LoRACSE: Contrastive Learning of Sentence Embeddings using LoRA

Sikai Li
University of Michigan, Ann Arbor
skevinci@umich.edu

Yilin Jia
University of Michigan, Ann Arbor
kirp@umich.edu

Yuqi Mai
University of Michigan, Ann Arbor
yqmai@umich.edu

1 Introduction

The development of universal sentence embeddings, which capture high-level semantic information in a way that is not specific to any particular task, is a crucial research area in the field of Natural Language Processing (NLP)[17]. These embeddings[2, 9, 14, 18] can greatly benefit a variety of applications, including information retrieval and question answering[20]. Recently, there has been significant success in using contrastive learning to fine-tune Pre-trained Language Models (PLMs) for the purpose of learning universal sentence representations[17]. Works show that fine-tuning PLMs using contrastive learning has proven to be an effective approach[11, 19]. Contrastive learning aims to bring semantically similar samples together and separate dissimilar samples, leading to improved representation of sentence semantics. In recent works, positive pairs are created through data augmentation or using supervised datasets, while negative pairs are formed from different sentences within the same mini-batch[17]. A typical example is the SimCSE[10] model, which uses standard dropout as a means of constructing positive pairs and demonstrates exceptional performance on seven standard Semantic Textual Similarity tasks[17].

Recently, Jiang et al. [16] have proposed a novel contrastive learning model for learning sentence representation PromptBERT, which adds hard prompt and prefix prompt to the sentences, which is also like a template, to make BERT and RoBERTa achieve better sentence embeddings. Apart from PromptBERT, PromCSE, proposed by Jiang et al.[17], achieved similar performance to PromptBERT and even better results on the same datasets. It changes hard prompt and prefix prompt to soft prompt (a set of trainable vectors) that is added to the model as a few layers. Moreover, PromCSE only needs to train this small-scale soft prompt while keeping pretrained language models fixed.

However, the models presented in prior works have exhibited a significant drawback in that they necessitate a substantial quantity of training parameters and epochs to attain convergence. Given the constraints of computational and temporal resources, it is desirable to obtain a model that demands fewer training parameters while simultaneously maintaining optimal performance.

In this paper, we propose *LoRACSE*, **c**ontrastive **l**earning **of s**entence **e**mbeddings using **LoRA**. Since the previous studies have demonstrated that a substantial amount of parameters and a considerable number of epochs are necessary to achieve convergence, the amount of trainable parameters in our LoRACSE is decreased significantly. In Table.2, we compare our model's training parameters to those of previous studies that used BERT$_{base}$ as a pretrained model. Our results reveal that our model has significantly fewer parameters, accounting for only 0.067% of the total model to train. Furthermore, our model requires only 2 epochs to converge, which is substantially less than the previous works. Despite using fewer parameters and training epochs, our model's performance remains comparable to that of the previous works. All three students contribute equally to the project.

Model	**LoRACSE**	**PromCSE**	**SimCSE**
Epoch	**3**	10	10
Trainable (%)	**0.06725**	0.27009	100
Avg Score	81.55	81.81	81.57

Table 1. Parameters Comparison on BERT$_{base}$

Model	**LoRACSE**	**PromCSE**	**SimCSE**
Epoch	**3**	10	10
Trainable (%)	**0.027656**	0.21838	100
Avg Score	84.69	84.76	83.76

Table 2. Parameters Comparison on RoBERTa$_{large}$

2 Related Work

2.1 Contrastive Learning

Recent works show that the principle of contrastive learning[12] is to draw positive sample pairs together and push away negative sample pairs. Well-designed Siamese networks have been already created to effectively carry out contrastive learning[6, 13, 23].

2.2 Contrastive Learning in Sentence Embedding

Recent years, contrastive learning have been broadly used in the sentence embedding task. Gao et al.[10] have presented

EECS 487, WN 2023, Ann Arbor, MI, USA

SimCSE, which fine-tunes all the parameters of BERT using the contrastive learning objective. Apart from SimCSE, Jiang et al.[17] have proposed PromCSE, a prompt-based contrastive learning model to learn the sentence embeddings. Moreover, Chuang et al.[7] have shown us DiffCSE, which develops a kind of sentence embedding that is able to differentiate between the original sentence and a modified version generated by randomly obscuring the original sentence and sampling from a masked language model. Their models achieved state-of-the-art performances in the sentence embedding area.

2.3 Pre-trained Language Model

Pre-trained language models, such as BERT, GPT-2, and GPT-3, have gained significant attention in the natural language processing (NLP) community in recent years. These models have set the stage for the development of even larger and more powerful language models, and have opened up new avenues for research in NLP.

2.4 LoRA

Compared to SimCSE that needs to tune all the parameters in the large language model, LoRA freezes the pretrained language model as PromCSE did and adds pairs of rank-decomposition weight matrices (called update matrices) to existing weights, matrix A with a dimension of $d \times r$ and matrix B with a dimension of $r \times d$, where d is the dimension of the current weights and r it the rank of LoRA that can be set while training and thus, control the number of trainable parameters. Then, it only trains those newly added weights and finally adds the multiplication of A and B to the fixed PLMs' weights.

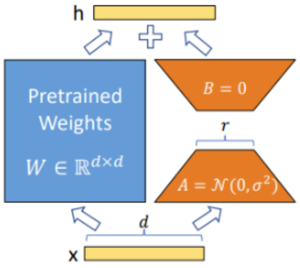
h
Pretrained Weights
$W \in \mathbb{R}^{d\times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Figure 1. LoRA architecture.[15]

3 Data Preprocessing

3.1 Training Dataset

In order to conduct a comparative analysis of our model with prior works, we utilized the identical dataset employed by PromCSE and SimCSE. Specifically, our training set consisted of a combination of MNLI and SNLI. Previous work[10] have shown that these two datasets is comprised of top-notch, crowd-sourced pairs, which task annotators with composing three sentences for a single premise: one that is unequivocally true (entailment), one that is potentially true (neutral), and one that is unquestionably false (contradiction). This means that for every premise and its corresponding entailment hypothesis, there is one similar hypothesis as well as a corresponding contradiction hypothesis. For simplicity, the similar hypothesis is tagged as *Sent1* and contradiction hypothesis is tagged as *Hard_neg* in the training dataset. As an illustration, one of such tuple is shown in Table.3

Label	Text
Sent0	*She smiled back*
Sent1	*She was happy*
Hard_neg	*She frowned*

Table 3. Training Data Sample

3.2 Evaluation & Test Dataset

3.2.1 Standard STS.

Dataset and Metric. We assess the effectiveness of our sentence embedding approach by employing seven well-established STS datasets, including STS tasks 2012-2016[1, 3–5], STS Benchmark[2], and SICK-Relatedness[21]. These datasets consist of pairs of sentences obtained from diverse sources, including news, forums, and lexical definitions. Each sentence pair is assigned a score between 0 to 5 indicating its semantic similarity. To evaluate our technique, we utilize the SentEval toolkit[8] and calculate the Spearman's correlation on the test sets, which is in line with previous research.

Baseline (need modification). We compare our unsupervised and plan to compare supervised LoRACSE with previous state-of-the-art sentence embedding methods for baseline comparison. Moreover, we introduce strong unsupervised baselines that leverage contrastive learning, which include CT-BERT, Mirror-BERT, SimCSE, DiffCSE, and Prompt-BERT [7, 10, 16, 17, 22]. In addition, we compare our model with methods that use extra supervision, such as InferSent, Universal Sentence Encoder, SBERT with BERT-flow, whitening, and CT, and SimCSE[7, 10, 22].

4 Methodology

4.1 Contrastive Learning Loss Function

For the contrastive learning objective, we choose the same loss function as SimCSE and PromCSE, as shown in Fig.1, which is the most widely adopted training objective NT-Xent loss[17].

$$\mathcal{L}_{CL} = -\log \frac{e^{\mathrm{sim}\left(\mathbf{h}_i, \mathbf{h}_i^+\right)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}\left(\mathbf{h}_i, \mathbf{h}_j^+\right)/\tau}} \quad (1)$$

If we have a collection of paired sentences $D = (X_i, X_i^+)_{i=1}^m$, where X_i and X_i^+ have similar meanings, we consider X_i^+ to be the positive pair of X_i. For the other sentences in the same mini-batch, we treat them as negative examples. We can represent the sentence embeddings of X_i and X_i^+ as h_i and h_i^+, respectively. Using these embeddings, we can formulate the NT-Xent loss for a single sample in a mini-batch of size N as shown in the above figure[17].

4.2 PEFT

Parameter-Efficient Fine-Tuning (PEFT) is the state-of-the-art parameter fine-tuning library, which facilitates the efficient adaptation of pre-trained language models (PLMs) to diverse downstream applications while circumventing the need to fine-tune all parameters of the model. Given that fine-tuning large-scale PLMs is often cost-prohibitive, PEFT methods offer a viable alternative by enabling the fine-tuning of only a small number of supplementary model parameters. This, in turn, leads to significant reductions in computational and storage expenses. Recently developed state-of-the-art PEFT techniques have demonstrated performance levels comparable to those achieved through full fine-tuning.

4.3 Model

Our models have been developed utilizing Huggingface's transformers, and we obtain pre-trained checkpoints of RoBERTa [19], another pretrain model from the same source. With the help of the PEFT, we employed the LoRA technique as illustrated in Fig. 2. Specifically, the LoRA architecture is appended to various matrices within each attention layer, with matched shapes. During training, we solely optimize the parameters in the Lora weight matrix, while keeping all the parameters in the original pretrained models unchanged.

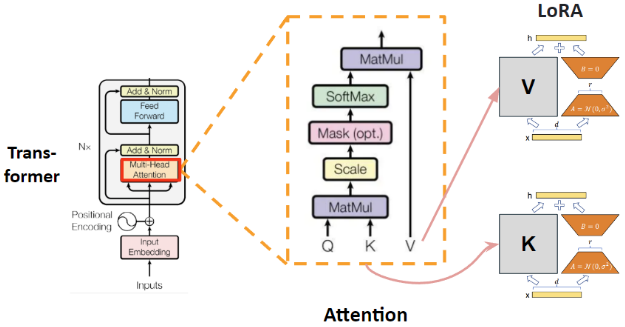

Figure 2. LoRACSE architecture.

5 Experiments

5.1 Experiments on Different Weight Matrices

Here we explore the difference of applying LoRA technique to different weight matrices, which are query, key and value matrices of every transformer layer and weight matrices in the final FFNN, with the rank or LoRA fixed as $r = 4$ and RoBERTa$_{base}$ being the PLM. The results is shown in Fig.2

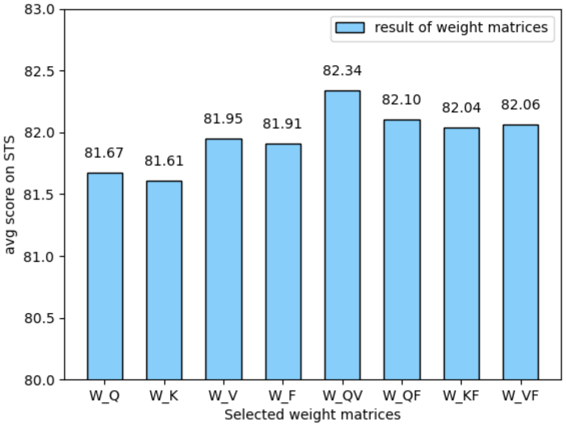

Figure 3. Apply LoRA on different weight matrices.

Since weight matrices of query, value and FFNN achieves relatively high average scores, we further conduct four experiments that apply LoRA to both query and value, query and FFNN, key and FFNN and value and FFNN. From Fig. 2, we can see that applying LoRA to both query and value matrices achieves highest average scores.

5.2 Experiments with Different Ranks

After completing the above experiments, we keep applying LoRA technique to query and value matrices of every transformer layer and conduct experiments with different ranks of LoRA, which means tuning different amounts of parameters in our model because the amounts of trainable parameters in LoRACSE is determined by the scale of ranks. We've tried rank = 1, 2, 4 and 8 here with RoBERTa$_{base}$ being the PLM and the corresponding results are shown in Figure.4.

5.3 Experiments with Different PLMs

5.3.1 Hyperparameters. After doing some basic experiments considering the amounts of trainable parameters and the matrices we apply LoRA to, we train our LoRACSE using the hyperparameters shown in Table. 4 and compare the results with SimCSE and PromCSE with the same PLMs. These hyperparameters are obtained through hyperparameter-search. Due to the lack of computational resources (especially GPU memory), we are only able to train the LoRACSE-RoBERTa-large model with batch size of 375.

5.3.2 Comparison Experiments. We train the model with the hyperparameters shown in Table.4 and apply LoRA technique to the $\boldsymbol{V}$ and $\boldsymbol{K}$ matrices in the Pretrained Language Models. The detailed result is shown in Table.6 and

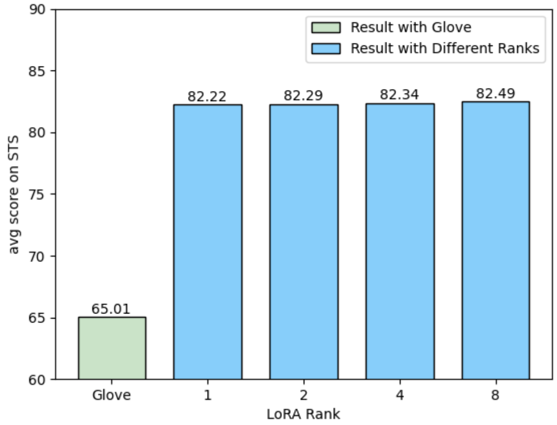

Figure 4. Apply LoRA with Different Ranks

Model	**BERT$_{base}$**	**RoBERTa$_{base}$**	**RoBERTa$_{large}$**
Batch Size	512	512	***375***
Learning Rate	2^{-4}	2^{-4}	2^{-4}
Epoch	3	3	3
Validation Steps	125	125	125
Rank	2	8	1

Table 4. Hyperparameters

Fig.5. The presented graph and table illustrate that our supervised LoRACSE model achieves a notably superior performance in comparison to the two baseline models, namely Glove embedding and Universal Sentence Encoder. Furthermore, despite training a substantially fewer number of parameters, our model outperforms SimCSE when we used Roberta$_{large}$ and performs comparably to SimCSE and PromCSE in other cases.

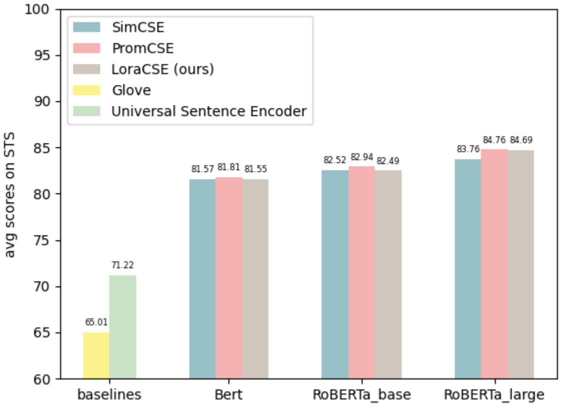

Figure 5. Result of Comparison Experiment

6 Discussion

6.1 Experiments on Different Weight Matrices

The figure presented in Figure 3 demonstrates that when implementing the LoRA technique on a single weight matrix, appending it to the V and FFNN matrices performs better than using the Q and K matrices. In addition, we conducted experiments by applying LoRA to two weight matrices simultaneously, selecting those that achieved high scores in the previous experiments(since we don't have sufficient computational resources), namely, Q & V, Q & FFNN, K & FFNN, and V & FFNN. The results reveal that the highest score of 82.34 was obtained when implementing LoRA on both Q and V matrices.

6.2 Experiments with Different Ranks

From Fig.4, we can see that with RoBERTa$_{base}$ being the PLM and applying LoRA to both query and value weight matrices of transformer's every layer, the average score increases as the rank of LoRA becomes larger. It's intuitive that the performance becomes better with larger amount of trainable parameters. However, when rank = 1, the score is 82.22 and it is 82.49 when the rank = 8. The difference between these scores is actually really small. Therefore, our thought is that the number of rank doesn't affect the performance of this task significantly, and choosing the rank between this range is reasonable for conducting the experiments.

6.3 Experiments with Different PLMs

From Table.4 and Figure.5, we can discern that our LoRACSE model performs comparably to SimCSE on all three pretrained language model on Bert and RoBERTa$_{base}$. And it outperforms SimCSE significantly on RoBERTa$_{large}$, which has an increment by nearly 1%. It is essential to note, though, that LoraCSE was trained using merely 0.02% of the model parameters utilized in SimCSE, and 10% of the model parameters of PromCSE, which leads to significant savings in training time and GPU memory. We also find that the model performs better on RoBERT$_{base}$ when the LoRA rank is 8, while on RoBERTa$_{large}$, LoRA rank equals to 1 is better as shown in Table.5. The reason may be that the large PLMs have already comsisted of large amount of parameters, and tuning with a higher LoRA rank will lead to overfitting. Therefore, larger PLMs prefer lower LoRA rank.

LoRA Rank	1	2	4	8
Avg Score	84.69	84.35	84.22	84.08

Table 5. Avg Score on RoBERTa$_{large}$ of Different LoRA Rank

References

[1] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Matthew Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio,

Model	**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**STS-B**	**SICK-R**	**Avg.**
				Supervised models				
InferSent-GloVe♣	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder♣	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
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
CT-SBERT$_{base}$◇	74.84	83.20	78.07	83.84	77.93	81.46	76.42	79.39
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
PromCSE-BERT$_{base}$ ♠	75.58	84.33	79.67	85.79	81.24	84.25	80.79	81.81
∗ LoraCSE-BERT$_{base}$ (ours)	75.17	84.18	79.67	86.25	81.30	84.24	80.02	81.55
SimCSE-RoBERTa$_{base}$◇	76.53	85.21	80.95	86.03	82.57	85.83	80.50	82.52
PromCSE-RoBERTa$_{base}$ ♠	76.75	85.86	80.98	86.51	83.51	86.58	80.41	82.94
∗ LoraCSE-RoBerTa$_{base}$(ours)	75.84	85.82	80.79	86.01	83.27	85.69	80.00	82.49
SimCSE-RoBERTa$_{large}$◇	77.46	87.27	82.36	86.66	83.93	86.70	81.95	83.76
PromCSE-RoBERTa$_{large}$ ♠	**79.14**	**88.64**	**83.73**	87.33	**84.57**	**87.84**	**82.07**	**84.76**
∗ LoraCSE-RoBerTa$_{large}$ (ours)	79.06	88.57	83.62	**87.80**	84.50	87.26	82.00	84.69

Table 6. The performance of different sentence embedding models on test sets of STS tasks (Spearman's correlation). The best performance and the second-best performance methods are denoted in bold and underlined fonts respectively. ♣: results from [22]; ◇: results from [10]; †: results from [19]; ♦: results from [7]; △: results from [16];+ EH: adding the Energy-based Hinge loss as shown in [16]; ♠: results [17]; ∗ : results from our experiments;

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 *SemEval@NAACL-HLT*.

[2] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Matthew 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 *International Workshop on Semantic Evaluation*.

[3] Eneko Agirre, Carmen Banea, Daniel Matthew 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 *International Workshop on Semantic Evaluation*.

[4] Eneko Agirre, Daniel Matthew Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In *International Workshop on Semantic Evaluation*.

[5] Eneko Agirre, Daniel Matthew Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *International Workshop on Semantic Evaluation*.

[6] Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning Representations by Maximizing Mutual Information Across Views. https://doi.org/10.48550/ARXIV.1906.00910

[7] 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. https://doi.org/10.48550/ARXIV.2204.10298

[8] Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (Eds.). European Language Resources Association (ELRA), Miyazaki, Japan.

[9] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. https://doi.org/10.48550/ARXIV.1705.02364

[10] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. https://doi.org/10.48550/ARXIV.2104.08821

[11] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2020. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. https://doi.org/10.48550/ARXIV.2006.03659

[12] R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. 1735–1742. https://doi.org/10.1109/CVPR.2006.100

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2019. Momentum Contrast for Unsupervised Visual Representation Learning. https://doi.org/10.48550/ARXIV.1911.05722

[14] Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning Distributed Representations of Sentences from Unlabelled Data. https://doi.org/10.48550/ARXIV.1602.03483

[15] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL]

[16] Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. PromptBERT: Improving BERT Sentence Embeddings with Prompts. https://doi.org/10.48550/ARXIV.2201.04337

[17] Yuxin Jiang, Linhan Zhang, and Wei Wang. 2022. Improved Universal Sentence Embeddings with Prompt-based Contrastive Learning and Energy-based Learning. https://doi.org/10.48550/ARXIV.2203.06875

[18] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. https://doi.org/10.48550/ARXIV.1506.06726

[19] Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021. Fast, Effective, and Self-Supervised: Transforming Masked Language Models into Universal Lexical and Sentence Encoders. https://doi.org/10.48550/ARXIV.2104.08027

[20] Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. https://doi.org/10.48550/ARXIV.1803.02893

[21] 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)* (26-31), Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (Eds.). European Language Resources Association (ELRA), Reykjavik, Iceland.

[22] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. https://doi.org/10.48550/ARXIV.1908.10084

[23] Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. 2019. Unsupervised Embedding Learning via Invariant and Spreading Instance Feature. https://doi.org/10.48550/ARXIV.1904.03436