Deep Continuous Prompt for Contrastive Learning of Sentence Embeddings

Anonymous ACL submission

Abstract

The performance of sentence representation has been remarkably improved by the framework of contrastive learning. However, recent works still require full fine-tuning, which is quite inefficient for large-scaled pre-trained language models. To this end, we present a novel method which freezes the whole language model and only optimizes the prefix deep continuous prompts. It not only tunes around 0.1% parameters of the original language model, but avoids the cumbersome computation of searching handcrafted prompts. Experimental results show that our proposed DCPCSE outperforms the state-of-the-art method SimCSE by a large margin. We raise the performance of unsupervised BERT$_{base}$ and supervised RoBERTa$_{large}$ by 2.24 and 1.00 points, respectively. Our code will be released at Github.

1 Introduction

Sentence representation learning is a vital problem in natural language processing (NLP) and has wide real-life applications including large-scale semantic similarity comparison, information retrieval, etc (Reimers and Gurevych, 2019a).

Benefited from large pre-trained language models, the performance of sentence representation learning has been further boosted with addition supervision. However, the naïve sentence embeddings derived from these over-parameterized models prone to be collapsed (Chen and He, 2021), resulting in high similarity between any two sentences. Recently, contrastive learning based on the idea of pulling semantically close samples together and pushing apart dissimilar samples in the vector space (Chen et al., 2020) has achieved extraordinary success in learning universal sentence embeddings. Works such as ConSERT (Yan et al., 2021) and SimCSE (Gao et al., 2021) apply various ways to construct proper positive pairs, and regard the in-batch examples as negatives. Nonetheless,

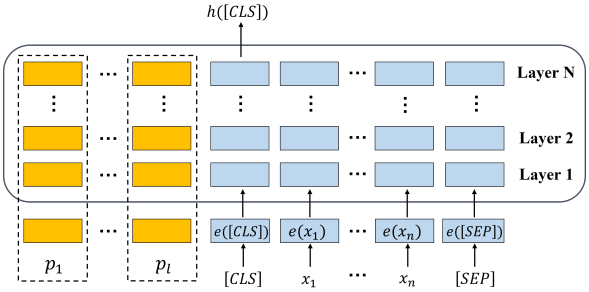

Figure 1: Deep continuous prompt framework for contrastive learning of sentence embeddings. We freeze the transformer parameters (the blue blocks) and only optimize the prefix deep continuous prompts (the orange blocks).

they still require to fine tune the whole pre-trained model, which is quite inefficient especially for models consisting of billions of parameters like T5-11B (Raffel et al., 2020). Considering the online setting where tasks arrive in a stream, it is particularly useful to store only a small number of parameters for each task rather than training an entire new model.

Prompting, which freezes all parameters of a pre-trained language model and adapts it as a predictor through completion of a cloze task, has become a new paradigm in NLP (Liu et al., 2021a). For example, in sentiment analysis, we can concatenate the text with a prompt "[X] the movie is [MASK]." and ask the pre-trained language model to predict the masked token. Then the predicted probabilities of "good" and "bad" being the masked token can be used to predict the sample's label. However, discovering the optimal prompt manually for specific tasks could be quite challenging, even for experienced prompt designers. To address this issue, plenty of prompt engineering methods have been proposed, which can be divided into two categories: discrete prompts and continuous prompts. Discrete prompts aim to search for a sequence of discrete trigger tokens through data-driven optimization (Schick and Schütze, 2020a,b; Shin et al.,

2020), while continuous prompts differentially optimize continuous token embeddings (Li and Liang, 2021a; Zhong et al., 2021; Liu et al., 2021b), whose effects will be propagated upward to all transformer activation layers and rightward to subsequent tokens. Compared with discrete prompts, continuous prompts are much more time-efficient and less likely to fall into local optima due to the expansion of the search space.

Inspired by continuous prompts, we propose DCPCSE, a deep continuous prompt framework for contrastive learning of sentence embeddings, as Figure 1 shows. By adding multi-layer trainable dense vectors as prompts to the input sequence, we train our whole architecture based on the idea of constractive learning, while keeping all parameters of the pre-trained model frozen. In other words, the input embeddings as well as each layer's hidden embeddings of continuous prompts are optimized, which enables more direct impact on the loss function and is easier to converge. Additionally, we find that multi-task learning by combining contrastive learning objective with an auxiliary masked langauge model (MLM) objective enables the language model to obtain a better sentence representation with a rich association among the continuous prompts, especially for the unsupervised setting.

We conduct comprehensive experiments on seven standard semantic textual similarity (STS) tasks. Our proposed DCPCSE substantially surpasses SimCSE with only 0.1% parameters tuned. Under the unsupervised setting, DCPCSE achieves a 78.49 and 77.93 averaged Spearman's correlation using BERT$_{base}$ and RoBERTa$_{base}$ respectively, a 2.24 and 1.36 points improvement compared to SimCSE. In the supervised setting, DCPCSE outperforms SimCSE by 0.78 on BERT$_{large}$ and 1.00 on RoBERTa$_{large}$.

2 Deep Continuous Prompt Framework

In this section, we illustrate how to encode sentences into embedding vectors through our proposed model and how to train it.

2.1 Sentence Embedding Encoder

Given a pre-trained language model $\mathcal{M}$, a common method to encode a sentence into an embedding vector is to map the sequence of tokens $\{x_1, ..., x_n\}$ to input embeddings $\{e(x_1), ..., e(x_n)\}$ first, and then feed these embeddings through multiple transformer layers (Vaswani et al., 2017). The sentence representation could be acquired by taking the [CLS] token embedding of the last layer or taking average of all token embeddings.

In our architecture depicted in Figure 1, l trainable dense vectors $\{p_1, ..., p_l\}$ are added as continuous prompts to the input sequence, whose dimensions are identical to $\mathcal{M}$'s input embeddings. Inspired by Perfix-Tuning (Li and Liang, 2021b), the hidden embeddings of these continuous prompts in all transformer layers are also optimized during training, which means they are independent to each other interlayers rather than being computed by previous layers. Trainable embeddings added to each layers can have more direct impact on the loss function, which benefits a smoother optimization. We choose to take the [CLS] representation from the last layer as the sentence embedding. Note that all the parameters of pre-trained language models are fixed, thus reducing the number of tunable parameters to around 0.1%.

2.2 Multi-task Learning

Contrastive learning objective We follow the contrastive learning framework in (Gao et al., 2021): given a set of paired sentences $\mathcal{D} = \{(X_i, X_i^+)\}_{i=1}^m$ where X_i and X_i^+ are semantically related, we regard X_i^+ as "positive" of X_i and other sentences in the same mini-batch as "negatives". Let $\mathbf{h}_i$ and $\mathbf{h}_i^+$ denote the representations of X_i and X_i^+, then the training objective for a single sample in a mini-batch of size N is:

$$\ell_{CL} = -\log \frac{\exp^{sim(\mathbf{h}_i, \mathbf{h}_i^+)} / \tau}{\sum_{j=1}^{N} \exp^{sim(\mathbf{h}_i, \mathbf{h}_j^+)} / \tau}$$

where τ is a temperature hyperparameter and $sim(\mathbf{h}_1, \mathbf{h}_2)$ is the cosine similarity function.

MLM objective To ensure the association among the pseudo prompt tokens $\{p_1, ..., p_l\}$, we also consider leveraging an auxiliary MLM objective proposed by (Devlin et al., 2019) and denote it as ℓ_{MLM}. That is, 15% tokens of each sequence are randomly chosen for prediction. The i-th chosen token x_i is replaced by (1) the [MASK] token 80% of the time (2) a random token 10% of the time (3) itself 10% of the time. The effectiveness of the auxiliary MLM objective is discussed in 3.3.

Finally, the overall training objective becomes:

$$\ell = \ell_{CL} + \lambda \ell_{MLM}$$

$$\lambda = 0.1 * decay\_rate^{\frac{global\_step}{decay\_step}}$$

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Unsupervised models								
BERT_{base}‡ (first-last-avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
BERT_{base}-flow‡	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT_{base}-whitening‡	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
ConSERT-BERT_{base}§	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE-BERT_{base}‡	68.40	82.41	74.38	80.91	78.56	76.85	**72.23**	76.25
SCPCSE-BERT_{base}	64.28	78.97	70.51	78.45	75.71	76.33	68.73	73.28
DCPCSE-BERT_{base}	**73.03**	**85.18**	**76.70**	**84.19**	**79.69**	**80.62**	70.00	**78.49**
SimCSE-BERT_{large}‡	70.88	84.16	76.43	84.50	79.76	79.26	**73.88**	78.41
DCPCSE-BERT_{large}	**73.34**	**85.90**	**77.10**	**85.26**	80.08	80.96	73.28	**79.42**
SimCSE-RoBERTa_{base}‡	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
DCPCSE-RoBERTa_{base}	70.57	81.91	74.60	82.90	**80.96**	**82.84**	71.70	77.93
Supervised models								
SBERT_{base}†	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT_{base}-flow‡	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
SBERT_{base}-whitening‡	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
ConSERT-BERT_{base}§	74.07	83.93	77.05	83.66	78.76	81.36	76.77	79.37
SimCSE-BERT_{base}‡	75.30	**84.67**	**80.19**	85.40	80.82	84.25	80.39	81.57
DCPCSE-BERT_{base}	**75.58**	84.33	79.67	**85.79**	**81.24**	**84.25**	**80.79**	**81.65**
SimCSE-BERT_{large}‡	75.78	86.33	80.44	86.60	80.86	84.87	81.14	82.21
DCPCSE-BERT_{large}	77.97	86.54	81.04	86.33	81.81	85.24	81.31	82.89
SimCSE-RoBERTa_{base}‡	76.53	85.21	80.95	86.03	82.57	85.83	80.50	82.52
DCPCSE-RoBERTa_{base}	76.75	85.86	80.98	86.51	83.51	86.58	80.41	82.94
SimCSE-RoBERTa_{large}‡	77.46	87.27	82.36	86.66	83.93	86.70	81.95	83.76
DCPCSE-RoBERTa_{large}	**79.14**	**88.64**	**83.73**	**87.33**	**84.57**	**87.84**	**82.07**	**84.76**

Table 1: The performance comparison of our DCPCSE and previous state-of-the-art models on seven STS tasks. The reported score is Spearman correlation magnified by a factor of 100. †: results from Reimers and Gurevych, 2019b; ‡: results from Gao et al., 2021; §: results from Yan et al., 2021.

where the weight of MLM loss λ decays exponentially as the training progresses, which forces the model to focus more and more on the main target. The decay_rate and decay_step are set to 0.95 and 100 empirically.

3 Experiments

3.1 Setups

Datasets We use seven standard STS datasets including STS tasks 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014) for our experiments. Each sample in these datasets contains a pair of sentence as well as a semantic similarity score ranging from 0 to 5.

Baselines To verify the validity of our proposed architecture, we mainly choose two post-process methods BERT-flow (Li et al., 2020) and BERT-whitening (Su et al., 2021) as well as two contrastive learning based methods ConSERT (Yan et al., 2021) and SimCSE (Gao et al., 2021) as baselines.

Training Details We obtain pre-trained checkpoints of BERT (Devlin et al., 2019) (uncased) or RoBERTa (Liu et al., 2019) (cased) from Huggingface [1]. Note that we only make the parameters of deep continuous prompts trainable, all parameters of pre-trained models are frozen during training. Following SimCSE (Gao et al., 2021), we use the same datasets to train our unsupervised models and supervised models. All the experiments are conducted on two Nvidia 3090 GPUs. More training details can be found in Appendix A.

[1] https://huggingface.co/models

3.2 Main Results

Table 1 summarizes the evaluation results on seven STS tasks. Our proposed DCPCSE can substantially surpass the previous state-of-the-art SimCSE in both unsupervised and supervised settings. Specifically, our unsupervised DCPCSE outperforms SimCSE by 2.24% on BERT$_{base}$, 1.36% on RoBERTa$_{base}$ and 1.01% on BERT$_{large}$ respectively. In terms of supervised setting, DCPCSE achieves slight improvements on base models (0.08% for BERT$_{base}$ and 0.42% for RoBERTa$_{base}$) but significant improvements on large models (0.78% for BERT$_{large}$ and 1.00% for RoBERTa$_{large}$). This is in line with the finding that prompt tuning can be more efficient as the model parameters scale up (Lester et al., 2021).

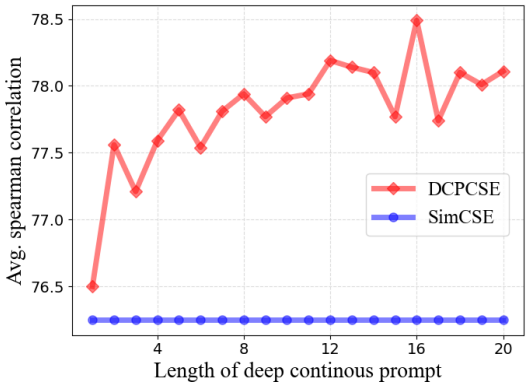

Figure 2: Test performance with various length of deep continuous prompts based on unsupervised DCPCSE-BERT$_{base}$.

3.3 Ablation Study

What if we only make the input embeddings of continuous prompts trainable? Following P-tuning (Liu et al., 2021c), we define "shallow" continuous prompt as follows:

$$[p_1] \ldots [p_m]\ [X]\ [p_{m+1}] \ldots [p_l]\ [MASK]$$

where X denotes the token sequence, $[p_1], ..., [p_l]$ are dense vectors with the same dimension as the language model's input embedding. After initializing each $[p_i]$ with the pre-trained input embedding, we keep all other model parameters fixed and only tune these shallow continuous prompts. Eventually, the output [MASK] representation is regarded as the sentence embedding. We apply this architecture to contrastive learning of sentence embeddings and name it as SCPCSE. The experimental settings are in Appendix A.

From Table 1, it can be clearly seen that SCPCSE-BERT$_{base}$ underperforms DCPCSE-BERT$_{base}$ by 5.21 points, which validates the necessity of multi-layer continuous prompts.

Prompt length Here we investigate how different prompt length affects our models. Figure 2 shows that at first the performance of the model rises steadily as the length of the prompt increases; after the length reaches 10, the score begins to fluctuate around 78%. It is interesting to observe that even if only one deep continuous prompt is added, our DCPCSE is still able to outperform SimCSE by 0.25 points.

Multi-task learning During experiments, we found that the auxiliary MLM objective is quite effective for RoBERTa models under the unsupervised setting, as Table 2 shows. Without the MLM loss, the performance of unsupervised DCPCSE-RoBERTa$_{base}$ even drops 8.69 points. It is reasonable that the MLM objective is capable of preventing the model from being trapped into local optima as the training progresses.

	BERT$_{base}$	RoBERTa$_{base}$
w/ MLM	78.10	77.93
w/o MLM	78.49	69.24

Table 2: Ablation study of the MLM auxiliary objective in unsupervised DCPCSE. The results are based on the test set of seven STS tasks.

4 Conclusion

In this paper, we present DCPCSE, a deep continuous prompt framework for constrastive learning of sentence embeddings. Compared with previous works which fine tune the whole language model, our architecture not only optimizes nearly 0.1% parameters, but avoids the cumbersome computation of searching handcrafted prompts. More importantly, our models can achieve new state-of-the-art performance, which significantly improves SimCSE in both unsupervised and supervised settings. DCPCSE has the potential to be a comprehensive alternative for fine-tuning and a strong baseline in the area of sentence representation.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 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@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 252–263. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 81–91. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 497–511. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 385–393. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA*, pages 32–43. Association for Computational Linguistics.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 1–14. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 15750–15758. Computer Vision Foundation / IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. Association for Computational Linguistics.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9119–9130. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021a. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Xiang Lisa Li and Percy Liang. 2021b. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4582–4597. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *CoRR*, abs/2110.07602.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021c. GPT understands, too. *CoRR*, abs/2103.10385.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 216–223. European Language Resources Association (ELRA).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Nils Reimers and Iryna Gurevych. 2019b. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020a. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Timo Schick and Hinrich Schütze. 2020b. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5065–5075. Association for Computational Linguistics.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [mask]: Learning vs. learning to recall. *arXiv preprint arXiv:2104.05240*.

A Experiment Details

For SCPCSE, we initialize the input embeddings with the manual template *This sentence : "[X]" means [MASK]*. The batch size, learning rate, epoch and valid steps we use are 256, 1e-3, 5 and 125, respectively. Other settings are the same as those in SimCSE.

For DCPCSE, the maximum sequence length is set to 32. We use the temperature $\tau = 0.05$ for all the experiments. Grid-search of batch size $\in \{64, 128, 256, 512\}$ and learning rate $\in \{$5e-3, 1e-2, 3e-2$\}$ is carried out on on STS-B development set. The hyperparameters of unsupervised setting and supervised setting are listed in Table 3 and 4, respectively. "Muiti-task" means whether the MLM objective is used.

Unsupervised	BERT		RoBERTa	
	base	large	base	large
Batch size	256	256	64	64
Learning rate	3e-2	3e-2	3e-2	1e-2
Prompt length	16	10	14	10
Muiti-task	False	False	True	True
Epoch	1	1	1	1
Valid steps	125	125	125	125

Table 3: Hyperparameters for our method in unsupervised setting.

Supervised	BERT		RoBERTa	
	base	large	base	large
Batch size	256	256	256	256
Learning rate	5e-3	5e-3	1e-2	5e-3
Prompt length	12	12	10	10
Muiti-task	False	False	False	False
Epoch	10	10	10	10
Valid steps	125	125	125	125

Table 4: Hyperparameters for our method in supervised setting.