

Bi-Granularity Contrastive Learning for Post-Training in Few-Shot Scene

Ruikun Luo, Guanhan Huang, Xiaojun Quan*

School of Computer Science and Engineering, Sun Yat-sen University, China
{luork, huanggh25}@mail2.sysu.edu.cn, quanxj3@mail.sysu.edu.cn

Abstract

The major paradigm of applying a pre-trained language model to downstream tasks is to fine-tune it on labeled task data, which often suffers instability and low performance when the labeled examples are scarce. One way to alleviate this problem is to apply post-training on unlabeled task data before fine-tuning, adapting the pre-trained model to target domains by contrastive learning that considers either token-level or sequence-level similarity. Inspired by the success of sequence masking, we argue that both token-level and sequence-level similarities can be captured with a pair of masked sequences. Therefore, we propose complementary random masking (CRM) to generate a pair of masked sequences from an input sequence for sequence-level contrastive learning and then develop contrastive masked language modeling (CMLM) for post-training to integrate both token-level and sequence-level contrastive learnings. Empirical results show that CMLM surpasses several recent post-training methods in few-shot settings without the need for data augmentation.

1 Introduction

The past few years have seen the rapid proliferation of large-scale pre-trained language models such as GPT (Radford et al., 2018), BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b). These models are generally characterized by pre-training on huge general-domain corpora and then fine-tuning on task-specific labeled examples when applied to a downstream task. Despite tremendous knowledge obtained from general-domain corpora through pre-training, sufficient labeled examples from the task domain are still needed. However, collecting them is often infeasible in many scenes, where it tends to be unstable and low-performing

*Corresponding author.

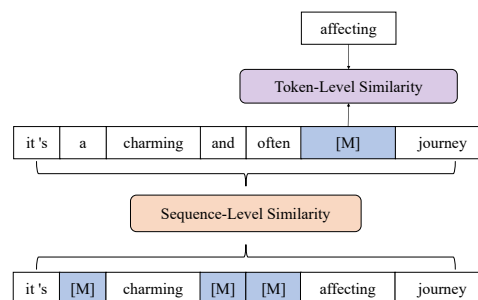


Figure 1: Demonstration of token-level similarity and sequence-level similarity, where the token “affecting” is close to the token “[MASK]” due to the latter’s context. The two sequences with different maskings are also close due to their semantic affinity.

when directly fine-tuning these pre-trained models (Zhang et al., 2021; Dodge et al., 2020).

Many efforts have been devoted to addressing the above issue. Firstly, it can be relieved by improving the fine-tuning process, such as introducing regularization (Jiang et al., 2020; Lee et al., 2020), re-initializing top layers (Zhang et al., 2021), and using debiased Adam optimizer (Mosbach et al., 2021). Besides, according to empirical results (Zhang et al., 2021; Mosbach et al., 2021), fine-tuning with a small learning rate and more fine-tuning epochs can also improve the situation. Secondly, additional data can be explored, for which two main genres of data might be helpful: labeled examples from related tasks and unlabeled task examples. The former has shown considerable success on the GLUE tasks (Phang et al., 2018; Liu et al., 2019a), whereas such labeled examples are not always easy to collect. By contrast, the latter is more feasible, especially in scenes where task examples are easy to collect but expensive to label.

Contrastive learning (Hadsell et al., 2006) is a recently re-emerged method for leveraging unlabeled data to enhance representation learning. The key to contrastive learning is to capture the similarity be-

tween samples. As shown in Figure 1, there are two sorts of similarities that can be captured for a natural language sequence: token-level similarity and sequence-level similarity. Masked language model (MLM) (Devlin et al., 2019), widely adopted in pre-trained language models, can be considered as token-level contrastive learning, as it maximizes the similarity between the “[MASK]” token and the original token before masking and minimizes the similarity with other tokens. As for sequence-level similarity, several works (Iter et al., 2020; Giorgi et al., 2020; Wu et al., 2020) introduce sequence-level contrastive learning into pre-training. While all these works focus on the pre-training phase, the focal point of this paper is to improve the performance of pre-trained models in downstream tasks through *post-training* especially for scenes where limited labeled data is available.

Speaking of pre-trained language models, Xu et al. (2019) and Gururangan et al. (2020) demonstrate improvement in various downstream tasks by training the models with MLM on task examples before fine-tuning, which, following Xu et al. (2019), is termed *post-training* in this paper even though Gururangan et al. (2020) term it *adaptive pre-training*. Fang and Xie (2020) also post-train their model on task examples by contrastive self-supervised learning (Chen et al., 2020), which pulls together two augmented sentences generated from the same sentence by back-translation (Edunov et al., 2018) while separating those otherwise. However, these works consider either token-level or sequence-level contrastive learning, without integrating them. Moreover, adopting back-translation to generate augmented sentences demands considerable computation and makes the effect of post-training dependent on the translation systems.

To capture both sequence-level and token-level similarities, we propose contrastive masked language modeling (CMLM) to achieve more effective knowledge transfer in post-training and to improve the performance of pre-trained language models in few-shot downstream tasks. For this purpose, we propose complementary random masking (CRM) to generate a pair of masked sequences from a single sequence for both sequence-level and token-level contrastive learnings. We conduct extensive experiments to compare CMLM with several recent post-training approaches, and the empirical results show that CMLM achieves superior or competitive performance in a wide range of downstream tasks.

Our contributions can be concluded as follows. First, we propose a new random masking strategy, CRM, to generate a pair of masked sequences favorable to sentence-level contrastive learning. Second, we propose a novel objective, CMLM, for post-training pre-trained language models, which realizes both sequence-level and token-level contrastive learnings on a pair of masked sequences. Third, we compare our approach with several post-training methods and obtain superior or competitive results in few-shot settings. Lastly, we compare two contrastive learning implementations, SimCLR (Chen et al., 2020) and SimSiam (Chen and He, 2020), in pre-trained language models. To our best knowledge, our work is the first effort to implement SimSiam in NLP and compare it with SimCLR.

2 Related Works

2.1 Pre-trained Language Model

Pre-trained language models such as GPT (Radford et al., 2018), BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) have become a new paradigm of NLP research, and been successfully applied in a wide range of tasks that used to be thorny. These models are generally structured with stacks of Transformer (Vaswani et al., 2017) and pre-trained on large-scale unlabeled corpora. Among them, GPT is pre-trained with a unidirectional language modeling objective, and BERT is with masked language modeling (MLM) and next sentence prediction (NSP). In RoBERTa, Liu et al. (2019b) turn the static MLM in BERT into a dynamic one and remove the NSP task, and pre-train it more intensively with larger corpora.

Despite tremendous knowledge learned from unlabeled corpora during pre-training, sufficient labeled task examples are still needed for fine-tuning, which can be challenging for some scenes. Plus, unlabeled task examples are not leveraged when sticking to the pre-training and fine-tuning paradigm.

2.2 Contrastive Learning

To take advantage of unlabeled or labeled data more effectively, contrastive learning (Hadsell et al., 2006) is re-emerged recently in computer vision (CV) and natural language processing (NLP). The key to contrastive learning is to pull positive samples together while separating negative samples apart. The construction of positive and negative sample pairs varies from tasks to tasks. In CV, Chen et al. (2020) take the augmented (e.g., by

random crop, color distortion, and Gaussian blur) images originated from the same image as positive pairs, and treat those otherwise as negative pairs. Khosla et al. (2020) take the images with the same label as positive pairs and take others as negative. In NLP, MLM in BERT can be viewed as contrastive learning on token level, which takes “[MASK]” and its original token before masking as a positive pair and the rest as negative. For sequence level, both Fang and Xie (2020) and Wu et al. (2020) follow Chen et al. (2020) to construct the sample pairs. Specifically, Fang and Xie (2020) utilize back-translation for sequence augmentation while Wu et al. (2020) use some easy deformation operations like deletion, reordering and synonym substitution. Besides, Giorgi et al. (2020) construct two spans as a positive pair if they overlap, subsume, or are adjacent. Gunel et al. (2021) introduce supervised contrastive learning proposed by Khosla et al. (2020) into NLP and treat the sequences with the same label as a positive pair. Li et al. (2021) augment the cross-entropy loss with a contrastive self-supervised learning term and a mutual information maximization term to deal with the cross-domain sentiment classification task.

2.3 Post-training

Post-training has been broadly applied in downstream tasks. For examples, Xu et al. (2019) post-train BERT with MLM on task examples to improve the sentiment analysis task. Gururangan et al. (2020) further divide post-training into two categories: domain-adaptive pre-training and task-adaptive pre-training, and evaluate them by extensive experiments. Phang et al. (2018) fine-tune their model on a related task before fine-tuning on the target task. Liu et al. (2019a) extend previous work into a multi-task learning fashion. Fang and Xie (2020) introduce contrastive self-supervised learning (CSSL) (Chen et al., 2020) to perform post-training and name it CSSL Pre-training.

3 Approach

3.1 Dynamic Random Masking

Masked language modeling (MLM) is firstly applied in BERT (Devlin et al., 2019), where some of the tokens in the input sequence are selected and replaced by a special token “[MASK]”. BERT uniformly selects 15% of the input tokens for replacement, and among the selected tokens, 80% are replaced with “[MASK]”, 10% are left un-

changed, and 10% are replaced by a randomly selected vocabulary token. In the original implementation of BERT, random masking and replacement are performed once in the beginning, and the sequences are kept unchanged through pre-training. Liu et al. (2019b) transform this static masking strategy into dynamic random masking (DRM) by generating a masking pattern every time a sequence is fed. That is to say, given an input sequence $T = \{t_1, t_2, \dots, t_N\}$, the probability of each token being selected is determined by p_m , which is fixed to 15% in BERT and RoBERTa.

$$P_{DRM}(t_n) = p_m, n \in [1, N] \quad (1)$$

3.2 Complementary Random Masking

It is straightforward to come up with an idea that generates a pair of masked sequences from a single sequence by random masking and applies MLM on each masked sequence to capture token-level similarity, and then perform sequence-level contrastive learning between the two sequences to capture sequence-level similarity. However, it faces a dilemma when applying this idea: setting a small p_m would make the pair of masked sequences too similar and make the contrastive learning loss drop to 0 quickly, harming sequence-level contrastive learning. On the other hand, setting a large p_m would make each masked sequence collapsed, making it hard for the model to recover the original tokens from “[MASK]” based on the context, which in turn harms token-level contrastive learning.

To address this issue, we decouple the pair of masked sequences, denoted by T^0 and T^1 , and assign them different masking probabilities p_m and p_c . Specifically, we obtain T^0 with a small masking probability p_m and T^1 with a larger probability p_c . Moreover, to avoid a single word being masked by both sequences, which cripples their relevance, we propose complementary random masking (CRM) to generate a pair of complementary masked sequences which maintain a complementary relationship. Concretely, in CRM, we first generate $T^0 = \{t_1^0, t_2^0, \dots, t_N^0\}$ with DRM (Liu et al., 2019b) from the original sequence T , and generate T^1 with an extra constraint: the masking probability $P_{CRM}(t_n)$ will be set to p_c if and only if t_n^0 has not been selected in T^0 . Otherwise, it will be set to 0. The process of CRM is described in Figure 2.

$$P_{CRM}(t_n) = \begin{cases} p_c, & t_n^0 \text{ was not selected in } T^0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

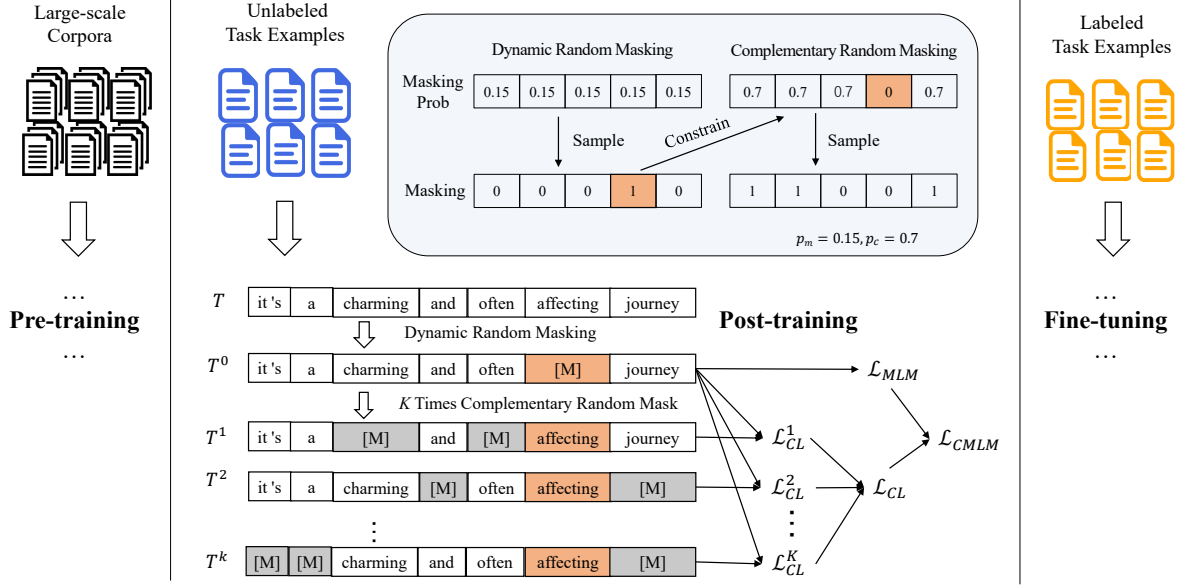


Figure 2: Demonstration of complementary random masking (CRM, in the blue box) and contrastive masked language modeling (CMLM).

CRM is aimed to generate a complementary pair of masked sequences: If $p_c = 1$, all tokens that are not selected in T^0 will be selected in T^1 . Reducing p_c can soften this complementary relationship and make the two sequences overlap increasingly.

3.3 Contrastive Masked Language Modeling

We propose contrastive masked language modeling, CMLM, based on CRM to realize domain transfer by masked language modeling (MLM) and sequence-level contrastive learning (CL) with pairs of masked sequences. The framework of CMLM is shown in Figure 2, which is described as follows.

Given a batch $\mathcal{T} = \{T_1, T_2, \dots, T_B\}$ of input sequences, we firstly apply dynamic random masking (DRM) on each sequence T_b to generate a masked sequence T_b^0 , and then apply CRM K times to generate $T_b^1, T_b^2, \dots, T_b^K$ based on T_b^0 and T_b :

$$T_b^0 = \text{DRM}(T_b), b \in [1, B] \quad (3)$$

$$T_b^k = \text{CRM}(T_b, T_b^0), k \in [1, K] \quad (4)$$

After obtaining $K + 1$ masked sequences from each sequence T_b in \mathcal{T} , we then compute their representations $H_b^k \in \mathbb{R}^{N \times d}$ by using an encoder, where d is the hidden size of the encoder:

$$H_b^k = \text{Encoder}(T_b^k), k \in [0, K] \quad (5)$$

Even though our approach is model-agnostic, in this paper we focus on the Transformer-based pre-trained language model RoBERTa, which is an enhanced version of BERT. Therefore, we employ RoBERTa to implement the $\text{Encoder}(\cdot)$ function.

To capture token-level similarity, We apply MLM on H_b^0 as Devlin et al. (2019) and Liu et al. (2019b), and compute the loss as follows:

$$\mathcal{L}_{MLM} = \frac{1}{B} \sum_{b=1}^B \text{MLM}(H_b^0) \quad (6)$$

To capture sequence-level similarity, we apply contrastive learning on each H_b^k and H_b^0 , and obtain the loss term \mathcal{L}_{CL} . We compare two different implementations of contrastive learning: SimCLR (Chen et al., 2020) and SimSiam (Chen and He, 2020). For SimCLR, \mathcal{L}_{CL} can be calculated as:

$$\mathcal{L}_{CL} = -\frac{1}{K \cdot B} \sum_{k=1}^K \sum_{b=1}^B \log \frac{e^{\text{sim}(H_b^k, H_b^0)/\tau}}{\sum_{i=1}^B e^{\text{sim}(H_i^k, H_b^0)/\tau}} \quad (7)$$

where τ is a temperature parameter.

Following Gunel et al. (2021), we take the first token representation $h_b^k \in \mathbb{R}^d$ of H_b^k to calculate the similarity between H_i^k and H_j^0 as follows.

$$\text{sim}(H_i^k, H_j^0) = \frac{h_i^k}{\|h_i^k\|_2} \cdot \frac{h_j^0}{\|h_j^0\|_2} \quad (8)$$

SimSiam is similar to SimCLR except without using negative pairs and has a negative loss value. To be consistent with \mathcal{L}_{MLM} and \mathcal{L}_{CL} , we define the loss function of SimSiam as follows:

$$\mathcal{L}_{CL} = \frac{1}{K \cdot B} \sum_{k=1}^K \sum_{b=1}^B e^{-\frac{1}{2}(D(z_b^k, h_b^0) + D(z_b^0, h_b^k))} \quad (9)$$

where z_b^k and $D(z, h)$ are defined as:

$$z_b^k = W_2 \cdot \text{gelu}(W_1 \cdot h_b^k) \quad (10)$$

$$D(z, h) = \text{sim}(z, \text{stopgrad}(h)). \quad (11)$$

Here, $W_1, W_2 \in R^{d \times d}$ are learnable parameters, $\text{sim}(\cdot)$ is similar to Equation 8, and $\text{Stopgrad}(\cdot)$ is a stop-gradient operation which is crucial for SimSiam (Chen and He, 2020).

Finally, we combine \mathcal{L}_{MLM} and \mathcal{L}_{CL} for contrastive masking language modeling:

$$\mathcal{L}_{CMLM} = \mathcal{L}_{MLM} + \alpha \cdot \mathcal{L}_{CL} \quad (12)$$

where α is a tunable hyper-parameter.

3.4 Relationship to Existing Approaches

Among existing approaches, the closest one to ours is CSSL Pre-training (Fang and Xie, 2020). We can implement CSSL Pre-training by slightly modifying Equation 3 and 12 to following ones:

$$T_b^k = \text{back}(T_b), k \in [0, 1] \quad (13)$$

$$\mathcal{L}_{CSSL} = \mathcal{L}_{CL} \quad (14)$$

where $\text{back}(T)$ means back-translation of T . And other equations stay the same. By comparing these equations, we note that CMLM can be considered as: (1) replacing back-translation with CRM, which not only reduces the computational cost but also prevents the model from depending on the translation systems; (2) adding \mathcal{L}_{MLM} to implement token-level contrastive learning, which is shown to be crucial in Section 5.2; (3) easily extending one pair of positive samples to K pairs, which can be attributed to the nature of random masking.

As for the loss term, both Giorgi et al. (2020) and Wu et al. (2020) use similar terms to ours for pre-training: $\mathcal{L}_{total} = \mathcal{L}_{MLM} + \mathcal{L}_{CL}$, where \mathcal{L}_{MLM} captures token-level similarity and \mathcal{L}_{CL} captures sequence-level similarity. The main difference between these methods and ours is that we use differently masked sequences from the same sequence as a positive pair, while Giorgi et al. (2020) use position-related segments (overlapping, adjacent or subsumed) and Wu et al. (2020) use sequences by different deformations as the positive pair.

4 Experiment

4.1 Tasks: GLUE

We evaluate our model on the GLUE benchmark (Wang et al., 2018), which contains 9 natural language understanding tasks that can be divided into three categories: (1) single sentence tasks: CoLA and SST-2; (2) similarity and paraphrase tasks: MRPC, QQP, and STS-B; (3) inference tasks: MNLI, QNLI, RTE, and WNLI. All of them are classification tasks except STS-B, so we eliminate it to focus on the classification tasks. WNLI has a small development set (70 examples) and is also ignored. MNLI contains two evaluation sets. One, denoted as MNLI, is from the same sources as the training set, and the other, denoted as MNLI-MM, is from different sources than the training set.

To simulate few-shot scenes of different degrees, we randomly select 20, 100, and 1000 examples respectively from these tasks as our training sets following recent work (Gunel et al., 2021). For each subset in each task, we sample 5 times with replacement and obtain 15 training sets for each task. As for the development set and test set, we randomly select 500 examples from the original development set as our development set and take the remaining as our test set. Since QQP contains too many examples (40k) in the original development set, we randomly select 2000 from the remaining examples after sampling our development set as our test set. Note that all the 15 training sets in each task share the same development and test sets.

4.2 Model: RoBERTa

As mentioned above, we take RoBERTa to implement our encoder in Equation 5. The base version of RoBERTa, Roberta-base, which contains 12 Transformer blocks with 12 self-attention heads, is employed. All the blocks have the same hidden size 768. The input sequence is either a segment or two segments separated by a special token “[\s]”, while “[s]” is always the first token. We take the implementation and pre-trained weights from Huggingface Transformers library (Wolf et al., 2020).

4.3 Training Details

For the fine-tuning of all approaches to be reported below, unless otherwise specified, we use AdamW (Loshchilov and Hutter, 2019) with a learning rate of 1e-5 and epochs of 350, 100, 10 for subsets sized 20, 100, 1000, respectively. This setting is

	CoLA	SST-2	MNLI	MNLI-MM	QNLI	RTE	MRPC	QQP
Metric	mcc	acc	acc	acc	acc	acc	acc	acc
data size = 20								
FT	0.0751±6.26	0.6604±6.38	0.3578±1.70	0.3652±1.96	0.6163±3.46	0.5281±4.07	0.6747±2.23	0.6777±3.67
SCL	0.1105±8.30	0.6964±6.24	0.3684±2.85	0.3751±3.41	0.6191±3.55	0.5082±7.75	0.6631±1.15	0.6947±2.35
CSSL	0.0795±4.13	0.6609±5.98	0.3640±2.06	0.3686±2.69	0.6064±3.26	0.5264±7.10	0.6638±1.64	0.6514±2.99
TAPT	0.0860±7.64	0.7326±4.99	0.3616±2.12	0.3689±2.40	0.6146±3.60	0.5437±5.06	0.6552±1.64	0.6584±3.33
CMLM (ours)	0.0902±8.65	0.7371±5.64	0.3633±2.15	0.3701±2.57	0.6231±3.60	0.5437±4.57	0.6586±1.41	0.6541±3.89
data size = 100								
FT	0.2176±7.78	0.8405±2.71	0.4361±2.50	0.4526±2.94	0.6820±2.56	0.5879±4.82	0.7099±1.72	0.7511±1.72
SCL	0.2467±5.46	0.8455±1.38	0.4499±3.30	0.4627±3.84	0.6765±2.47	0.5835±6.41	0.7063±1.45	0.7461±1.63
CSSL	0.1719±7.90	0.8401±1.71	0.4185±2.98	0.4298±3.55	0.6701±1.89	0.5532±5.10	0.7038±1.58	0.7274±1.85
TAPT	0.2626±6.03	0.8496±2.52	0.4508±2.60	0.4682±2.80	0.6970±1.63	0.6095±6.60	0.6987±1.77	0.7429±2.13
CMLM (ours)	0.2663±6.97	0.8525±1.95	0.4530±2.75	0.4683±3.00	0.6980±1.67	0.6147±6.36	0.6933±1.90	0.7479±2.16
data size = 1000								
FT	0.4216±3.13	0.8996±0.97	0.7048±1.19	0.7168±1.17	0.7681±1.07	0.7472±2.50	0.8223±1.22	0.7934±0.90
SCL	0.2758±11.94	0.8991±1.04	0.5020±3.81	0.5089±4.09	0.7449±1.22	0.7100±5.44	0.7157±6.83	0.7853±0.77
CSSL	0.4069±3.53	0.8993±1.38	0.6900±1.37	0.7048±1.39	0.7760±0.97	0.7082±5.30	0.8261±1.70	0.7881±1.00
TAPT	0.4362±3.46	0.9016±0.70	0.7074±1.79	0.7203±1.68	0.7689±0.72	0.7524±4.06	0.8214±1.14	0.7890±0.82
CMLM (ours)	0.4374±2.06	0.9023±0.88	0.7110±2.00	0.7247±1.84	0.7719±0.91	0.7610±3.28	0.8223±0.82	0.7891±0.90

Table 1: Results on the GLUE benchmark with 20, 100 and 1000 training examples, respectively, and compared with baseline (FT: (Liu et al., 2019b)) and several recent post-training or contrastive learning methods (SCL (Gunel et al., 2021), CSSL (Fang and Xie, 2020), TAPT (Gururangan et al., 2020)). Unit of standard deviation is 10^{-2} .

based on previous empirical results (Zhang et al., 2021; Mosbach et al., 2021), which show that fine-tuning with a small learning rate and more epochs stabilizes the performance of a model in few-shot scenes. We set the batch size to 16 and dropout rate to 0.1, and save model parameters every 100 update steps and pick the best based on validation.

For post-training of CMLM, we apply AdamW with a learning rate of $1e-5$ and epochs of 200, 50, 5 for subsets sized 20, 100, 1000, respectively. For a fair comparison with other approaches, we set K in Equation 3 to 1 and the batch size to 8, where the maximum GPU memory usage is approximately equal to that of fine-tuning. For the implementation of \mathcal{L}_{CL} , we choose SimSiam for it consumes less computation. For p_m in Equation 1, we follow (Liu et al., 2019b) and set it to 0.15. We conduct a grid-based search for hyper-parameters with $\alpha \in \{0.01, 0.1, 0.3, 0.5, 0.7, 1\}$ (Equation 12) and $p_c \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ (Equation 2), and find that the combination of $\alpha = 0.5$ and $p_c = 0.7$ performs the best on the development set.

For the baselines to be introduced below, we follow the same fine-tuning and post-training settings as our CMLM, with only several method-specific hyper-parameters unchanged.

4.4 Baseline Approaches

As mentioned in Section 2.2 and 2.3, there have been works trying to add extra loss terms in fine-tuning or to insert a post-training phase in between pre-training and fine-tuning. To make a com-

prehensive comparison, we employ the following approaches as our baselines: (1) fine-tuning (FT) (Liu et al., 2019b), which directly fine-tunes a model with cross-entropy loss; (2) fine-tuning with SCL (SCL) (Gunel et al., 2021), which fine-tunes a model with cross-entropy loss and supervised contrastive loss; (3) post-training with CSSL (CSSL) (Fang and Xie, 2020), which post-trains a model with contrastive self-supervised learning loss; (4) post-training with MLM (TAPT) (Gururangan et al., 2020), which post-trains a model with MLM loss and is equal to CMLM when $\alpha = 0$. Comparing with recent works (Fang and Xie, 2020; Gunel et al., 2021) that take only the conventional either BERT or RoBERTa as their baseline, we consider a few more baselines to obtain more conclusive results.

4.5 Evaluation Details

In few-shot scenes, the distribution of the training set may deviate from the test set seriously. Gunel et al. (2021) pick the top-3 results from all combinations of training sets and model seeds for each task. Differently, for each data size of 20, 100, and 1000 described in Section 4.1, we train our model with random seeds $\{31, 42, 53\}$ for the 5 training subsets, and calculate the mean and standard deviation of the 15 test results. We assume this is a better way to evaluate the overall effect of our model.

4.6 Few-Shot Results

In Table 1, we report our few-shot results on the GLUE tasks with 20, 100, and 1000 training exam-

ples, respectively. Five observations can be made from the table. First, CMLM obtains superior performance on the datasets with 100 and 1000 examples, surpassing the baselines in 13 of 16 tasks. Since we use the same hyper-parameters for these approaches and report the average results over 3 random seeds and 5 randomly sampled training sets, these results are convincing. Second, on the dataset with 20 training examples, CMLM only surpasses the other approaches in 3 of 8 tasks. Training a model with only 20 examples is very unstable, and the test results of the baseline approaches indeed show large deviations across different training sets. Third, we find that post-training with only L_{MLM} (Xu et al., 2019; Gururangan et al., 2020) can achieve competitive results with the baselines, showing the effectiveness of this widely-used approach. Fourth, SCL (Gunel et al., 2021) has extremely poor performance on CoLA, MNLI, and MRPC when the data size is 1000, which is beyond our expectation. In the original paper, the authors of SCL only report the top-3 results from combinations of model seeds and train sets. So we speculate this under-performance might come from the instability of SCL in few-shot settings. Fifth, CSSL (Fang and Xie, 2020) performs even worse than FT when the data size is either 20 or 100 but achieves competitive results when the data size is 1000. CSSL is designed for full-size GLUE tasks and might not be suitable for the few-shot scenes.

4.7 Full-Size Results

To verify whether post-training with CMLM can still achieve desirable results when sufficient labeled examples are available, we conduct experiments on the RTE (2.5k), MRPC (3.7k), CoLA (8.5k), SST-2 (67k), and QNLI (106k) tasks with their full-size training sets. We set the learning rate to $3e-5$ for both post-training and fine-tuning and set the epoch to 3. Other hyper-parameters remain the same as in Section 4.3. Experiment results are shown in Table 2, from which we can note that CMLM maintains its superiority on RTE and CoLA but fails on MRPC, SST-2, and QNLI. TAPT performs better on tasks with more training examples, which can be explained by the better generalizability of token-level representation, though it demands more training steps to learn well. Note that CMLM is specifically proposed for few-shot settings, so the experiments in the full-size setting are only to evaluate it from different perspectives and make a

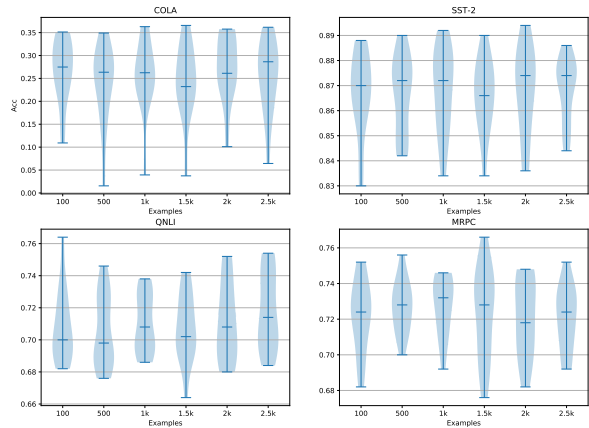


Figure 3: Results of our model on development sets with increasing numbers of unlabeled examples.

comprehensive comparison with baselines.

	RTE	MRPC	CoLA	SST-2	QNLI
metric	acc	acc	mcc	acc	acc
data-size	2.5k	3.7k	8.5k	67k	106k
FT	0.7403	0.8623	0.5552	0.9319	0.9043
SCL	0.6753	0.7393	0.5329	0.9373	0.9002
CSSL	0.6623	0.8713	0.5217	0.9310	0.8904
TAPT	0.7403	0.8541	0.5519	0.9355	0.9063
CMLM (ours)	0.7446	0.8574	0.5714	0.9310	0.9039

Table 2: Results on the RTE, MRPC, CoLA, SST-2 and QNLI tasks with full-size training sets, and average results over 3 random seeds are reported.

4.8 Additional Unlabeled Examples

We consider the scene where additional unlabeled task examples are provided. We evaluate CMLM on CoLA, SST-2, QNLI, and MRPC with 100 labeled examples for fine-tuning and increase unlabeled examples from 100 to 2500 for post-training. We depict the results on the development sets in Figure 3, from which two observations can be made. First, the performance generally increases with the number of unlabeled examples grows, showing the helpfulness of unlabeled task examples, which is also confirmed by Gururangan et al. (2020). Second, there are certain fluctuations in the results. We assume they come from the random nature of these additional unlabeled examples, which are sampled from a much larger training set and might severely deviate from the original training set. Moreover, we should acknowledge that adding more unlabeled examples in post-training gains limited improvement compared with adding more labeled examples in fine-tuning. Labeled examples are more treasurable in classification learning.

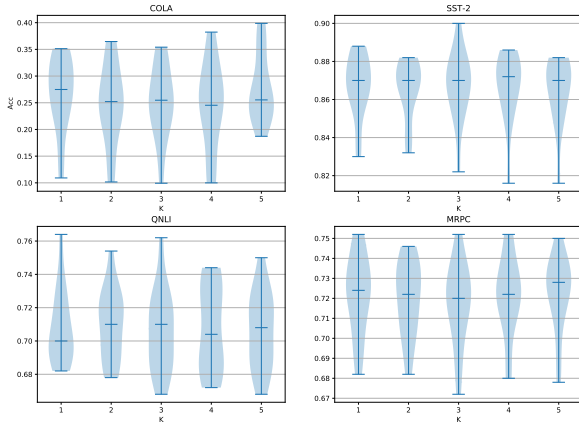


Figure 4: Results of our model on development sets while the parameter K changes.

4.9 Hyper-parameter K

We evaluate whether increasing K (in Equation 7) can lead to improvement of our model in the few-shot setting. We evaluate CMLM on the CoLA, SST-2, QNLI, and MRPC tasks with 100 training examples by increasing K , and the results are depicted in Figure 4. Similar to increasing unlabeled examples, the performance slightly improves on the 5 tasks but has some fluctuations, which is within our expectation. Intuitively, exposing the model to different forms of masked sequences can better reflect the distribution of examples sampled from a large training set, but cannot narrow the deviation between these examples and the original train set.

5 Ablation Studies

5.1 SimCLR vs SimSiam

As described above, \mathcal{L}_{CL} can be implemented by either Equation 7 or Equation 10, although we implement the latter to conduct the above experiments for its less computational cost. According to Chen and He (2020), SimSiam performs better than SimCLR on ImageNet (Deng et al., 2009). It is thus interesting to verify whether the same holds in our situation. We compare SimSiam (CMLM) and SimCLR (w/ SimCLR) on SST-2, CoLA, QNLI and RTE, and the results are reported in Table 3. From the results, we cannot easily conclude which one is better due to their comparable performances, yet further investigation is beyond the scope of this paper. However, we prefer SimSiam due to it consumes less computation and is easier to implement.

	SST-2	CoLA	QNLI	RTE
Metric	acc	mcc	acc	acc
data size = 100				
CMLM	0.8525	0.2663	0.6980	0.6147
w/ SimCLR	0.8586	0.2511	0.6885	0.6355
w/o CL	0.8496	0.2626	0.6980	0.6095
w/o MLM	0.8280	0.2492	0.6873	0.5913
w/o CRM	0.8511	0.2621	0.6920	0.6242
data size = 1000				
CMLM	0.9023	0.4374	0.7719	0.7610
w/ SimCLR	0.9041	0.4446	0.7696	0.7732
w/o CL	0.9016	0.4362	0.7689	0.7524
w/o MLM	0.8927	0.3983	0.7623	0.7039
w/o CRM	0.9013	0.4434	0.7698	0.7654

Table 3: Results of ablation study for CMLM. *w/ SimCLR* means replacing SimSiam with SimCLR, *w/o CL* and *w/o MLM* mean removing \mathcal{L}_{CL} and \mathcal{L}_{MLM} terms from \mathcal{L}_{CMLM} , respectively, and *w/o CRM* means replacing CRM with dynamic random masking (DRM).

5.2 Are MLM & CL Critical for CMLM?

One of the improvements of CMLM over previous works is combining \mathcal{L}_{MLM} and \mathcal{L}_{CL} to implement both token-level and sequence-level contrastive learnings. Here, we verify how the bi-granularity contrastive learnings contribute to the performance differently. We remove \mathcal{L}_{MLM} and \mathcal{L}_{CL} alternatively from \mathcal{L}_{CMLM} and evaluate the resulting model on SST-2, CoLA, QNLI, and RTE with 100 and 1000 training examples, respectively. The results are reported in Table 3. As we can see, the results suffer severe deterioration by up to 7.5% after removing \mathcal{L}_{MLM} , while removing \mathcal{L}_{CL} only leads to a drop by up to 1.4%. Although both \mathcal{L}_{MLM} and \mathcal{L}_{CL} contribute to the improvement of CMLM, MLM tends to play a more essential role.

5.3 Complementary Random Masking vs Dynamic Random Masking

We propose a complementary random masking (CRM) strategy to generate complementary masked sequences T^k , $k \in [1, K]$, based on T^0 , which is generated by dynamic random masking (DRM). Here, we verify whether this complementary nature of T^k benefits contrastive learning. We replace CRM in Equation 3 by DRM, and conduct experiments on SST-2, CoLA, QNLI and RTE with 100 and 1000 training examples, respectively. As shown in Table 3, CRM still surpasses DRM on all 8 tasks, with improvement by up to 1.6%. The superiority of CRM mainly comes from fact that it avoids tokens to be masked in both T^0 and T^k .

6 Conclusion

In this paper, we proposed a novel post-training objective, CMLM, for pre-trained language models in downstream few-shot scenes. CMLM attempts to combine both token-level and sequence-level contrastive learnings for more efficient domain transfer during post-training. For sentence-level contrastive learning, we developed a random masking strategy, CRM, to generate a pair of complementary masked sequences for an input sequence. Empirical results show that post-training with our CMLM outperforms other recent approaches on the GLUE tasks with 100 and 1000 labeled training examples, respectively. We also conducted extensive ablation studies and showed that both token-level and sequence-level contrastive learnings contribute to the results of CMLM, and that CRM achieves favorable sequence-level contrastive learning over the previous masking strategy. In future work, we will further investigate how token-level and sequence-level contrastive learnings affect domain transfer in post-training and explore more effective methods for sequence-level contrastive learning.

Acknowledgments

The paper was supported by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No.2017ZT07X355).

References

- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607, Virtual. PMLR.
- Xinlei Chen and Kaiming He. 2020. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*.
- J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. 2009. *Imagenet: A large-scale hierarchical image database*. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of deep bidirectional transformers for language understanding*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. *Understanding back-translation at scale*. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.
- Hongchao Fang and Pengtao Xie. 2020. CERT: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766*.
- John M Giorgi, Osvald Nitski, Gary D Bader, and Bo Wang. 2020. DeCLUTR: Deep contrastive learning for unsupervised textual representations. *arXiv preprint arXiv:2006.03659*.
- Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. *Supervised contrastive learning for pre-trained language model fine-tuning*. In *International Conference on Learning Representations*.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. *Don’t stop pretraining: Adapt language models to domains and tasks*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- 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)*, volume 2, pages 1735–1742.
- Dan Iter, Kelvin Guu, Larry Lansing, and Dan Jurafsky. 2020. *Pretraining with contrastive sentence objectives improves discourse performance of language models*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4859–4870, Online. Association for Computational Linguistics.
- Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. *SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. *Supervised contrastive learning*. In *Advances in Neural*

- Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.
- Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020. [Mixout: Effective regularization to finetune large-scale pretrained language models](#). In *International Conference on Learning Representations*.
- Tian Li, Xiang Chen, Shanghang Zhang, Zhen Dong, and Kurt Keutzer. 2021. [Cross-domain sentiment classification with contrastive learning and mutual information maximization](#). In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8203–8207.
- Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. [Multi-task deep neural networks for natural language understanding](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. [RoBERTa: A robustly optimized BERT pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. [On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines](#). In *International Conference on Learning Representations*.
- Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. [Sentence encoders on STILTs: Supplementary training on intermediate labeled-data tasks](#). *arXiv preprint arXiv:1811.01088*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. [Improving language understanding by generative pre-training \(2018\)](#).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. [CLEAR: Contrastive learning for sentence representation](#). *arXiv preprint arXiv:2012.15466*.
- Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. [BERT post-training for review reading comprehension and aspect-based sentiment analysis](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.
- Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. [Revisiting few-sample BERT fine-tuning](#). In *International Conference on Learning Representations*.