T}, \mathbf{E}_{L_T}) \tag{8}$$

$$\mathcal{H}_0 = ITL_{in}(\mathcal{H}_{L_T}, \theta_{ITL_{in}}) \tag{9}$$

$$LM(\mathcal{T}, \Theta') = ITL_{out}(\mathcal{H}_C, \theta_{ITL_{out}}), \tag{10}$$

where $\Theta' = [\mathbf{E}_{L_T}, \theta_{ITL_{in}}, \theta_{ITL_{out}}, \theta_1, \theta_2, \cdots, \theta_C]$. $\mathcal{H}_C$ is computed using Equation (3), and the token vectors in Equations (5) and (6) are also replaced with respective vectors from $\mathbf{E}_{L_T}$. This configuration allows a more flexible mapping compared to modules that operate on a token level such as Adapters. It is a great advantage as multilingual contextualized representations are known to be highly sensitive to word order [37].

3.4. Two-Phase Post-Training

The parameters $\mathbf{E}_{L_T}$, $\theta_{ITL_{in}}$, and $\theta_{ITL_{out}}$ are randomly initialized and learned during the post-training phase. The noise introduced by this randomness can negatively impact the tuned parameters from L_S. To prevent this, we split the post-training into two phases, similar to gradual unfreezing [50,51]. In the first phase, the parameters copied from the L_S model are frozen, and only the L_T embeddings and ITLs are learned using the training examples in L_T. This is analogous of the method used for zero-shot transfer learning in the multilingual PLM literature, where only the parameters responsible in learning L_T are updated.

Phase two of our proposed method proceeds further and completely adapts the language model to L_T. This is achieved by unfreezing the parameters from L_S and fine-tuning the entire model using data in L_T. Here, it is assumed that the randomness and the resulting noise is minimized in the first phase. Each phase inherits the training objective from L_S training, and the model is optimized until convergence using the unlabeled data in L_T.

4. Experimental Setup

4.1. Overview

We conduct a case study for Korean, transferring from English RoBERTa [3] as the PLM in L_S. This model has almost the same structure as BERT, but incorporates improved training techniques such as dynamic masking, longer training, and larger batch size. The *BASE* configuration is used, which has 768-dimensional word embedding and 12 encoder layers. Applying our proposed XPT results in 14 encoder layers in total. To understand the effect of two-phase training, we also train a variation of XPT, termed XPT-SP, where the entire model is post-trained in a single phase without freezing any parameters. We skip the pretraining in L_S and use the model checkpoint released by the authors instead [52].

4.2. Baselines

For intrinsic evaluations, we compare our proposed method to the following two baseline models.

Scratch—This model does not utilize the knowledge from L_1 and is trained from scratch using data from L_2. To match the number of parameters with our proposed method, two additional encoder layers are added to this model.

Adapters—Instead of using ITL, Adapter modules are added as in [43] in the encoder layers. This setting is similar to MonoTrans [44], except that the entire model is post-trained without freezing to maximize the performance in L_2.

In addition to the aforementioned baselines, we consider the following models as baselines for the extrinsic evaluations.

mBERT—The massively multilingual version of BERT, trained on the Wikipedia dump for the top 100 languages with the largest number of documents.

XLM-R—Another massively multilingual language model, which is based on RoBERTa and trained on the cleaned Common Crawl dataset [53] with 295 billion tokens covering 100 languages.

KoBERT—Publicly available monolingual BERT trained from scratch using Korean corpora [54]. The training corpora consists of Korean Wikipedia dump with 54 million tokens and Korean news dataset with 270 million tokens.

4.3. Dataset

We use Korean Wikipedia (Wiki-ko) for post-training in L_2. The Wikipedia dump from September 2020 was downloaded and extracted using the WikiExtractor [55] tool. This raw text is split into sentences and tokenized using SentencePiece [56] with a vocabulary size of 50 K tokens. This resulted in 4.19 M sentences with 61 M words before tokenization. The dataset is split into 4 M/100 K/88 K sentences to be used as train/valid/test splits, respectively.

For extrinsic evaluations, we use the following four tasks to quantitatively compare different models. Detailed statistics of each dataset is summarized in Table 1.

Table 1. Statistics of the datasets used in the downstream evaluations.

Task	Train		Validation		Test	
	# Examples	# Tokens	# Examples	# Tokens	# Examples	# Tokens
PAWS-X	49,401	1,413,443	2000	50,292	2000	50,599
KorSTS	5749	86,253	1499	26,022	1378	21,066
KQP	6136	46,007	682	5067	758	5589
KHS	7896	129,422	471	7755	N/A	N/A

PAWS-X—We use the Korean portion of the PAWS-X [57] dataset. The goal is to identify whether the given sentence pairs are a paraphrasing of each other or not. We report the classification accuracy (%) for this dataset.

KorSTS—The Semantic Textual Similarity (STS) [58] dataset aims at assessing the semantic similarity between a pair of sentences. Each pair is assigned with a score ranging from 0 to 5, and the model's performance is measured using Spearman's rank correlation coefficient against the gold labels. The KorSTS [59] dataset provides the machine-translated train examples as well as the human-translated development and test examples of the STS dataset.

KQP—The Korean Question Pairs (KQP) [60] is a dataset analogous to the Quora Question Pairs (QQP) [61] dataset, in which the model needs to identify if the given two questions convey the same meaning. KQP consists of 7.6 K human-annotated question pairs. Accuracy (%) is used as the evaluation metric.

KHS—The Korean Hate Speech (KHS) [62] dataset is a collection of 8367 news comments, labeled as one of "hate", "offensive", or "none" by human annotators. As gold labels for the test split are unavailable, we report the best f1-score on the validation split for this task.

4.4. Implementation Details

We implement our proposed method and the baselines using PyTorch [63], Fairseq [64], and Hugging Face Transformers [65]. All models are trained until convergence with early stopping. We mostly use the suggested hyperparameters from [3] with a few exceptions, which are summarized in Tables A1 and A2.

5. Results and Discussion

In this section, we summarize the key findings from quantitative evaluations, both intrinsic and extrinsic. For the intrinsic evaluations and the experiments testing the data efficiency (i.e., Figures 3 and 4, and Table 2), the results for mBERT, XLM-R, and KoBERT are not available as these are not trained from scratch and instead the publicly available model checkpoints are used.

5.1. Intrinsic Evaluation

To quantitatively measure the learning process of each model, we first perform intrinsic evaluations. While performing better on intrinsic tests does not guarantee the model to be better at downstream tasks as well, they are good estimates and relatively cheap to evaluate. In this study, perplexity, hits@k, mean rank, and mean reciprocal rank are calculated on the masked tokens and used as intrinsic metrics.

The intrinsic performance of each model after using all available 4M training examples is summarized in Table 2. It can be seen that XPT performs the best, followed by Adapters and Scratch. Further, the improvement from Scratch to Adapters is as large as the improvement from Adapters to XPT, indicating that our proposed XPT performs much favorably to Adapters.

Table 2. Intrinsic evaluation results after using all available 4 million training examples in L_2. Best values are highlighted in bold. PPL = perplexity. MRR = mean reciprocal rank. MR = mean rank.

Model	PPL	Hits@1	Hits@3	Hits@5	Hits@10	MRR	MR
Scratch	5.88	63.91	76.18	80.41	85.18	0.7145	58.87
Adapters	5.46	65.05	77.28	81.40	86.04	0.7253	53.93
XPT-SP (Ours)	5.41	65.28	77.49	81.60	86.18	0.7273	54.03
XPT (Ours)	**5.07**	**66.13**	**78.31**	**82.38**	**86.90**	**0.7354**	**50.41**

To understand the data efficiency during the post-training process, we exponentially vary the number of training examples from 100 K to 3200 K. The result for each metric is plotted in Figure 3. As can be seen, the performance increases linearly as the dataset size increases exponentially. This shows how data-hungry these language models are, demonstrating the importance of increasing the data efficiency. Adapters and XPT perform comparably, with XPT performing slightly better when there are more than 400 K training examples and slightly worse with less than 400 K examples.

On the other hand, we find that transferring from English (i.e., Adapters and XPT) is roughly as effective as doubling the number of training examples in the Scratch setting. Further, the difference between transferred and non-transferred settings are most pronounced when the amount of data is minimal, suggesting that low-resource languages are likely to benefit the most from XPT. The common underlying hypothesis in the multilingual language modeling literature is that simultaneously learning from multiple languages is the key to improving performance in low-resource languages by learning polyglot representation [12,13,36–38]. However, our results suggests that some part of the knowledge acquired by monolingual models is language-agnostic and thus can be effectively transferred to other languages. Based on this, we argue that more emphasis should be put on transferring monolingual representation [44,45] as these are more sustainable than multilingual training.

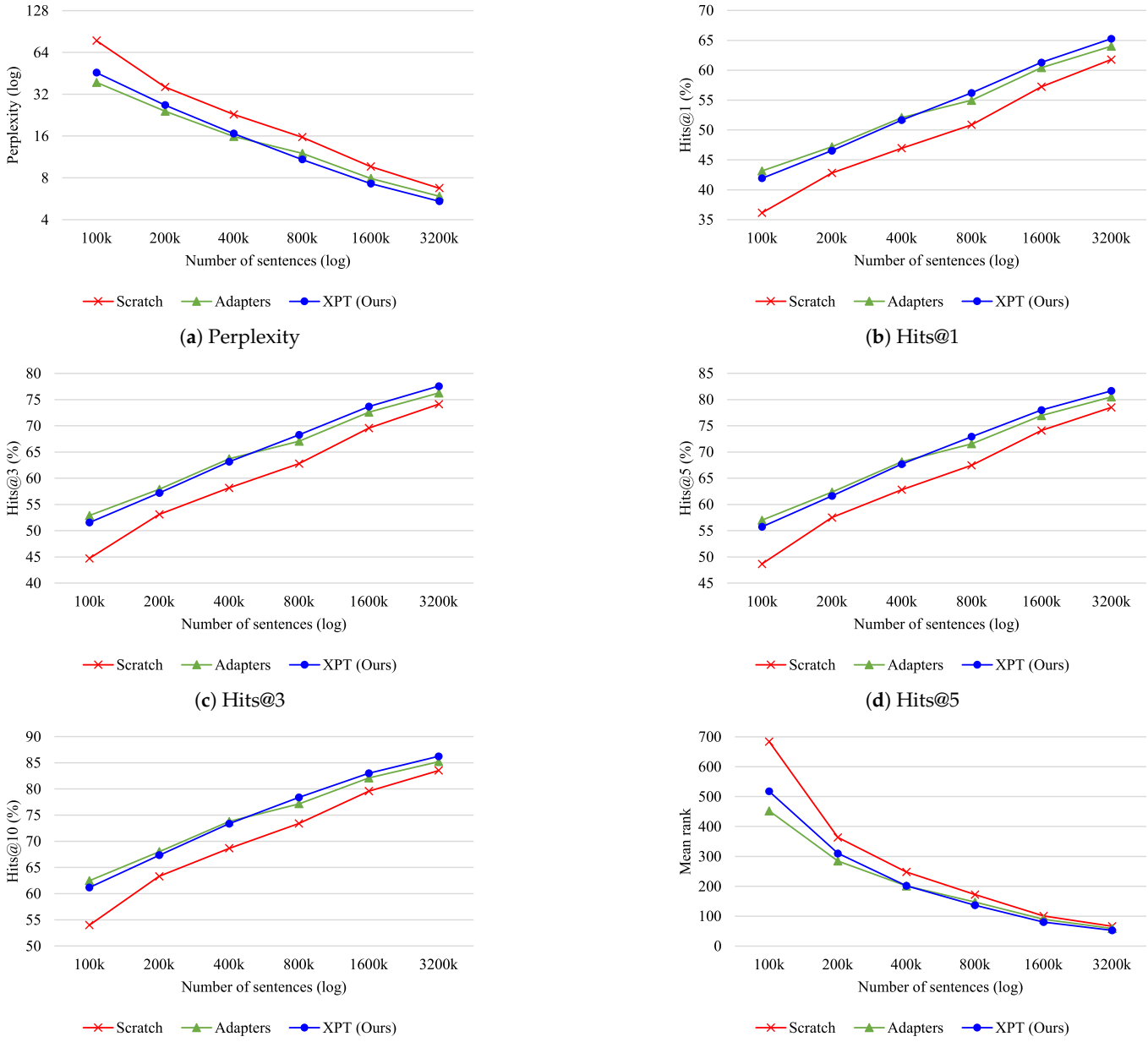


Figure 3. Intrinsic evaluation results as a function of the number of training examples.

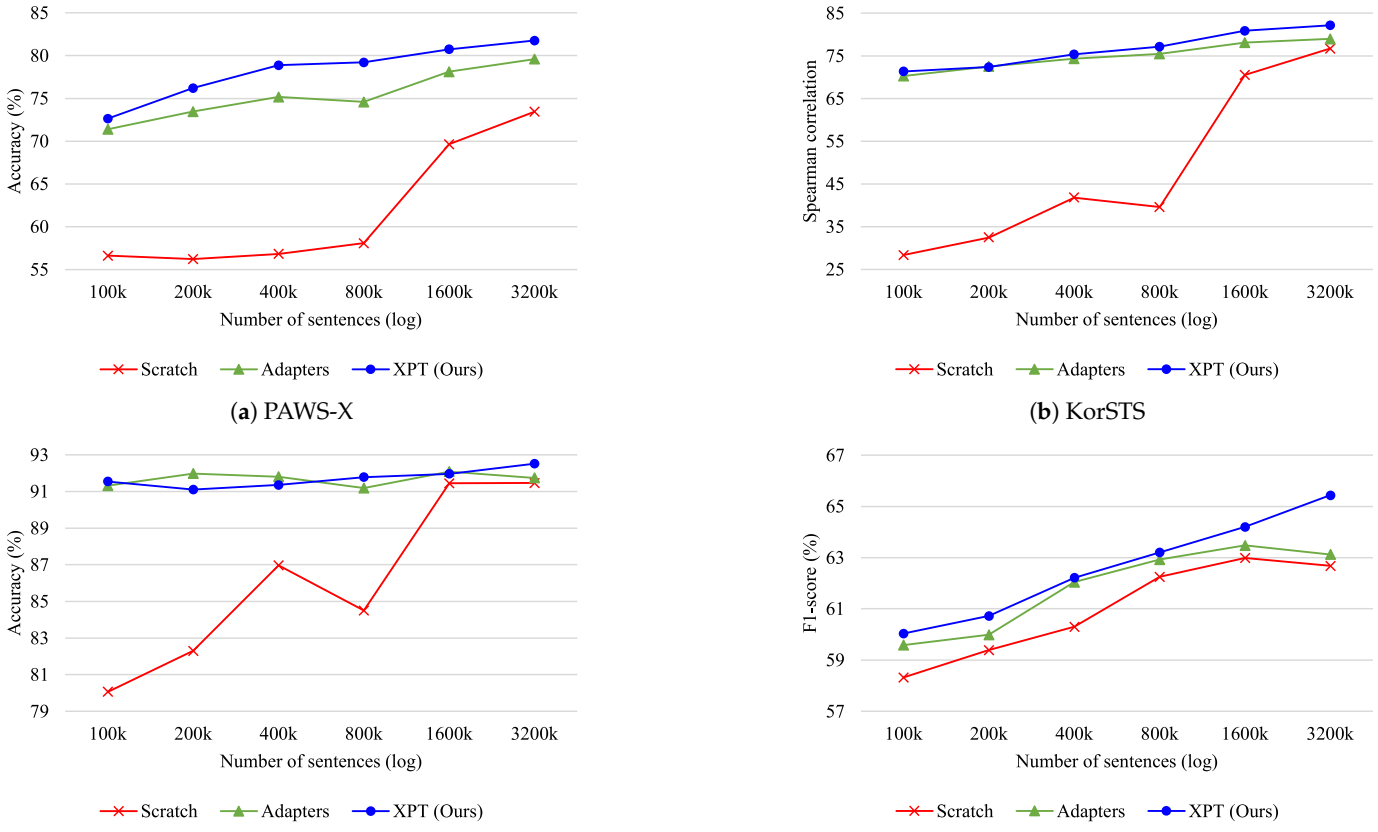
5.2. Extrinsic Evaluation

The results after using all 4 M post-training examples are summarized in Table 3, alongside the results from fine-tuning publicly available models. All models are trained 10 times with different random seeds, and we report the mean and standard deviation. It can be seen that the proposed XPT outperforms Adapters and Scratch by a large margin across all tasks, demonstrating its effectiveness given the same amount of training data. Further, it also outperforms mBERT in all tasks and XLM-R in three out of four tasks. The fact that XPT is post-trained with only a fraction of the data used to train these massively multilingual language models makes this result more encouraging. When compared to KoBERT, a monolingual model trained from scratch with approximately 5.3 times more training examples, XPT still performs better in all tasks except KQP, with a 22.30% relative error reduction in KorSTS. Interestingly, the Scratch model outperforms all models in the KHS dataset. We believe that this is caused by a domain mismatch between the pre/post-training data and the fine-tuning data, as the KHS dataset is collected from social media.

Table 3. Downstream evaluation results. Best values in each block are highlighted in bold, and the overall bests are highlighted with underline. * Estimated value by extrapolation.

Model	Pre/Post-Train Data		Task			
	Domain	# Tokens	PAWS-X	KorSTS	KQP	KHS
mBERT	Wiki-100	6 B*	81.37 ± 0.63	77.48 ± 0.73	92.30 ± 0.70	61.66 ± 1.56
XLM-R	CC-100	295 B	**81.54** ± 0.34	**78.63** ± 0.63	93.21 ± 0.48	63.45 ± 1.09
KoBERT	Wiki/news-ko	324 M	79.68 ± 0.75	77.67 ± 2.86	**93.54** ± 0.75	**64.68** ± 0.64
Scratch	Wiki-ko	61 M	73.40 ± 0.42	74.38 ± 0.73	91.65 ± 0.55	**65.55** ± 1.12
Adapters	Wiki-ko	61 M	79.69 ± 0.87	79.78 ± 0.85	91.94 ± 0.34	62.90 ± 1.79
XPT-SP (Ours)	Wiki-ko	61 M	80.46 ± 0.50	80.98 ± 0.40	92.22 ± 0.46	64.22 ± 1.24
XPT (Ours)	Wiki-ko	61 M	**81.62** ± 0.38	**82.65** ± 0.39	**92.88** ± 0.29	64.85 ± 0.53

Similar to the intrinsic evaluations, we also experiment with varying the number of post-training sentences in the downstream evaluations and plot the results in Figure 4. From Figure 4a,b, it can be seen that the Scratch setting shows minimal gains by increasing the data size from 100 K sentences to 800 K sentences. On the other hand, the models transferred from English show consistent improvements with more data. This shows that without any prior knowledge, these tasks cannot benefit from pretraining until a certain amount of data is available. We find a similar yet less pronounced trend in the KQP task as well. However, the results from the KHS task is different from the other three tasks, with all models performing comparably across all dataset sizes. Again, this is likely caused by the domain shift.



(**a**) PAWS-X (**b**) KorSTS

(**c**) Korean Question Pairs (**d**) Korean Hate Speech

Figure 4. Downstream evaluation results as a function of the number of pre/post-training examples.

Investigating the change in data efficiency, it can be seen that to reach the performance of transferred models post-trained with 100 K examples, the Scratch model requires approximately 32, 16, and 16 times more pre-training data in the PAWS-X, KorSTS, and KQP tasks, respectively. Between the transferred models, XPT consistently outperforms Adapters in PAWS-X, KorSTS, and KHS, while performing on par in KQP. This indicates that regardless of the amount of available post-training data, XPT can be expected to perform better than Adapters.

Existing cross-lingual language model pretraining approaches force the model to maintain its multi-linguality [13,40,44,45]. However, in a realistic and practical scenario, the goal is often maximizing the performance on a single language at interest. Our results demonstrates that under this scenario, XPT and completely adapting the model to a single language in general are superior to polyglot models.

5.3. Effect of Two-Phase Training

To understand the effect of two-phase training, we trained XPT-SP, where the entire model is post-trained without freezing any parameters. The intrinsic results and downstream results are shown in Tables 2 and 3, respectively. Overall, XPT-SP outperforms Adapters in all cases, but performs suboptimally compared to two-phase training. This demonstrates that ITL is better than Adapter modules at learning linguistic differences. Incorporating two-phase training to Adapters could bring some improvements. However, based on the fact that XPT-SP performs better than Adapters, we expect this variant to be less effective than XPT.

6. Conclusions and Future Directions

While being highly effective across a wide range of NLP tasks, pretrained Transformers are extremely data-hungry. For the majority of the over 7000 languages spoken worldwide, it is difficult to secure sufficient data for training such models. In this paper, we tackled this problem by proposing an approach for data-efficient training of pretrained language models in a low-resource language. Our approach, termed XPT, achieves this goal by post-training a PLM from another high-resource language. Language-agnostic parameters of a model trained in the high-resource language are selectively reused and tuned while learning the language-specific parameters in the target language. We also proposed ITL, which is designed to learn linguistic differences between the two languages at a sequence level instead of a token level.

To evaluate our method in a challenging and controlled scenario, we conducted a case study for Korean by post-training English RoBERTa with a varying amount of post-training examples. Intrinsic results have shown that post-training an English model in L_T is roughly as effective as using twice as much data in L_T and training from scratch. Further, downstream evaluations on four natural language understanding tasks demonstrated that our approach can improve the data efficiency by a factor of up to 32. When compared to monolingual and massively multilingual PLMs trained with several orders of magnitude more data, XPT still outperformed or matched the performance. This suggests that completely adapting a model to a single language of interest is more effective and efficient, and more focus should be put on this direction of research.

As for future directions, experimenting with other target languages with different resource availability and linguistic characteristics is an important step. Building a systematic approach for selecting the source language depending on the target language is also a promising research direction.

Author Contributions: Conceptualization, C.L. and T.W.; data curation, C.P.; formal analysis, C.L. and A.M.; funding acquisition/project administration/supervision, H.L.; investigation, C.L.; methodology/software, K.Y. and C.P.; review and editing, T.W. and A.M.; original draft, C.L. and K.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute for Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques). Additionally, it was also supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2018-0-01405) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The data can be found here: PAWS-X: https://github.com/google-research-datasets/paws/tree/master/pawsx (accessed on 22 February 2021). KorSTS: https://github.com/kakaobrain/KorNLUDatasets/tree/master/KorSTS (accessed on 22 February 2021). KQP: https://github.com/songys/Question_pair (accessed on 22 February 2021). KHS: https://github.com/kocohub/korean-hate-speech (accessed on 22 February 2021).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Hyperparameters

Table A1. Hyperparameters used for post-training.

Hyperparameters	Training Data Size		
	100 K to 400 K	**800 K to 1.6 M**	**3.2 M to 4 M**
Batch size	256	2048	2048
Learning rate	1×10^{-4}	3×10^{-4}	3×10^{-4}
Total steps	60 K	60 K	100 K
Warmup steps	1 K	1 K	1 K

Table A2. Hyperparameters used for downstream evaluations.

Hyperparameters	Task			
	PAWS-X	**KorSTS**	**KQP**	**KHS**
Batch size	32	32	32	32
Learning rate	1×10^{-5}	5×10^{-5}	1×10^{-5}	5×10^{-5}
Training epochs	15	10	10	10
Warpup proportion	0.06	0.06	0.06	0.06

References

1. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186.
2. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
3. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
4. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* **2019**, arXiv:1909.11942.
5. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 7871–7880.
6. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. ELECTRA: Pre-Training Text Encoders as Discriminators Rather Than Generators. *arXiv* **2020**, arXiv:2003.10555.
7. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.

8. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *arXiv* **2020**, arXiv:2005.14165.
9. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
10. He, P.; Liu, X.; Gao, J.; Chen, W. DeBERTa: Decoding-Enhanced BERT with Disentangled Attention. *arXiv* **2020**, arXiv:2006.03654.
11. Ethnologue: Languages of the World. Available online: https://www.ethnologue.com (accessed on 22 February 2021).
12. Lample, G.; Conneau, A. Cross-lingual language model pretraining. *arXiv* **2019**, arXiv:1901.07291.
13. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised cross-lingual representation learning at scale. *arXiv* **2019**, arXiv:1911.02116.
14. Artetxe, M.; Ruder, S.; Yogatama, D.; Labaka, G.; Agirre, E. A Call for More Rigor in Unsupervised Cross-lingual Learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 7375–7388.
15. Joshi, P.; Santy, S.; Budhiraja, A.; Bali, K.; Choudhury, M. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 6282–6293.
16. Kim, N.K. *International Encyclopedia of Linguistics*; Oxford University Press: Oxford, UK, 1992; Volume 2, pp. 282–286.
17. Song, J.J. *The Korean Language: Structure, Use and Context*; Routledge: Oxfordshire, UK, 2006; p. 15.
18. Campbell, L. *Glossary of Historical Linguistics*; Edinburgh University Press: Edinburgh, UK, 2007; pp. 90–91.
19. Lauscher, A.; Ravishankar, V.; Vulić, I.; Glavaš, G. From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 4483–4499.
20. Rogers, A.; Kovaleva, O.; Rumshisky, A. A primer in bertology: What we know about how bert works. *Trans. Assoc. Comput. Linguist.* **2021**, *8*, 842–866. [CrossRef]
21. Xia, P.; Wu, S.; Van Durme, B. Which* BERT? A Survey Organizing Contextualized Encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 7516–7533.
22. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
23. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems—Volume 2*; Curran Associates Inc.: Red Hook, NY, USA, 2013, pp. 3111–3119.
24. Mikolov, T.; Le, Q.V.; Sutskever, I. Exploiting similarities among languages for machine translation. *arXiv* **2013**, arXiv:1309.4168.
25. Gouws, S.; Bengio, Y.; Corrado, G. Bilbowa: Fast bilingual distributed representations without word alignments. *arXiv* **2014**, arXiv:1410.2455.
26. Lample, G.; Conneau, A.; Ranzato, M.; Denoyer, L.; Jégou, H. Word translation without parallel data. *arXiv*, **2017**, arXiv:1710.04087.
27. Lample, G.; Conneau, A.; Denoyer, L.; Ranzato, M. Unsupervised Machine Translation Using Monolingual Corpora Only. *arXiv* **2017**, arXiv:1711.00043.
28. Wang, Z.; Xie, J.; Xu, R.; Yang, Y.; Neubig, G.; Carbonell, J.G. Cross-lingual Alignment vs Joint Training: A Comparative Study and A Simple Unified Framework. *arXiv* **2019**, arXiv:1910.04708.
29. Schuster, T.; Ram, O.; Barzilay, R.; Globerson, A. Cross-Lingual Alignment of Contextual Word Embeddings, with Applications to Zero-shot Dependency Parsing. 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)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 1599–1613.
30. Wang, Y.; Che, W.; Guo, J.; Liu, Y.; Liu, T. Cross-Lingual BERT Transformation for Zero-Shot Dependency Parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 5725–5731.
31. Artetxe, M.; Schwenk, H. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 597–610. [CrossRef]
32. Mulcaire, P.; Kasai, J.; Smith, N.A. Polyglot Contextual Representations Improve Crosslingual Transfer. 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)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 3912–3918.
33. Huang, H.; Liang, Y.; Duan, N.; Gong, M.; Shou, L.; Jiang, D.; Zhou, M. Unicoder: A Universal Language Encoder by Pre-Training with Multiple Cross-lingual Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 2485–2494.
34. Feng, F.; Yang, Y.; Cer, D.; Arivazhagan, N.; Wang, W. Language-agnostic bert sentence embedding. *arXiv* **2020**, arXiv:2007.01852.
35. Multilingual BERT. Available online: https://github.com/google-research/bert/blob/master/multilingual.md (accessed on 22 February 2021).

36. Wu, S.; Dredze, M. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 833–844.
37. Pires, T.; Schlinger, E.; Garrette, D. How Multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Florence, Italy, 2019; pp. 4996–5001.
38. Conneau, A.; Wu, S.; Li, H.; Zettlemoyer, L.; Stoyanov, V. Emerging Cross-lingual Structure in Pretrained Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; 2020; pp. 6022–6034. Available online: https://www.aclweb.org/anthology/2020.acl-main.0 (accessed on 22 February 2021).
39. Üstün, A.; Bisazza, A.; Bouma, G.; van Noord, G. UDapter: Language Adaptation for Truly Universal Dependency Parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 2302–2315.
40. Pfeiffer, J.; Vulić, I.; Gurevych, I.; Ruder, S. 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)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 7654–7673.
41. Rebuffi, S.A.; Bilen, H.; Vedaldi, A. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA; 2017; Volume 30, pp. 506–516.
42. Rebuffi, S.A.; Bilen, H.; Vedaldi, A. Efficient parametrization of multi-domain deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8119–8127.
43. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. *arXiv* **2019**, arXiv:1902.00751.
44. Artetxe, M.; Ruder, S.; Yogatama, D. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 4623–4637.
45. Tran, K. From english to foreign languages: Transferring pre-trained language models. *arXiv* **2020**, arXiv:2002.07306.
46. Jwa, H.; Oh, D.; Park, K.; Kang, J.M.; Lim, H. exBAKE: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Appl. Sci.* **2019**, *9*, 4062. [CrossRef]
47. Whang, T.; Lee, D.; Lee, C.; Yang, K.; Oh, D.; Lim, H. An Effective Domain Adaptive Post-Training Method for BERT in Response Selection. *arXiv* **2020**, arXiv:1908.04812.
48. Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N.A. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 8342–8360.
49. Luo, H.; Ji, L.; Li, T.; Jiang, D.; Duan, N. GRACE: Gradient Harmonized and Cascaded Labeling for Aspect-based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2020*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020, pp. 54–64.
50. Felbo, B.; Mislove, A.; Søgaard, A.; Rahwan, I.; Lehmann, S. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 1615–1625.
51. Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 328–339.
52. Published English RoBERTa model. Available online: https://dl.fbaipublicfiles.com/fairseq/models/roberta.base.tar.gz (accessed on 22 February 2021).
53. Wenzek, G.; Lachaux, M.A.; Conneau, A.; Chaudhary, V.; Guzmán, F.; Joulin, A.; Grave, É. CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data. *arXiv* **2019**, arXiv:1911.00359.
54. KoBERT: Korean BERT pretrained cased. Available online: https://github.com/SKTBrain/KoBERT (accessed on 22 February 2021).
55. Attardi, G. WikiExtractor. 2015. Available online: https://github.com/attardi/wikiextractor (accessed on 22 February 2021).
56. Kudo, T.; Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*; Association for Computational Linguistics: Brussels, Belgium; 2018; pp. 66–71.
57. Yang, Y.; Zhang, Y.; Tar, C.; Baldridge, J. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, China, 3–7 November 2019.
58. Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; Specia, L. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 1–14.
59. Ham, J.; Choe, Y.J.; Park, K.; Choi, I.; Soh, H. KorNLI and KorSTS: New Benchmark Datasets for Korean Natural Language Understanding. *arXiv* **2020**, arXiv:2004.03289.
60. Korean Question Pairs Dataset. Available online: https://github.com/songys/Question_pair (accessed on 22 February 2021).

61. First Quora Dataset Release: Question Pairs. Available online: https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs (accessed on 22 February 2021).
62. Moon, J.; Cho, W.I.; Lee, J. BEEP! Korean Corpus of Online News Comments for Toxic Speech Detection. In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 25–31.
63. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
64. Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; Auli, M. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In Proceedings of the NAACL-HLT 2019: Demonstrations, Minneapolis, MN, USA, 2–7 June 2019.
65. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*; Association for Computational Linguistics: Minneapolis, MN, USA, 2020; pp. 38–45.