An Ensemble Distillation Framework for Sentence Embeddings with Multilingual Round-Trip Translation

Tianyu Zong* **, Likun Zhang**

University of Chinese Academy of Sciences
{zongtianyu20,zhanglikun20}@mails.ucas.ac.cn

Abstract

In this work, we propose a novel unsupervised contrastive learning framework to improve state-of-the-art sentence embeddings. First, we train a set of contrastive submodels which take multilingual round-trip translation(RTT) as data augmentation. The RTT naturally changes the length of the same sentence and replaces Synonyms simultaneously. Then we incorporate them into a single model through knowledge distillation. Specifically, it takes an input sentence and predicts the ensemble output of all submodels via a contrastive objective. Thus we preserve nearly the same semantic expressiveness as the ensemble model without increasing the test cost. We evaluate our framework on standard semantic textual similarity (STS) tasks. Experimental results show the advantage of our framework that we achieve an average of 79.27% Spearman's correlation, a 3.02% improvement compared to the previous best results using BERT-base.

Introduction

In natural language processing tasks, sentence embedding representation is an efficient and general tool to convert raw text data into a numerical vector representation, which can be used in a wide range of natural language processing applications such as semantic similarity computation and text classification. Though BERT(Devlin et al. 2019) is of great significance in natural language processing, its performance in unsupervised semantic representation is not very well. Based on BERT, Sentence-BERT(Reimers and Gurevych 2019) uses a contrastive learning method similar to SimCLR(Chen et al. 2020) to implement sentence embedding. ConSERT(Yan et al. 2021) emphasizes the importance of data augmentation, SimCSE(Gao, Yao, and Chen 2021) uses the dropout that BERT itself has to achieve data augmentation, and DiffCSE(Chuang et al. 2022) learns sentence embeddings that are sensitive to the difference between the original sentence and the edited sentence. However, the above work focuses on optimizing a single model and is prone to over-fitting. Compared with a single model, the sentence embedding representation of the ensemble models can avoid falling into overfitting. The ensemble model is to combine multiple trained models and realizes multi-model fusion of test data in a certain way so that the final result can learn from each other's strengths, integrate the learning ability of each model, and improve the generalization ability of the final model.

To further improve the state-of-the-art sentence embeddings, in this paper, we propose a novel unsupervised contrastive learning framework based on ensemble learning and knowledge distillation. Ensemble learning solves a single prediction problem by building several submodels. It works by multiple submodels predicting independently. The combined prediction is strengthened by the combination of the models to prevent overfitting, so it is better than any single classification to make predictions. Knowledge distillation is a model compression method based on the "teacher-student network idea", which is simple and efficient. Suppose the two processes of ensemble learning and knowledge distillation can be effectively combined. In that case, it can not only improve the prediction accuracy through model integration but also realize model compression through distillation to improve the accuracy without increasing the model volume. Based on the above points, we propose a simple but efficient sentence embedding representation for unsupervised learning of semantic similarity computation. The basic idea is that we first combine the trained n submodels to obtain a relatively large ensemble model and then compress the model through knowledge distillation so that the ensemble model can be compressed without losing the sentence embedding representation ability of the ensemble model. The volume is compressed to the original $\frac{1}{n}$. In this process, the integrated model combines the advantages of each submodel and improves the generalization ability. Simultaneously, constrains each other to avoid the over-fitting phenomenon of each submodel to the greatest extent. Therefore, the actual performance of the ensemble model is better than any of the submodels. However, the ensemble model requires much computational overhead, so we compress it through knowledge distillation to make it consistent with a submodel in volume. Through this method, we not only achieve the ability to improve sentence embedding representation of the model but also do not increase the computational overhead of using the model, the best of both worlds.

In terms of data augmentation, SimCSE uses the dropout operation that BERT itself has: by sending a sentence into BERT twice, two different encoding vectors for the same

*Corresponding author

sentence are obtained. These two vectors are positive samples of each other to realize self-supervised(Xie et al. 2021) learning. ESimCSE(Wu et al. 2021a) pointed out that the model trained by SimCSE may mistakenly regard two sentences of the same length are of high similarity, which is definitely not the case. In data augmentation, SimCSE uses the dropout operation that BERT itself has: by sending a sentence into BERT twice, two different encoding vectors for the same sentence are obtained. These two vectors are positive samples of each other to realize self-supervised(Xie et al. 2021) learning. ESimCSE(Wu et al. 2021a) pointed out that the model trained by SimCSE may mistakenly regard two sentences of the same length are of high similarity, which is not the case. To mitigate the problem, random word copying(Ghiasi et al. 2020), synonym replacement(McCrae et al. 2019) are proposed. Inspired by the above work, we adopt round-trip translation(RTT) as the data augmentation method based on SimCSE. The natural advantage of RTT is that it is possible to change the length of the same sentence and can simultaneously replace synonyms. Before RTT, we finished cleaning and compensating the train set. By round-trip translating the training set from multiple languages and using the unsupervised learning scheme of SimCSE, we obtained several submodels for ensemble learning. The realization of each submodel is better than the original SimCSE.

In order to verify the effectiveness, we evaluate the submodels trained with data augmentation, ensemble learning, and knowledge distillation on seven test sets of STS12-16, STS-B, and SICK-R for the calculation of semantic similarity. The experimental results show that our model outperforms the previous state-of-the-art.

The main contributions of our work are as follows:

- Our proposed model introduces ensemble learning and knowledge distillation training method into the unsupervised semantic similarity computation field. Ensemble learning can maximize the advantages of each submodel. We propose a solution for model compression, which significantly reduces the model's size without losing the model's sentence embedding representation ability.
- Based on unsup-SimCSE, we propose a data augmentation method of cleaning, compensation, and RTT. This method can reduce the interference of meaningless training corpus on the model training process and generate many new training data, which can be used to train multiple submodels.
- We test our model on seven semantic similarity computation tasks, STS12-16, STS-Benchmark, and SICK-R, for semantic similarity. The experimental results show that the average scores of our proposed ensemble model and distilled model on BERT-base on these test sets reach 79.04% and 79.27%, respectively, compared with the previous state-of-the-art model SimCSE-BERT-base, it has increased by 2.79% and 3.02% respectively.

The rest of our paper is organized as follows. The related work introduces recent research on data augmentation, model integration, and knowledge distillation. Background of unsup-SimCSE introduces how unsupervised SimCSE is trained. The model section introduces our proposed data augmentation method and the overall structure of the model. The experimental section presents the performance of our model on the dataset. The experimental analysis section presents the test results of our model and ablation studies on the model architecture.

Related Work

Sentence Embedding

Sentence embeddings have been extensively studied in previous works. In the field of unsupervised learning or supervised learning, after BERT was proposed, a large number of models on BERT emerged, such as RoBERTa(Ott et al. 2019), IS-BERT(Zhang et al. 2020b), CT-BERT(Müller, Salathé, and Kummervold 2020), etc. Sentence-BERT(Reimers and Gurevych 2019) is trained using the Siamese network, CMLM(Yang et al. 2021) introduces conditional masks, PASER(Wu and Zhao 2022) compares generative and contrastive methods in sentence representation learning. Kim, Yoo, and Lee (2021) redesigned the contrast learning objectives and applied them to sentence representation learning. ConSERT(Yan et al. 2021) is randomly deleted. Combining multiple data augmentation methods such as adversarial generation. SimCSE(Gao, Yao, and Chen 2021) proposes a simple data augmentation scheme through dropout masks. Based on SimCSE, ESimCSE adopts the data augmentation scheme of synonym replacement and random copying of words. Finally, DiffCSE(Chuang et al. 2022) learns sentence embeddings sensitive to differences between original and edited sentences.

Data Augmentation

Data augmentation, to some extent, solves the problem of over-fitting caused by the lack of training data. In natural language processing, there have been a line of data augmentation methods. Synonym replacement(An et al. 2022) randomly replaces a few words from a sentence with their corresponding synonyms. Typically, Wordnet(Nhut Lam, Al Tarouti, and Kalita 2022) is usually used to look up the synonyms. Random deletion(Chuang et al. 2022) randomly deletes some words with a fixed probability in one sentence. Adversarial generation(Yilmaz 2022) adds adversarial examples to the training set to improve the robustness of the embedding model.

Ensemble Learning and Knowledge Distillation

Ensemble learning(Zhou 2014) incorporates multiple learners to give the final prediction. According to the generation method of individual learners, current ensemble learning techniques can be roughly divided into two categories. The first type is serialization, where individual learners have such a strong dependency on each other that it must be generated serially, such as Boosting(Schapire 1989). The second type is parallelization, where there is no reliability among learners, and they can be generated simultaneously, such as Bagging(Breiman 1996) and Random Forests(Pavlov 1997).

Knowledge distillation(Anil et al. (2018), Hinton, Vinyals, and Dean (2015)) is a technique of model compression, which mainly includes five types: network pruning, knowledge distillation, parameter quantization, struc-

ture design, and dynamic calculation. In our work, the traditional weighted combination of submodels is used in the ensemble learning part, and the "teacher-student modelHinton, Vinyals, and Dean (2015)" is used in the knowledge distillation part.

Background of Unsup-SimCSE

The unsupervised SimCSE model adopts a self-supervised learning method. Its data augmentation scheme uses BERT as the dropout function of the encoder itself. A sentence x is input into the encoder twice, and two embedded vector representations about this sentence can be obtained, denoted as $f(x)$ and $f(x)^+$. Since these two vectors represent the same sentence, these two vectors are positive samples of each other. During training, the encoding vectors of other sentences are used as negative samples of sentence x. The positive and negative samples are sent into the loss function of the contrastive learning so as to realize the self-supervised constraint, as shown in Eq. 1.

$$\mathcal{L}_i = -\log \frac{e^{sim\left(f(x_i), f(x_i)^+\right)/\tau}}{\sum_{j=1}^{N}\left(e^{sim(f(x_i), f(x_j))/\tau}\right)}, \tag{1}$$

where $f(x)$ represents the encoder output of for an input sentence x, $sim(\cdot,\cdot)$ calculates the cosine similarity between two vectors. τ is the temperature parameter. SimCSE empirically demonstrated that it performs best when $\tau = 0.05$, so we follow the same setting of $\tau = 0.05$ when training the submodels.

The Proposed Framework

Data Augmentation

The data augmentation method we adopt is to clean, compensate, and round-trip translate the original dataset[1]. In the original dataset, we found some non-English characters and many repetitions of punctuation marks. In order to avoid affecting the pre-training model's encoding of the sentences, we removed some non-English characters and some punctuation marks with a large number of repetitions. The removal of these training data resulted in a certain amount of missing training data, and to supplement this gap, we added some unsupervised data to the training set. Please see the Implementation Details section for cleansing and compensating the training data.

Round-trip Translation(Pham et al. 2021) is the translation of sentences from the source language into another language and back into the source language. Compared with data augmentation methods such as random replacement, deletion, and order shuffling, RTT not only realizes the replacement of synonyms of certain words in the data set but also preserves the actual semantics of the original data to the greatest extent. Another advantage of RTT is that it will not distort the meaning of sentences after data augmentation due to the replacement or deletion of some key words.

We propose two data augmentation methods based on RTT[2]. The first is data augmentation based on SimCSE, replacing the original data set of SimCSE with the RTT training set for unsupervised training. The second is also based on SimCSE. The cleaned and compensated training set is compared with the training set after RTT. Each sentence and its corresponding RTT sentence are used as a pair of positive examples, and then the training is carried out utilizing unsupervised learning. In the RTT process, we adopted the five most widely used languages in the world, including English-Chinese, English-French, English-Spanish, English-Dutch, and English-Russian.

Training Submodels Using Augmented Data

Since the SimCSE model is trained with a single training set, the generalization ability of SimCSE is limited. We use the training sets of five language RTTs to train separately and get five SimCSE-based submodels. In our model architecture, if the first data augmentation method is adopted, the training set that has been cleaned, compensated, and round-trip translated directly replaces the training set of SimCSE. Through the dropout operation of BERT, sentence x uses itself as a positive example for contrastive learning, takes other samples as negative examples, and sends them into the InfoNCE loss function. If the second data augmentation method is adopted, the data in the cleaned-compensated training set and the corresponding data in the cleaned-compensated-round-trip translated training set are used as a pair of positive examples, and other samples are used as negative examples for the InfoNCE loss function. Using these two RTT methods, we select the top five submodels plus a submodel trained using the training set with only cleansing and compensation operation without RTT.

Using Ensemble Learning and Knowledge Distillation

Through pre-training based on SimCSE, we get five submodels. By comparing the experimental results of each submodel. In order to avoid the influence of overfitting caused by a single training set on the training process of submodels to the greatest extent, we introduce the scheme of ensemble learning: submodels are integrated in a weighted manner to obtain an ensemble model. It can be formulated as:

$$f_I(x) = \sum_{k=1}^{K} \alpha_k \cdot f_k(x), \tag{2}$$

where f_k denotes the k-th submodel for ensemble, f_I represents the integrated model and $\alpha \in [0, 1]$ is the weights for each submodel.

We use two integration methods, respectively. The first method is to directly average the outputs of several submodels as the output of the integrated model. The second weighting method sends the average score of all submodels on the test set into the $softmax$ function to obtain a set of normalized weight coefficients. This set of coefficients is used to weight the submodels into the ensemble model.

[1]https://huggingface.co/datasets/princeton-nlp/datasets-for-simcse/resolve/main/wiki1m_for_simcse.txt

[2]We use the Google Translate system to obtain round-trip translation data.

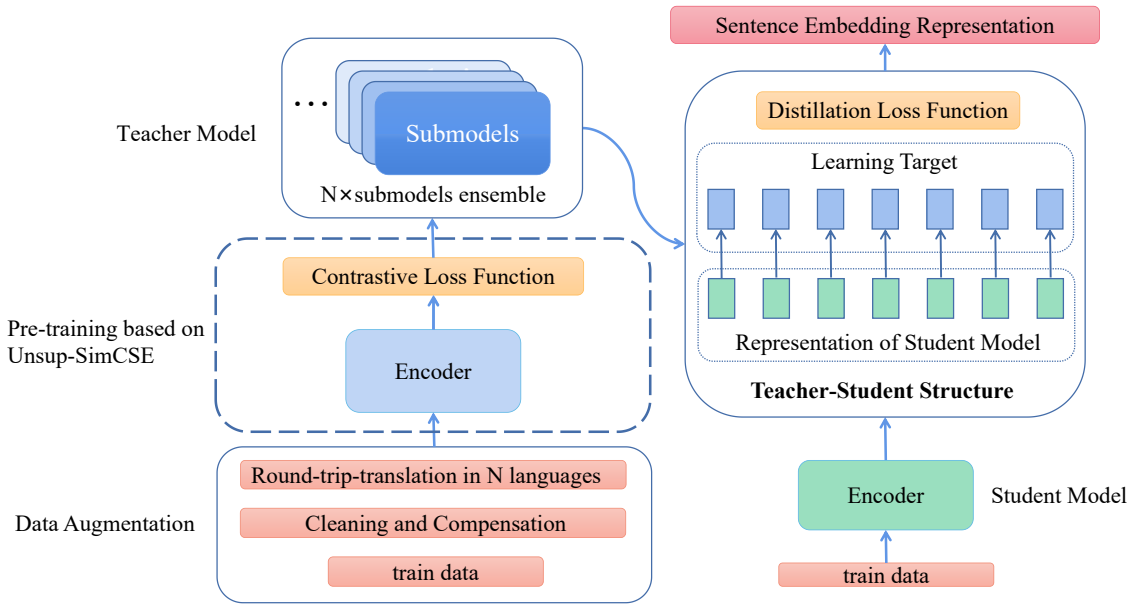

Figure 1: Training Pipeline of Our Proposed Framework

The sentence embedding representation ability of the ensemble model is better than that of the submodel. However, the ensemble model is jointly contributed by several submodels, so we need to reduce the volume of the ensemble model. Our goal is to compress the volume of the ensemble model into a model comparable in size to BERT-base without losing too much of the ensemble model's sentence embedding representation performance. The primary method of knowledge distillation is to use a teacher model to train the student model. Inspired by this view, we take the ensemble model as the teacher model and the pre-trained encoder, such as BERT-base or RoBERTa-base as the student model. Formally, we denote $f_T(\cdot) = f_I(\cdot)$ as the teacher model, $f_S(\cdot) = Encoder(\cdot)$ as the student model.

The student model is optimized towards outputting the same sentence embedding representations as the ensemble model, such that the sentence embedding representation of the student model is approaching the teacher model. During such distillation, we use the the training set of original SimCSE training set with only cleansing and compensation without RTT. Since the InfoNCE loss function can play a good role in unsupervised and self-supervised learning, it is not entirely suitable for such strongly supervised tasks. In contrast, the MAE loss function can measure the minimum absolute value deviation of the outputs of the two models. Intuitively, the more minor MAE loss, the closer the two models are. Therefore, we construct the loss function in the distillation stage by weighting the two loss functions. We give the loss function in the process of knowledge distillation, as shown in Eq. 3. MAE loss function is shown in Eq. 4, where M represents the $batch\ size$ during training.

$$\mathcal{L}_{\text{Distil}} = \lambda \mathcal{L}_{\text{InfoNCE}} + (1-\lambda)\mathcal{L}_{MAE}, \tag{3}$$

$$\mathcal{L}_{MAE} = \frac{1}{M}\sum_{i=1}^{M} \left\| f_T\left(x_i\right) - f_S\left(x_i\right) \right\|, \tag{4}$$

where λ represents the hyperparameter controlling the weights of two optimization objectives. Here we set $\lambda = 0.1$. During distillation, we give more significant weight to the MA loss function because the teacher model can achieve intense supervision of the student model through the MA loss function to quickly converge. The meaning of the InfoNCE loss function is to create a slack space during distillation. This slack space creates the possibility for students to outperform the teacher's training results.

Experimental Settings

Datasets

To verify the effectiveness of our proposed method, we implemented it on seven test sets for semantic similarity computation, including STS 2012-2016(Agirre et al. 2012, 2013, 2014, 2015, 2016), STS-Benchmark(Cer et al. 2017) and SICK-Relatedness(Marelli et al. 2014)[3]. Each sample consists of a sentence pair and the corresponding similarity score(0-5). We use Spearman similarity coefficient as the evaluation metric.[4].

Baselines

In order to demonstrate the sentence embedding representation ability of our model, we choose a lot of models for comparison, including some powerful state-of-the-art models. Such as ConSERT(Yan et al. 2021), SimCSE(Gao, Yao, and Chen 2021), ESimCSE(Wu et al. 2021a) and DiffCSE(Chuang et al. 2022), which have recently refreshed state-of-the-art. PT-BERT(Tan et al. 2022), CT-BERT(Müller, Salathé, and Kummervold 2020), IS-BERT(Kao and Lee 2021), SCPCSE(Tan, Yao, and Liu 2022) and DCPCSP(Jiang and Wang 2022) also performed very well. We also compare with a newly launched model

[3]Our code is based on https://github.com/yangjianxin1/SimCSE.

[4]All the experiments were implemented on an NVIDIA RTX3060 and an NVIDIA RTX3090Ti.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
BERT-base								
BERT(first-last avg.)♢	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
BERT-flow♢	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT-whitening♢	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-BERT♢	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
SG-OPT♢	66.87	80.13	71.23	81.56	77.17	77.23	68.16	74.62
PT-BERT♠	71.20	83.76	76.34	82.63	78.90	79.42	71.94	77.74
CT-BERT♢	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
ConSERT♠	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE♠	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
SimCSE+GS-InfoCSE♠	70.12	82.57	75.21	82.89	80.23	79.70	72.70	77.63
ArcCSE♠	72.08	84.27	76.25	82.32	79.54	79.92	72.89	78.11
DiffCSE♢	72.28	84.43	76.47	83.90	**80.54**	80.59	71.23	78.49
SCPCSE♣	64.28	78.97	70.51	78.45	75.71	76.33	68.73	73.28
DCPCSE♣	73.03	**85.18**	76.70	84.19	79.69	80.62	70.00	78.49
ESimCSE♠	73.40	83.27	**77.25**	82.66	78.81	80.17	72.30	78.27
Ours-Ensemble model	74.48	83.14	76.39	**84.45**	80.02	**81.97**	72.83	**79.04**
Ours-Distilled model	**74.50**	83.61	76.24	84.02	80.44	81.94	**74.16**	**79.27**
RoBERTa-base								
RoBERTa(first-last avg.)♢	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
DeCLUTR♢	52.41	75.19	65.52	77.12	78.63	72.41	68.62	69.99
RoBERTa-whitening♢	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
SimCSE♠	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
SimCSE+GS-InfoCSE♠	71.12	83.24	75.00	82.61	81.36	81.26	69.62	77.74
DiffCSE♢	70.05	83.43	75.49	82.81	**82.12**	82.38	71.19	78.21
ESimCSE♠	69.90	82.50	74.68	83.19	80.30	80.99	70.54	77.44
DCPCSE♣	70.57	81.91	74.60	82.90	80.96	**82.84**	71.70	77.93
CARDS♠	72.49	**84.09**	**76.19**	82.98	82.11	82.25	70.65	78.68
Ours-Ensemble model	**74.65**	82.15	75.61	**83.90**	81.11	82.05	73.88	**79.05**
Ours-Distilled model	71.04	81.08	77.04	83.08	81.96	82.36	**74.54**	**78.73**

Table 1: Performance of different sentence embedding models on the semantic similarity task (Spearman correlation). ⋄ refers to the results given by (Chuang et al. 2022). ♣ refers to the results given by (Jiang and Wang 2022). ♠ refers to results given by their original papers.

CARDS(Wang et al. 2022), SimCSE+GS-InfoCSE(Wu et al. 2021b) and ArcCSE(Zhang et al. 2022). For fairness, we also implement naive models such as BERT(first-last avg.)(Devlin et al. 2019), BERT-whitening(Su et al. 2021), BERT-flow(Li et al. 2020), CMLM(Yang et al. 2021), and DeCLUTR(Yang et al. 2020).

Implementation Details

In terms of data cleaning and compensation, we clean the training data whose character length is less than or equal to 3, and retain some important punctuation. Through experiments, we found that different RTT languages are sensitive to compensation data, so we compensated the unsupervised part of the SICK-R or STS-B training set when training the submodels. We did not supplement the supervised part of STS-R or STS-B training sets;. However, the training sets of STS-R and STS-B already gave the degree of similarity between the two sentences contained in a sentence pair, we still treat each sentence pair as two independent sentences to supplement, ensuring that we are still supplementing in an unsupervised manner.

In terms of pooling methods, we train the submodel using $[CLS]$, $First-last\,avg.$, $Top2\,avg.$ and $Avg.$ pooling, the ensemble model adopts the optimal pooling of each submodel, and when training the distilled model, we use the $[CLS]$ pooling.

In terms of RTT methods, we have done many experiments on the two RTT methods based on BERT and RoBERTa-base, respectively. For BERT-base submodels, the performance of submodels for different RTT languages is sensitive to both RTT methods, and we choose the optimal RTT method for each submodel. For the submodel on RoBERTa-base, no matter which RTT language is used, the second way of RTT is stronger than the first, so we adopt the second way of RTT method for the submodel on RoBERTa-base.

During the ensemble learning process, we try all combinations of the five submodels on BERT-base and RoBERTa-base. We found that the optimal combination on BERT-base is the $softmax$-weighted ensemble of submodels with English-Chinese, English-Dutch, English-French, English-Russian and English-Spanish. The optimal combination on RoBERTa-base is the average ensemble of submodels with English-Chinese and English-Dutch. For other hyperparameter settings, please refer to Appendix A. .

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
BERT-base								
Unsup-SimCSE	68.40	82.41	**74.38**	80.91	78.56	76.85	72.23	76.25
Ours-En-Chinese	71.65	78.69	71.97	**83.59**	77.72	79.22	72.37	76.46(+0.21)
Ours-En-Dutch	**73.05**	79.00	72.34	83.23	77.18	**79.95**	70.24	76.43(+0.18)
Ours-En-French	71.89	**82.52**	74.08	82.98	78.98	78.35	70.74	**77.08**(+0.83)
Ours-En-Russian	71.37	78.45	71.95	82.51	76.71	79.73	**73.10**	76.27(+0.02)
Ours-En-Spanish	69.74	81.55	74.31	82.19	**79.25**	78.52	71.29	76.69(+0.44)
RoBERTa-base								
Unsup-SimCSE	70.16	**81.77**	73.24	81.36	**80.65**	80.22	68.56	76.57
Ours-En-Chinese	**74.20**	78.63	72.22	82.16	76.67	79.45	**76.07**	77.06(+0.49)
Ours-En-Dutch	74.07	78.77	**74.17**	**82.22**	78.68	80.18	73.30	**77.34**(+0.77)
Ours-En-French	71.97	78.92	73.18	81.27	79.19	**80.26**	74.22	77.00(+0.43)
Ours-En-Russian	71.75	78.80	73.21	81.86	79.41	80.22	73.86	77.02(+0.45)
Ours-En-Spanish	71.65	78.20	72.30	82.01	78.76	79.53	73.67	76.59(+0.02)

Table 2: The average similarity score of SimCSE and submodels in different language RTT on test sets

Experimental Analysis

Comparison Results

We compare the performance of our proposed ensemble model and distilled model on BERT-base and RoBERTa-base with baseline models on seven test sets. From Table 1, our BERT-base ensemble model and distilled model achieved average scores of 79.04% and 79.27% on seven test sets, respectively, 2.79% and 3.02% higher than the SimCSE-BERT-base model. Compared with SimCSE-RoBERTa-base, our ensemble model achieves an average score of 79.05% on the test set, and the distilled model achieves an average score of 78.73%, an improvement of 2.48% and 2.16%, respectively.

Ablation Studies

Effect of Submodel Numbers In this section, we investigate the relationship between the performance of the ensemble model and the number of submodels when only the random seed setting is changed, without changing any other hyperparameter settings. To be fair, we use the original code provided by SimCSE[5].

We have conducted experiments about generating submodels by varying the random seed. We followed the hyperparameters given in SimCSE with BERT-base to generate 15 submodels by varying only the random seed, from which we randomly selected 2-12 submodels for ensemble, respectively. The results indicate that the integration of the models grows and then flattens out as the number of submodels increases, with the ensemble model performing close to the upper limit when the number of submodels is 5 (which is the main reason we used 5 RTT submodels for integration), with an average correct rate of 77.67% on the test set; the ensemble model perform best when the number of submodels is 10, with an average correct rate of 77.73%.

In addition, we have conducted ablation experiments on RoBERTa-base. The results show that the ensemble model performs essentially close to the best when the number of submodels is 4, the ensemble model perform best when the number of submodels is 10. Some of the experimental results are shown in the Figure 2.

[5]https://github.com/princeton-nlp/SimCSE

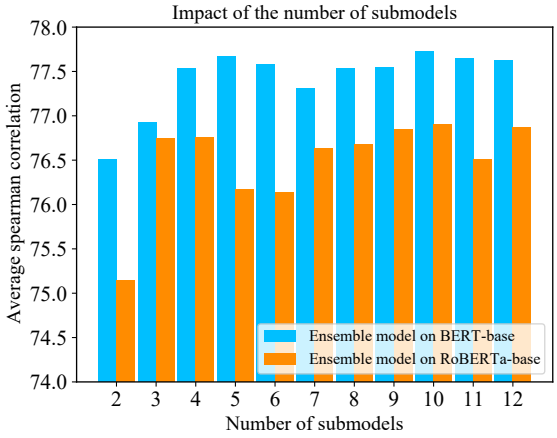

Figure 2: Effect of number of submodels

Effect of Data Augmentation. In this section, we investigate the impact on the training sub-model through different methods of data augmentation. The data augmentation method we use is "clean, compensate, and RTT", so we ablate the use of "clean only", "compensate only" and "RTT only", and all the two-by-two combinations between them. We used BERT-base as the pre-trained language model, and the English-French RTT data were obtained by Google Translate. The experimental results are shown in Table 5. It can be seen that the effect of the data augmentation method we used is optimal.

Effect of Pooling Method. Following SimCSE, our work also gives the performance of the model on different pooling methods, including $[CLS]$, $Top2\,avg$, $First-last\,avg$ and Avg. Due to space limitations, the pooling performance of the submodel is given in the Appendix B. Here we give the sentence embedding performance of the distilled model in the above pooling in Table 6, which shows the scores and average scores of our BERT-base distilled model on STS12-16, STS-B, SICK-R through four pooling methods. The $[CLS]$ pooling method performs the best.

Effect of Distillation Temperature Factor. We research the impact of temperature factor when training the distilled

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
ConSERT-large♠	70.69	82.96	74.13	82.78	76.66	77.53	70.37	76.45
SimCSE-BERT-large♣	70.88	**84.16**	**76.43**	84.50	79.76	79.26	73.88	78.41
SimCSE-RoBERTa-large♠	72.86	83.99	75.62	**84.77**	**81.80**	**81.98**	71.26	78.90
Ours-Distilled-BERT-base	**74.50**	83.61	76.24	84.02	80.44	81.94	**74.16**	**79.27**

Table 3: Comparison between our distilled model on BERT-base and baseline models on BERT/RoBERTa-large. ♣ refers to the results given by (Zhang et al. 2022). ♠ refers to results given by their original papers.

Loss for distillation	STS12-16	STS-B
MSE	79.13	80.87
MSE + InfoNCE	79.26	81.09
KLDiv	79.22	81.26
KLDiv + InfoNCE	79.32	81.53
SmoothL1	78.85	80.61
SmoothL1 + InfoNCE	78.98	80.95
MAE	79.23	81.87
MAE + InfoNCE(ours)	**79.76**	**81.94**

Table 4: Effect of loss function for distillation

Methods of Data Augmentation	Avg.
Dropout(SimCSE)	76.25
Dropout+clean	76.77
Dropout+compensate	76.67
Dropout+RTT	76.89
Dropout+clean+compensate	76.10
Dropout+clean+RTT	75.88
Dropout+compensate+RTT	76.56
Dropout+clean+compensate+RTT(ours)	**77.08**

Table 5: This table presents the impact caused by different data augmentation methods.

model on BERT-base. In Table 7, we studied the effect of the temperature factor on the distillation process. When other parameters were consistent, the distillation effect performed best on the STS-B test set when $\tau = 0.3$.

Effect of Distillation Loss Functions. We test the effect of MAE loss and InfoNCE loss on model distillation on BERT-base and RoBERTa-base models. We found that the distillation effect of using MAE only and InfoNCE only was slightly worse than using MAE with InfoNCE weighting. This result is because the fitting effect of MAE is better than InfoNCE. However, if only MAE is used, the student model will be too "tight" in the process of fitting the teacher. Suppose a specific weight of InfoNCE is added as a "relaxation factor." In that case, the model has a certain margin of slight variation in the distillation process, which may be positive feedback to the training of the distilled model, so this explains why the sentence embedding performance of the BERT-base distilled model is better than the ensemble model. However, if only InfoNCE is used, this results in a "relaxation factor" that is too large; the performance is worse than MAE only or MAE and InfoNCE weighted.

Pooling	STS12-16	STS-B	SICK-R
First-last avg.	78.45	80.30	71.47
Top2 avg.	79.28	81.44	73.59
Avg.	79.06	80.74	72.53
[CLS]	**79.76**	**81.94**	**74.16**

Table 6: Effect of pooling methods

τ	0.05	0.1	**0.3(ours)**	0.5	1.0
STS-B	81.27	81.40	**81.66**	81.43	81.21

Table 7: Effect of temperature factor

In addition, we also apply other loss functions such as MSE(Ren et al. 2022), KL-Divergence(Huang et al. 2019)), Smooth$L1$(Zhang et al. 2020a) to the ablation experiments, as shown in Table 4. Experimental results show that MAE and InfoNCE loss function weighting are the most suitable for distillation learning.

Discussion

Due to space constraints, we will present the advantages and disadvantages of our model and the baseline model when using more different pre-trained language models, evaluation metrics, and hyperparameter settings in appendix A and C. With limited computational resources, we did not implement our model on RoBERT-large and BERT-large. However, we have achieved the state-of-the-art results with only BERT-base and RoBERT-base models. Surprisingly, our model on BERT-base even outperforms SimCSE-BERT-large, SimCSE-RoBERTa-large, and ConSERT-large, as shown in Table 3. Therefore, we believe our model can also achieve state-of-the-art results on BERT-large and RoBERTa-large.

Conclusion

In this work, we propose a sentence embedding framework based on multilingual round-trip translation ensemble and knowledge distillation, significantly improving the semantic expressiveness of the state-of-the-art sentence embedding model. Combining ensemble learning and knowledge distillation, we incorporate sentence embeddings from multiple contrastive models trained with round-trip translation sentence pairs in different languages into one single model, improving the efficiency at test time. The ablation experiments demonstrate the effectiveness and efficiency of the proposed multilingual round-trip translation ensemble technique and knowledge distillation. We believe that our sentence embedding models build a foundation for various downstream application scenarios and will motivate more ideas for researchers in the field of natural language processing.

References

Agirre, E.; Banea, C.; Cardie, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W.; Lopez-Gazpio, I.; Maritxalar, M.; Mihalcea, R.; Rigau, G.; Uria, L.; and Wiebe, J. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 252–263. Denver, Colorado: Association for Computational Linguistics.

Agirre, E.; Banea, C.; Cardie, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W.; Mihalcea, R.; Rigau, G.; and Wiebe, J. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 81–91. Dublin, Ireland: Association for Computational Linguistics.

Agirre, E.; Banea, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Mihalcea, R.; Rigau, G.; and Wiebe, J. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 497–511. San Diego, California: Association for Computational Linguistics.

Agirre, E.; Cer, D.; Diab, M.; and Gonzalez-Agirre, A. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 385–393. Montréal, Canada: Association for Computational Linguistics.

Agirre, E.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; and Guo, W. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, 32–43. Atlanta, Georgia, USA: Association for Computational Linguistics.

An, C.; Han, E.; Noh, D.; Kwon, O.; Lee, S.; and Han, H. 2022. Building Korean Sign Language Augmentation (KoSLA) Corpus with Data Augmentation Technique. *arXiv e-prints*, arXiv:2207.05261.

Anil, R.; Pereyra, G.; Passos, A.; Ormandi, R.; Dahl, G. E.; and Hinton, G. E. 2018. Large scale distributed neural network training through online distillation. In *International Conference on Learning Representations*.

Breiman, L. 1996. Bagging predictors. *Machine Learning*.

Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 1–14. Vancouver, Canada: Association for Computational Linguistics.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv e-prints*, arXiv:2002.05709.

Chuang, Y.-S.; Dangovski, R.; Luo, H.; Zhang, Y.; Chang, S.; Soljacic, M.; Li, S.-W.; Yih, W.-t.; Kim, Y.; and Glass, J. 2022. DiffCSE: Difference-based Contrastive Learning for Sentence Embeddings. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.-Y.; Cubuk, E. D.; Le, Q. V.; and Zoph, B. 2020. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. *arXiv e-prints*, arXiv:2012.07177.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv e-prints*, arXiv:1503.02531.

Huang, Y.; Song, T.; Xu, J.; Chen, Y.; and Zhuang, X. 2019. KLDivNet: An unsupervised neural network for multi-modality image registration. *arXiv e-prints*, arXiv:1908.08767.

Jiang, Y.; and Wang, W. 2022. Deep Continuous Prompt for Contrastive Learning of Sentence Embeddings. arXiv:2203.06875.

Kao, W.-T.; and Lee, H.-Y. 2021. Is BERT a Cross-Disciplinary Knowledge Learner? A Surprising Finding of Pre-trained Models' Transferability. *arXiv e-prints*, arXiv:2103.07162.

Kim, T.; Yoo, K. M.; and Lee, S.-g. 2021. Self-Guided Contrastive Learning for BERT Sentence Representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2528–2540. Online: Association for Computational Linguistics.

Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; and Li, L. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Marelli, M.; Menini, S.; Baroni, M.; Bentivogli, L.; Bernardi, R.; and Zamparelli, R. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Language Resources and Evaluation*.

McCrae, J. P.; Rademaker, A.; Bond, F.; Rudnicka, E.; and Fellbaum, C. 2019. English WordNet 2019 – An Open-Source WordNet for English. In *Proceedings of the 10th Global Wordnet Conference*, 245–252. Wroclaw, Poland: Global Wordnet Association.

Müller, M.; Salathé, M.; and Kummervold, P. E. 2020. COVID-Twitter-BERT: A Natural Language Processing

Model to Analyse COVID-19 Content on Twitter. *arXiv e-prints*, arXiv:2005.07503.

Nhut Lam, K.; Al Tarouti, F.; and Kalita, J. 2022. Automatically constructing Wordnet synsets. *arXiv e-prints*, arXiv:2208.03870.

Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Pavlov, Y. L. 1997. Random Forests. *Karelian Centre Russian Acad.sci.petrozavodsk*, 45(1): 5–32.

Pham, H.; Wang, X.; Yang, Y.; and Neubig, G. 2021. Meta Back-Translation. In *International Conference on Learning Representations*.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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)*, 3982–3992. Hong Kong, China: Association for Computational Linguistics.

Ren, J.; Zhang, M.; Yu, C.; and Liu, Z. 2022. Balanced MSE for Imbalanced Visual Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7926–7935.

Schapire, R. E. 1989. The strength of weak learnability. *Proceedings of the Second Annual Workshop on Computational Learning Theory*, 5(2): 197–227.

Su, J.; Cao, J.; Liu, W.; and Ou, Y. 2021. Whitening Sentence Representations for Better Semantics and Faster Retrieval. *arXiv preprint arXiv:2103.15316*.

Tan, H.; Shao, W.; Wu, H.; Yang, K.; and Song, L. 2022. A Sentence is Worth 128 Pseudo Tokens: A Semantic-Aware Contrastive Learning Framework for Sentence Embeddings. In *Findings of the Association for Computational Linguistics: ACL 2022*, 246–256. Association for Computational Linguistics.

Tan, W.; Yao, Q.; and Liu, J. 2022. Two-Stage COVID19 Classification Using BERT Features. *arXiv e-prints*, arXiv:2206.14861.

Wang, W.; Ge, L.; Zhang, J.; and Yang, C. 2022. Improving Contrastive Learning of Sentence Embeddings with Case-Augmented Positives and Retrieved Negatives. *arXiv e-prints*, arXiv:2206.02457.

Wu, B.; and Zhao, H. 2022. Generative or Contrastive? Phrase Reconstruction for Better Sentence Representation Learning. *arXiv e-prints*, arXiv:2204.09358.

Wu, X.; Gao, C.; Zang, L.; Han, J.; Wang, Z.; and Hu, S. 2021a. ESimCSE: Enhanced Sample Building Method for Contrastive Learning of Unsupervised Sentence Embedding. *arXiv e-prints*, arXiv:2109.04380.

Wu, X.; Gao, C.; Zang, L.; Han, J.; Wang, Z.; and Hu, S. 2021b. Smoothed Contrastive Learning for Unsupervised Sentence Embedding. *arXiv e-prints*, arXiv:2109.04321.

Xie, Z.; Lin, Y.; Yao, Z.; Zhang, Z.; Dai, Q.; Cao, Y.; and Hu, H. 2021. Self-Supervised Learning with Swin Transformers. *arXiv e-prints*, arXiv:2105.04553.

Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; and Xu, W. 2021. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 5065–5075.

Yang, Z.; Yang, Y.; Cer, D.; Law, J.; and Darve, E. 2020. Universal Sentence Representation Learning with Conditional Masked Language Model. *arXiv e-prints*, arXiv:2012.14388.

Yang, Z.; Yang, Y.; Cer, D.; Law, J.; and Darve, E. 2021. Universal Sentence Representation Learning with Conditional Masked Language Model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6216–6228. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Yilmaz, I. 2022. FIGO: Enhanced Fingerprint Identification Approach Using GAN and One Shot Learning Techniques. *arXiv e-prints*, arXiv:2208.05615.

Zhang, H.; Chang, H.; Ma, B.; Wang, N.; and Chen, X. 2020a. Dynamic R-CNN: Towards High Quality Object Detection via Dynamic Training. *arXiv e-prints*, arXiv:2004.06002.

Zhang, Y.; He, R.; Liu, Z.; Lim, K. H.; and Bing, L. 2020b. An Unsupervised Sentence Embedding Method by Mutual Information Maximization. *arXiv e-prints*, arXiv:2009.12061.

Zhang, Y.; Zhu, H.; Wang, Y.; Xu, N.; Li, X.; and Zhao, B. 2022. A Contrastive Framework for Learning Sentence Representations from Pairwise and Triple-wise Perspective in Angular Space. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4892–4903. Dublin, Ireland: Association for Computational Linguistics.

Zhou, Z.-H. 2014. Ensemble methods. *Combining pattern classifiers. Wiley, Hoboken*, 186–229.