Don't Judge a Language Model by Its Last Layer: Contrastive Learning with Layer-Wise Attention Pooling

Dongsuk Oh[*‡] **Yejin Kim**[*§] **Hodong Lee**[‡] **H. Howie Huang**[†§] **and Heuiseok Lim**[†‡]
[‡]Computer Science and Engineering, Korea University
[§]Graph Lab., George Washington University
{inow3555,bigshane319,limhseok}@korea.ac.kr
{yejinjenny, howie}@gwu.edu

Abstract

Recent pre-trained language models (PLMs) achieved great success on many natural language processing tasks through learning linguistic features and contextualized sentence representation. Since attributes captured in stacked layers of PLMs are not clearly identified, straightforward approaches such as embedding the last layer are commonly preferred to derive sentence representations from PLMs. This paper introduces the attention-based pooling strategy, which enables the model to preserve layer-wise signals captured in each layer and learn digested linguistic features for downstream tasks. The contrastive learning objective can adapt the layer-wise attention pooling to both unsupervised and supervised manners. It results in regularizing the anisotropic space of pre-trained embeddings and being more uniform. We evaluate our model on standard semantic textual similarity (STS) and semantic search tasks. As a result, our method improved the performance of the base contrastive learned BERT$_{base}$ and variants.

1 Introduction

Pre-trained language models (PLMs) (Kenton and Toutanova, 2019; Liu et al., 2019; Radford et al., 2019; Raffel et al., 2019) have shown competitive performance on many natural language processing (NLP) tasks. Also, contrastive learning using the PLMs shows the highest performance in sentence representation. Contrastive learning is to learn effective representations by staying semantically close sample pairs together while dissimilar ones are far apart(Hadsell et al., 2006).

In general, PLMs use either $[CLS]$ tokens in the last layer, AVG which is the average representation of tokens in the last layer(Reimers et al., 2019; Li et al., 2020), or AVG_{FL} which is the average representation of tokens in the first and last layers(Gao et al., 2021), to pool out sentence representation from word representations. However, since language models show performance gaps by domain when trained on different objectives, the fixed pooling strategy has limitations in performance improvement.

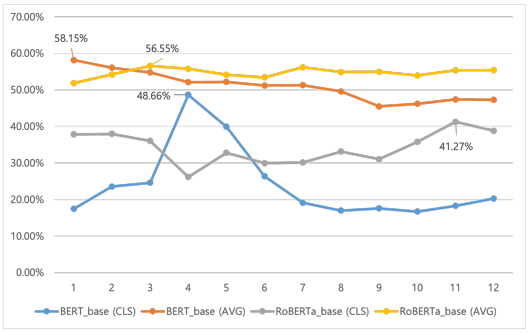

Figure 1: Spearsman's correlation score of each layer evaluated on STS-B test set

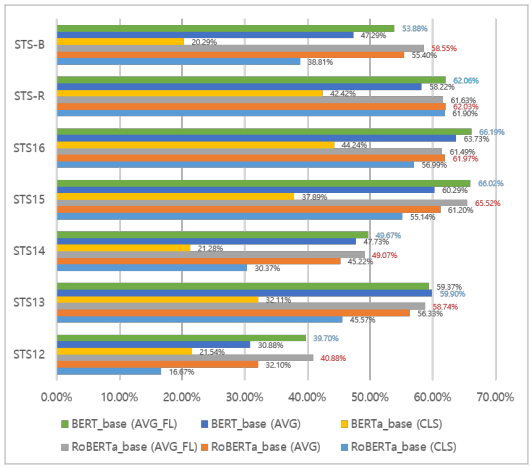

Figure 2: Spearsman's correlation score depending on the pooling methods of PLMs for each domain

Figure 1 and 2 show the Spearman's correlation score of each layer or pooling method in PLMs. We evaluated the test set of the standard semantic textual similarity (STS) dataset(Cer et al., 2017; Agirre et al., 2012, 2013, 2014, 2015, 2016; Marelli

[*] These authors contributed equally.
[†] These authors are corresponding authors.

Proceedings of the 29th International Conference on Computational Linguistics, pages 4585–4592
October 12–17, 2022.

et al., 2014).

The comparison of performance when pooling each layer shown in Figure 1 indicates that using only a specific layer for pooling is insufficient. Other layers other than the last layer may contain substantial information for sentence representation. For example, for the STS benchmark (STS-B) task(Cer et al., 2017), BERT$_{base}$ with $[CLS]$ embedding scored the highest at the fourth layer (48.66%), which is about 20% higher than the last layer.

Figure 2 shows that simply pooling from more layers impedes the performance by comparing models pooled from the first and last layer and the last layer. In addition, there is no consistent tendency to compare effectiveness for a given layer between $[CLS]$ pooling and average pooling.

Motivated by this point, we designed the attention networks and task-agnostic pooling methods to assign more weights to spots that need more focus in the layer and lead to representation vector optimization. Our proposed method outperforms previously fixed pooling strategies in contrastive learning. In addition, contrastive learning models with layer-wise attention pooling show a higher semantic search performance with the same parameters.

In summary, the contributions of this paper are as follows:

- We proposed layer-wise attention pooling* to assign weights to each layer and learn sentence representation fitted to a given task.
- To our knowledge, our pooling strategy shows the best performance out of all InfoNCE-based loss functions for the sentence embedding tasks.
- For the semantic search evaluation, we excluded the proposed pooling method in the inference phase and obtained better performance.

2 Method

In this section, we present a layer-wise pooling strategy based on attention mechanisms to improve the quality of sentence representations from language models. In addition, we describe the process of applying the proposed pooling strategy to be leveraged on three contrastive learning schemes.

*https://github.com/nlpods/LayerAttPooler

2.1 Layer-Wise Attention Pooling

This paper proposes a new layer-wise pooling based on a multiplicative attention mechanism (Luong et al., 2015). As shown in Figure 1, the performance with $[CLS]$ pooling varied dramatically according to which layer to pool from. There is no significant performance gap between layers when using AVG pooling. It can be explained that each layer can contain different information for sentence representation, while average pooling can mitigate the information gap between layers.

In Equation 1, h^a is the AVG representation, which is the mean vector of tokens in the sentence, and h^c is the input representation $[CLS]$ of each layer. α_i is the importance of the i-th layer. In Equation 2, h^l is the representation with the importance score per layer. In Equation 3, h^L is the mean vector of h^l and is the representation that contains the relevance of all layers (N is the number of layers). W_k, W_q and W_v are learnable parameters.

$$\alpha_i = \frac{W_q h_i^c W_k h_i^a}{\sum_{j \in N} W_q h_i^c W_k h_j^a} \tag{1}$$

$$h_i^l = \sum_{j \in N} \alpha_i W_v h_j^a \tag{2}$$

$$h^L = \frac{1}{N} \sum_{i}^{N} h_i^l \tag{3}$$

We add a Multi-Layer Perceptron (MLP) layer randomly initialized after pooling, following the method in the Gao et al. (2021), and keep it with random initialization. As for Equation 4, h_{last}^c is the input representation $[CLS]$ of the last layer. We concatenate the input representation h_{last}^c with the layer representation h^L as the input of an MLP. Finally, h is represented in the same dimension as the sentence representation dimension of the original language model through the MLP layer.

$$h^{CL} = [h_{last}^c; h^L] \tag{4}$$

$$h = MLP(h^{CL}) \tag{5}$$

2.2 Contrastive Learning with Layer-wise Attention Pooling

We prove that the proposed pooling strategy is effective with three training objectives l_i.

Basic Supervised Contrastive Learning We use the basic supervised contrastive learning model proposed by Chen et al. (2020). This model learns the premise(x_i) and entailment(x_i^+) of the NLI(SNLI(Bowman et al., 2015) + MNLI(Williams et al., 2018)) datasets. When $D = (x_i, x_i^+)_{i=1}^m$ is a set of paired samples, where x_i and x_i^+ are semantically related. And, it takes the cross-entropy objective with an in-batch negative(Chen et al., 2017; Henderson et al., 2017). h_i and h_i^+ are representations of x_i and x_i^+ through proposed pooling strategy. the training objective l_i is :

$$l_i = -log \frac{e^{sim(h_i, h_i^+)/\tau}}{\sum_{j=1}^{M} e^{sim(h_i, h_j^+)/\tau}} \quad (6)$$

M is the mini-batch, and τ is the temperature hyperparameter and $sim(\cdot, \cdot)$ is the cosine similarity.

Unsupervised Contrastive Learning Unsupervised contrastive learning uses $x_i^+ = x_i$ in the collection of sentences $\{x_i\}_{i=1}^m$. The idea is to use an independently sampled dropout mask for x_i and x_i^+ which gets this to work as identical positive pairs during training. And, unsupervised contrastive learning denotes $h_i^z = f(x_i, z)$ using h obtained in Equation 5. z is a random mask for dropout. It gets two embeddings with different dropout masks z, $z^{'}$ from the encoder with the same input twice, and the training objective l_i is represented:

$$l_i = -log \frac{e^{sim(h_i^{z_i}, h_i^{z_i^{'}})/\tau}}{\sum_{j=1}^{M} e^{sim(h_i^{z_i}, h_j^{z_j^{'}})/\tau}} \quad (7)$$

In Equation 7, z is the standard dropout of the transformer.

Supervised Contrastive Learning with Hard Negative Supervised contrastive learning with hard negative trains natural language inference (NLI) datasets. The NLI datasets are labeled, given one premise, as true(entailment), neutral, and definitely false (contradiction). The model predicts whether the relationship between two sentences is entailment, neutral, or contradiction. The positive pairs (x_i, x_i^+) use the entailment of the NLI(SNLI + MNLI) datasets. Next, contradiction pairs (x_i, x_i^-) from the NLI datasets are used as hard negatives. Thus, it expands from (x_i, x_i^+) to (x_i, x_i^+, x_i^-). And, (x_i, x_i^+, x_i^-) is represented as (h_i, h_i^+, h_i^-) through Equation 5. As a result, in Equation 8, the training objective l_i is :

$$-log \frac{e^{sim(h_i, h_i^+)/\tau}}{\sum_{j=1}^{M} (e^{sim(h_i, h_j^+)/\tau} + e^{sim(h_i, h_j^-)/\tau})} \quad (8)$$

3 Experiments

3.1 Experimental Setup

Our main experiments uses the STS(Cer et al., 2017; Agirre et al., 2012, 2013, 2014, 2015, 2016; Marelli et al., 2014) dataset. This data set consists of sentence pairs labeled with a similarity score between 0 and 5. The evaluation is done by the SentEval toolkit. The parameter setting of the model used in the experiment is written in Table 4 of the Appendix. Additionally, to measure the search effect and efficiency of the proposed model, it is evaluated on the same parameters as the original language model. We evaluate the performance of the semantic search† with FAISS‡ using the Quora Duplicate Questions Dataset(Shankar et al., 2021) containing more than 400,000 pairs of questions.

3.2 Main Results

In Table 1, we investigate whether the proposed layer-wise attention pooling of language models performs better in contrastive learning. The experiment compares performance by training on language models with three training objectives. All results evaluate sentence embeddings on all STS tasks. Equation 6 is basic supervised learning proposed by (Chen et al., 2020). And, Equations 7 and 8 are unsupervised, supervised learning proposed by Gao et al. (2021). However, in this paper, we could not experiment with the same parameters due to hardware. Therefore, as specified in Table 4 of the Appendix, there is a difference from the original performance because it learns by choosing a low mini-batch size. † is the original performance, and ‡ is our reimplementations. As a result, the proposed pooling strategy shows higher performance in different language models and in all domains.

3.3 Ablation Studies

We investigate performance differences according to different pooling strategies in supervised contrastive learning. All results are reported in this

†https://github.com/autoliuweijie/BERT-whitening-pytorch

‡https://github.com/facebookresearch/faiss

Unsupervised Models								
Model	**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**STS-B**	**SICK-R**	**Avg**
BERT$_{base}$(CLS$_{Last}$)(Equation 7)†	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
RoBERTa$_{base}$(CLS$_{Last}$)(Equation 7)†	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
RoBERTa$_{large}$(CLS$_{Last}$)(Equation 7)†	72.86	83.99	75.62	84.77	81.80	81.98	71.23	78.89
Our Reimplementations								
BERT$_{base}$(CLS$_{Last}$)(Equation 7)‡	69.53	78.98	75.50	80.07	79.01	78.28	71.35	76.10
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**70.27**	**80.22**	**75.65**	**80.71**	**79.74**	**79.51**	**72.18**	**76.90**
RoBERTa$_{base}$(CLS$_{Last}$)(Equation 7)‡	68.72	78.29	74.35	80.40	80.83	80.14	68.71	75.92
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**68.96**	**78.83**	**75.37**	**81.05**	**81.53**	**80.99**	**69.03**	**76.54**
RoBERTa$_{large}$(CLS$_{Last}$)(Equation 7)‡	70.82	79.66	76.26	83.25	81.86	81.25	71.09	77.74
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**71.52**	**79.86**	**76.86**	**83.50**	**82.38**	**84.56**	**71.46**	**78.59**
Supervised Models								
BERT$_{base}$(CLS$_{Last}$)(Equation 8)†	75.30	84.67	80.19	85.40	80.82	84.25	80.39	81.57
RoBERTa$_{base}$(CLS$_{Last}$)(Equation 8)†	76.53	85.21	80.95	86.03	82.57	85.83	80.50	82.52
RoBERTa$_{large}$(CLS$_{Last}$)(Equation 8)†	77.46	87.27	82.36	86.66	83.93	86.70	81.95	83.76
Our Reimplementations								
BERT$_{base}$(CLS$_{Last}$)(Equation 8)‡	70.50	80.77	79.52	83.82	81.17	84.34	79.04	79.88
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**71.34**	**80.84**	**79.76**	**83.86**	**81.42**	**86.76**	**79.80**	**80.54**
RoBERTa$_{base}$(CLS$_{Last}$)(Equation 8)‡	70.80	81.31	79.60	83.48	82.86	85.71	79.77	80.50
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**71.35**	**81.44**	**79.82**	**83.79**	**83.89**	**87.42**	**80.11**	**81.12**
RoBERTa$_{large}$(CLS$_{Last}$)(Equation 8)‡	72.36	83.06	81.99	85.39	85.51	87.11	80.46	82.27
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**72.65**	**84.41**	**82.31**	**86.38**	**85.54**	**87.58**	**81.56**	**82.92**
BERT$_{base}$(CLS$_{Last}$)(Equation 6)	69.29	78.69	76.45	80.87	79.82	79.41	76.41	77.28
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**69.58**	**78.84**	**76.70**	**81.13**	**80.11**	**87.23**	**76.45**	**78.58**
RoBERTa$_{base}$(CLS$_{Last}$)(Equation 6)	68.85	77.28	74.67	80.11	80.80	87.42	76.51	77.95
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**69.51**	**78.72**	**75.97**	**81.32**	**81.47**	**89.58**	**76.83**	**79.06**
RoBERTa$_{large}$(CLS$_{Last}$)(Equation 6)	70.82	80.33	77.79	82.03	83.04	85.38	76.84	79.46
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**71.15**	**81.45**	**78.04**	**83.03**	**83.09**	**88.88**	**77.39**	**80.43**

Table 1: Performance of sentence embedding on all STS tasks (Spearman's correlation). †: published in Gao et al. (2021); and ‡: models from our reimplementations. We are shown in bold the highest performance among models from our reimplementation.

section using the STS-B test set. All models extract sentence embeddings by adding an MLP layer as suggested in Gao et al. (2021). Table 2 shows the performance difference between the fixed pooling method and the layer-wise attention pooling. Additionally, we compare the representation concatenated between fixed pooling because we construct the h representation by concatenating h^c_{last} and h^L. The layer-wise attention pooling shows the results of ablation studies with $[CLS]$ and AVG. For $[CLS]_{All}$ and AVG_{All}, h^l computes the importance between each layer and the others. In addition, $[CLS]_{All}$ + AVG_{All} represent h^l by calculating the importance between $[CLS]$ and AVG of all layers. All of these methods show higher performance than the fixed pooling strategy. However, as described in Section 2, the pooling strategy concatenated with $[CLS]_{Last}$ shows the highest performance.

Model	**STS-B**
BERT$_{base}$(Equation 8)	
w/ (CLS$_{Last}$)	84.34
w/ (AVG$_{Last}$)	84.84
w/ (AVG$_{FL}$)	84.76
w/ (AVG$_{Last}$+AVG$_{FL}$ concat)	84.93
w/ (CLS$_{Last}$+AVG$_{Last}$ concat)	85.11
w/ LayerAttPooler(CLS$_{All}$ attention)	85.45
w/ LayerAttPooler(AVG$_{All}$ attention)	85.72
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention)	86.57
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**86.76**

Table 2: Ablation studies of different pooling methods in supervised model on STS-B task (Spearman's correlation)

3.4 Semantic Search Results

In Table 3, we compare semantic search speed and performance on the same parameters. This experiment proves that the proposed pooling strategy is effective for training the language models and also for semantic search performance with the same parameters during inference. Sentence embeddings for all supervised learning models use $[CLS]_{Last}$. MRR@10 is used to measure the performance of semantic search, and Average Retrieval Time (ms) measures retrieval efficiency. Memory Usage (GB) shows memory usage. FAISS experiments in CPU mode. nlist = 1024 and the CPU is Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz. Result shows that the performance of semantic search is higher when the proposed pooling strategy is used during training.

Model	MRR@10	Average Retrieval Time (ms)	Memory Usage (GB)
BERT$_{base}$(Equation 8)			
w/ (CLS$_{Last}$)	63.48	1.46	0.25
w/ LayerAttPooler (train)	**64.32**	**1.45**	0.25
RoBERTa$_{base}$(Equation 8)			
w/ (CLS$_{Last}$)	63.89	1.56	0.25
w/ LayerAttPooler (train)	**65.05**	**1.48**	0.25
RoBERTa$_{large}$(Equation 8)			
w/ (CLS$_{Last}$)	65.85	2.22	0.33
w/ LayerAttPooler (train)	**66.32**	**2.21**	0.33

Table 3: Performance of semantic search evaluation using the Quora Duplicate Questions Dataset with FAISS. w/ LayerAttPooler (train) : remove layer-wise attention pooling after training

4 Conclusion

In this work, we propose layer-wise attention pooling to capture the importance of the weight in each layer for the pre-trained language models (PLMs). Training layer-wise attention layer with contrastive learning objectives outperforms BERT and variants of PLMs. No matter what pooling method is used, our model achieved higher scores than prior state-of-the-art models. In addition, this layer-wise attention technique also can be exploited in semantic search tasks, in which more cost-efficient computation (i.e. less latency and memory usage) is required. The model trained with our method obtained higher performance with the same or less time and memory usage, even if the added attention layer is detached in the inference stage.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques) and by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2022-2018-0-01405) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) In addition, This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425)

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 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)*, pages 252–263.

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 *SemEval@ COLING*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 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)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 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*, pages 32–43.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 767–776.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James Glass. 2022. Diffcse: Difference-based contrastive learning for sentence embeddings. *arXiv preprint arXiv:2204.10298*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

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)*, pages 9119–9130.

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*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

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'14)*, pages 216–223.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, Iryna Gurevych, Nils Reimers, Iryna Gurevych, et al. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 671–688. Association for Computational Linguistics.

Iyer Shankar, Dandekar Nikhil, and Csernai Kornél. 2021. First quora dataset release: Question pairs. *Retrieved July 7th*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, pages 1112–1122.

A Training Details

Unsupervised Models		
Models	**Batch Size**	**Learning Rate**
BERT$_{base}$(Equation 7)	64	3e-5
w/ LayerAttPooler	64	3e-5
RoBERTa$_{base}$(Equation 7)	256	1e-5
w/ LayerAttPooler	256	1e-5
RoBERTa$_{large}$(Equation 7)	256	3e-5
w/ LayerAttPooler	256	3e-5
BERT$_{base}$(DiffCSE)	64	7e-6
w/ LayerAttPooler	64	3e-5
Supervised Models		
BERT$_{base}$(Equation 6)	256	5e-5
w/ LayerAttPooler	256	1e-5
RoBERTa$_{base}$(Equation 6)	256	5e-5
w/ LayerAttPooler	256	3e-5
RoBERTa$_{large}$(Equation 6)	256	1e-5
w/ LayerAttPooler	256	5e-5
BERT$_{base}$(Equation 8)	256	5e-5
w/ LayerAttPooler	256	2e-5
RoBERTa$_{base}$(Equation 8)	256	5e-5
w/ LayerAttPooler	256	3e-5
RoBERTa$_{large}$(Equation 8)	256	1e-5
w/ LayerAttPooler	256	5e-5

Table 4: Batch sizes and learning rate for each models

Due to hardware problems, Equations 7 and 8 train at a smaller batch size than the Gao et al. (2021) paper. The GPU used in the experiment is RTX 8000, and the the hyperparameters are specified in the Table 4.

B Experiments on Different Model

DiffCSE model	
Model	**STS-B**
BERT$_{base}$(CLS$_{Last}$) (w/o BatchNorm)†	83.23
w/ LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)	**83.87**

Table 5: Development set results of STS-B. †: published in Chuang et al. (2022); Bold shows the highest performance among models from our reimplementation.

We experiment with whether the proposed pooling strategy is effective for a contrastive learning model with a different structure. DiffCSE model (Chuang et al., 2022) improves the performance of sentence representation by adding generator and discriminator structures of ELECTRA (Clark et al., 2020). While training, DiffCSE freezes the generator's weight and updates the sentence encoder and discriminator for sentence embedding with the contrastive learning objective. However, the discriminator is not used for inference since only representations from the sentence encoder and generator are needed. We applied our proposed pooling strategy to the sentence encoder with a contrastive learning objective. As a result, layer-wise attention pooling improves the performance of the DiffCSE model (Table 5). We use the one linear layer with the tanh activation function following SimCSE as in Equation 5, while DiffCSE uses a two-layer pooler with Batch Normalization (BatchNorm) (Ioffe and Szegedy, 2015). However, BatchNorm is not used for a fair comparison of results.

C Analysis Attention Weights over Layers

(a) Attention scores of LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention) + (CLS$_{Last}$ concat)

(b) Attention scores of LayerAttPooler(CLS$_{All}$ + AVG$_{All}$ attention)

Figure 3: Attention scores of layer-wise pooling only (b) and concatenating the $[CLS]_{Last}$ representation of the last layer (a) on a sentence "You should do it." (short) and "People on motorcycles wearing racing gear ride around a racetrack." (long) sentences. These scores are implemented on BERT$_{base}$.

We also analyze the layer-wise attention scores depending on the length of sentences. Figure 3 (a) case explains that the last layer relatively contains more information than other layers by the $[CLS]$ token of the last layer. However, the attention score of the last layer is calculated differently for the long and short sentences. Figure 3 (b) case indicates that other layers than the last layer have substantial information for the same sentence, and the balanced attention weight per layer supports it.