WEIGHTED SAMPLING FOR MASKED LANGUAGE MODELING

Linhan Zhang†, Qian Chen‡, Wen Wang‡, Chong Deng‡, Xin Cao†, Kongzhang Hao†, Yuxin Jiang*, Wei Wang*

† University of New South Wales, School of Computer and Engineering
‡ Speech Lab of DAMO Academy, Alibaba Group
∗ Hong Kong University of Science and Technology (Guangzhou), China

{linhan.zhang,xin.cao}@unsw.edu.au
{tanqing.cq,w.wang,dengchong.d}@alibaba-inc.com
weiwcs@ust.hk

ABSTRACT

Masked Language Modeling (MLM) is widely used to pretrain language models. The standard random masking strategy in MLM causes the pre-trained language models (PLMs) to be biased towards high-frequency tokens. Representation learning of rare tokens is poor and PLMs have limited performance on downstream tasks. To alleviate this frequency bias issue, we propose two simple and effective **Weighted Sampling** strategies for masking tokens based on token frequency and training loss. We apply these two strategies to BERT and obtain **Weighted-Sampled BERT (WSBERT)**. Experiments on the Semantic Textual Similarity benchmark (STS) show that WSBERT significantly improves sentence embeddings over BERT. Combining WSBERT with calibration methods and prompt learning further improves sentence embeddings. We also investigate fine-tuning WSBERT on the GLUE benchmark and show that Weighted Sampling also improves the transfer learning capability of the backbone PLM. We further analyze and provide insights into how WSBERT improves token embeddings.

Index Terms— Weighted Sampling, Mask Language Model, Sentence Representation, GLUE Evaluation

1. INTRODUCTION

Early language models model context unidirectionally, either left-to-right or right-to-left. In contrast, Masked Language Modeling (MLM) replaces a subset of tokens in the input sequence with a special token [MASK] and trains the model to predict the masked tokens using their bidirectional context. MLM has been widely adopted as a self-supervised pre-training objective for learning bidirectionally contextualized language representations, such as BERT [1] and RoBERTa [2]. BERT and its extensions as pre-trained language models (PLMs) have shown remarkable performance on various downstream NLP tasks.

Nevertheless, recent studies reveal critical problems in MLM. [3, 4] find the contextualized word representations of BERT and other PLMs are not isotropic as they are not uniformly distributed w.r.t. direction; instead, they are anisotropic as word representations occupy a narrow cone. The token frequency in the pre-training data usually follows a long-tailed distribution. The conventional masking strategy for MLM selects tokens to mask with a uniform distribution [1, 2]. This random masking strategy for MLM unavoidably encounters the frequency bias issue, that is, high-frequency tokens will be masked frequently, while more informative tokens, typically with lower frequencies, will be masked much less frequently during pre-training, which would greatly harm the efficiency of pre-training,

Fig. 1. An example from WikiText. Randomly selected tokens are in blue while Frequency Weighted Sampled tokens are in pink.

lower the quality of representations of rare tokens and limit the performance of PLMs. As shown in Figure 1, tokens selected based on their frequency (in pink, see Eqn. 2) are apparently more informative than tokens selected randomly (in blue) which are mostly high-frequency tokens but not essential to the semantics of the sentence. [5] investigates the embedding space of MLM-trained PLMs and confirms that embeddings are biased by token frequency and rare tokens are distributed sparsely in the embedding space. [6] demonstrates that frequency bias indeed harms the performance of sentence embeddings generated by MLM-trained PLMs. As shown in these studies, alleviating the frequency bias issue is essential for improving effectiveness of MLM and performance of resulting PLMs.

Several recent studies focus on improving efficiency of pre-training, including mixed-precision training [7], parameter distillation for different layers [8], introducing a note dictionary for saving information of rare tokens [9], designing different training objectives [10, 11, 12], and dropping redundant tokens during pre-training [13]. However, most of these approaches focus on modifying model architecture or optimization for pre-training.

Our work focuses on alleviating the frequency bias issue in MLM and improving quality of PLMs. We propose two **Weighted Sampling** methods for masking tokens based on token frequency or training loss. The latter one can dynamically adjust sampling weights and achieve a good balance between masking probabilities of *common tokens* and *rare tokens*[1] based on the learning status of PLMs. Our Weighted Sampling methods can be applied to *any* MLM-pretrained PLMs. In this work, we focus on investigating the effectiveness of applying Weighted Sampling to BERT as the backbone. We initialize from BERT and continue pre-training with

---

[1]We denote high-frequency and low-frequency tokens by *common tokens* and *rare tokens* in the rest of the paper.

Weighted Sampling. We denote the resulting PLM by **WSBERT**. We hypothesize that since Weighted Sampling could alleviate frequency bias, it could improve representation learning of rare tokens and also improve the overall quality of language representations. Quality of pre-trained language representations is generally evaluated on sentence representations generated by PLMs, commonly evaluated on the Semantic Textual Similarity (STS) benchmark [14, 15, 16, 17, 18, 19, 20]; and evaluated on transfer learning capability of PLMs, commonly evaluated on fine-tuning and testing on the GLUE benchmark [21]. Recent efforts on sentence representation modeling include calibration methods [5, 22], prompt learning[23, 24, 25, 26, 27, 6], and sentence-level contrastive learning (CL) based models such as SimCSE [28] and its variants [29, 30]. Although SimCSE and its variants achieve state-of-the-art (SOTA) performance on STS, they degrade the transfer learning capability on tasks such as SQuAD since they do not target improving token-level representation learning [31]. We also observe absolute 0.5 performance degradation on GLUE from SimCSE-BERT compared to BERT.

In this work, to investigate whether the proposed Weighted Sampling could improve the quality of token embeddings, we evaluate sentence representations generated by WSBERT on STS and the transferability of WSBERT on GLUE. We also analyze the embedding space of WSBERT and BERT to understand how Weighted Sampling improves the quality of token embeddings. Our contributions can be summarized as follows:

- We propose two **Weighted Sampling** methods to alleviate the frequency bias issue in conventional masked language modeling.
- We develop a new PLM, WSBERT, by applying Weighted Sampling to BERT. Different from SOTA sentence representation models, we find WSBERT outperforms BERT on **both sentence representation quality and transfer learning capability**. We also find integrating calibration methods and prompts into WSBERT further improve sentence representations.
- We design ablation approaches to analyze the embedding space of WSBERT and BERT. We find that with Weighted Sampling, rare tokens are more concentrated with common tokens and common tokens are more concentrated in the embedding space than BERT. We also find that both common and rare tokens are closer to the origin in WSBERT than BERT and token embeddings of WSBERT are less sparse than BERT. We believe these improvements in token embeddings caused by Weighted Sampling contribute to the improvements in sentence representations and transferability.

2. METHOD

In this section, we first describe traditional Masked Language Modeling (MLM). Then we propose **Weighted Sampling** for MLM to alleviate the frequency bias problem.

2.1. Masked Language Modeling

For a sentence $S = \{t_1, t_2, \ldots, t_n\}$, where n is the number of tokens and t_i is a token, the standard masking strategy as in [1] randomly chooses 15% of tokens to mask. The language model learns to predict the masked tokens with bidirectional context. To make the model compatible with fine-tuning, for a chosen token, 10% of the time it is replaced by a random token from the corpus, 10% of the time it remains unchanged, and 80% of the time it is replaced by a special token [MASK].

Fig. 2. Illustration of the proposed **Dynamic Weighted Sampling** for mask language modeling (MLM). The sampling weight of choosing a token to mask is computed based on the prediction loss of this token by the current PLM. We store the sampling weights of each token in the weight dictionary.

2.2. Weighted Sampling

In order to tackle the frequency bias problem, we propose two weighted sampling strategies, namely, **Frequency Weighted Sampling** and **Dynamic Weighted Sampling**, to compute the masking probability for each token, based on statistical signals and model-based signals, respectively.

2.2.1. Frequency Weighted Sampling

A natural statistical signal characterizing the informativeness of a token w is its frequency $\mathrm{freq}(w)$ in the pre-training corpus. We first apply the following transformation to remove the excessive influences of extremely rare tokens, which are usually noise.

$$\mathrm{freq}^*(w) = \begin{cases} \mathrm{freq}(w) & , \text{if } \mathrm{freq}(w) > \theta \\ \theta & , \text{otherwise.} \end{cases} \quad (1)$$

Then we compute the sampling weight $wt(w)$ for w as follows.

$$\mathrm{wt}(w) = (\mathrm{freq}^*(w))^{-\alpha} \quad (2)$$

In our experiments, we set the hyperparameters $\theta = 10$ and $\alpha = 0.5$ based on optimizing performance on the development set.

For each token t_i in a sentence $S = \{t_1, t_2, \ldots, t_n\}$, where n is the number of tokens, we compute the sampling probability $p(t_i)$ for masking t_i by normalizing $wt(t_i)$.

$$p(t_i) = \frac{wt(t_i)}{\sum_{j=1}^{n} wt(t_j)} \quad (3)$$

2.2.2. Dynamic Weighted Sampling

Frequency Weighted Sampling produces constant sampling probabilities for tokens and does not consider the learning status of the backbone masked language model that it applies for. We hypothesize that the signal of informativeness of a token w may also be derived from how poorly a masked language model predicts it. Therefore, we propose the Dynamic Weighted Sampling strategy shown in Figure 2. We use a weight dictionary in memory to store the sampling weights of each token after each batch in each iteration instead of updating the sampling weights after processing all batches in an iteration as the sampling only happens once in the latter case. Firstly, we set an initial sampling weight $wt(t_i) = 1$ for each token $t_i \in T$ in

Method	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
BERT	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
BERT-CP	41.00	60.02	51.11	68.43	64.59	56.32	62.07	57.65
WSBERT_Freq	42.60	61.32	52.04	69.84	66.61	59.89	61.94	59.18
WSBERT_Dynamic	47.80	67.28	57.13	71.41	68.87	65.28	64.90	63.24
BERT-Whitening	54.28	**78.07**	**65.44**	64.83	70.16	71.43	62.23	66.43
WSBERT-Whitening	55.14	<u>78.45</u>	<u>66.13</u>	65.47	**70.68**	**71.98**	61.91	67.10
BERT + Prompt†	**60.96**	73.83	62.18	**71.54**	68.68	70.60	**67.16**	**67.85**
WSBERT + Prompt	<u>63.03</u>	71.66	63.80	<u>75.32</u>	<u>76.67</u>	<u>74.79</u>	65.32	<u>70.08</u>

Table 1. Sentence representation performance on STS tasks. The reported score is Spearman's correlation coefficient between the predicted similarity and the gold standard similarity scores. The best results are both underlined and in bold. WSBERT without a subscript refers to WSBERT_Dynamic. Performance of BERT-Whitening is from the model learned on the full target dataset with embedding size 256 (train+development+test) [22]. † denotes the best manual-prompt results cited from [6].

the weight dictionary, where T denotes all tokens in the pre-training dataset. Then, we compute the sampling probabilities using sampling weights in the weight dictionary based on Eqn. 3 and train a masked language model. During each mini-batch, the masked tokens are predicted by the current model and we compute the total cross-entropy loss for token t_i as:

$$L_{t_i} = -logP(t_i \mid x, \theta) \qquad (4)$$

where x denotes the input masked sequence and θ denotes the parameters of the current masked language model. Then, we use L_{t_i} to compute the sampling weight $wt(t_i)$ based on Eqn. 5.

$$wt(t_i) = exp(\frac{L_{t_i}}{\tau}) \qquad (5)$$

where τ is a temperature parameter with default as 0.2. Finally, for the next mini-batch, we compute the sampling probability $p(t_i)$ for token t_i by normalizing $wt(t_i)$ following Eqn. 3.

The sampling weights computed by Eqn. 5 are larger for tokens with higher cross-entropy prediction loss, i.e., tokens that are poorly learned in the current masked language model and are often rare tokens; the sampling weights are smaller for tokens with lower cross-entropy loss, i.e., tokens that are relatively better learned. We design the sampling weight function $wt(t_i)$ as Eqn. 5 to enlarge the variance of sampling weights between different tokens and to further boost up sampling probabilities of rare tokens. During each iteration in pre-training, the weight dictionary is updated with the latest sampling weights $wt(t_i)$ for each token t_i. For the next iteration, for a sequence $s = t_1, t_2, ..., t_n, s \in S$, the sampling probability of choosing to mask each token is computed using the updated weight dictionary by Eqn. 3.

3. EXPERIMENTS

We conduct two experiments: evaluating unsupervised sentence representations using WSBERT on STS tasks, and evaluating the efficacy of Weighted Sampling on BERT's transfer learning capability by fine-tuning WSBERT on the GLUE benchmark. Ablation studies are also designed to analyze the impact of WSBERT on token embeddings for rare and common tokens.

3.1. Datasets and Implementation Details

For WSBERT, we continue pre-training on `bert-base-uncased` (BERT)[2], with Weighted Sampling on the WikiText dataset (+100M

Dataset	BERT	BERT-CP	WSBERT
MNLI	$84.30_{\pm 0.26}$	$84.26_{\pm 0.19}$	$\mathbf{84.42_{\pm 0.35}}$
QQP	$91.31_{\pm 0.04}$	$90.94_{\pm 0.59}$	$\mathbf{91.43_{\pm 0.05}}$
QNLI	$\mathbf{91.47_{\pm 0.01}}$	$91.32_{\pm 0.17}$	$91.14_{\pm 0.17}$
SST-2	$\mathbf{92.86_{\pm 0.13}}$	$92.78_{\pm 0.43}$	$91.35_{\pm 0.47}$
CoLa	$56.47_{\pm 0.65}$	$57.44_{\pm 0.95}$	$\mathbf{58.29_{\pm 0.33}}$
STS-B	$89.68_{\pm 0.26}$	$89.52_{\pm 0.37}$	$\mathbf{89.86_{\pm 0.18}}$
MRPC	$86.13_{\pm 1.63}$	$85.13_{\pm 0.53}$	$\mathbf{88.20_{\pm 2.39}}$
RTE	$69.23_{\pm 0.4}$	$67.25_{\pm 1.84}$	$\mathbf{70.89_{\pm 0.17}}$
AVG	$82.68_{\pm 0.33}$	$82.33_{\pm 0.32}$	$\mathbf{83.20_{\pm 0.10}}$

Table 2. GLUE Validation results from *BERT-base-uncased* (BERT-base), *BERT-base-uncased* continually pre-trained (BERT-CP), and Weighted-Sampled BERT (WSBERT). BERT-CP and WSBERT both continually train on BERT with the same training settings. WSBERT refers to WSBERT_Dynamic. The best results for each dataset and AVG are in bold.

tokens)[3]. We set the learning rate 5×10^{-5} and train 10 epochs. We use 4 NVIDIA V100 GPUs to train WSBERT with batch size 8 per device and gradient accumulation as 8. We use the WordPiece tokenizer as in [1] and token frequency is based on the tokenized Wikitext. STS tasks contain STS 2012-2016, STS benchmark, and SICK-Relatedness datasets. GLUE includes eight datasets [21]. To analyze the effect of continual pre-training without Weighted Sampling, we also continue pre-training on BERT with the same random sampling as for BERT on the same Wiki-Text data and with the same pre-training setting as WSBERT. We denote the resulting model **BERT-CP**. We compare fine-tuning performance of `bert-base-uncased` (BERT), BERT-CP, and WSBERT on GLUE. For each model on each GLUE task, we run three runs with different random seeds; for each run, we conduct a grid search on hyperparameters on GLUE validation set among $2 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}$ learning rate and $5, 10$ epochs. The other hyperparameters are the same for the three models: we use 1 V100 with batch size 32 per device, warm-up ratio 0.06, and weight-decay 0.01. We then report the mean and standard deviation of the best results from three runs in Table 2.

3.2. Main Results

Semantic Textual Similarity Table 1 shows the main STS results. All models in the table are of BERT base size. We report results of WSBERT with *Frequency Weighted Masking* and *Dynamic Weighted Masking*, denoted WSBERT_Freq and WSBERT_Dynamic. The first group in Table 1 shows that WSBERT_Dynamic outperforms BERT and BERT-CP by **6.54** and **5.59** absolute, significantly improving the quality of sentence embeddings of PLMs. WSBERT_Dynamic outperforms WSBERT_Freq by 4.06 absolute, showing that Dynamic Weighted Sampling is more effective than sampling only based on token frequency. BERT_Whitening [22], as a calibration method, is compatible with WSBERT. The second group in Table 1 shows that although WSBERT_Dynamic yields a lower average score on STS compared to BERT_Whitening, WSBERT_Dynamic could be effectively combined with Whitening and further improve the performance of WSBERT_Dynamic to **67.10**. We also investigate enhancing BERT and WSBERT with prompt. Different from previous works using a single [MASK] in prompts [6], we transform a sentence using manual prompt templates with multiple [MASK][4].

[4]The best prompt is designed as The sentence: $[X]$ means $[MASK]$ and also means $[MASK]$.

Method	Rare Tokens	Common Tokens
BERT	14.97	64.82
BERT-CP	14.86	64.95
WSBERT	**15.60**	**65.54**

Table 3. Portion of *common tokens* (high-frequency tokens) in the nearest neighbors of *rare tokens* (low-frequency tokens) and *common tokens*. We first sort tokens in the WikiText vocabulary by frequency in descending order. Then we select the tokens with ranks ranging from 10K-20K as rare tokens while choosing the Top-10K tokens as common tokens. We choose the 10 nearest neighbors decided by the Euclidean distance between representations of the target token and other tokens.

Furthermore, instead of extracting and averaging representations of the masked tokens as final sentence embeddings [6], we encode the whole transformed sentence and compute sentence embedding by average pooling all token embeddings of the sentence. The third group in Table 1 shows that prompt-enhanced WSBERT achieves **70.08**. These results demonstrate that Weighted Sampling improves sentence representations generated by PLMs and combining WSBERT with Whitening and prompts further improves sentence embeddings. We did not compare WSBERT to the SOTA models on STS, i.e., sentence-level contrastive learning (CL) based models such as SimCSE [28], since prior works [31] and our studies show sentence-level CL based models hurt transfer learning capability. We observe absolute 0.5 degradation on GLUE AVG score from SimCSE-BERT compared to BERT. However, as shown in the following GLUE experiments, WSBERT both enhances sentence representations of BERT and improves transfer learning capability.

GLUE Evaluation As shown in Table 2, WSBERT achieves the best average GLUE score compared to BERT and BERT-CP, outperforming BERT by **0**.52 absolute. WSBERT maintains competitive performance on MNLI and QQP and outperforms BERT on all other tasks. In contrast to models such as SimCSE, Dynamic Weighted Sampling improves the transfer learning capability while enhancing sentence representations. Compared to BERT, BERT-CP degrades GLUE AVG by 0.35 absolute while WSBERT outperforms BERT-CP by 0.87 absolute. These results prove that the gain of WSBERT over BERT is from continual pre-training with Dynamic Weighted Sampling instead of just continuing pre-training BERT using random sampling with more steps on the same WikiText dataset. BERT-CP degrades GLUE performance compared to BERT, probably because WikiText (373.28M data size) used for continual pre-training is much smaller and less diverse than the standard BERT pre-training dataset (Wikipedia and Bookscorpus, 16GB data size), which could hurt generalizability of PLMs.

3.3. Analysis

As observed in [5, 32], token embeddings of MLM-pretrained PLMs can be biased by token frequency, causing embeddings of high-frequency tokens to concentrate densely and low-frequency tokens to disperse sparsely. Inspired by these works, to analyze whether Weighted Sampling could indeed alleviate the frequency bias problem, we propose two approaches to analyze distributions of BERT, BERT-CP, and WSBERT in the representation space. We also discuss the training time for training MLM with and without weighted sampling.

Nearest Neighbors We investigate the portion of common tokens in the nearest neighbors (NN) of rare tokens, denoted P_{rare}, and the portion of common tokens in NN of common tokens, denoted

Rank of token frequency		0-100	100-500	500-5k	5k-10k
Mean ℓ2-norm	BERT	0.9655	1.0462	1.2150	1.3639
	BERT-CP	0.9597	1.0428	1.2141	1.3647
	WSBERT	**0.9562**	**1.0385**	**1.2112**	**1.3621**
Mean k-NN ℓ2-norm (k=3)	BERT	0.6972	0.7782	0.8188	0.8953
	BERT-CP	0.6913	0.7750	0.8180	0.8963
	WSBERT	**0.6883**	**0.7724**	**0.8154**	**0.8929**
Mean k-NN ℓ2-norm (k=5)	BERT	0.8007	0.8868	0.9327	1.0083
	BERT-CP	0.7936	0.8833	0.9319	1.0096
	WSBERT	**0.7899**	**0.8800**	**0.9287**	**1.0056**
Mean k-NN ℓ2-norm (k=7)	BERT	0.8590	0.9458	0.9932	1.0671
	BERT-CP	0.8513	0.9422	0.9924	1.0685
	WSBERT	**0.8471**	**0.9386**	**0.9888**	**1.0642**

Table 4. The mean ℓ2-norm calculated for each bin of tokens with ranking ranges based on token frequency in WikiText. Common tokens occupy a higher ranking while rare tokens are in low rankings. A lower mean ℓ2-norm suggests that the token embeddings in that bin are more concentrated.

P_{common} (rare and common tokens are defined in Table 3 caption). The larger portion of common tokens in NN of rare/common tokens indicates more concentrated token distributions and hence smaller frequency bias in the token embeddings. In Table 3, P_{rare}/P_{common} for WSBERT increase by 0.63/0.72 over those of BERT and 0.74/0.59 over those of BERT-CP, suggesting that WSBERT has more concentrated token distributions and smaller frequency bias in token embeddings compared to BERT and BERT-CP, and common tokens are also more concentrated in WSBERT than BERT and BERT-CP.

Token Distribution Inspired by [5], we compute the mean ℓ2-norm between token embeddings and the origin for the three models to analyze token distributions. As shown in the first row of Table 4, although common tokens are close to the origin and rare tokens are distributed far away from the origin, the smaller mean ℓ2-norm indicates the token embeddings of WSBERT are more concentrated than BERT and BERT-CP. The ℓ2-norms of WSBERT are smaller on all the bins than those of BERT, suggesting that both common tokens and rare tokens are closer to the origin in WSBERT than BERT. Furthermore, WSBERT tokens in each bin are more compact than BERT and BERT-CP as shown by the smaller mean k-NN ℓ2-norm for each K in Table 4, which indicates that the embedding space of WSBERT is less sparse than BERT and BERT-CP. Sparsity in the embedding space may cause poorly defined semantic meanings [5], hence the gains in sentence embeddings and transfer learning capability from WSBERT over BERT may also be attributed to sparsity reduction by WSBERT in the embedding space.

Training Time Calculating the sampling weight for each token during training takes extra time. Training a MLM without weighted sampling takes 11 hours while training with weighted sampling takes 20 hours. The extra time can be reduced through optimizations such as implementing parallel writing to the dictionary and using an in-memory vector to accelerate the reading and writing processes.

4. CONCLUSION

We propose two Weighted Sampling methods to alleviate the frequency bias issue in Masked Language Modeling for pre-training language models. Extensive experiments show Weighted Sampling improves both sentence representations and the transferability of pre-trained models. We also analyze the token embeddings to explain how Weighted Sampling works. Future work includes investigating other dynamic sampling methods and exploring training objectives with a penalty for frequency bias.

5. REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*. 2019, pp. 4171–4186, Association for Computational Linguistics.

[2] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[3] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu, "Representation degeneration problem in training natural language generation models," in *ICLR*. 2019, OpenReview.net.

[4] Kawin Ethayarajh, "How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings," in *EMNLP-IJCNLP*. 2019, pp. 55–65, Association for Computational Linguistics.

[5] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li, "On the sentence embeddings from pre-trained language models," in *EMNLP*. 2020, pp. 9119–9130, Association for Computational Linguistics.

[6] Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang, "Promptbert: Improving BERT sentence embeddings with prompts," in *EMNLP*. 2022, pp. 8826–8837, Association for Computational Linguistics.

[7] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro, "Megatron-lm: Training multibillion parameter language models using model parallelism," *CoRR*, vol. abs/1909.08053, 2019.

[8] Linyuan Gong, Di He, Zhuohan Li, Tao Qin, Liwei Wang, and Tie-Yan Liu, "Efficient training of BERT by progressively stacking," in *ICML*, Kamalika Chaudhuri and Ruslan Salakhutdinov, Eds. 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 2337–2346, PMLR.

[9] Qiyu Wu, Chen Xing, Yatao Li, Guolin Ke, Di He, and Tie-Yan Liu, "Taking notes on the fly helps language pre-training," in *ICLR*. 2021, OpenReview.net.

[10] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *ICLR*. 2020, OpenReview.net.

[11] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning, "ELECTRA: pre-training text encoders as discriminators rather than generators," in *ICLR*. 2020, OpenReview.net.

[12] Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song, "COCO-LM: correcting and contrasting text sequences for language model pretraining," in *NeurIPS*, 2021, pp. 23102–23114.

[13] Le Hou, Richard Yuanzhe Pang, Tianyi Zhou, Yuexin Wu, Xinying Song, Xiaodan Song, and Denny Zhou, "Token dropping for efficient BERT pretraining," in *ACL*. 2022, pp. 3774–3784, Association for Computational Linguistics.

[14] Eneko Agirre, Daniel M. Cer, Mona T. Diab, and Aitor Gonzalez-Agirre, "Semeval-2012 task 6: A pilot on semantic textual similarity," in *NAACL-HLT*. 2012, pp. 385–393, The Association for Computer Linguistics.

[15] Eneko Agirre, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo, "*sem 2013 shared task: Semantic textual similarity," in *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM*. 2013, pp. 32–43, Association for Computational Linguistics.

[16] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe, "Semeval-2014 task 10: Multilingual semantic textual similarity," in *COLING*. 2014, pp. 81–91, The Association for Computer Linguistics.

[17] 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, "Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability," in *NAACL-HLT*. 2015, pp. 252–263, The Association for Computer Linguistics.

[18] Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe, "Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in *NAACL-HLT*. 2016, pp. 497–511, The Association for Computer Linguistics.

[19] Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia, "Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *ACL*. 2017, pp. 1–14, Association for Computational Linguistics.

[20] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli, "A SICK cure for the evaluation of compositional distributional semantic models," in *LREC*. 2014, pp. 216–223, European Language Resources Association (ELRA).

[21] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *ICLR*. 2019, OpenReview.net.

[22] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou, "Whitening sentence representations for better semantics and faster retrieval," *CoRR*, vol. abs/2103.15316, 2021.

[23] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al., "Improving language understanding by generative pre-training," 2018.

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, pp. 9, 2019.

[25] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei, "Language models are few-shot learners," in *NeurIPS*, 2020.

[26] Timo Schick and Hinrich Schütze, "Exploiting cloze-questions for few-shot text classification and natural language inference," in *EACL*. 2021, pp. 255–269, Association for Computational Linguistics.

[27] Tianyu Gao, Adam Fisch, and Danqi Chen, "Making pre-trained language models better few-shot learners," in *ACL/IJCNLP*. 2021, pp. 3816–3830, Association for Computational Linguistics.

[28] Tianyu Gao, Xingcheng Yao, and Danqi Chen, "Simcse: Simple contrastive learning of sentence embeddings," in *EMNLP*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, Eds. 2021, pp. 6894–6910, Association for Computational Linguistics.

[29] Yuxin Jiang, Linhan Zhang, and Wei Wang, "Improved universal sentence embeddings with prompt-based contrastive learning and energy-based learning," in *Findings of EMNLP*. 2022, pp. 3021–3035, Association for Computational Linguistics.

[30] Xing Wu, Chaochen Gao, Zijia Lin, Jizhong Han, Zhongyuan Wang, and Songlin Hu, "Infocse: Information-aggregated contrastive learning of sentence embeddings," in *Findings of EMNLP*. 2022, pp. 3060–3070, Association for Computational Linguistics.

[31] Yixuan Su, Fangyu Liu, Zaiqiao Meng, Tian Lan, Lei Shu, Ehsan Shareghi, and Nigel Collier, "Tacl: Improving BERT pre-training with token-aware contrastive learning," in *Findings of NAACL*. 2022, pp. 2497–2507, Association for Computational Linguistics.

[32] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu, "Consert: A contrastive framework for self-supervised sentence representation transfer," in *ACL/IJCNLP*. 2021, pp. 5065–5075, Association for Computational Linguistics.