TRANS-ENCODER: Unsupervised sentence-pair modelling through self- and mutual-distillations

Fangyu Liu[1*] **Yunlong Jiao**[2] **Jordan Massiah**[2] **Emine Yilmaz**[2] **Serhii Havrylov**[2]
[1]University of Cambridge [2]Amazon
fl399@cam.ac.uk {jyunlong,jormas,eminey,havrys}@amazon.com

Abstract

In NLP, a large volume of tasks involve pairwise comparison between two sequences (e.g. sentence similarity and paraphrase identification). Predominantly, two formulations are used for sentence-pair tasks: bi-encoders and cross-encoders. Bi-encoders produce fixed-dimensional sentence representations and are computationally efficient, however, they usually underperform cross-encoders. Cross-encoders can leverage their attention heads to exploit inter-sentence interactions for better performance but they require task fine-tuning and are computationally more expensive. In this paper, we present a completely unsupervised sentence representation model termed as TRANS-ENCODER that combines the two learning paradigms into an iterative joint framework to simultaneously learn enhanced bi- and cross-encoders. Specifically, on top of a pre-trained Language Model (PLM), we start with converting it to an unsupervised bi-encoder, and then alternate between the bi- and cross-encoder task formulations. In each alternation, one task formulation will produce pseudo-labels which are used as learning signals for the other task formulation. We then propose an extension to conduct such self-distillation approach on multiple PLMs in parallel and use the average of their pseudo-labels for mutual-distillation. TRANS-ENCODER creates, to the best of our knowledge, the first completely unsupervised cross-encoder and also a state-of-the-art unsupervised bi-encoder for sentence similarity. Both the bi-encoder and cross-encoder formulations of TRANS-ENCODER outperform recently proposed state-of-the-art unsupervised sentence encoders such as Mirror-BERT and SimCSE by up to 5% on the sentence similarity benchmarks.

1 Introduction

Comparing pairwise sentences is fundamental to a wide spectrum of tasks in NLP such as information retrieval (IR), natural language inference (NLI), sentence textual similarity (STS) and clustering. Two general architectures usually used for sentence-pair modelling are bi-encoders and cross-encoders.

In a cross-encoder, two sequences are concatenated and sent into the model (usually deep Transformers like BERT/RoBERTa) in one pass. The attention heads of Transformers could directly model the inter-sentence interactions and output a classification/relevance score. However, a cross-encoder needs to recompute the encoding for different combinations of sentences in each unique sequence pair, resulting in a heavy computational overhead. It is thus impractical in tasks like IR and clustering where massive pairwise sentence comparisons are involved. Also, task fine-tuning is always required for converting PLMs to cross-encoders. By contrast, in a bi-encoder, each sequence is encoded separately and mapped to a common embedding space for similarity comparison. The encoded

[*] Work done during internship at Amazon.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

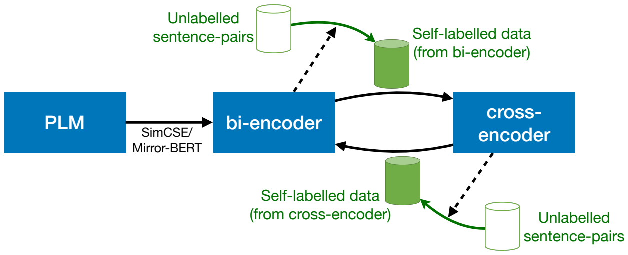


Figure 1: A graphical illustration of the self-distillation learning scheme in TRANS-ENCODER. Notice that the blue boxes represent the same model trained sequentially.

sentences can be cached and reused. A bi-encoder is thus much more efficient. Also, the output of a bi-encoder can be used off-the-shelf as sentence embeddings for other downstream tasks. That said, it is well-known that bi-encoders underperform cross-encoders (Humeau et al., 2020; Thakur et al., 2021) since the former could not explicitly model the interactions between sentences but could only compare them in the embedding space in a *post hoc* manner.

In this work, we ask the question: can we leverage the advantages of both bi- and cross-encoders and bootstrap knowledge from them in an unsupervised manner? Our proposed TRANS-ENCODER addresses this question with the following intuition: As a starting point, we can use bi-encoder representations to tune a cross-encoder. With more powerful inter-sentence modelling, the cross-encoder should resurface more knowledge from the PLMs than the bi-encoder given the same data. In turn, the more powerful cross-encoder can distil its knowledge back to the bi-encoder. We can repeat this cycle to iteratively bootstrap from both the bi- and cross-encoders.

2 TRANS-ENCODER

The general idea of TRANS-ENCODER is simple yet extremely effective. We first transform an off-the-shelf PLM to a strong bi-encoder, serving as an initialisation point. Then, the bi-encoder produces pseudo-labels and the PLM subsequently learns from these pseudo-labels in a cross-encoder manner. Consecutively, the cross-encoder further produces more accurate pseudo-labels for bi-encoder learning. This self-distillation process is visualised in Fig. 1. Then, we propose an extension called mutual-distillation that stabilises training and boosts the encoder performance even more.

Transform PLMs into Effective Bi-encoders. Off-the-shelf PLMs are unsatisfactory bi-encoders. To have a reasonably good starting point, we leverage a simple contrastive tuning procedure to transform existing PLMs to bi-encoders. This approach is concurrently proposed in both Mirror-BERT (Liu et al., 2021) and SimCSE (Gao et al., 2021). We use the checkpoints released by them directly. Please refer two the two papers for details.

Self-distillation: Bi- to Cross-encoder. After obtaining a sufficiently good bi-encoder, we leverage it to label sentence pairs sampled from the task of interest. Specifically, for a given sentence-pair (`sent1`, `sent2`), we input them to the bi-encoder separately and get two embeddings (we use the embedding of `[CLS]` from the last layer). The cosine similarity between them is regarded as their relevance score. In this way we have constructed a self-labelled sentence-pair scoring dataset in the format of (`sent1`, `sent2`, score). We then employ the same model architecture to learn from these score, but with a cross-encoder formulation. The cross-encoder weights are initialised from the original PLM. For the sentence-pair (`sent1`, `sent2`), we concatenate them to produce "`[CLS] sent1 [SEP] sent2 [SEP]`" and input it to the cross-encoder. A linear layer (newly initialised) then map the sequence's encoding (embedding of the `[CLS]` token) to a scalar. The learning objective of the cross-encoder is minimising the KL divergence between its predictions and the self-labelled scores from the bi-encoder:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{n=1}^{N}\Big(y_n \cdot \log(\sigma(x_n)) + (1-y_n)\cdot \log(1-\sigma(x_n))\Big) \tag{1}$$

where N is the data-batch size; $\sigma(\cdot)$ is the sigmoid activation; x_n is the prediction of the cross-encoder; y_n is the self-labelled ground-truth score from the bi-encoder.

Note that while the cross-encoder is essentially learning from the data produced by itself (in a bi-encoder form), usually, it outperforms the original bi-encoder on held-out data. The cross-encoder directly discovers the similarity between two sentences through its attention heads, finding more accurate cues to justify the relevance score. The ability of discovering such cues could then generalise

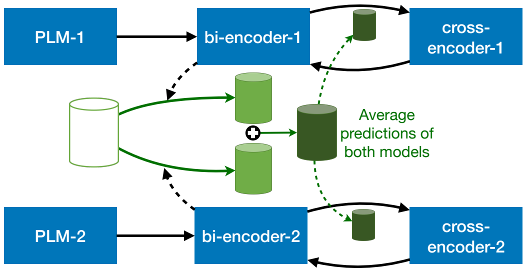


Figure 2: A graphical illustration of the mutual-distillation learning scheme in TRANS-ENCODER. Note that, for simplicity, only the bi- to cross-encoder mutual-distillation is shown. We also conduct cross- to bi-encoder mutual-distillation in the same manner.

to unseen data, resulting in stronger sentence-pair scoring capability than the original bi-encoder. From a knowledge distillation perspective, we can view the bi- and cross-encoder as the teacher and student respectively. In this case the student outperforms the teacher, not because of stronger model capacity, but smarter task formulation. By leveraging this simple yet powerful observation, we are able to design a learning scheme that iteratively boosts the performance of both bi- and cross-encoder.

Self-distillation: Cross- to bi-encoder. With the more powerful cross-encoder at hand, a natural next step is distilling the extra gained knowledge back to a bi-encoder form, which is more useful for downstream tasks. Besides, and more importantly, a better bi-encoder could produce even more self-labelled data for cross-encoder learning. In this way we could repeat bi- to cross-encoder distillation and *vice versa*, continually bootstrapping the encoder performance.

We create the self-labelled sentence-scoring dataset in the same way as bi-to-cross distillation except that the cross-encoder is used for producing the relevance score. The bi-encoder is initialised with the weights after Mirror-BERT/SimCSE training. For every sentence pair, two sentence embeddings are produced separately by the bi-encoder. The cosine similarity between the two embeddings are regarded as their predictions of the relevance score. The aim is to regress the predictions to the self-labelled scores by the cross-encoder. We use a mean square error (MSE) loss: $\mathcal{L}_{\text{MSE}} = -\frac{1}{N}\sum_{n=1}^{N}\Big(x_n - y_n\Big)^2$ where N is the batch size; x_n is the cosine similarity between a sentence pair; y_n is the self-labelled ground-truth. In experiments, we will show that this resulting bi-encoder is more powerful than the initial bi-encoder. Sometimes, the bi-encoder will even outperform its teacher (i.e. the cross-encoder).

Mutual-distillation. The aforementioned self-learning approach has a drawback: since the model regresses to its previous predictions, it tends to reinforce its errors. To mitigate this problem, we design a simple mutual-distillation approach to smooth out the errors/biases originated from PLMs. Specifically, we conduct self-distillation on multiple PLMs in parallel (for brevity, we use two in this paper: BERT and RoBERTa, however the framework itself is compatible with any number of PLMs). Each PLM does not communicate/synchronise with each other except when producing the self-labelled scores. In mutual-distillation, we use the average predictions of all models as the ground-truth for the next round of learning. A graphical illustration is shown in Fig. 2.

In the following, we call all self-distillation models TRANS-ENCODER (or TENC for short); all mutual-distillation models TRANS-ENCODER-mutual (or TENC-mutual for short).

3 Results and Discussion

STS and binary classification results (Tab. 1; Tab. 2). The main results for STS are listed in Tab. 1. Compared with the baseline SimCSE, TRANS-ENCODER has brought significant improvements across the board. With various variants of the SimCSE as the base model, TRANS-ENCODER consistently enhances the average score by approximately 4-5%.[2] Self-distillation usually brings an improvement of 2-3%. Further, mutual-distillation brings another 1-3%. For binary classification tasks (i.e., On QQP, QNLI, and MRPC), we observe similar trends as the STS tasks (Tab. 2).

Domain transfer setup (Tab. 3). One of the key questions we are keen to find out is how much of TRANS-ENCODER's success on in-domain tasks generalises/transfers to other tasks/domains. Specifically, does TRANS-ENCODER create universally better bi- and cross-encoders, or is it only

[2] Similar trends observed on Mirror-BERT, see Appendix (App. §A.3).

dataset→	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	avg.
SimCSE-RoBERTa-base	68.34	80.96	73.13	80.86	80.61	80.20	68.62	76.10
+ TEnc (bi)	73.36	82.47	76.39	83.96	82.67	82.05	67.63	78.36
+ TEnc (cross)	72.59	83.24	76.83	84.20	82.82	82.85	69.51	78.86
+ TEnc-mutual (bi)	75.01	85.22	78.26	85.16	83.22	83.88	**72.56**	80.47
+ TEnc-mutual (cross)	**76.37**	**85.87**	**79.03**	**85.77**	**83.77**	**84.65**	72.62	**81.15**
SimCSE-RoBERTa-large	71.40	84.60	75.94	84.36	82.22	82.67	71.23	78.92
+ TEnc (bi)	77.92	86.69	79.29	87.23	84.22	86.10	68.36	81.40
+ TEnc (cross)	**78.32**	86.20	79.61	86.88	82.93	84.48	67.90	80.90
+ TEnc-mutual (bi)	78.15	**88.39**	81.76	88.38	84.95	86.55	**72.31**	**82.93**
+ TEnc-mutual (cross)	78.28	88.31	**81.94**	**88.63**	**85.03**	**86.70**	71.63	**82.93**

Table 1: English STS. Spearman's ρ rank correlations are reported. TEnc models use only self-distillation while TEnc-mutual models use mutual-distillation as well. Blue and red denotes mutual-distillation models that are trained in the base and large group respectively. Models without colour are not co-trained with any other models. Full table see Appendix (Tab. 8).

dataset→	QQP	QNLI	MRPC	avg.
SimCSE-BERT-base	80.38	71.38	75.02	75.59
+ TEnc (bi)	82.10	75.30	75.71	77.70
+ TEnc (cross)	82.10	75.61	76.21	77.97
+ TEnc-mutual (bi)	84.00	76.93	76.62	79.18
+ TEnc-mutual (cross)	**84.29**	**77.11**	**77.77**	**79.72**

Table 2: Binary classification task results. AUC scores are reported. We only demonstrate results for one model for brevity. Full table can be found in Appendix (Tab. 9).

task/domain-specific? To answer the question, we directly test models trained with the STS task data on binary classification tasks. For bi-encoders, i.e. TEnc (bi), the results are inconsistent across setups. These results hint that training on in-domain data is important for optimal performance gains for unsupervised bi-encoders. This is in line with the finding of (Liu et al., 2021). However, for cross-encoders, surprisingly, we see extremely good transfer performance. W/ or w/o mutual-distillation, TEnc (cross) models outperform the SimCSE baselines by large margins, despite the fact that tasks like QQP and QNLI are of a very different domain compared with STS. In fact, they are sometimes even better than models tuned on the in-domain task data (c.f. Tab. 9). The finding hints that the cross-encoder architecture is by design very suitable for sentence-pair modelling (i.e. strong inductive bias is already in the architecture design), and once its sentence-pair modelling capability is 'activated', it is an extremely powerful universal representation that can be transferred to other tasks.

4 Conclusion

We propose Trans-Encoder, an unsupervised approach of training bi- and cross-encoders for sentence-pair tasks. The core idea of Trans-Encoder is self-distillation in a smart way: alternatively training a bi-encoder and a cross-encoder (of the same architecture) with pseudo-labels created from the other. We also propose a mutual-distillation extension to mutually bootstrap two self-distillation models trained in parallel. On sentence-pair tasks including sentence similarity, question de-duplication, question-answering entailment, and paraphrase identification, we show strong empirical evidence verifying the effectiveness of Trans-Encoder. We also found the surprisingly strong generalisation capability of our trained cross-encoders across domains and tasks. Finally, we

dataset→	QQP	QNLI	MRPC	avg.
SimCSE-RoBERTa-base	81.82	73.54	75.06	76.81
+ TEnc (bi)	82.56	71.67	74.24	76.16
+ TEnc (cross)	83.66	79.38	79.53	80.86
+ TEnc-mutual (bi)	81.71	72.78	75.51	76.67
+ TEnc-mutual (cross)	**83.92**	**79.79**	**79.96**	**81.22**

Table 3: A domain transfer setup: testing Trans-Encoder models trained with STS data directly on binary classification tasks. We only demonstrate results for one model for brevity. Full table can be found in Appendix (Tab. 10).

conduct thorough ablation studies and analysis to verify our design choices and shed insight on the model mechanism.

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 *SemEval 2015*, pages 252–263.

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 *SemEval 2014*, pages 81–91.

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 *SemEval-2016*, pages 497–511.

Agirre, E., Cer, D., Diab, M., and Gonzalez-Agirre, A. (2012). SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012*, pages 385–393.

Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W. (2013). *SEM 2013 shared task: Semantic textual similarity. In **SEM 2013*, pages 32–43.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *EMNLP 2015*, pages 632–642.

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 *SemEval 2017*, pages 1–14.

Gao, T., Yao, X., and Chen, D. (2021). Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP 2021*.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. (2020). Momentum contrast for unsupervised visual representation learning. In *CVPR 2020*, pages 9726–9735.

Humeau, S., Shuster, K., Lachaux, M.-A., and Weston, J. (2020). Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *ICLR 2020*.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., and Li, L. (2020). On the sentence embeddings from pre-trained language models. In *EMNLP 2020*, pages 9119–9130.

Liu, F., Vulić, I., Korhonen, A., and Collier, N. (2021). Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *EMNLP 2021*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *ICLR 2019*.

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 *LREC 2014*, pages 216–223.

Mobahi, H., Farajtabar, M., and Bartlett, P. L. (2020). Self-distillation amplifies regularization in hilbert space. *arXiv preprint arXiv:2002.05715*.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP 2016*, pages 2383–2392, Austin, Texas.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP-IJCNLP 2019*, pages 3982–3992.

Thakur, N., Reimers, N., Daxenberger, J., and Gurevych, I. (2021). Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *NAACL 2021*, pages 296–310.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR 2019*.

A Appendix

A.1 Evaluation Tasks

Evaluation task: sentence textual similarity (STS). Following prior works (Reimers and Gurevych, 2019; Liu et al., 2021; Gao et al., 2021), we consider seven STS datasets: SemEval STS 2012-2016 (STS12-16, Agirre et al. 2012, 2013, 2014, 2015, 2016), STS Benchmark (STS-B, Cer et al. 2017) and SICK-Relatedness (SICK-R, Marelli et al. 2014). In all these datasets, sentence pairs are given a human-judged relevance score from 1 to 5. We normalise them to 0 and 1 and models are asked to predict the scores. We report Spearman's ρ rank correlation between the two.

Evaluation task: binary classification. We also test TRANS-ENCODER on sentence-pair binary classification tasks where the model has to decide if a sentence-pair has certain relations. We choose (1) the Quora Question Pair (QQP) dataset, requiring a model to judge whether two questions are duplicated; (2) a question-answering entailment dataset QNLI (Rajpurkar et al., 2016; Wang et al., 2019) in which given a question and a sentence the model needs to judge if the sentence answers the question; (3) the Microsoft Research Paraphrase Corpus (MRPC) which asks a model to decide if two sentences are paraphrases of each other. The ground truth labels of all datasets are either 0 or 1. Following (Li et al., 2020; Liu et al., 2021), we compute Area Under Curve (AUC) scores using the binary labels and the cosine similarity scores of sentence-pair embeddings.

A.2 Experimental Setup

Training and evaluation details. For each task, we use all available sentence pairs (from train, development and test sets of all datasets combined) without their labels as training data. The original QQP and QNLI datasets are extremely large. We thus downsample QQP to have 10k, 1k and 10k pairs for train, dev and test; QNLI to have 10k train set. QNLI does not have public ground truth labels for testing. So, we use its first 1k examples in the official dev set as our dev data and the rest in the official dev set as test data. The dev set for MRPC is its official dev sets. The dev set for STS12-16, STS-B and SICK-R is the dev set of STS-B. We save one checkpoint for every 200 training steps and at the end of each epoch. We use the dev sets to select the best model for testing. Dev sets are also used to tune the hyperprameters in each task.

For clear comparison with SimCSE and Mirror-BERT, we use their released checkpoints as initialisation points (i.e., we do not train them ourselves). We consider four SimCSE variants. Two base variants: SimCSE-BERT-base, SimCSE-RoBERTa-base; and two large variants: SimCSE-BERT-large, SimCSE-RoBERTa-large. We consider two Mirror-BERT variants: Mirror-RoBERTa-base and Mirror-RoBERTa-base-drophead.[3] For brevity, our analysis in the main text focuses on SimCSE models. We list Mirror-BERT results in Appendix. We train TRANS-ENCODER models for 3 cycles on the STS task and 5 cycles on the binary classification tasks. Within each cycle, all bi- and cross-encoders are trained for 10 and 1 epochs respectively for the STS task; 15 and 3 epochs for binary classification.[4] All models use AdamW (Loshchilov and Hutter, 2019) as the optimiser. In all tasks, unless noted otherwise, we create final representations using `[CLS]`. We train our base models on a server with 4 * V100 (16GB) GPUs and large models on a server with A100 (40GB) GPUs. All main experiments have the same fixed random seed (2021).[5]

Mutual-distillation setup. For the two models used for mutual-distillation, they are either (1) the base variant of SimCSE-BERT and SimCSE-RoBERTa or (2) the large variant of the two. Theoretically, we could mutually distil even more models but we keep the setup simple for fast and clear comparison. Also, since mutual-distillation models use information from both PLMs, we also list *ensemble results* that use the average of the predictions from two TRANS-ENCODERS.

[3] We do not consider BERT-based checkpoints by Liu et al. (2021) since they adopt mean pooling over all tokens. For brevity, we only experiment with encoders trained with `[CLS]`.

[4] We use fewer epochs for cross-encoders since they usually converge much faster than bi-encoders.

[5] All other hparams listed in Appendix.

dataset→	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	avg.
Mirror-RoBERTa-base	64.67	81.53	73.05	79.78	78.16	78.36	**70.03**	75.08
+ TEnc (bi)	71.99	81.85	75.73	83.32	**79.97**	81.19	69.47	77.64
+ TEnc (cross)	**73.10**	**82.31**	**76.49**	**83.71**	79.64	**81.70**	69.70	**78.09**
Mirror-RoBERTa-base-drophead	68.15	82.55	73.47	82.26	79.44	79.63	71.58	76.72
+ TEnc (bi)	74.95	83.86	77.50	85.80	83.22	83.94	72.56	80.26
+ TEnc (cross)	**75.70**	**84.58**	**78.35**	**86.49**	**83.96**	**84.26**	**72.76**	**80.87**

Table 4: English STS (Mirror-BERT models). Spearman's ρ scores are reported.

dataset→	QQP	QNLI	MRPC	avg.
Mirror-RoBERTa-base	78.89	73.73	75.44	76.02
+ TEnc (bi)	**82.34**	79.57	77.08	79.66
+ TEnc (cross)	82.00	**79.87**	**78.40**	**80.09**
Mirror-RoBERTa-base-drophead	78.36	75.56	77.18	77.03
+ TEnc (bi)	82.04	82.12	79.63	81.26
+ TEnc (cross)	**82.90**	**82.52**	**81.38**	**82.27**

Table 5: Binary classification (Mirror-BERT models). AUC scores are reported.

A.3 Mirror-BERT Results

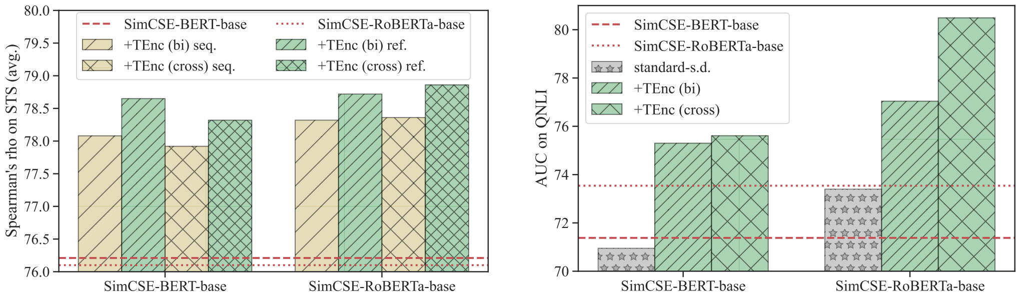
As seen in Tab. 4 and Tab. 5, Trans-Encoder brings significant improvement to the base Mirror-BERT models, similar to what we have observed on SimCSE models. After Trans-Encoder training, the Mirror-BERT-based models perform even slightly better than SimCSE-based models, on both the STS and binary classification tasks. We suspect it is due to that SimCSE checkpoints are trained for more iterations (with the contrastive learning objective) and thus are more prone to overfit the training corpus.

A.4 Discussions and Analysis

In this section we discuss some interesting phenomena we observed, justify design choices we made in an empirical way, and also show a more fine-grained understanding of what exactly happens when using Trans-Encoder.

Initiate with the original weights or train sequentially? (Fig. 3a) As mentioned in §2, we initiate bi-encoders with SimCSE/Mirror-BERT weights and cross-encoders with PLMs' weights (we call this strategy *refreshing*). We have also tried maintaining the weights of bi- and cross-encoders sequentially. I.e., we initiate the cross-encoder's weights with the bi-encoder which just created the pseudo labels and vice versa (we call this strategy *sequential*). The benefit of doing so is keeping all the legacy knowledge learned in the process of self-distillation in the weights. As suggested in Fig. 3a (also in Appendix Tab. 11), *refreshing* is slightly better than *sequential*. We suspect it is because initiating with original weights alleviate catastrophic forgetting problem in self-supervised learning, similar to the moving average and stop-gradient strategy used in contrastive SSL methods such as MoCo (He et al., 2020).

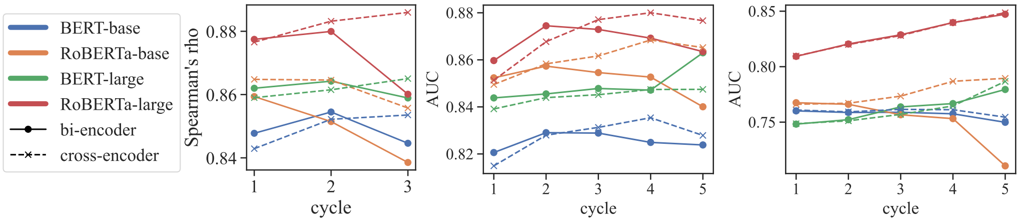
Self-distillation without alternating between different task formulations? (Fig. 3b) If we disregard the bi- and cross-encoder alternating training paradigm but using the same bi-encoder architecture all along, the model is then similar to the standard self-distillation model (c.f. Figure 1 by Mobahi et al. (2020)). Theoretically, standard self-distillation could also have helped, especially considering that it adapts the model to in-domain data. However, as seen in Fig. 3b (and also Appendix Tab. 12), standard self-distillation lags behind Trans-Encoder by a significant amount, and sometimes underperforms the base model. Standard self-distillation essentially ensembles different checkpoints (i.e., the same model at different training phase provides different views of the data). Trans-Encoder can be seen as a type of self-distillation model, but instead of using previous models to provide 'views' of data, we use different task formulations (i.e. bi- and cross-encoder) to provide



(a) Sequential vs. refreshing (on STS).

(b) TENC vs.standard self-distillation (on QNLI).

Figure 3: Ablation studies of TRANS-ENCODER regarding two design choices.



(a) STS

(b) QQP

(c) QNLI

Figure 4: TRANS-ENCODER's performance against distillation cycles under different base models and tasks.

different views of the data. The fact that standard self-distillation underperforms TRANS-ENCODER by large margins suggest the effectiveness of our proposed bi- and cross-encoder training scheme.

How many cycles is optimal? (Fig. 4) Our approach iteratively bootstrap the performance of both bi- and cross-encoders. But how many iterations (cycles) are enough? We find that it heavily depends on the model and dataset, and could be unpredictable (since the 'optimality' depends on a relatively small dev set). In Fig. 4, we plot the TRANS-ENCODER models' (self-distillation only) performances on dev sets in three tasks: STS, QQP, and QNLI. In general, the patterns are different across datsets and models.

Does more training data help? (Fig. 5) We control the number of training samples drawn and test the TRANS-ENCODER model (using SimCSE-BERT-base) on STS test sets. The average performance on all STS tasks are reported in Fig. 5, with three runs of different random seeds. There are in total 37,081 data points from STS. We only draw from these in-domain data until there is none left. It can be seen that for in-domain training, more data points are usually always beneficial. Next, we test with more samples drawn from another task, SNLI (Bowman et al., 2015), containing abundant sentence-pairs. The performance of the model further increases till 30k extra data points and starts to gradually decrease after that point. However, it is worth mentioning that there will be discrepancy of such trends across tasks and models. And when needed, one can mine as many sentence-pairs as desired from a general corpus (such as Wikipedia) for TRANS-ENCODER learning.

How robust is TRANS-ENCODER? (Tab. 6) As mentioned, our experiments used one fixed random seed. Due to the scale of the experiments, performing multiple runs with different random seeds for all setups is overly expensive. We pick STS and the base variants of BERT and RoBERTa as two examples for showing results of five runs (shown in Tab. 6).

STS and SICK-R as different tasks (Tab. 7, Fig. 6). It is worth noting that in the STS main results table (Tab. 1), SICK-R is the only dataset where TRANS-ENCODER models lead to performance worse than the baselines in some settings (e.g., on SimCSE-BERT-large). We believe it is due to that SICK-R is essentially a different task from STS2012-2016 and STS-B since SICK-R is labelled

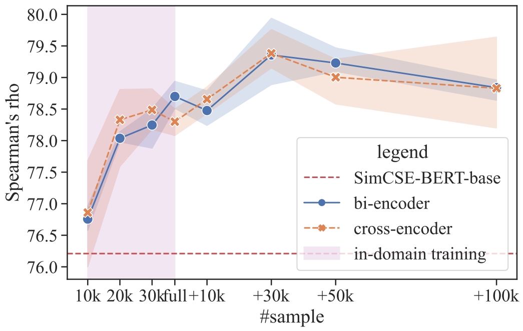


Figure 5: STS performance against number of training samples. 'full' means all STS in-domain data are used. After 'full', the samples are drawn from the SNLI dataset. Note that each point in the graph is an individual run.

dataset→	mean	S.D.
SimCSE-BERT-base		
+ TEnc (bi)	78.72	0.16
+ TEnc (cross)	78.21	0.26
SimCSE-RoBERTa-base		
+ TEnc (bi)	78.38	0.26
+ TEnc (cross)	78.90	0.34

Table 6: Mean and standard deviation (S.D.) of five runs on STS.

Data→		STS		SICK-R	
model→	off-the-shelf	+TEnc (bi)	+TEnc (cross)	+TEnc (bi)	+TEnc (cross)
SimCSE-BERT-base	72.22	71.84	71.16	74.13	74.43
SimCSE-RoBERTa-base	68.62	67.63	69.51	68.39	70.38
SimCSE-BERT-large	73.88	71.46	70.90	74.92	74.98
SimCSE-RoBERTa-large	71.23	68.36	67.90	72.63	73.13

Table 7: Compare Trans-Encoder models trained with all STS data (using STS-B's dev set) and SICK-R data only (using SICK-R's dev set). Large performance gains can be obtained when treating SICK-R as a standalone task.

with a different aim in mind. It focuses specifically on compositional distributional semantics.[6] To verify our claim, we train SICK-R as a standalone task using its official dev set instead of STS-B. All sentence pairs from SICK-R are used as raw training data. The results are shown in Tab. 7 and Fig. 6. As shown, around 3-4% gain can be obtained by switching to training on SICK-R only, confirming the different nature of the two tasks. This suggests that generalisation is also dependent on the standard of how relevance is defined between sequences in the target dataset, and training with an in-domain dev set maybe crucial when the domain shift between training and testing data is significant.

A.5 Full Tables

Here, we list the full tables of SimCSE STS (Tab. 8) and binary classification results (Tab. 9); SimCSE domain transfer results (Tab. 10); and full tables for ablation studies (Tab. 11, Tab. 12), which are presented as figures in the main texts in Fig. 3.

[6] SICK-R "includes a large number of sentence pairs that are rich in the lexical, syntactic and semantic phenomena that Compositional Distributional Semantic Models (CDSMs) are expected to account for (e.g., contextual synonymy and other lexical variation phenomena, active/passive and other syntactic alternations, impact of negation, determiners and other grammatical elements), but do not require dealing with other aspects of existing sentential data sets (e.g., STS, RTE) that are not within the scope of compositional distributional semantics." (see `http://marcobaroni.org/composes/sick.html`).

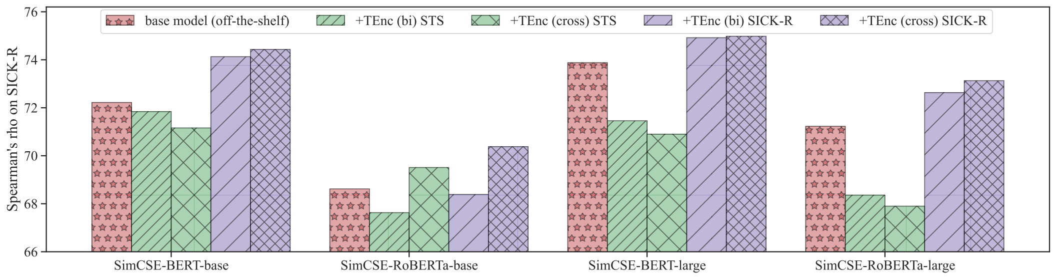


Figure 6: Graphical presentation of Tab. 7.

#	dataset→	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	avg.
	single-model results								
1	SimCSE-BERT-base*	68.64	82.39	74.30	80.69	78.71	76.54	72.22	76.21
1.1	+ TEnc (bi)	72.17	84.40	76.69	83.28	80.91	81.26	71.84	78.65
1.2	+ TEnc (cross)	71.94	84.14	76.39	82.87	80.65	81.06	71.16	78.32
1.3	+ TEnc-mutual (bi)	75.09	85.10	77.90	**85.08**	**83.05**	**83.90**	**72.76**	**80.41**
1.4	+ TEnc-mutual (cross)	**75.44**	**85.59**	**78.03**	84.44	82.65	83.61	69.52	79.90
2	SimCSE-RoBERTa-base	68.34	80.96	73.13	80.86	80.61	80.20	68.62	76.10
2.1	+ TEnc (bi)	73.36	82.47	76.39	83.96	82.67	82.05	67.63	78.36
2.2	+ TEnc (cross)	72.59	83.24	76.83	84.20	82.82	82.85	69.51	78.86
2.3	+ TEnc-mutual (bi)	75.01	85.22	78.26	85.16	83.22	83.88	**72.56**	80.47
2.4	+ TEnc-mutual (cross)	**76.37**	**85.87**	**79.03**	**85.77**	**83.77**	**84.65**	72.62	**81.15**
3	SimCSE-BERT-large	71.30	84.32	76.32	84.28	79.78	79.04	**73.88**	78.42
3.1	+ TEnc (bi)	75.55	84.08	77.01	85.43	81.37	82.88	71.46	79.68
3.2	+ TEnc (cross)	75.81	84.51	76.50	85.65	82.14	83.47	70.90	79.85
3.3	+ TEnc-mutual (bi)	**78.19**	**88.51**	**81.37**	**88.16**	**84.81**	**86.16**	71.33	**82.65**
3.4	+ TEnc-mutual (cross)	77.97	88.31	81.02	88.11	84.40	85.95	71.92	82.52
4	SimCSE-RoBERTa-large	71.40	84.60	75.94	84.36	82.22	82.67	71.23	78.92
4.1	+ TEnc (bi)	77.92	86.69	79.29	87.23	84.22	86.10	68.36	81.40
4.2	+ TEnc (cross)	**78.32**	86.20	79.61	86.88	82.93	84.48	67.90	80.90
4.3	+ TEnc-mutual (bi)	78.15	**88.39**	81.76	88.38	84.95	86.55	**72.31**	**82.93**
4.4	+ TEnc-mutual (cross)	78.28	88.31	**81.94**	**88.63**	**85.03**	**86.70**	71.63	**82.93**
	ensemble results (average predictions of two models)								
1+2	SimCSE-base ensemble	70.71	83.49	76.45	83.13	81.79	81.51	71.94	78.43
1.1+2.1	TEnc-base (bi) ensemble	73.58	85.01	78.35	85.02	83.21	84.07	70.93	80.03
1.2+2.2	TEnc-base (cross) ensemble	74.25	84.92	78.57	85.16	83.25	83.52	70.73	80.06
1.3+2.3	TEnc-mutual-base (bi) ensemble	75.29	85.34	78.49	85.28	83.43	84.09	**72.75**	80.67
1.4+2.4	TEnc-mutual-base (cross) ensemble	**76.60**	**86.18**	**79.09**	**85.49**	**83.64**	**84.67**	71.96	**81.09**
3+4	SimCSE-large ensemble	73.22	86.03	78.15	85.95	82.83	83.05	**73.86**	80.66
3.1+4.1	TEnc-large (bi) ensemble	78.49	87.02	80.31	87.85	84.31	**86.54**	72.39	82.41
3.2+4.2	TEnc-large (cross) ensemble	77.93	86.91	79.83	87.82	83.74	85.58	72.02	81.98
3.3+4.3	TEnc-mutual-large (bi) ensemble	78.42	88.60	**81.91**	88.38	**85.01**	86.52	72.23	83.01
3.4+4.4	TEnc-mutual-large (cross) ensemble	**78.52**	**88.70**	81.90	**88.67**	84.96	86.70	72.03	**83.07**

Table 8: English STS. Spearman's ρ rank correlations are reported. TEnc models use only self-distillation while TEnc-mutual models use mutual-distillation as well. Blue and red denotes mutual-distillation models that are trained in the base and large group respectively. Models without colour are not co-trained with any other models. *Note that for base encoders, our results can slightly differ from numbers reported in (Gao et al., 2021) since different evaluation packages are used.

#	dataset→	QQP	QNLI	MRPC	avg.
	single-model results				
	SimCSE-BERT-base	80.38	71.38	75.02	75.59
	+ TEnc (bi)	82.10	75.30	75.71	77.70
	+ TEnc (cross)	82.10	75.61	76.21	77.97
	+ TEnc-mutual (bi)	84.00	76.93	76.62	79.18
	+ TEnc-mutual (cross)	**84.29**	**77.11**	**77.77**	**79.72**
	SimCSE-RoBERTa-base	81.82	73.54	75.06	76.81
	+ TEnc (bi)	84.13	77.08	75.29	78.83
	+ TEnc (cross)	**85.16**	**80.49**	76.09	**80.58**
	+ TEnc-mutual (bi)	84.36	77.29	76.90	79.52
	+ TEnc-mutual (cross)	84.73	77.32	**78.47**	80.17
	SimCSE-BERT-large	82.42	72.46	75.93	76.94
	+ TEnc (bi)	84.53	78.22	78.42	80.39
	+ TEnc (cross)	84.18	79.71	79.43	81.11
	+ TEnc-mutual (bi)	85.72	**81.61**	79.59	82.31
	+ TEnc-mutual (cross)	**86.55**	81.55	**79.69**	**82.60**
	SimCSE-RoBERTa-large	82.99	75.76	77.24	78.66
	+ TEnc (bi)	85.65	84.49	79.34	83.16
	+ TEnc (cross)	86.16	**84.69**	81.00	**83.95**
	+ TEnc-mutual (bi)	85.65	81.87	79.53	82.35
	+ TEnc-mutual (cross)	**86.38**	83.14	**81.77**	83.76
	ensemble results (average predictions of two models)				
	SimCSE-base ensemble	81.10	72.46	75.04	76.20
	TEnc-base (bi) ensemble	83.11	76.19	75.45	78.25
	TEnc-base (cross) ensemble	84.15	**78.55**	**78.90**	**80.53**
	TEnc-mutual-base (bi) ensemble	84.18	77.11	76.76	79.35
	TEnc-mutual-base (cross) ensemble	**84.68**	77.92	78.22	80.27
	SimCSE-large ensemble	82.47	74.09	76.55	77.70
	TEnc-large (bi) ensemble	85.09	81.35	78.88	81.77
	TEnc-large (cross) ensemble	86.22	**83.78**	81.06	**83.69**
	TEnc-mutual-large (bi) ensemble	86.10	81.74	79.56	82.47
	TEnc-mutual-large (cross) ensemble	**86.26**	82.86	**81.41**	83.51

Table 9: Binary classification task results (SimCSE models; full table). AUC scores are reported.

A.6 Data Statistics

A complete listing of train/dev/test stats of all used datasets can be found in Tab. 13. Note that for STS 2012-2016, we dropped all sentence-pairs without a valid score. And the train sets include all sentence pairs (w/o labels) regardless of split in each task.

dataset→	QQP	QNLI	MRPC	avg.
SimCSE-BERT-base	80.38	71.38	75.02	75.59
+ TEnc (bi)	80.80	72.35	74.53	75.89
+ TEnc (cross)	80.84	**78.81**	**79.42**	79.69
+ TEnc-mutual (bi)	**83.24**	72.88	75.86	77.33
+ TEnc-mutual (cross)	82.40	78.30	78.72	**79.81**
SimCSE-RoBERTa-base	81.82	73.54	75.06	76.81
+ TEnc (bi)	82.56	71.67	74.24	76.16
+ TEnc (cross)	83.66	79.38	79.53	80.86
+ TEnc-mutual (bi)	81.71	72.78	75.51	76.67
+ TEnc-mutual (cross)	**83.92**	**79.79**	**79.96**	**81.22**
SimCSE-BERT-large	82.42	72.46	75.93	76.94
+ TEnc (bi)	81.86	71.99	74.99	76.28
+ TEnc (cross)	**83.31**	**79.62**	79.93	80.95
+ TEnc-mutual (bi)	82.30	72.47	76.74	77.17
+ TEnc-mutual (cross)	83.04	79.58	**81.18**	**81.27**
SimCSE-RoBERTa-large	82.99	75.76	77.24	78.66
+ TEnc (bi)	82.34	70.88	76.05	76.42
+ TEnc (cross)	85.98	80.07	81.20	82.42
+ TEnc-mutual (bi)	85.28	71.56	76.81	77.88
+ TEnc-mutual (cross)	**86.31**	**81.77**	**81.86**	**83.31**

Table 10: Full table for domain transfer setup: testing Trans-Encoder (SimCSE models) trained with STS data directly on binary classification tasks.

dataset→	STS (avg.)
SimCSE-BERT-base	76.21
+ TEnc (bi) sequential	78.08
+ TEnc (cross) sequential	77.92
+ TEnc (bi) refreshing	**78.65**
+ TEnc (cross) refreshing	78.32
SimCSE-RoBERTa-base	76.10
+ TEnc (bi) sequential	78.32
+ TEnc (cross) sequential	78.72
+ TEnc (bi) refreshing	78.36
+ TEnc (cross) refreshing	**78.86**

Table 11: Ablation: sequential training with the same set of weights vs. refreshing weights for all models.

dataset→	STS (avg.)	QNLI
SimCSE-BERT-base	76.21	71.38
+ standard-self-distillation	77.16	70.95
+ TEnc (bi)	78.65	75.30
+ TEnc (cross)	78.32	75.61
SimCSE-RoBERTa-base	76.10	73.54
+ standard-self-distillation	76.25	73.40
+ TEnc (bi)	78.36	77.04
+ TEnc (cross)	78.86	80.49

Table 12: Ablation: compare Trans-Encoder with standard self-distillation.

Dataset	|Train|	|Dev|	|Test|
STS 2012	-	-	3,108
STS 2013	-	-	1,500
STS 2014	-	-	3,750
STS 2015	-	-	3,000
STS 2016	-	-	1,186
STS-B	-	1,500	1,379
SICK-R	-	495	4,906
STS train (full)⋆	37,081	-	-
QQP	21,000	1,000	10,000
QNLI	15,463	1,000	4,463
MRPC	5,801	408	1,725

Table 13: A listing of train/dev/test stats of all used datasets. ⋆: a collection of all individual sentence-pairs from all STS tasks.