Implementing and Extending minBERT for Sentence-Level Tasks

Katherine Chen Douglas Li William Zhang

Department of Computer Science, Stanford University

We combine ***contrastive learning*** and ***regularization methods*** to improve the performance of minBERT on **sentiment classification**, **paraphrase detection**, and **semantic textual similarity**.

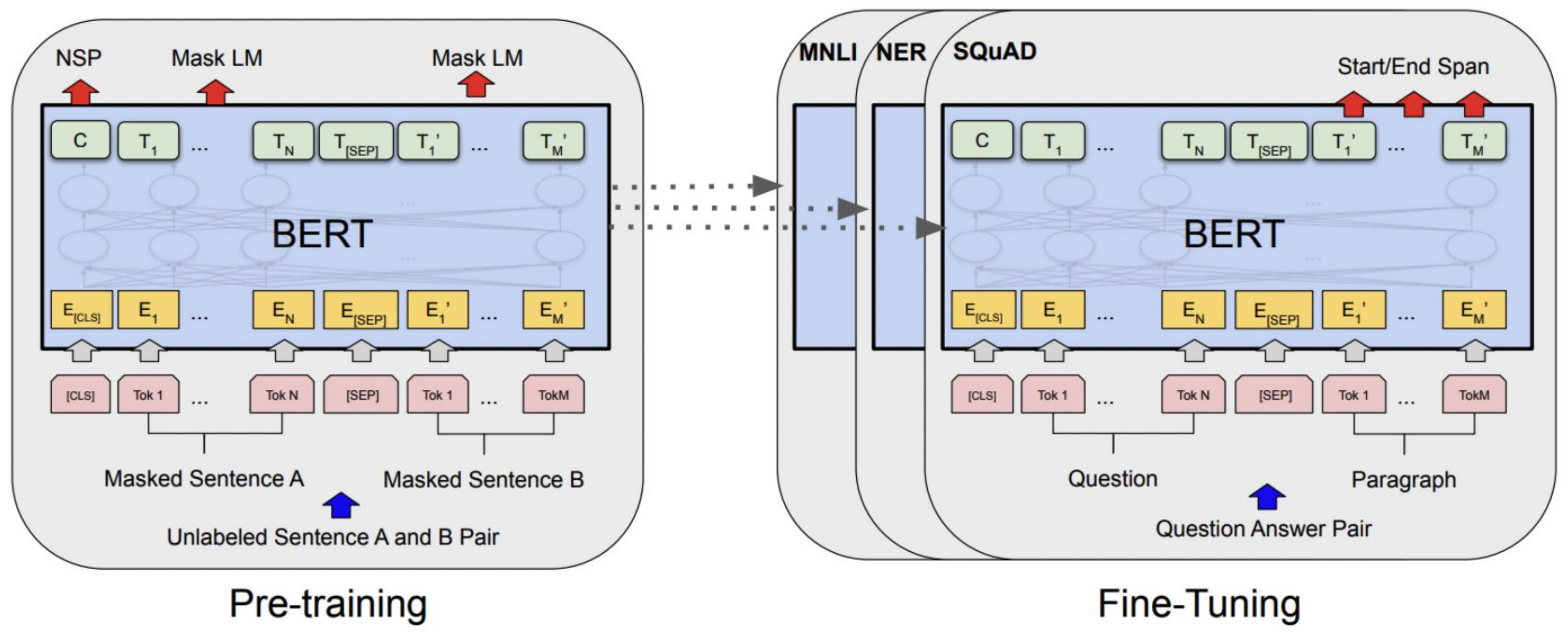

Figure 1. Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both. The pre-trained model parameters are used to initialize models for different downstream tasks.

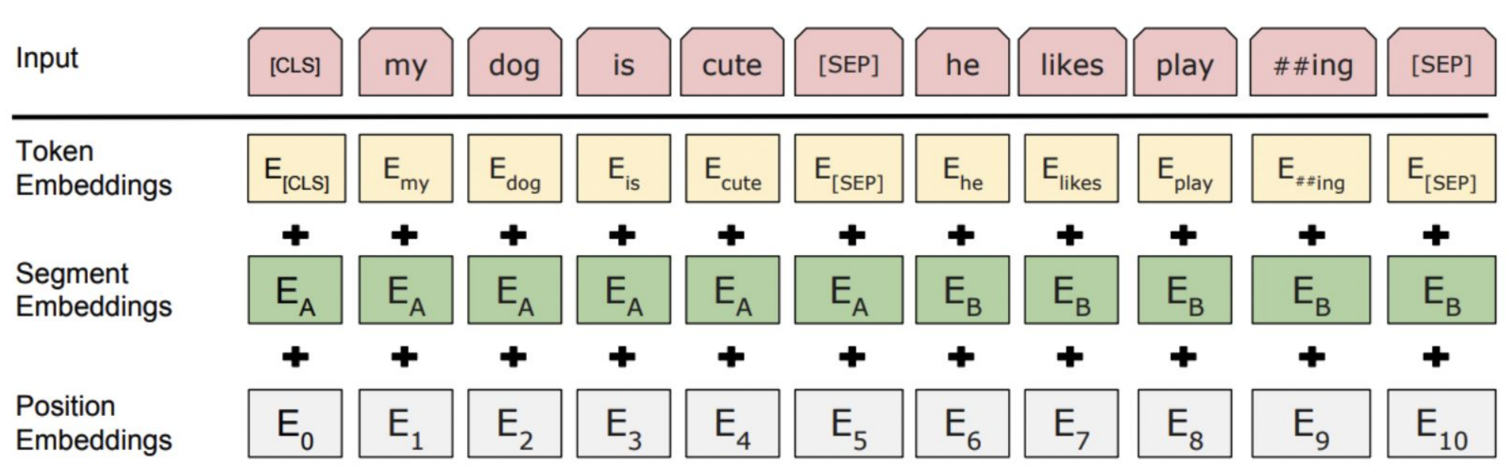

Figure 2. The input embeddings utilized in BERT are the sum of the token embeddings, the segmentation embeddings, and the position embeddings.

Problem

- Fine-tuning pre-trained BERT for downstream tasks is challenging
- Limited task-specific labeled data, too many model parameters
- Overfitting and poor generalization

Background

Bidirectional Encoder Representations from Transformers [Devlin et al., 2019]

- 12 encoder transformer layers
- Pre-trained on 2 **unsupervised** tasks using Wikipedia articles
  - Masked token prediction
  - Next sentence prediction
- Learns **contextual** info of sentences

Methods

Pre-training with vanilla minBERT

- Multi-head self-attention
- Transformer layer
- Adam optimizer: compute adaptive learning rates for params by estimating the first & second moments of the gradient

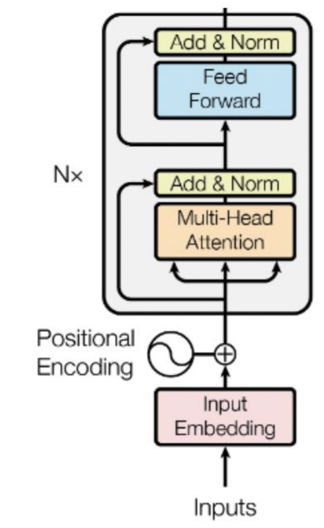

Figure 3. BERT transformer layer.

Contrastive learning (Unsupervised SimCSE) [Gao et al., 2021]

- Embeddings map sentence pairs $(x_i, x^+_i) \rightarrow$ vector representations (h_i, h^+_i)
- Initialized with minBERT → fine-tuned to maximize cosine similarity *sim*(h_i, h^+_i)
- Goal: distinguish positive (similar) and negative (dissimilar) sentence pairs
- Learns **semantic** relationships between sentences

$$\ell_i = -\log \frac{e^{\mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_j^+)/\tau}}$$

Fine-tuning with regularized optimization [Jiang et al., 2019]

- $L'(\theta) = L(\theta) + \lambda_s R_s(\theta)$, where $L(\theta)$ is cross-entropy/MSE loss, $R_s(\theta)$ is the smoothness inducing adversarial regularizer, and λ_s is a hyperparameter (tuned with dropout prob.)

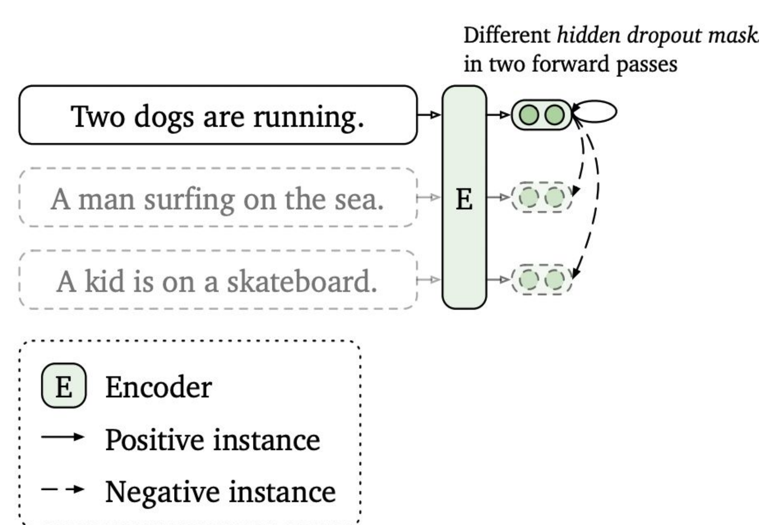

Figure 4. Unsupervised SimCSE predicts the input sentence itself from in-batch negatives, with different hidden dropout masks applied.

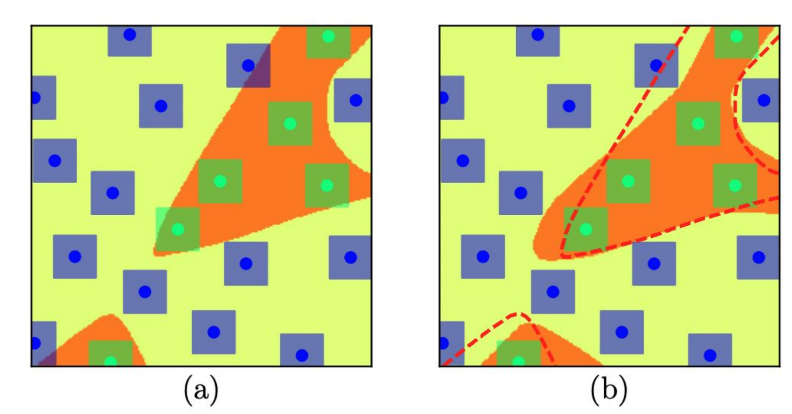

Figure 5. Decision boundaries learned without (a) and with (b) regularized optimization. The red dotted line in (b) represents the boundary in (a).

Multi-task fine-tuning

- Experimented as a step before single-step fine-tuning
- Optimize the params for each of the 3 tasks on each epoch
- Loss function:

$$L_{total} = L_{SST} + L_{Quora} + L_{STS}$$

Hyperparameter tuning

- Learning rate
- Dropout probability

Single-task fine-tuning

- **Sentiment classification**
  - "Positive, negative, or neutral?"
  - Stanford Sentiment Treebank (SST)
  - Linear layer
- **Paraphrase detection**
  - "Do 2 sentences reword each other?"
  - Quora Dataset
  - Linear/LSTM + cosine similarity
- **Semantic textual similarity (STS)**
  - "Degree of semantic equivalence?"
  - SemEval STS Benchmark
  - Linear/LSTM + cosine similarity

Experiments

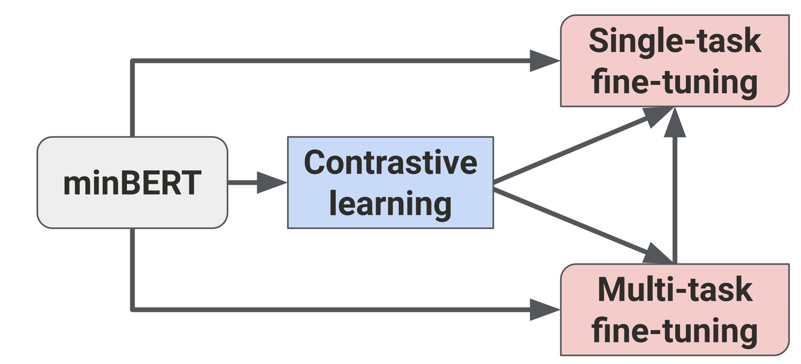

Figure 6. Experiment flow diagram.

Model configuration	SST acc	Paraphrase acc	STS corr
Baseline	0.424	0.416	0.202
LSTM	0.504	0.526	0.183
Linear	**0.532**	**0.788**	**0.769**
CL+LSTM	0.509	0.375	0.557
CL+Linear	0.514	0.787	0.759
Linear+RegOpt	0.522	0.739	0.727
CL+Linear+RegOpt	0.517	0.724	0.755

Table 1. Single-task fine-tuning results. The Linear configuration without CL/RegOpt performed the best on the dev set, with substantial gains above the baseline for Paraphrase/STS.

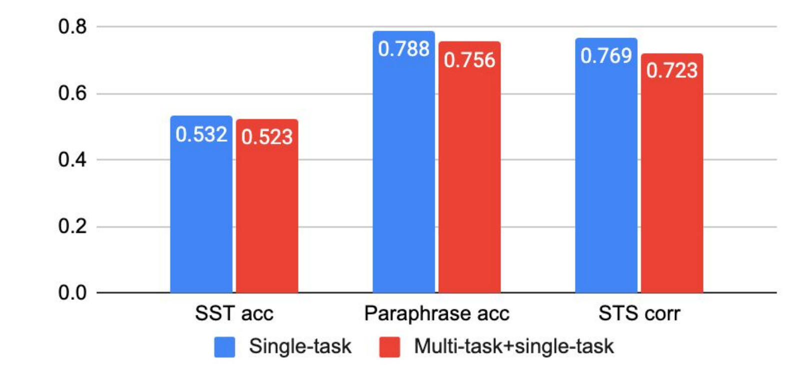

Table 2. Multi-task fine-tuning results for the Linear configuration. Did not improve single-task fine-tuning.

Learning rate	SST acc	Paraphrase acc	STS corr
1E-05	0.432	0.624	0.634
2E-05	**0.532**	**0.788**	**0.769**
5E-05	0.524	0.735	0.743
1E-04	0.508	0.733	0.728
2E-04	0.483	0.719	0.623
5E-04	0.439	0.716	0.612

Table 3. Learning rate tuning results. Dropout probability was fixed as 0.3.

Dropout probability	SST acc	Paraphrase acc	STS corr
0.1	0.529	0.773	0.721
0.2	0.522	0.781	0.715
0.3	**0.532**	**0.788**	**0.769**
0.4	0.516	0.765	0.684
0.5	0.489	0.643	0.543

Table 4. Dropout prob. tuning results. Learning rate was fixed at $2 \cdot 10^{-5}$.

Analysis

- Since we only used **unsupervised** CL, for Paraphrase/STS, only the embeddings for the first sentence in each pair were improved → little effect on predicting pair similarity
- RegOpt did not improve scores due to **reduced expressiveness** of our model when biased with the weight decay parameter
- The linear model may need **more parameters**, since CL/RegOpt/Multi-task might work better when the model is more expressive

Conclusion

- Our approach improves the performance of minBERT
  - Moderately for **SST dataset**
  - Significantly for **Paraphrase**/**STS**
- Best model scores obtained using a **linear** architecture **without** contrastive learning, regularized optimization or multi-task fine-tuning

References

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1*, pp. 4171-4186.

[2] Gao, T., Yao, X., and Chen, D. 2021. SimCSE: Simple contrastive learning of sentence embeddings. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910.

[3] Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Zhao, T. 2019. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*.