

From English to Foreign Languages: Transferring Pretrained Language Models

Ke Tran

Amazon Alexa AI

trnke@amazon.com

Abstract

Pretrained models have demonstrated their effectiveness in many downstream natural language processing (NLP) tasks. The availability of multilingual pretrained models enables zero-shot transfer of NLP tasks from high resource languages to low resource ones. However, recent research in improving pretrained models focuses heavily on English. While it is possible to train the latest neural architectures for other languages from scratch, it is *undesirable* due to the required amount of compute. In this work, we tackle the problem of transferring an existing pretrained model from English to other languages under a *limited computational budget*. With a single GPU, our approach can obtain a foreign BERT_{BASE} model within a day (20 hours) and a foreign BERT_{LARGE} within two days (46 hours). Furthermore, evaluating our models on six languages, we demonstrate that our models are better than multilingual BERT on two zero-shot tasks: natural language inference and dependency parsing. Our code is available at <https://github.com/anonymized>.

1 Introduction

Pretrained models (Devlin et al., 2019; Peters et al., 2018) have received much of attention recently thanks to their impressive results in many downstream NLP tasks. Additionally, multilingual pretrained models enable many NLP applications for other languages via zero-shot cross-lingual transfer. Zero-shot cross-lingual transfer has shown promising results for rapidly building applications for low resource languages. Wu and Dredze (2019) show the potential of multilingual BERT (Devlin et al., 2019) in zero-shot transfer for a large number of languages from different language families on five NLP tasks, namely, natural language inference, document classification, named entity recognition, part-of-speech tagging, and dependency parsing.

Although multilingual models are an important piece for building up language technology in many languages, recent research on improving pretrained models puts much emphasis on English (Radford et al., 2019; Dai et al., 2019; Yang et al., 2019). The current state of affairs makes it difficult to translate developments in pre-training from English to non-English languages. To our best knowledge, there are only three publicly available multilingual pretrained models to date:

1. The multilingual BERT (mBERT) model (Devlin et al., 2019) that supports 104 languages;
2. Cross-lingual language model (XLM-R)¹ (Lample and Conneau, 2019; Conneau et al., 2019) that supports 100 languages;
3. Language Agnostic SEntence Representations model (LASER)² (Artetxe and Schwenk, 2019) that supports 93 languages.

Among the three models, LASER is based on neural machine translation approach and strictly requires parallel data to train.

A common practice to train a large-scale multilingual model is to do so from scratch. *But do multilingual models always need to be trained from scratch? Can we transfer linguistic knowledge learned by English pretrained models to other languages?* In this work, we develop a technique to rapidly transfer an existing pretrained model from English to other languages. As the first step, we focus on building a *bilingual* language model (LM) of English and a target language. Starting from a pretrained English LM, we learn the target language specific parameters (*i.e.*, word embeddings), while keeping the encoder layers of the pretrained

¹<https://github.com/facebookresearch/XLM>

²<https://github.com/facebookresearch/LASER>

English LM fixed. We then fine-tune both English and target model to obtain the bilingual LM. We apply our approach to autoencoding language models with masked language model objective and show the advantage of the proposed approach in zero-shot transfer. Our main contributions in this work are:

- We propose a *fast* adaptation method for obtaining a bilingual BERT_{BASE} of English and a target language within a day using one Tesla V100 16GB GPU (§3).
- We evaluate our bilingual LMs for six languages on two zero-shot cross-lingual transfer tasks, namely natural language inference (XNL) (Conneau et al., 2018) and universal dependency parsing. We show that our models offer competitive performance or even better than mBERT (§4).
- We illustrate that our bilingual LMs can serve as an excellent feature extractor in supervised dependency parsing task (§5).

Concurrent to our work, Artetxe et al. (2019) transfer pretrained English model to other languages by fine-tuning only *randomly initialized* target word embeddings while keeping the Transformer encoder fixed. Their approach is simpler than ours but requires more compute (64 TPUv3 chips) to achieve good results.

2 Bilingual Pretrained LMs

We first provide some background of pretrained language models. Let E_e be English word embeddings and $\text{enc}(\theta)$ be the Transformer (Vaswani et al., 2017) encoder with parameters θ . Let e_{w_i} denote the embedding of word w_i (i.e., $e_{w_i} = E_e[w_i]$). We omit positional embeddings and bias for clarity. A pretrained LM typically performs the following computations:

- transform a sequence of input tokens to contextualized representations $[c_{w_1}, \dots, c_{w_n}] = \text{enc}(e_{w_1}, \dots, e_{w_n}; \theta)$
- predict an output word y_i at i^{th} position $p(y_i | c_{w_i}) \propto \exp(c_{w_i}^\top e_{y_i})$

Autoencoding LM (Devlin et al., 2019) corrupts some input tokens w_i by replacing them with a special token [MASK]. It then predicts the original

tokens $y_i = w_i$ from the corrupted tokens. Autoregressive LM (Radford et al., 2019) predicts the next token ($y_i = w_{i+1}$) given all the previous tokens. The recently proposed XLNet model (Yang et al., 2019) is an autoregressive LM that factorizes output with all possible permutations, which shows empirical performance improvement over GPT-2 due to the ability to capture bidirectional context. Here we assume that the encoder performs necessary masking with respect to each training objective.

Given an English pretrained LM, we wish to learn a bilingual LM for English and a given target language ℓ under a limited computational resource budget. To quickly build a bilingual LM, we directly adapt the English pre-trained model to the target model. Our approach consists of two steps. First, we initialize target language word embeddings E_ℓ in the English embedding space such that embeddings of a target word and its English equivalents are close together (§2.1). Next, we construct a bilingual LM of E_e , E_ℓ , and $\text{enc}(\theta)$ and fine-tune all the parameters (§2.2). Figure 1 illustrates the two steps in our approach.

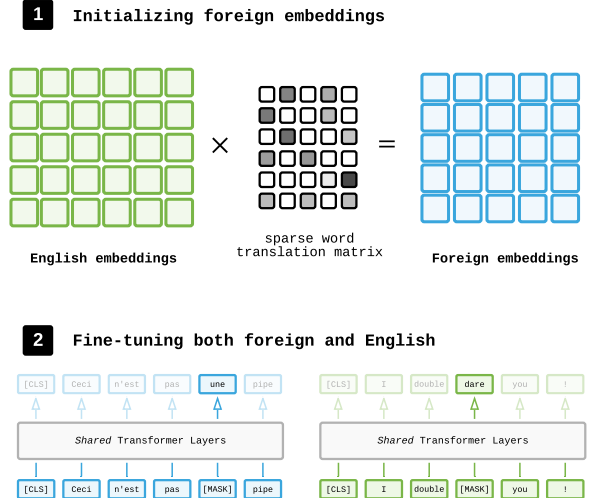


Figure 1: Illustration of our two-step approach. In the first step, foreign embeddings are initialized in English space (§2.1). In the second step, we jointly fine-tune both English and foreign models (§2.2).

2.1 Initializing Target Embeddings

Our approach to learn the initial foreign word embeddings $E_\ell \in \mathbb{R}^{|V_\ell| \times d}$ is based on the idea of mapping the trained English word embeddings $E_e \in \mathbb{R}^{|V_e| \times d}$ to E_ℓ such that if a foreign word and an English word are similar in meaning then their embeddings are similar. We represent each foreign

word embedding $E_\ell[i] \in \mathbb{R}^d$ as a linear combination of English word embeddings $E_e[j] \in \mathbb{R}^d$

$$E_\ell[i] = \sum_{j=1}^{|V_e|} \alpha_{ij} E_e[j] = \alpha_i E_e \quad (1)$$

where $\alpha_i \in \mathbb{R}^{|V_e|}$ is a sparse vector and $\sum_j \alpha_{ij} = 1$.

In this step of initializing foreign embeddings, having a good estimation of α could speed of the convergence when tuning the foreign model and enable zero-shot transfer (§5). In the following, we discuss how to estimate $\alpha_i \forall i \in \{1, 2, \dots, |V_\ell|\}$ under two scenarios: (i) we have parallel data of English-foreign, and (ii) we only rely on English and foreign monolingual data.

Learning from Parallel Corpus Given an English-foreign parallel corpus, we can estimate word translation probability $p(e_j | \ell_i)$ for any (English-foreign) pair (e_j, ℓ_i) using popular word-alignment (Brown et al., 1993) toolkits such as fast-align (Dyer et al., 2013). We then assign

$$\alpha_{ij} = p(e_j | \ell_i) \quad (2)$$

Since α_i is estimated from word alignment, it is a sparse vector.

Learning from Monolingual Corpus For low resource languages, parallel data may not be available. In this case, we rely only on monolingual data (e.g., Wikipedias). We estimate word translation probabilities from word embeddings of the two languages. Word vectors of these languages can be learned using fastText (Bojanowski et al., 2017) and then are aligned into a shared space with English (Lample et al., 2018b; Joulin et al., 2018). Unlike learning contextualized representations, learning word vectors is fast and computationally cheap. Given the aligned vectors \bar{E}_ℓ of foreign and \bar{E}_e of English, we calculate the word translation matrix $A \in \mathbb{R}^{|V_\ell| \times |V_e|}$ as

$$A = \text{sparsemax}(\bar{E}_\ell \bar{E}_e^\top) \quad (3)$$

Here, we use sparsemax (Martins and Astudillo, 2016) instead of softmax. sparsemax is a sparse version of softmax and it puts zero probabilities on most of the words in the English vocabulary except few English words that are similar to a given foreign word. This property is desirable in our approach since it leads to a better initialization of the foreign word embeddings.

2.2 Fine-tuning Bilingual LM

We create a bilingual LM by plugging foreign language specific parameters to the pretrained English LM (Figure 1). The new model has two separate word embedding layers (and output layers), one for English and one for foreign language. The encoder layer in between is shared. We then fine-tune this model using English and foreign monolingual data. Here, we keep tuning the model on English to ensure that it does not forget what it has learned in English and that we can use the resulting model for zero-shot transfer (§3). In this step, the encoder is updated so that it can learn syntactic aspects (i.e., word order, morphological agreement) of the target languages.

3 Zero-shot Experiments

In the scope of this work, we focus on transferring autoencoding LMs trained with masked language model objective. We choose BERT and RoBERTa (Liu et al., 2019) as the source models for building our bilingual language models, named RAMEN³ for the ease of discussion. We implement our models on top of HuggingFace’s Transformer (Wolf et al., 2019).⁴ For each pretrained model, we experiment with 12 layers (BERT_{BASE} and RoBERTa_{BASE}) and 24 layers (BERT_{LARGE} and RoBERTa_{LARGE}) variants. Using BERT_{BASE} allows us to compare the results with mBERT model. Using BERT_{LARGE} and RoBERTa allows us to investigate whether the performance of the target LM correlates with the performance of the source pretrained model. RoBERTa is a recently published model that is similar to BERT architecturally but with an improved training procedure. By training for longer time, with bigger batches, on more data, and on longer sequences, RoBERTa matched or exceeded previously published models including XLNet. We include RoBERTa in our experiments to validate the motivation of our work: *with similar architecture, does a stronger pretrained English model result in a stronger bilingual LM?* We evaluate our models on two cross-lingual zero-shot tasks: (1) Cross-lingual Natural Language Inference (XNLI) and (2) dependency parsing.

³The author likes ramen.

⁴<https://github.com/huggingface/transformers>

3.1 Data

We evaluate our approach for six target languages: French (fr), Russian (ru), Arabic (ar), Chinese (zh), Hindi (hi), and Vietnamese (vi). These languages belong to four different language families. French, Russian, and Hindi are Indo-European languages, similar to English. Arabic, Chinese, and Vietnamese belong to Afro-Asiatic, Sino-Tibetan, and Austro-Asiatic family respectively. The choice of the six languages also reflects different training conditions depending on the amount of monolingual data. French and Russian, and Arabic can be regarded as high resource languages whereas Hindi has far less data and can be considered as low resource.

For experiments that use parallel data to initialize foreign specific parameters, we use the same datasets in the work of Lample and Conneau (2019). Specifically, we use United Nations Parallel Corpus (Ziems et al., 2016) for en-ru, en-ar, en-zh, and en-fr. We collect en-hi parallel data from IIT Bombay corpus (Kunchukuttan et al., 2018) and en-vi data from OpenSubtitles 2018.⁵ For experiments that use only monolingual data to initialize foreign parameters, instead of training word-vectors from the scratch, we use the pretrained word vectors⁶ from fastText (Bojanowski et al., 2017) to estimate word translation probabilities (Eq. 3). We align these vectors into a common space using orthogonal Procrustes (Artetxe et al., 2016; Lample et al., 2018b; Joulin et al., 2018). We only use *identical words* between the two languages as the supervised signal. We use WikiExtractor⁷ to extract raw sentences from Wikipedias as monolingual data for fine-tuning target embeddings and bilingual LMs (§2.2). We *do not* lowercase or remove accents in our data preprocessing pipeline.

We tokenize English using the provided tokenizer from pretrained models. For target languages, we use fastBPE⁸ to learn 30,000 BPE codes and 50,000 codes when transferring from BERT and RoBERTa respectively. We truncate the BPE vocabulary of foreign languages to match the size of the English vocabulary in the source models. Precisely, the size of foreign vocabulary is set to

32,000 when transferring from BERT and 50,000 when transferring from RoBERTa.

We use XNLI dataset (Conneau et al., 2018) for classification task and Universal Dependencies v2.4 (Nivre et al., 2019) for dependency parsing task. Since a language might have more than one treebank in Universal Dependencies, we use the following treebanks: en_ewt (English), fr_gsd (French), ru_syntagrus (Russian), zh_gsd (Chinese), vi_vtb (Vietnamese), hi_hdtb (Hindi), and ar_padt (Arabic).

Remark on BPE (Lample et al., 2018a) show that sharing subwords between languages improves alignments between embedding spaces. Wu and Dredze (2019) observe a strong correlation between the percentage of overlapping subwords and mBERT’s performances for cross-lingual zero-shot transfer. On the other hand, K et al., (2020) report in their control experiments that subword overlapping has minimal contribution cross-lingual ability of BERT. Artetxe et al., (2019) confirms in their study that a shared vocabulary is not necessary for multilingual models. In our current approach, subwords between source and target are *not* shared. A subword that is in both English and foreign vocabulary has two different embeddings.

3.2 Estimating translation probabilities

Since pretrained models operate on subword level, we need to estimate subword translation probabilities. Therefore, we subsample 2M sentence pairs from each parallel corpus and tokenize the data into subwords before running fast-align (Dyer et al., 2013).

Estimating subword translation probabilities from aligned word vectors requires an additional processing step since the provided vectors from fastText are not at subword level.⁹ We use the following approximation to obtain subword vectors: the vector e_s of subword s is the weighted average of all the aligned word vectors e_{w_i} that have s as an subword

$$e_s = \sum_{w_j: s \in w_j} \frac{p(w_j)}{n_s} e_{w_j} \quad (4)$$

where $p(w_j)$ is the unigram probability of word w_j estimated from monolingual data and $n_s = \sum_{w_j: s \in w_j} p(w_j)$.

⁹In our preliminary experiments, we learned the aligned subword vectors but it results in poor performances.

⁵<http://opus.nlpl.eu>

⁶<https://fasttext.cc/docs/en/crawl-vectors.html>

⁷<https://github.com/attardi/wikiextractor>

⁸<https://github.com/glample/fastBPE>

	🌐	fr	vi	zh	ru	ar	hi	avg
(Conneau et al., 2018)	⊖	67.7	66.4	65.8	65.4	64.8	64.1	65.7
(Artetxe and Schwenk, 2019)	⊕	71.9	72.0	71.4	71.5	71.4	65.5	70.6
(Lample and Conneau, 2019) (MLM)	⊖	76.5	72.1	71.9	73.1	68.5	65.7	71.3
(Lample and Conneau, 2019) (MLM+TLM)	⊕	78.7	76.1	76.5	75.3	73.1	69.6	74.9
mBERT (Wu and Dredze, 2019)	⊖	73.8	69.5	69.3	69.0	64.9	60.0	67.8
<hr/>								
RAMEN _{BASE}								
+ BERT	⊖	75.2	71.8	70.7	71.1	69.3	62.8	70.1
	⊕	75.1	72.5	71.9	70.8	69.7	63.5	70.6
+ RoBERTa	⊖	79.9	75.9	73.7	73.6	71.9	65.6	73.4
	⊕	80.3	75.6	76.2	75.8	73.1	68.1	74.9
<hr/>								
RAMEN _{LARGE}								
+ BERT	⊖	78.1	74.8	74.5	73.7	70.8	64.5	72.7
	⊕	78.0	75.1	71.3	74.0	71.8	66.1	72.7
+ RoBERTa	⊖	81.3	76.2	76.3	75.6	73.5	64.5	74.6
	⊕	81.0	76.2	76.8	75.0	72.9	68.2	75.0

Table 1: Zero-shot classification results on XNLI. ⊕ indicates parallel data is used. RAMEN only uses parallel data for initialization. The best results are marked in **bold**.

We take the top 50,000 words in each aligned word-vectors to compute subword vectors.

In both cases, not all the words in the foreign vocabulary can be initialized from the English word-embeddings. Those words are initialized randomly from a Gaussian $\mathcal{N}(0, 1/d^2)$.

3.3 Hyper-parameters

In all the experiments, we tune RAMEN_{BASE} for 120,000 updates and RAMEN_{LARGE} for 300,000 updates. The sequence length is set to 256. For tuning bilingual LMs, we use a mini-batch size of 112 for RAMEN_{BASE} and 24 for RAMEN_{LARGE} where half of the batch are English sequences and the other half are foreign sequences. This strategy of balancing mini-batch has been used in multilingual neural machine translation (Firat et al., 2016; Lee et al., 2017).

We optimize RAMEN using Adam optimizer. We linearly increase the learning rate from 10^{-7} to 10^{-4} in the first 4000 updates and then follow an inverse square root decay. When fine-tuning RAMEN on XNLI and UD, we use a mini-batch size of 32, Adam’s learning rate of 10^{-5} . The number of epochs are set to 4 and 50 for XNLI and UD tasks respectively.

All experiments are carried out on a single Tesla V100 16GB GPU. Each RAMEN_{BASE} model is trained within a day and each RAMEN_{LARGE} is trained within two days.¹⁰

¹⁰22 and 46 GPU hours, to be precise. Learning alignment with fast-align takes less than 2 hours and we do not account for training time of fastText vectors.

4 Results

In this section, we present the results of our models for two zero-shot cross lingual transfer tasks: XNLI and universal dependency parsing.

4.1 Cross-lingual Natural Language Inference

Table 1 shows the XNLI test accuracy. For reference, we also include the scores from the previous work, notably the state-of-the-art system XLM (Lample and Conneau, 2019). Before discussing the results, we spell out that the fairest comparison in this experiment is the comparison between mBERT and RAMEN_{BASE}+BERT trained with monolingual only. We first discuss the transfer results from BERT. Initialized from fastText vectors, RAMEN_{BASE} slightly outperforms mBERT by 2.6 points on average and widens the gap of 4.4 points on Arabic. RAMEN_{BASE} gains extra 0.3 points on average when initialized from parallel data. With triple number of parameters, RAMEN_{LARGE} offers an additional boost in term of accuracy and initialization with parallel data consistently improves the performance. It has been shown that BERT_{LARGE} significantly outperforms BERT_{BASE} on eleven English NLP tasks (Devlin et al., 2019), the strength of BERT_{LARGE} also shows up when adapted to foreign languages.

Transferring from RoBERTa leads to better zero-shot accuracies. With the same initializing condition, RAMEN_{BASE}+RoBERTa outperforms RAMEN_{BASE}+BERT on average by 3.1 and 4.3 points when initializing from monolingual and par-

allel data respectively. This result show that with similar number of parameters, our approach benefits from a better English pretrained model. When transferring from RoBERTa_{LARGE}, we obtain state-of-the-art results for five languages. It is worth mentioning that these models are still underfit and they have potential to push the XNLI performance further.

Currently, RAMEN only uses parallel data to initialize foreign embeddings. RAMEN can also exploit parallel data through translation objective proposed in XLM. We believe that by utilizing parallel data during the fine-tuning of RAMEN would bring additional benefits for zero-shot tasks. We leave this exploration to future work. In summary, starting from BERT_{BASE}, our approach obtains competitive bilingual LMs with mBERT for zero-shot XNLI. Our approach shows the accuracy gains when adapting from a better pretrained model.

4.2 Universal Dependency Parsing

We build on top of RAMEN a graph-based dependency parser (Dozat and Manning, 2016). For the purpose of evaluating the contextual representations learned by our model, we *do not* use part-of-speech tags. Contextualized representations are directly fed into Deep-Biaffine layers to predict arc and label scores. Table 2 presents the Labeled Attachment Scores (LAS) for zero-shot dependency parsing.

	🇬🇧	fr	vi	zh	ru	ar	hi	avg
mBERT	🇬🇧	71.6	35.7	26.6	65.2	36.4	30.4	44.3
RAMEN _{BASE}								
+ BERT	🇬🇧	78.0	38.3	30.1	67.2	40.6	38.6	48.8
	🇬🇧	77.0	36.7	30.8	66.8	40.9	40.9	48.9
+ RoBERTa	🇬🇧	79.1	38.9	31.5	67.7	42.2	41.2	50.1
	🇬🇧	78.5	39.4	31.3	65.1	41.4	44.2	50.0
RAMEN _{LARGE}								
+ BERT	🇬🇧	78.2	39.5	30.5	65.6	45.3	44.0	50.5
	🇬🇧	78.9	38.4	30.5	66.3	43.4	44.1	50.3
+ RoBERTa	🇬🇧	79.7	40.0	32.2	65.7	44.1	44.6	51.1
	🇬🇧	79.1	39.2	30.5	65.4	43.8	46.8	50.8

Table 2: LAS scores for zero-shot dependency parsing. 🇬🇧 indicates parallel data is used for initialization. Punctuation are removed during the evaluation. The best results are marked in **bold**.

We first look at the fairest comparison between mBERT and monolingually initialized RAMEN_{BASE}+BERT. The latter outperforms the former on all the languages with the average of 4.5 LAS point. We observe the largest gain of +8.2 LAS point. We observe the largest gain of +8.2 LAS point. We observe the largest gain of +8.2 LAS point for Hindi and +6.2 LAS for French. Arabic enjoys

+4.2 LAS from our approach. With similar architecture (12 or 24 layers) and initialization (using monolingual or parallel data), RAMEN+RoBERTa performs better than RAMEN+BERT for most of the languages. Arabic and Hindi benefit the most from bigger models. For the other four languages, RAMEN_{LARGE} renders a modest improvement over RAMEN_{BASE}.

5 Analysis

Throughout this section, we carry out the main analysis for the RAMEN_{BASE}+BERT model that only use monolingual data.

5.1 How does linguistic knowledge transfer happen through each training stages?

We evaluate the performance of RAMEN+BERT_{BASE} (initialized from monolingual data) at each 20K training steps. The results are presented in Figure 2.

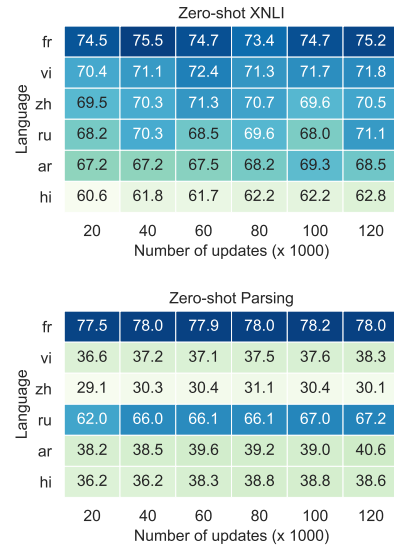


Figure 2: Accuracy and LAS evaluated at each checkpoints.

We observe the general trend that further fine-tuning RAMEN improves the performances of two zero-shot tasks. Just after 40,000 fine-tuning updates—corresponding to seven training hours—RAMEN_{BASE}+BERT surpasses mBERT on both XNLI (Table 1) and universal dependency parsing (Table 2). Interestingly, we find that the English encoder can quickly adapt to the foreign syntax that has different word order (ar and hi) with a little training data.

Language similarities seem to have more impact on transferring syntax than semantics. French en-

joys 78.0 LAS for being closely related to English, whereas Arabic and Hindi, SOV languages, modestly reach 38.2 and 36.2 points using the SVO encoder. Although Chinese has SVO order, it is often seen as head-final while English is strong head-initial. Perhaps, this explains the poor performance for Chinese.

5.2 Impact of initialization

Initializing foreign embeddings is the backbone of our approach. A good initialization leads to better zero-shot transfer results and enables fast adaptation. To verify the importance of a good initialization, we train a $\text{RAMEN}_{\text{BASE}}+\text{BERT}$ with foreign word-embeddings that are initialized randomly from $\mathcal{N}(0, 1/d^2)$. For a fair comparison, we use the same hyper-parameters in §3.3. Table 3 shows the results of XNLI and UD parsing of random initialization. In comparison to the initialization using aligned fastText vectors, random initialization decreases the zero-shot performance of $\text{RAMEN}_{\text{BASE}}$ by 10.3% for XNLI and 11.6 points for UD parsing on average. We also see that zero-shot parsing of SOV languages (Arabic and Hindi) suffers random initialization.

	⌘	fr	vi	zh	ru	ar	hi	avg
XNLI	rnd	66.9	67.1	65.7	59.1	50.8	48.5	59.7
	scr	71.1	58.0	62.8	63.8	62.1	47.5	60.8
	vec	75.2	71.8	70.5	71.1	68.5	62.8	70.0
UD	rnd	71.0	33.2	25.5	60.3	19.7	13.4	37.2
	scr	63.8	17.0	13.6	61.0	15.5	11.2	30.3
	vec	78.0	38.3	30.1	67.2	40.6	38.6	48.8

Table 3: Comparison between random initialization (rnd) of language specific parameters and initialization using aligned fastText vectors (vec) and bilingual BERT trained from scratch (scr) for 400 hours.

To highlight the efficiency of our transferring approach, we compare $\text{RAMEN}_{\text{BASE}}+\text{BERT}$ with a bilingual BERT (bBERT) trained from scratch (scr). For a fair comparison, we use the hyper-parameters described in section §3.3. All bBERT models are trained for 2,300,000 updates (400 GPU hours), which is more than sixteen times longer than RAMEN. We emphasize two important observations from Table 3. First, the randomly initialized foreign embeddings RAMEN performs on par with bBERT on XNLI task and significantly better than bBERT on UD parsing. This suggests that a large amount of linguistic knowledge can be transferred through reusing the pretrained BERT encoder. Secondly, with a careful initialization of

foreign embeddings, our $\text{RAMEN}_{\text{BASE}}+\text{BERT}$ outshines bBERT with even just 20,000 updates (3.5 GPU hours) as shown in Figure 2.

5.3 Are contextual representations from RAMEN also good for supervised parsing?

All the RAMEN models are built from English and tuned on English for zero-shot cross-lingual tasks. It is reasonable to expect RAMEN s do well in those tasks as we have shown in our experiments. *But are they also a good feature extractor for supervised tasks?* We offer a partial answer to this question by evaluating our model for supervised dependency parsing on UD datasets.

	fr	vi	zh	ru	ar	hi	avg
mBERT	92.1	62.2	85.1	93.1	83.6	91.3	84.6
$\text{RAMEN}_{\text{BASE}}$							
+ BERT	92.2	63.3	85.0	93.3	83.9	92.2	85.0
+ RoBERTa	92.8	65.3	86.0	93.7	84.9	92.5	85.9
$\text{RAMEN}_{\text{LARGE}}$							
+ BERT	92.8	64.7	86.2	93.9	84.9	92.0	85.7
+ RoBERTa	93.0	66.4	87.3	94.0	85.3	92.8	86.5

Table 4: Evaluation in supervised UD parsing. The scores are LAS.

We used train/dev/test splits provided in UD to train and evaluate our RAMEN -based parser. Table 4 summarizes the results (LAS) of our supervised parser. For a fair comparison, we choose mBERT as the baseline and all the RAMEN models are initialized from aligned fastText vectors. With the same architecture of 12 Transformer layers, $\text{RAMEN}_{\text{BASE}}+\text{BERT}$ performs competitive to mBERT and outshines mBERT by +1.1 points for Vietnamese. The best LAS results are obtained by $\text{RAMEN}_{\text{LARGE}}+\text{RoBERTa}$ with 24 Transformer layers. Overall, our results indicate the potential of using contextual representations from RAMEN for supervised tasks.

6 Conclusions

In this work, we have presented a simple and effective approach for rapidly building a bilingual LM under a limited computational budget. Using BERT as the starting point, we demonstrate that our approach performs better than mBERT on two cross-lingual zero-shot sentence classification and dependency parsing. We find that the performance of our bilingual LM, RAMEN , correlates with the performance of the original pretrained English models. We also find that RAMEN is also

a powerful feature extractor in supervised dependency parsing. Finally, we hope that our work sparks of interest in developing fast and effective methods for transferring pretrained English models to other languages.

References

- Joakim Nivre et al. 2019. Universal dependencies 2.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. [Learning principled bilingual mappings of word embeddings while preserving monolingual invariance](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.
- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. [On the Cross-lingual Transferability of Monolingual Representations](#). *arXiv e-prints*, page arXiv:1910.11856.
- Mikel Artetxe and Holger Schwenk. 2019. [Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond](#). *Transactions of the Association for Computational Linguistics*, 7:597–610.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. [Enriching word vectors with subword information](#). *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Unsupervised cross-lingual representation learning at scale](#).
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*.
- 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.
- Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. In *ICLR*.
- Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.
- Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. [Multi-way, multilingual neural machine translation with a shared attention mechanism](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.
- Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.
- 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*.
- Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).
- Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *NeurIPS*.
- Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.
- Guillaume Lample, Alexis Conneau, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018b. Word translation without parallel data. In *International Conference on Learning Representations*.

- Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. [Fully character-level neural machine translation without explicit segmentation](#). *Transactions of the Association for Computational Linguistics*, 5:365–378.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- André F. T. Martins and Ramón F. Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML’16, pages 1614–1623. JMLR.org.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- 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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [Huggingface’s transformers: State-of-the-art natural language processing](#).
- Shijie Wu and Mark Dredze. 2019. [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)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *NeurIPS*.
- Michał Ziemiński, Marcin Junczys-Dowmunt, and Bruno Poulliquen. 2016. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).