UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

SEMINAR

Improving BERT Pre-training with Token-aware Contrastive Learning

Antonio Lukić
Mentor: *prof. dr. sc. Jan Šnajder, mag. ing. David Dukić*

Zagreb, November 2024.

CONTENTS

1. Reproducing the paper

1.1. Task and Model Description

Transformer models such as BERT come pre-trained, which is essential for their high performance across various natural language processing tasks. By leveraging unsupervised learning on large corpora, these models learn deep contextual representations before being fine-tuned on specific downstream tasks. However, pre-trained Masked Language Models like BERT (Devlin et al., 2018) output an anisotropic distribution of token representations that live in a narrow subset of the entire representation space. Anisotropic token representation is not ideal for discriminative tasks.

The paper by (Su et al., 2021) introduces a continual novel and effective pre-training approach called TaCL (Token-aware Contrastive Learning).

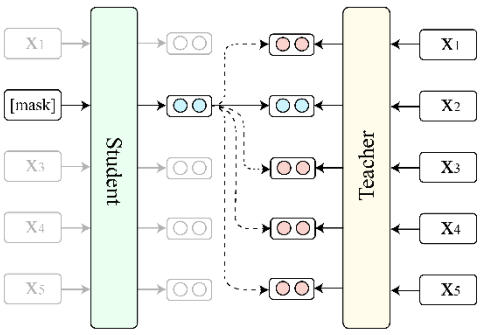


Figure 1.1: TaCL objective

Continual pre-training is done by using two BERT models. The first one is called the student and the second one is called the teacher. They start as the same pre-trained BERT model. The concept behind this approach is to freeze the teacher and optimize the student model. The student model is optimized with the original BERT MLM, next sentence prediction (NSP) objective (Devlin et al., 2018) and a new TaCL objective. TaCL objective, shown in Figure 1.1, is to contrast the masked token from the student to the same unmasked token from the teacher. By doing this, it gets closer to the teacher's unmasked token and gets further from the student's reference tokens in the

same sequence. If the ith token in the input sequence is represented by x_i, $\tilde{x_i}$ as the masked input token, h_i as the ith output token of the teacher and $\tilde{h}_i$ as the ith token output of the student, the TaCL objective can be described as:

$$\mathcal{L}_{TaCL} = -\sum_{i=1}^{n} \mathbb{1}(\tilde{x}_i) log \frac{exp(sim(\tilde{h}_i, h_i)/\tau)}{\sum_{j=1}^{n} exp(sim(\tilde{h}_i, h_j)/\tau)} \tag{1.1}$$

where $\mathbb{1}$ is the indicator function being 1 when $\tilde{x}_i$ is a masked token, the *sim* is a cosine similarity, and τ a hyperparameter. Another measurement that could be used is *self-similarity*, which is the sum of all cosine similarities of each token to all other tokens in the model output. The lower the self-similarity the more discriminative the tokens are.

1.2. Reproducing the results

1.2.1. Experimental setup

The code and pre-trained models were referenced in the paper and are publicly available.[1] The repository contains code both for English and Chinese benchmarks of which I have only focused on English benchmarks. The first difference between the provided and reimplemented code is the environment. The authors utilized torch version $1.6.0$, torchvision version $0.7.0$ and transformers version $4.7.0$. I found that these versions were incompatible so I have used torch version $2.0.0$, torchvision version $0.15.1$ and transformers version $4.39.3$. The initial step was to download the raw Wikipedia text as pre-training data through `download_rawdata.py` module. The provided Wikipedia dataset `20200501.en` was not available so I have used the closest available one `20220301.en`. [2] Furthermore, the data was processed through the `tokenize_data.py` module.

1.2.2. Pre-training and fine-tuning BERT

Given the computationally intensive nature of the task, which demands substantial processing power, I have used Srce's supercomputer Supek (University Computing Centre (Srce), 2023). The pre-training was done on 2 NVIDIA A100 GPUs with 40 GB of memory and 4 AMD Epyc 7763 CPUs operating at 2.45 GHz. Access to Supek's resources were limited to only 24 hours per job. Consequently, the pre-training of

[1] https://github.com/yxuansu/TaCL

[2] https://huggingface.co/datasets/wikipedia

the *bert-base-uncased* model using the `train.py` module was done for only 130000 iterations instead of the original 150000. For fine-tuning, only 1 GPU was used. However, the fine-tuning hyperparameters were not specified. It was done using Hugging Face's `run_qa.py` module for SQuAD (Rajpurkar et al., 2016) and `run_glue.py` module for GLUE (Wang et al., 2019) benchmarks.

1.3. Result comparison

The base TaCL model outperformed BERT base model on all benchmarks except SST-2, MNLI and RTE. The results of the original TaCL base model compared to my reimplemented model can be seen in Table 1.1 and Table 1.2. The model's performance was evaluated on the test set through the GLUE benchmark submission.[3] In these tables, the metric legends are represented as follows: *A* for accuracy, *MC* for Matthews correlation, *SC* for Spearman correlation, *MMA* for matched/mismatched accuracy and *EM* for exact match percentage.

Table 1.1: Comparison of Original and My Results for GLUE

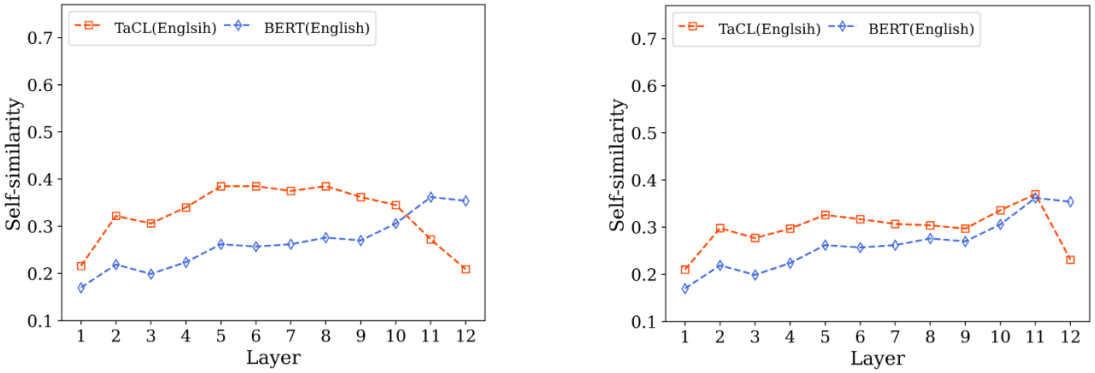
	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE
metric	MC	A	A	SC	A	MMA	A	A
Original	52.4	92.3	90.8	89.0	80.7	84.4/84.3	91.1	62.8
My	44.6	91.7	84.1	86.4	88.2	82.9/81.7	89.1	60.9

Table 1.2: Comparison of Original and My Results for SQuAD

	SQuAD 1.1		SQuAD 2.0	
metric	EM	F1	EM	F1
Original	81.6	89.0	74.4	77.5
My	77.6	85.9	64.4	67.5

From the data we can see that reimplemented results are, on average, a few percentages lower than the original results. The most significant declines are in the CoLA and SQuAD 2.0 datasets. Although the reduction in pre-training iterations by 20000 could be the cause of the decline, I think that the main reason behind this is the absence of specified fine-tuning hyperparameters. However, it can also be seen that there is a

[3] https://gluebenchmark.com/submit



(a) Original **(b)** My reimplementation

Figure 1.2: Layer-wise representation of self-similarity

7.5% increase in the QQP dataset. This could be attributed to the model's enhanced ability to capture semantic nuances between question pairs, potentially benefiting from the contrastive learning approach which emphasizes token distinctiveness crucial for tasks like QQP.

As mentioned before, another possible approach for analysis of learned token representations is self-similarity. Figure 1.2 shows the self-similarity of 50000 sampled sentences computed over different layers. It can be observed that over almost all layers, BERT's tokens have lower self-similarity then TaCl's. However, at the last layer, it can be seen that the output tokens of TaCL are less self-similar making them more discriminative. My results follow the same trend with a more sudden drop at the end probably due to fewer iterations in pre-training.

1.4. Conclusion

From my perspective, the paper provides a good view of the limitations of token representations from BERT. It presents an innovative yet simple idea that requires no additional data that could improve the model. However, I encountered challenges in replicating the results. The provided code was overall clear and well-structured. Furthermore, conducting experiments with models like BERT and large datasets required additional resources like the supercomputer Supek to conduct these experiments. While the authors gave a clear explanation of methods, objectives, and what their goal was, they did not provide the exact hyperparameters for fine-tuning. In my opinion, the lack of hyperparameters and environment differences is the main reason for results deviation. Additionally, there is no indication that the authors have conducted hyperparameter

tuning. Potential improvement could be expanding the contrastive learning framework to include not only positive examples (teacher's unmasked tokens) but also carefully selected negative examples from contexts that are not only found in the input sequence. This enhancement could further refine the model's ability to distinguish between relevant and irrelevant token features. Lastly, the incorporation of an attention-based mechanism that weighs the importance of different tokens in the contrastive learning objective could provide more useful token embeddings.

2. Bibliography

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

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. *arXiv preprint arXiv:2111.04198*, 2021.

University Computing Centre (Srce). Advanced computing. `https://www.srce.unizg.hr/napredno-racunanje`, 2023. Accessed: 03.05.2024.

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. U *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rJ4km2R5t7`.