Multi-stage Distillation Framework for Cross-Lingual Semantic Similarity Matching
Kunbo Ding\(^1\), Weijie Liu\(^{1,2}\), Yuejian Fang\(^1\), Zhe Zhao\(^2\), Qi Ju\(^2\), Xuefeng Yang\(^2\), Rong Tian\(^2\), Tao Zhu\(^2\), Haoyan Liu\(^2\), Han Guo\(^2\), Xingyu Bai\(^2\), Weiquan Mao\(^2\), Yudong Li\(^2\), Weigang Guo\(^2\), Taiqiang Wu\(^2\), and Ningyuan Sun\(^2\)

\(^1\)Peking University, Beijing, China \(^2\)Tencent Research, Beijing, China

Abstract

Previous studies have proved that cross-lingual knowledge distillation can significantly improve the performance of pre-trained models for cross-lingual similarity matching tasks. However, the student model needs to be large in this operation. Otherwise, its performance will drop sharply, thus making it impractical to be deployed to memory-limited devices. To address this issue, we delve into cross-lingual knowledge distillation and propose a multi-stage distillation framework for constructing a small-size but high-performance cross-lingual model. In our framework, contrastive learning, bottleneck, and parameter recurrent strategies are combined to prevent performance from being compromised during the compression process. The experimental results demonstrate that our method can compress the size of XLM-R and MiniLM by more than 50%, while the performance is only reduced by about 1%.

1 Introduction

On the internet, it is widespread to store texts in dozens of languages in one system. Cross-lingual similar text matching in multilingual systems is a great challenge for many scenarios, e.g., search engines, recommendation systems, question-answer robots, etc. (Cer et al., 2017; Hardalov et al., 2020; Asai et al., 2021).

In the monolingual scenario, benefiting from the robust performance of the pre-trained language models (PLMs) (e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2020), etc.), significant success has been achieved in text-similarity matching tasks. For example, Reimers and Gurevych (2019) proposed the SBERT model trained with similar text pairs and achieved the state-of-the-art performance in the supervised similarity matching. In unsupervised scenarios, Gao et al. (2021) proposed the SimCSE model, which was trained on Wiki corpus through contrastive learning task.

Drawing on the success in the monolingual scenario, researchers began to introduce pre-training technology into cross-lingual scenarios and proposed a series of multilingual pre-trained models, e.g., mBERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019), XLM-R (Conneau et al., 2020), etc. Due to the vector collapse issue (Li et al., 2020), the performances of these cross-lingual models on similarity matching tasks are still not satisfactory. Reimers and Gurevych (2020) injected the similarity matching ability of SBERT into the cross-lingual model through knowledge distillation, which alleviated the collapse issue and improved the performance of cross-lingual matching tasks.

Although the cross-lingual matching tasks have achieved positive results, the existing cross-lingual models are huge and challenging to be deployed in devices with limited memory. We try to distill the SBERT model into an XLM-R with fewer layers following Reimers and Gurevych (2020). However, as shown in Figure 1, the performance will be significantly reduced as the number of layers decreases. This phenomenon indicates that cross-lingual capabilities are highly dependent on the
model size, and simply compressing the number of layers will bring a serious performance loss.

In this work, we propose a multi-stage distillation compression framework to build a small-size but high-performance model for cross-lingual similarity matching tasks. In this framework, we design three strategies to avoid semantic loss during compression, i.e., multilingual contrastive learning, parameter recurrent, and embedding bottleneck. We further investigate the effectiveness of the three strategies through ablation studies. Besides, we respectively explore the performance impact of reducing the embedding size and encoder size. Experimental results demonstrate that our method effectively reduces the size of the multilingual model with minimal semantic loss. Finally, our code is publicly available.

The main contributions of this paper can be summarized as follows:

- We validate that cross-lingual capability requires a larger model size and explore the semantic performance impact of shrinking the embedding or encoder size.
- A multi-stage distillation framework is proposed to compress the size of cross-lingual models, where three strategies are combined to reduce semantic loss.
- Extensive experiments examine the effectiveness of these three strategies and multi-stages used in our framework.

2 Related work

2.1 Multilingual models

Existing multilingual models can be divided into two categories, namely Multilingual general model and Cross-lingual representation model.

In the first category, transformer-based pre-trained models have been massively adopted in multilingual NLP tasks (Huang et al., 2019; Chi et al., 2021; Luo et al., 2021; Ouyang et al., 2021). mBERT (Devlin et al., 2019) was pre-trained on Wikipedia corpus in 104 languages, achieved significant performance in the downstream task. XLM (Conneau and Lample, 2019) presented the translation language modeling (TLM) objective to improve the cross-lingual transferability by leveraging parallel data. XLM-R (Conneau et al., 2020) was built on RoBERTa (Liu et al., 2019) using CommonCrawl Corpus.

In the second category, LASER (Artetxe and Schwenk, 2019) used an encoder-decoder architecture based on a Bi-LSTM network and was trained on the parallel corpus obtained by neural machine translation. Multilingual Universal Sentence Encoder (mUSE) (Chidambaram et al., 2019; Yang et al., 2020) adopted a bi-encoder architecture and was trained with an additional translation ranking task. LaBSE (Feng et al., 2020) turned the pre-trained BERT into a bi-encoder mode and was optimized with the objectives of mask language model (MLM) and TLM. Recently, Mao et al. (2021) presented a lightweight bilingual sentence representation method based on the dual-transformer architecture.

2.2 Knowledge distillation

However, Multilingual models do not necessarily have cross-lingual capabilities, especially in the first category, in which vector spaces of different languages are not aligned. Knowledge distillation (Hinton et al., 2015) used knowledge from a teacher model to guide the training of a student model, which can be used to compress the model and align its vector space at the same time.

For model compression, knowledge distillation aimed to transfer knowledge from a large model to a small model. BERT-PKD (Sun et al., 2019) extracted knowledge from both last layer and intermediate layers at fine-tuning stage. DistilBERT (Sanh et al., 2019) performed distillation at pre-training stage to halve the depth of BERT. TinyBERT (Jiao et al., 2020) distilled knowledge from BERT at both pre-training and fine-tuning stages. MobileBERT (Sun et al., 2020) distilled bert into a model with smaller dimensions at each layer. MiniLM (Wang et al., 2021) conducted deep self-attention distillation.

Unlike previous works presenting general distillation frameworks, we focus on compressing multilingual pre-trained models while aligning their cross-lingual vector spaces. In addition, we take inspiration from Reimers and Gurevych (2020), which successfully aligned the vector space of the multilingual model through cross-lingual knowledge distillation (X-KD). Our framework combines the advantages of X-KD for aligning vectors and introduces three strategies and an assistant model to prevent performance from being compromised.
Figure 2: The overview of the model architecture and the multi-stage distillation. It consists of four stages and aims to obtain a small multilingual student model. For convenience, we take the English SBERT as the teacher model, XLM-R as the assistant model, \(<s_1, t_1>\) is a pair of parallel sentences in two different language. \(N\) is the batch size. MSE is the mean squared error loss function.

3 Method

In this section, we will introduce our method in detail. First, we exhibit the model architecture, and then introduce the multi-stage distillation strategy for the model training. An overview of our approach is shown in Figure 2.

3.1 Model architecture

Given a large-size monolingual model as teacher \(T\) and a small-size multilingual model as student \(S\), our goal is to transfer semantic similarity knowledge from \(T\) to \(S\) and simultaneously compress the size of \(S\) with \(m\) parallel sentences \(P = \{<s_1, t_1>, <s_2, t_2>, \ldots, <s_m, t_m>\}\).

3.1.1 Teacher model

In this work, we use SBERT (Reimers and Gurevych, 2019) as the teacher model, which has been proven to perform well on monolingual semantic similarity tasks. SBERT adopts a siamese network structure to fine-tune a BERT (Devlin et al., 2019) encoder, and applies a mean pooling operation to its output to derive sentence embedding.

3.1.2 Assistant model

Mirzadeh et al. (2020) proved that when the gap between the student and teacher is large, the performance of the student model will decrease. We hope to get a small student model with cross-lingual capabilities, while the teacher is a large monolingual model. To alleviate the gaps, we introduce an assistant model \(A\) (Mirzadeh et al., 2020), which is a large multilingual model with cross-lingual ability.

3.1.3 Student model

Inspired by ALBERT (Lan et al., 2020), we design the student model with Parameter Recurrent and Embedding Bottleneck strategy. Since there is no available multilingual ALBERT, we need to design from scratch.

Parameter recurrent. We choose the first \(M\) layers of the assistant model as a recurring unit (RU). The role of RU is to initialize the student model with layers from the assistant model. Concretely, the RU is defined as,

\[
RU = \{L_i | i \in [1, M]\},
\]

(1)
where $L_i$ is the $i^{th}$ transformer layer.

**Embedding bottleneck.** Multilingual pre-trained models usually require a large vocabulary $V$ to support more languages, which leads to large embedding layer parameters. We add a bottleneck layer (He et al., 2016; Lan et al., 2020; Sun et al., 2020) of size $B$ between embedding layer and hidden layer $H$. In this way, the embedding layer is reduced from $O(V \times H)$ to $O(V \times B + B \times H)$.

### 3.2 Multi-stage distillation

Multi-stage Distillation is the key for enabling the small-size student model with cross-lingual matching ability.

**Stage 1. Teaching assistant**

As the **Stage 1** in Figure 2, we use the teacher model and parallel corpus to align vector space between different languages through the loss function in (2), enabling its cross-lingual ability (Reimers and Gurevych, 2020).

$$\ell_{stage1} = \frac{1}{|N|} \sum_{i}^{N} \left[ (h_{T}^{si} - h_{A}^{si})^2 + (h_{T}^{ti} - h_{A}^{ti})^2 \right],$$

where $N$ is the batch size, and $s_i$ and $t_i$ denotes the parallel sentences in a mini batch.

**Stage 2. Align student embedding**

As the **Stage 2** in Figure 2, we align the embedding bottleneck layer with the assistant embedding space through the loss function in (3),

$$\ell_{stage2} = \frac{1}{|N|} \sum_{i}^{N} \left[ (h_{Be}^{si} - h_{Be}^{Ti})^2 + (h_{Be}^{ti} - h_{Be}^{Ti})^2 \right],$$

where $h_{Be}^{si}, h_{Be}^{ti}$ denotes the output of assistant embedding layer, $h_{Be}^{si}, h_{Be}^{ti}$ denotes the output of embedding bottleneck layer.

**Stage 3. Teaching student**

In the **Stage 3**, the student model is trained to imitate the output of the assistant model with loss function in (4),

$$\ell_{stage3} = \frac{1}{|N|} \sum_{i}^{N} \left[ (h_{A}^{si} - h_{S}^{si})^2 + (h_{A}^{ti} - h_{S}^{ti})^2 \right],$$

where $h_{A}^{si}, h_{A}^{ti}$ denotes the output of assistant model, $h_{S}^{si}, h_{S}^{ti}$ denotes the output of student model.

**Stage 4. Multilingual contrastive learning**

After the above three stages, we can get a small multilingual sentence embedding model. However, as shown in Figure 1, when the model size decrease, its cross-lingual performance decreases sharply. Therefore, in this stage, we propose multilingual contrastive learning (MCL) task further to improve the performance of the small student model.

Assuming the batch size is $N$, for a specific translation sentence pair $(s_i, t_i)$ in one batch, the mean-pooled sentence embedding of the student model is $(h_{S}^{si}, h_{S}^{ti})$. The MCL task takes parallel sentence pair $(h_{T}^{si}, h_{T}^{ti})$ as positive one, and other sentences in the same batch $\{(h_{S}^{si}, h_{S}^{ti})|j \in [1, N], j \neq i\}$ as negative samples. Considering that the MCL task needs to be combined with knowledge distillation. Unlike the previous work (Yang et al., 2019; Feng et al., 2020; Mao et al., 2021), the MCL task does not directly apply the temperature-scaled cross-entropy loss function.

Here, we introduce the implementation of the MCL task. For each pair of negative examples $(s_i, t_j)$ in the parallel corpus, the MCL task first unifies $(s_i, t_j)$ into the source language $(s_i, s_j)$, then uses the fine-grained distance between $h_{T}^{si}$ and $h_{T}^{sj}$ in the teacher model to push away the semantic different pair $(h_{S}^{si}, h_{S}^{sj})$ in the student model. For positive examples, the MCL task pull semantically similar pair $(h_{S}^{si}, h_{S}^{sj})$ together. The MCL task loss is (5),

$$\ell_1 = \frac{1}{N^2} \sum_{i}^{N} \sum_{j}^{N} \left[ \phi(h_{T}^{si}, h_{T}^{sj}) - \phi(h_{S}^{si}, h_{S}^{sj}) \right]^2,$$

where $\phi$ is the distance function. Following prior work (Yang et al., 2019; Feng et al., 2020), we set $\phi(x, y) = \cosine(x, y)$. we also add the knowledge distillation task for multilingual sentence representation learning. The knowledge distillation loss is defined as,

$$\ell_2 = \frac{1}{|N|} \sum_{i}^{N} \left[ (h_{T}^{si} - h_{S}^{si})^2 + (h_{T}^{ti} - h_{S}^{ti})^2 \right],$$

In stage 4, the total loss function is added by $\ell_1$ and $\ell_2$.

$$\ell_{stage4} = \ell_1 + \ell_2.$$
4 Experimental results

4.1 Evaluation setup

Dataset. The semantic text similarity (STS) task requires models to assign a semantic similarity score between 0 and 5 to a pair of sentences. Following Reimers and Gurevych (2020), we use the paradigm of two multilingual STS tasks, i.e., STS2017 (Cer et al., 2017) and STS2017-extend (Reimers and Gurevych, 2020), which contain three monolingual tasks (EN-EN, AR-AR, ES-ES) and six cross-lingual tasks (EN-AR, EN-ES, EN-TR, EN-FR, EN-IT, EN-NL).

Parallel corpus. In stage 1, stage 2 and stage 3, we use TED2020 (Reimers and Gurevych, 2020), WikiMatrix (Schwenk et al., 2021), Europarl (Koehn, 2005) and NewsCommentary (Tiedemann, 2012) as parallel corpus for training. In stage 4, TED2020 is used for contrastive learning. In this way, the student model first learns generalized multilingual knowledge and then possesses semantic similarity capabilities.

Metric. Spearman’s rank correlation \( \rho \) is reported in our experiments. Specifically, we first compute the cosine similarity score between two sentence embeddings, then calculate the Spearman rank correlation \( \rho \) between the cosine score and the golden score.

4.2 Implementation details

Mean pooling is applied to obtain sentence embeddings, and the max sequence length is set to 128. We use AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 2e-5 and a warm-up of 0.1. In stage 1, stage 2, and stage 3, the models are trained for 20 epochs with a batch size of 64, while in stage 4, the student model is trained for 60 epochs. The mBERT, XLM-R used in this work are base-size model obtained from Huggingface’s transformers package (Wolf et al., 2020), and the MiniLM refers to MiniLM-L12-H384\(^2\). Our implementation is based on UER (Zhao et al., 2019).

4.3 Performance comparison

We compare the model obtained from our multi-stage distillation with the previous state-of-the-art models, and results are shown in Table 1 and Table 2. In Pre-trained Model, mBERT(mean) and XLM-R(mean) are mean pooled mBERT and XLM-R models. mBERT-nli-stsb and XLM-R-nli-stsb are mBERT and XLM-R fine-tuned on the NLI and STS training sets. LASER and LaBSE are obtained from Artetxe and Schwenk (2019) and Feng et al. (2020). In Knowledge Distillation, we use the paradigm of Student\(\rightarrow\)Teacher to represent the student model distilled from the teacher model. There

Table 1: Spearman rank correlation \((\rho \times 100)\) between the cosine similarity of sentence representations and the gold labels for STS 2017 monolingual dataset. \(b\) indicates whether to use the Embedding Bottleneck strategy, \(bs\) indicates the hidden size of Bottleneck layer. \([RU]\) indicates the first \([RU]\) layers from the basic model are taken as Recurrent Unit, the recurrent times \(=\) basic model layers\(/ [RU]\).

<table>
<thead>
<tr>
<th>Model</th>
<th>AR-AR</th>
<th>ES-ES</th>
<th>EN-EN</th>
<th>Avg.</th>
<th>Embedding size</th>
<th>Encoder size</th>
</tr>
</thead>
<tbody>
<tr>
<td>mBERT</td>
<td>50.9</td>
<td>56.7</td>
<td>54.4</td>
<td>54.0</td>
<td>92.20M</td>
<td>85.05M</td>
</tr>
<tr>
<td>XLM-R</td>
<td>25.7</td>
<td>51.8</td>
<td>50.7</td>
<td>42.7</td>
<td>92.20M</td>
<td>85.05M</td>
</tr>
<tr>
<td>mBERT-nli-stsb</td>
<td>65.3</td>
<td>83.9</td>
<td>80.2</td>
<td>76.5</td>
<td>92.20M</td>
<td>85.05M</td>
</tr>
<tr>
<td>XLM-R-nli-stsb</td>
<td>64.4</td>
<td>83.1</td>
<td>78.2</td>
<td>75.3</td>
<td>92.20M</td>
<td>85.05M</td>
</tr>
<tr>
<td>LASER</td>
<td>68.9</td>
<td>79.7</td>
<td>77.6</td>
<td>75.4</td>
<td>23.56M</td>
<td>17.06M</td>
</tr>
<tr>
<td>LaBSE</td>
<td>69.1</td>
<td>80.8</td>
<td>79.4</td>
<td>76.4</td>
<td>385.28M</td>
<td>85.05M</td>
</tr>
</tbody>
</table>

Knowledge Distillation

<table>
<thead>
<tr>
<th>Model</th>
<th>AR-AR</th>
<th>ES-ES</th>
<th>EN-EN</th>
<th>Avg.</th>
<th>Embedding size</th>
<th>Encoder size</th>
</tr>
</thead>
<tbody>
<tr>
<td>mBERT(\rightarrow)SBERT-paraphrases</td>
<td>78.8</td>
<td>83.0</td>
<td>82.5</td>
<td>81.4</td>
<td>92.20M</td>
<td>85.05M</td>
</tr>
<tr>
<td>XLM-R(\rightarrow)SBERT-paraphrases</td>
<td>79.9</td>
<td>83.5</td>
<td>82.5</td>
<td>82.0</td>
<td>92.20M</td>
<td>85.05M</td>
</tr>
<tr>
<td>mBERT(\rightarrow)SBERT-paraphrases</td>
<td>79.1</td>
<td>86.5</td>
<td>88.2</td>
<td>84.6</td>
<td>92.20M</td>
<td>85.05M</td>
</tr>
<tr>
<td>DistilmBERT(\rightarrow)SBERT-paraphrases</td>
<td>77.7</td>
<td>85.8</td>
<td>88.5</td>
<td>84.0</td>
<td>92.20M</td>
<td>46.10M</td>
</tr>
<tr>
<td>XLM-R(\rightarrow)SBERT-paraphrases</td>
<td>79.6</td>
<td>86.3</td>
<td>88.8</td>
<td>84.6</td>
<td>192.40M</td>
<td>85.05M</td>
</tr>
<tr>
<td>MiniLM(\rightarrow)SBERT-paraphrases</td>
<td>80.3</td>
<td>84.9</td>
<td>85.4</td>
<td>83.5</td>
<td>96.21M</td>
<td>21.29M</td>
</tr>
</tbody>
</table>

LaBSE and LASER are obtained from Artetxe and Schwenk (2019) and Feng et al. (2020). In Knowledge Distillation, we use the paradigm of Student\(\rightarrow\)Teacher to represent the student model distilled from the teacher model. There

\(^2\)https://huggingface.co/microsoft/Multilingual-MiniLM-L12-H384
are two teacher models, i.e., SBERT-nli-stsb and SBERT-paraphrases, which are released by UKPLab\textsuperscript{3}. The former is fine-tuned on the English NLI and STS training sets, and the latter is trained on more than 50 million English paraphrase pairs. The student models include mBERT, XLM-R, DistilMBERT (Sanh et al., 2019) and MiniLM (Wang et al., 2021).

Table 1 and Table 2 show the evaluation results on monolingual and multilingual STS task, respectively. For the XLM-R, our method compresses the embedding size by 83.2\% with 0.3\% worse monolingual performance and 0.9\% worse cross-lingual performance, compresses the encoder size by 75\% with slightly higher (0.4\%) monolingual performance and 0.5\% worse cross-lingual performance. When compressing the embedding layer and the encoder simultaneously, the model size is reduced by 80.6\%, its monolingual performance drop by 2\% and cross-lingual performance drop by 4\%, but it still outperforms the pre-trained models.

For comparison with other distillation methods, MiniLM←SBERT-paraphrases is taken as a strong baseline. Our framework can further compress its embedding size by 66.7\% with 0.6\% worse in monolingual performance and 1.1\% worse in cross-lingual performance. Its encoder size is further compressed by 75\% with slightly higher monolingual (0.1\%) and cross-lingual (0.4\%) performance. In addition, our compressed XLM-R(b = True, bs = 128, |RU| = 12) achieves higher monolingual(0.8\%) and cross-lingual(1.9\%) performance with the same model size.

### 4.4 Ablation study

Among the three key strategies, multilingual contrastive learning (MCL) and parameter recurrent (Rec.) are two crucial mechanisms to improve model performance. The bottleneck is used to compress the model. In this section, ablation studies is performed to investigate the effects of MCL and Rec.. The effects of the bottleneck will be dis-

---

\textsuperscript{3}\url{https://github.com/UKPLab/sentence-transformers}
discussed in section 4.7.

XLM-R (b=True, bs=128, |RU| = 3) is selected as the basic model. We consider three different settings: 1) training without MCL task. 2) training without parameter recurrent. 3) training without both. The monolingual results and multilingual results are presented in Table 3 and Table 4.

It can be observed that: 1) without MCL task, the model performs poorer on the cross-lingual tasks. 2) without parameter sharing, the model performs poorer on all datasets. 3) MCL task can significantly improve the cross-lingual performance on EN-AR, EN-ES, EN-FR, EN-TR, EN-NL task. It can be concluded that both MCL task and parameter recurrent play a key role in our method.

4.5 Effect of contrastive learning

To investigate the effects of contrastive learning in stage 4, we select XLM-R (b=True, bs=128, |RU| = 3), modify the original objective in (5) into three different settings, namely, *Bool*, *CE* and w/o *CL*.

In the *Bool* setting, the soft label in (5) is replaced with hard label (0 or 1), as (8),

$$\ell_{\text{Bool}} = \frac{1}{N^2} \sum_{i,j} N \sum_{i,j} (\delta(h_{\phi i}, h_{\phi j}) - \phi(h_{S}^{i}, h_{S}^{j}))^2,$$

where \(\delta(x, y) = 1\), if \(x = y\), otherwise 0.

In the *CE* setting, the objective in (5) is replaced with temperature-scaled cross-entropy, as (9),

$$\ell_{\text{CE}} = \frac{1}{N^2} \sum_{i,j} \phi_T \log \frac{e^{\phi_{S}/\tau}}{\sum_{k=1}^{N} e^{\phi_{S}/\tau}},$$

where \(\phi_T = \cos(h_{\phi i}^{s}, h_{\phi j}^{s})\), \(\phi_S = \cos(h_{S}^{s}, h_{S}^{j})\), \(\tau = 0.05\) is a hyperparameter called temperature.

In the w/o *CL* setting, the contrastive learning is removed in Stage 4.

Table 5 presents the model performance of cross-lingual semantic similarity task with different settings. It can be observed that all the above training objectives can improve the model performance on the cross-lingual task, compared with the w/o *CL* settings. Model trained with (8) and (9) underperform that trained with (5), especially on EN-AR, EN-ES, EN-FR, EN-NL task.

We plot the convergence process of different settings in Figure 3. On EN-AR, EN-ES, EN-FR tasks, our setting outperform other settings. It is worth mentioning that on the EN-TR task, our setting underperform the *CE* setting according to Table 5. However, our setting reaches the same level as *CE* setting during the 30 to 40 epoch.

Table 5: Evaluation results of XLM-R (b = True, bs = 128, |RU| = 3) on the STS2017-extend cross-lingual task with different contrastive learning settings.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>78.0</td>
<td>79.8</td>
<td>73.9</td>
<td>80.5</td>
<td>82.1</td>
<td>80.3</td>
<td>81.2</td>
<td>79.4</td>
</tr>
<tr>
<td>CE</td>
<td>77.0↑</td>
<td>80.5↑</td>
<td>73.5↑</td>
<td>80.8↑</td>
<td>80.3↑</td>
<td>80.7↑</td>
<td>81.2↑</td>
<td>79.0↑</td>
</tr>
<tr>
<td>w/o CL</td>
<td>75.4↑</td>
<td>79.7↑</td>
<td>73.2↑</td>
<td>79.9↑</td>
<td>80.4↑</td>
<td>80.5↑</td>
<td>78.9↑</td>
<td>78.5↑</td>
</tr>
</tbody>
</table>

Table 6: Comparison of using different stage settings on monolingual and multilingual STS task. XLM-R is the basic model. The first three layers from XLM-R are taken as a Recurrent Unit, bottleneck hidden size is 128.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Avg. (Monolingual)</th>
<th>Avg. (Cross-lingual)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-stage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Initialize</td>
<td>78.1</td>
<td>71.1</td>
</tr>
<tr>
<td>+ Pre-Distillation</td>
<td>79.0</td>
<td>73.8</td>
</tr>
<tr>
<td>Multi-stage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>stage 1 + 2</td>
<td>48.4</td>
<td>20.8</td>
</tr>
<tr>
<td>stage 1 + 2 + 3</td>
<td>75.2</td>
<td>70.6</td>
</tr>
<tr>
<td>stage 1 + 2 + 3 + 4</td>
<td>82.6</td>
<td>79.4</td>
</tr>
</tbody>
</table>
Table 7: The performance of STS2017 monolingual task based on XLM-R(b=True, bs=128, |RU| = 3) and MiniLM(b=True, bs=128, |RU| = 3). We evaluated the effect of increasing bs or |RU|.

<table>
<thead>
<tr>
<th>Model</th>
<th>EN-AR</th>
<th>ES-ES</th>
<th>EN-EN</th>
<th>Avg.</th>
<th>Embedding size</th>
<th>Encoder size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teacher model=SBERT-paraphrases, Student model=XLM-R,</td>
<td>RU</td>
<td>= 3</td>
<td>b = True, bs = 128</td>
<td>76.7</td>
<td>84.5</td>
<td>86.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b = True, bs = 256</td>
<td>76.2</td>
<td>84.9</td>
<td>87.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b = False</td>
<td>79.9</td>
<td>86.8</td>
<td>88.4</td>
</tr>
</tbody>
</table>

| Teacher model=SBERT-paraphrases, Student model=XLM-R, | RU | = 3 | b = True, bs = 128 | 76.7 | 84.5 | 86.6 | 82.6 | 32.49M | 21.26M |
| | | | | 6 | 84.3 | 85.5 | | 84.3 | 32.49M | 85.05M |

| Teacher model=SBERT-paraphrases, Student model=MiniLM, | RU | = 3 | b = True, bs = 128 | 72.8 | 79.3 | 84.4 | 78.8 | 32.05M | 5.32M |
| | | | b = True, bs = 256 | 72.2 | 81.2 | 85.2 | 79.5 | 64.10M | 5.32M |
| | | | b = False | 79.9 | 85.3 | 86.7 | 83.6 | 96.21M | 5.32M |

Table 8: The performance of STS2017-extend cross-lingual task based on XLM-R(b=True, bs=128, |RU| = 3) and MiniLM(b=True, bs=128, |RU| = 3). We evaluated the effect of increasing bs or |RU|.

<table>
<thead>
<tr>
<th>Model</th>
<th>EN-AR</th>
<th>EN-DE</th>
<th>EN-TR</th>
<th>EN-ES</th>
<th>EN-FR</th>
<th>EN-IT</th>
<th>EN-NL</th>
<th>Avg.</th>
<th>Embedding size</th>
<th>Encoder size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teacher model=SBERT-paraphrases, Student model=XLM-R,</td>
<td>RU</td>
<td>= 3</td>
<td>b = True, bs = 128</td>
<td>78.0</td>
<td>79.8</td>
<td>73.9</td>
<td>80.5</td>
<td>82.1</td>
<td>80.3</td>
<td>81.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b = True, bs = 256</td>
<td>79.2</td>
<td>81.8</td>
<td>73.8</td>
<td>82.3</td>
<td>82.7</td>
<td>81.6</td>
<td>82.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b = False</td>
<td>81.1</td>
<td>84.3</td>
<td>79.8</td>
<td>82.6</td>
<td>84.5</td>
<td>84.8</td>
<td>85.4</td>
</tr>
</tbody>
</table>

| Teacher model=SBERT-paraphrases, Student model=XLM-R, | RU | = 3 | b = True, bs = 128 | 78.0 | 79.8 | 73.9 | 80.5 | 82.1 | 80.3 | 81.2 | 79.4 | 32.49M | 21.26M |
| | | | | 6 | 80.0 | 74.7 | 82.9 | 83.5 | 83.4 | 84.6 | 81.1 | 32.49M | 42.52M |
| | | | | 12 | 79.4 | 83.6 | 78.7 | 83.3 | 84.2 | 85.6 | 84.8 | 82.8 | 32.49M | 85.05M |

| Teacher model=SBERT-paraphrases, Student model=MiniLM, | RU | = 3 | b = True, bs = 128 | 73.0 | 76.0 | 63.7 | 71.4 | 71.8 | 72.1 | 74.7 | 71.8 | 32.05M | 5.32M |
| | | | b = True, bs = 256 | 69.7 | 77.1 | 66.2 | 73.5 | 73.5 | 74.3 | 75.6 | 72.6 | 64.10M | 5.32M |
| | | | b = False | 82.3 | 82.8 | 76.9 | 82.1 | 80.5 | 82.3 | 82.4 | 81.3 | 96.21M | 5.32M |

Table 6. But the benefit of increasing unit layers |b| is less than the previous unless the entire embedding layer is retained. The MiniLM(b=False) can outperform its original model in Table 1 and Table 2. But the benefit of increasing bs is not obvious unless the entire embedding layer is retained.

<table>
<thead>
<tr>
<th>Model</th>
<th>AR-AR</th>
<th>ES-ES</th>
<th>EN-EN</th>
<th>Avg.</th>
<th>Embedding size</th>
<th>Encoder size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teacher model=SBERT-paraphrases, Student model=XLM-R,</td>
<td>RU</td>
<td>= 3</td>
<td>b = True, bs = 128</td>
<td>76.7</td>
<td>84.3</td>
<td>86.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b = True, bs = 256</td>
<td>76.2</td>
<td>84.9</td>
<td>87.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b = False</td>
<td>79.9</td>
<td>86.8</td>
<td>88.4</td>
</tr>
</tbody>
</table>

4.6 Effect of multi-stages

To verify the effectiveness of multi-stages, we show the performance comparison of using different stage settings in Table 6. In the Single-stage setting, we first initialize the shrunk student model in two ways: (1) Random Initialize: Adding the untrained embedding bottleneck layers to the student model. (2) Pre-Distillation: The student model with bottleneck layer is initialized by distillation using XLM-R and the same corpus as section 4.1. Then we follow Reimers and Gurevych (2020) to align vector space between different languages. In the Multi-stage setting, the performance of the student model is reported after each stage.

As shown in Table 6, the Multi-stage setting outperforms the single-stage one, indicating that our multi-stage framework with an assistant model is effective. Adding stage3 and stage4 further improves the student model performance, suggesting that multi-stage training are necessary.

4.7 Effect of bottleneck and recurrent unit

In this section, we study the impact of embedding bottleneck and recurrent unit strategies on multilingual semantic learning. We consider three settings for each strategy, as shown in Table 7 and Table 8.

First, we found that both XLM-R and MiniLM perform better as the bottleneck hidden size bs increases. The performance is best when the entire embedding layer is retained, The MiniLM(b=False) can outperform its original model in Table 1 and Table 2. But the benefit of increasing bs is not obvious unless the entire embedding layer is retained.

Second, by increasing the number of recurrent unit layers |b|, XLM-R and MiniLM have been steadily improved on these two tasks. The increase in model size caused by the |b| is less than the bs. For example, the performance of MiniLM on cross-lingual tasks increased by 8%, while its size increased by 256%.
only increased by 15.9M.

Finally, it can be observed that when using the bottleneck layer ($b=True$), the model performance will increase steadily as $|RU|$ increases. The smaller the encoder hidden size, the more significant effect caused by $|RU|$ increasing ($\Delta\text{MiniLM}$$>$$\Delta\text{XLM-R}$). However, the increase of $bs$ can not improve performance significantly but make the embedding size larger. Therefore, an effective way to compress the multilingual model is reducing $bs$ while increasing $|RU|$. In this way, we shrink XLM-R by 58%, MiniLM by 55%, with less than 1.1% performance degradation.

5 Conclusion

In this work, we realize that the cross-lingual similarity matching task requires a large model size. To obtain a small-size model with cross-lingual matching ability, we propose a multi-stage distillation framework. Knowledge distillation and contrastive learning are combined in order to compress model with less semantic performance loss.

Our experiments demonstrate promising STS results with three monolingual and six cross-lingual pairs, covering eight languages. The empirical results show that our framework can shrink XLM-R or MiniLM by more than 50%. In contrast, the performance is only reduced by less than 0.6% on monolingual and 1.1% on cross-lingual tasks. If we slack the tolerated loss performance in 4%, the size of XLM-R can be reduced by 80%.

References


