

# CLEAR: Contrastive Learning for Sentence Representation

Zhuofeng Wu<sup>1\*</sup> Sinong Wang<sup>2</sup> Jiatao Gu<sup>2</sup>  
Madian Khabsa<sup>2</sup> Fei Sun<sup>3</sup> Hao Ma<sup>2</sup>

<sup>1</sup>School of Information, University of Michigan  
zhuofeng@umich.edu

<sup>2</sup>Facebook AI

{sinongwang, jgu, mkhabasa, haom}@fb.com

<sup>3</sup>Institute of Computing Technology, Chinese Academy of Sciences  
ofey.sunfei@gmail.com

## Abstract

Pre-trained language models have proven their unique powers in capturing implicit language features. However, most pre-training approaches focus on the word-level training objective, while sentence-level objectives are rarely studied. In this paper, we propose Contrastive LEARNING for sentence Representation (CLEAR), which employs multiple sentence-level augmentation strategies in order to learn a noise-invariant sentence representation. These augmentations include word and span deletion, reordering, and substitution. Furthermore, we investigate the key reasons that make contrastive learning effective through numerous experiments. We observe that different sentence augmentations during pre-training lead to different performance improvements on various downstream tasks. Our approach is shown to outperform multiple existing methods on both SentEval and GLUE benchmarks.

## 1 Introduction

Learning a better sentence representation model has always been a fundamental problem in Natural Language Processing (NLP). Taking the mean of word embeddings as the representation of sentence (also known as mean pooling) is a common baseline in the early stage. Later on, pre-trained models such as BERT (Devlin et al., 2019) propose to insert a special token (i.e., [CLS] token) during the pre-training and take its embedding as the representation for the sentence. Because of the tremendous improvement brought by BERT (Devlin et al., 2019), people seemed to agree that CLS-token embedding is better than averaging word embeddings. Nevertheless, a recent paper SentenceBERT (Reimers and Gurevych, 2019) observed

that averaging of all output word vectors outperforms the CLS-token embedding marginally. Sentence-BERT’s results suggest that models like BERT learn a better representation at the token level. One natural question is how to better learn sentence representation.

Inspired by the success of contrastive learning in computer vision (Zhuang et al., 2019; Tian et al., 2019; He et al., 2020; Chen et al., 2020; Misra and Maaten, 2020), we are interested in exploring whether it could also help language models generate a better sentence representation. The key method in contrastive learning is augmenting positive samples during the training. However, data augmentation for text is not as fruitful as for image. The image can be augmented easily by rotating, cropping, resizing, or cutouting, etc. (Chen et al., 2020). In NLP, there are minimal augmentation ways that have been researched in literature (Giorgi et al., 2020; Fang and Xie, 2020). The main reason is that every word in a sentence may play an essential role in expressing the whole meaning. Additionally, the order of the words also matters.

Most existing pre-trained language models (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2019) are adding different kinds of noises to the text and trying to restore them at the word-level. Sentence-level objectives are rarely studied. BERT (Devlin et al., 2019) combines the word-level loss, masked language modeling (MLM) with a sentence-level loss, next sentence prediction (NSP), and observes that MLM+NSP is essential for some downstream tasks. RoBERTa (Liu et al., 2019) drops the NSP objective during the pre-training but achieves a much better performance in a variety of downstream tasks. ALBERT (Lan et al., 2019) proposes a self-supervised loss for Sentence-Order Prediction (SOP), which models the

\* Work done while the author was an intern at Facebook AI.

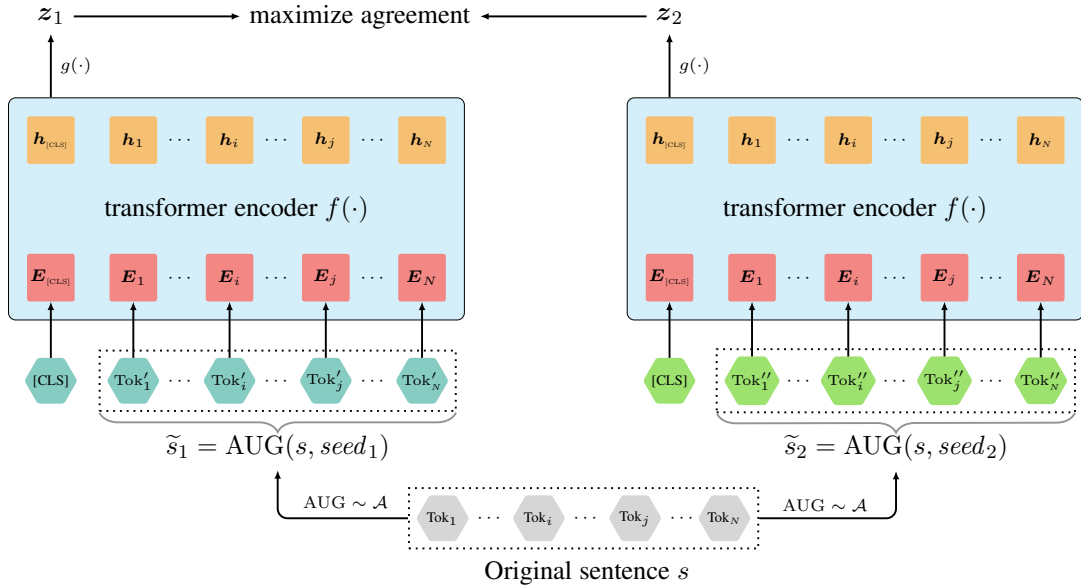


Figure 1: The proposed contrastive learning framework CLEAR.

inter-sentence coherence. Their work shows that coherence prediction is a better choice than the topic prediction, the way NSP uses. DeCLUTR (Giorgi et al., 2020) is the first work to combine Contrastive Learning (CL) with MLM into pre-training. However, it requires an extremely long input document, i.e., 2048 tokens, which restricts the model to be pre-trained on limited data. Further, DeCLUTR trains from existing pre-trained models, so it remains unknown whether it could also achieve the same performance when it trains from scratch.

Drawing from the recent advances in pre-trained language models and contrastive learning, we propose a new framework, CLEAR, combining word-level MLM objective with sentence-level CL objective to pre-train a language model. MLM objective enables the model capture word-level hidden features while CL objective ensures the model with the capacity of recognizing similar meaning sentences by training an encoder to minimize the distance between the embeddings of different augmentations of the same sentence. In this paper, we present a novel design of augmentations that can be used to pre-train a language model at the sentence-level. Our main findings and contributions can be summarized as follows:

- We proposed and tested four basic sentence augmentations: random-words-deletion, spans-deletion, synonym-substitution, and reordering, which fills a large gap in NLP

about what kind of augmentations can be used in contrastive learning.

- We showed that model pre-trained by our proposed method outperforms several strong baselines (including RoBERTa and BERT) on both GLUE (Wang et al., 2018) and SentEval (Conneau and Kiela, 2018) benchmark. For example, we showed +2.2% absolute improvement on 8 GLUE tasks and +5.7% absolute improvement on 7 SentEval semantic textual similarity tasks compared to RoBERTa model.

## 2 Related Work

There are three lines of literatures that are closely related to our work: sentence representation, large-scale pre-trained language representation models, contrastive learning.

### 2.1 Sentence Representation

Learning the representation of sentence has been studied by many existing works. Applying various pooling strategies onto word embeddings as the representation of sentence is a common baseline (Iyyer et al., 2015; Shen et al., 2018; Reimers and Gurevych, 2019). Skip-Thoughts (Kiros et al., 2015) trains an encoder-decoder model trying to reconstruct surrounding sentences. Quick-Thoughts (Logeswaran and Lee, 2018) trains a encoder-only model with the ability to select the correct context of the sentence out of other contrastive sentences. Later

on, many pre-trained language models such as BERT (Devlin et al., 2019) propose to use the manually-inserted token (the [CLS] token) as the representation of the whole sentence and become the new state-of-the-art in a variety of downstream tasks. One recent paper SentenceBERT (Reimers and Gurevych, 2019) compares the average BERT embeddings with the CLS-token embedding and surprisingly finds that computing the mean of all output vectors at the last layer of BERT outperforms the CLS-token marginally.

## 2.2 Large-scale Pre-trained Language Representation Models

The deep pre-trained language models have proven their powers in capturing implicit language features even with different model architectures, pre-training tasks, and loss functions. Two of the early works that are GPT (Radford et al., 2018) and BERT (Devlin et al., 2019): GPT uses a left-to-right Transformer while BERT designs a bidirectional Transformer. Both created an incredible new state of the art in a lot of downstream tasks.

Following this observation, recently, a tremendous number of research works are published in the pre-trained language model domain. Some extend previous models to a sequence-to-sequence structure (Song et al., 2019; Lewis et al., 2019; Liu et al., 2020), which enforces the model’s capability on language generation. The others (Yang et al., 2019; Liu et al., 2019; Clark et al., 2020) explore the different pre-training objectives to either improve the model’s performance or accelerate the pre-training.

## 2.3 Contrastive Learning

Contrastive Learning has become a rising domain because of its significant success in various computer vision tasks and datasets. Several researchers (Zhuang et al., 2019; Tian et al., 2019; Misra and Maaten, 2020; Chen et al., 2020) proposed to make the representations of the different augmentation of an image agree with each other and showed positive results. The main difference between these works is their various definition of image augmentation.

Researchers in the NLP domain have also started to work on finding suitable augmentation for text. CERT (Fang and Xie, 2020) applies the back-translation to create augmentations of original sentences, while DeCLUTR (Giorgi et al.,

2020) regards different spans inside one document are similar to each others. Our model differs from CERT in adopting an encoder-only structure, which decreases noise brought by the decoder. Further, unlike DeCLUTR, which only tests one augmentation and trains the model from an existing pre-trained model, we pre-train all models from scratch, which provides a straightforward comparison with the existing pre-trained models.

## 3 Method

This section proposes a novel framework and several sentence augmentation methods for contrastive learning in NLP.

### 3.1 The Contrastive Learning Framework

Borrow from SimCLR (Chen et al., 2020), we propose a new contrastive learning framework to learn the sentence representation, named as CLEAR. There are four main components in CLEAR, as outlined in Figure 1.

- An augmentation component  $AUG(\cdot)$  which apply the random augmentation to the original sentence. For each original sentence  $s$ , we generate two random augmentations  $\tilde{s}_1 = AUG(s, seed_1)$  and  $\tilde{s}_2 = AUG(s, seed_2)$ , where  $seed_1$  and  $seed_2$  are two random seeds. Note that, to test each augmentation’s effect solely, we adopt the same augmentation to generate  $\tilde{s}_1$  and  $\tilde{s}_2$ . Testing the mixing augmentation models requests more computational resources, which we plan to leave for future work. We will detail the proposed augmentation set  $\mathcal{A}$  at Section 3.3.
- A transformer-based encoder  $f(\cdot)$  that learns the representation of the input augmented sentences  $H_1 = f(\tilde{s}_1)$  and  $H_2 = f(\tilde{s}_2)$ . Any encoder that learns the sentence representation can be used here to replace our encoder. We choose the current start-of-the-art (i.e., transformer (Vaswani et al., 2017)) to learn sentence representation and use the representation of a manually-inserted token as the vector of the sentence (i.e., [CLS], as used in BERT and RoBERTa).
- A nonlinear neural network projection head  $g(\cdot)$  that maps the encoded augmentations  $H_1$  and  $H_2$  to the vector  $z_1 = g(H_1)$ ,  $z_2 = g(H_2)$  in a new space. According to observations in SimCLR (Chen et al., 2020), adding

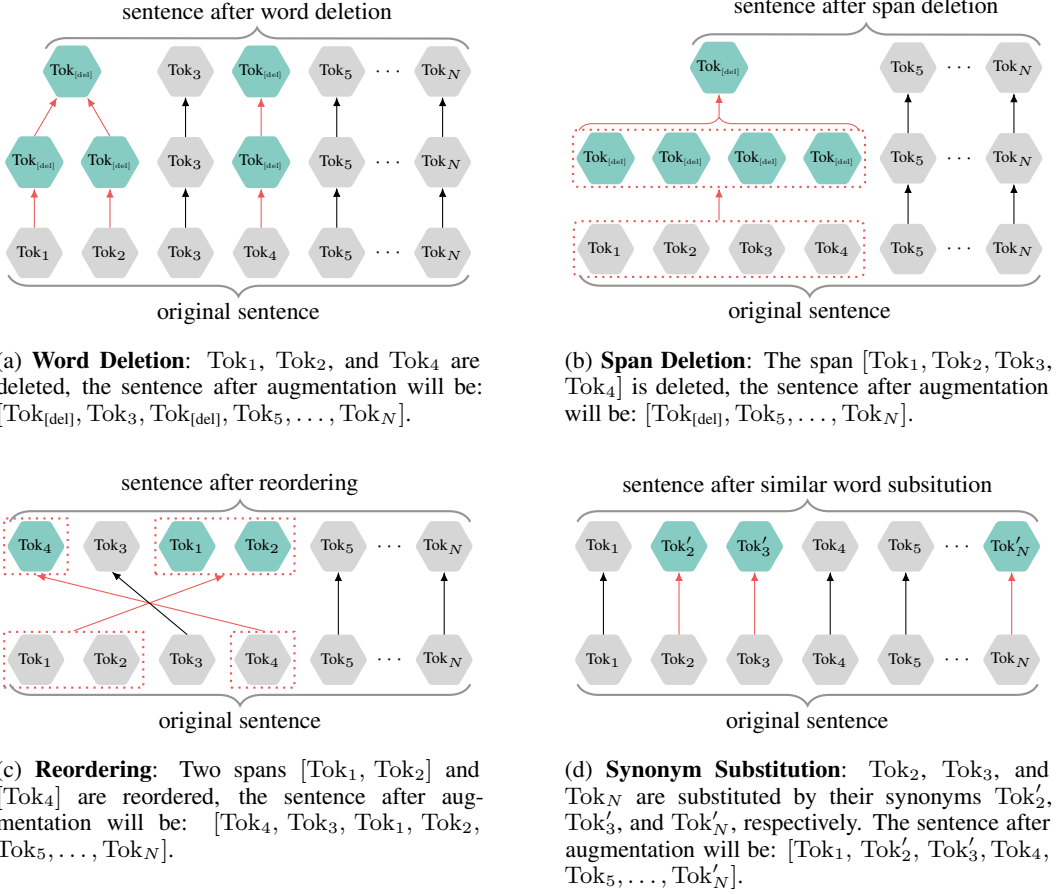


Figure 2: Four sentence augmentation methods in proposed contrastive learning framework CLEAR.

a nonlinear projection head can significantly improve representation quality of images.

- A contrastive learning loss function defined for a contrastive prediction task, i.e., trying to predict positive augmentation pair  $(\tilde{s}_1, \tilde{s}_2)$  in the set  $\{\tilde{s}\}$ . We construct the set  $\{\tilde{s}\}$  by randomly augmenting twice for all the sentences in a minibatch (assuming a minibatch is a set  $\{s\}$  size  $N$ ), getting a set  $\{\tilde{s}\}$  with size  $2N$ . The two variants from the same original sentence form the positive pair, while all other instances from the same minibatch are regarded as negative samples for them. The contrastive learning loss has been tremendously used in previous work (Wu et al., 2018; Chen et al., 2020; Giorgi et al., 2020; Fang and Xie, 2020). The loss function for a positive pair is defined as:

$$l(i, j) = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \quad (1)$$

where  $\mathbb{1}_{[k \neq i]}$  is the indicator function to judge

whether  $k \neq i$ ,  $\tau$  is a temperature parameter,  $\text{sim}(u, v) = u^\top v / (\|u\|_2 \|v\|_2)$  denotes the cosine similarity of two vector  $u$  and  $v$ . The overall contrastive learning loss is defined as the sum of all positive pairs' loss in a minibatch:

$$\mathcal{L}_{\text{CL}} = \sum_{i=1}^{2N} \sum_{j=1}^{2N} m(i, j) l(i, j) \quad (2)$$

where  $m(i, j)$  is a function returns 1 when  $i$  and  $j$  is a positive pair, returns 0 otherwise.

### 3.2 The Combined Loss for Pre-training

Similar to (Giorgi et al., 2020), for the purpose of grabbing both token-level and sentence-level features, we use a combined loss of MLM objective and CL objective to get the overall loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{CL}} \quad (3)$$

where  $\mathcal{L}_{\text{MLM}}$  is calculated through predicting the random-masked tokens in set  $\{s\}$  as described in BERT and RoBERTa (Devlin et al., 2019; Liu et al., 2019). Our pre-training target is to minimize the  $\mathcal{L}_{\text{total}}$ .

### 3.3 Design Rationale for Sentence Augmentations

The data augmentation is crucial for learning the representation of image (Tian et al., 2019; Jain et al., 2020). However, in language modeling, it remains unknown whether data (sentence) augmentation would benefit the representation learning and what kind of data augmentation could apply to the text. To answer these questions, we explore and test four basic augmentations (shown in Figure 2) and their combinations in our experiment. We do believe there exist more potential augmentations, which we plan to leave for future exploration.

One type of augmentation we consider is **deletion**, which bases on the hypothesis that some deletion in a sentence wouldn't affect too much of the original semantic meaning. In some case, it may happen that deleting some words leads the sentence to a different meaning (e.g., the word *not*). However, we believe including proper noise can benefit the model to be more robust. We consider two different deletions, i.e., **word deletion** and **span deletion**.

- Word deletion (shown in Figure 2a) randomly selects tokens in the sentence and replace them by a special token [DEL], which is similar to the token [MASK] in BERT (Devlin et al., 2019).
- Span deletion (shown in Figure 2b) picks and replaces the deletion objective on the span-level. Generally, span-deletion is a special case of word-deletion, which puts more focus on deleting consecutive words.

To avoid the model easily distinguishing the two augmentations from the remaining words at the same location, we eliminate the consecutive token [DEL] into one token.

**Reordering** (shown in Figure 2c) is another widely-studied augmentation that can keep the original sentence's features. BART (Lewis et al., 2019) has explored restoring the original sentence from the random reordered sentence. We randomly sample several pairs of span and switch them pairwise to construct the reordering augmentation in our implementation.

**Substitution** (shown in Figure 2d) has been proven efficient in improving model's robustness (Jia et al., 2019). Following their work, we

sample some words and replace them with synonyms to construct one augmentation. The synonym list comes from a vocabulary they used. In our pre-training corpus, there are roughly 40% tokens with at least one similar-meaning token in the list.

## 4 Experiment

This section presents empirical experiments that compare the proposed methods with various baselines and alternative approaches.

### 4.1 Setup

**Model configuration:** We use the Transformer (12 layers, 12 heads and 768 hidden size) as our primary encoder (Vaswani et al., 2017). Models are pre-trained for 500K updates, with mini-batches containing 8,192 sequences of maximum length 512 tokens. For the first 24,000 steps, the learning rate is warmed up to a peak value of  $6e-4$ , then linearly decayed for the rest. All models are optimized by Adam (Kingma and Ba, 2014) with  $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1e-6$ , and  $L_2$  weight decay of 0.01. We use 0.1 for dropout on all layers and in attention. All of the models are pre-trained on 256 NVIDIA Tesla V100 32GB GPUs.

**Pre-training data:** We pre-train all the models on a combination of BookCorpus (Zhu et al., 2015) and English Wikipedia datasets, the data BERT used for pre-training. For more statistics of the dataset and processing details, one can refer to BERT (Devlin et al., 2019).

**Hyperparameters for MLM:** For calculating MLM loss, we randomly mask 15% tokens of the input text  $\mathbf{s}$  and use the surrounding tokens to predict them. To fill the gap between fine-tuning and pre-training, we also adopt the 10%-random-replacement and 10%-keep-unchanged setting in BERT for the masked tokens.

**Hyperparameters for CL:** To compute CL loss, we set up different hyperparameters:

- For **Word Deletion (del-word)**, we delete 70% tokens.
- For **Span Deletion (del-span)**, we delete 5 spans (each with 5% length of the input text).
- For **Reordering (reorder)**, we randomly pick 5 pairs of spans (each with roughly 5% length as well) and switch spans pairwise.

Table 1: Performance of competing methods evaluated on GLUE dev set. Following GLUE’s setting (Wang et al., 2018), unweighted average accuracy on the matched and mismatched dev sets is reported for MNLI. The unweighted average of accuracy and F1 is reported for MRPC and QQP. The unweighted average of Pearson and Spearman correlation is reported for STS-B. The Matthews correlation is reported for CoLA. For all other tasks we report accuracy.

Method	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS	Avg
Baselines									
BERT-base (Devlin et al., 2019)	84.0	89.0	89.1	61.0	93.0	86.3	57.3	89.5	81.2
RoBERTa-base (Liu et al., 2019)	87.2	93.2	88.2	71.8	94.4	87.8	56.1	89.4	83.5
MLM+1-CL-objective									
<b>MLM+ del-word</b>	86.8	93.0	90.2	79.4	94.2	89.7	62.1	<b>90.5</b>	<b>85.7</b>
<b>MLM+ del-span</b>	<b>87.3</b>	92.8	90.1	<b>79.8</b>	94.4	89.9	59.8	90.3	85.6
MLM+2-CL-objective									
<b>MLM+ subs+ del-word</b>	<b>87.3</b>	93.1	90.0	73.3	93.7	90.2	62.1	90.1	85.0
<b>MLM+ subs+ del-span</b>	87.0	<b>93.4</b>	<b>90.3</b>	74.4	94.3	90.5	63.3	<b>90.5</b>	85.5
<b>MLM+ del-word+ reorder</b>	87.0	92.7	89.5	76.5	<b>94.5</b>	<b>90.6</b>	59.1	90.4	85.0
<b>MLM+ del-span+ reorder</b>	86.7	92.9	90.0	78.3	<b>94.5</b>	89.2	<b>64.3</b>	89.8	<b>85.7</b>

- For **Substitution (subs)**, we randomly select 30% tokens and replace each token with one of their similar-meaning tokens.

Some of the above hyperparameters are slightly-tuned on the WikiText-103 dataset (Merity et al., 2016) (trained for 100 epochs, evaluated on the GLUE dev benchmark). For example, we find 70% deletion model perform best out of {30%, 40%, 50%, 60%, 70%, 80%, 90%} deletion models. For models using mixed augmentations, like MLM+2-CL-objective in Table 1, they use the same optimized hyperparameters as in the single model. For instance, our notation MLM+subs+del-span represents a model combining the MLM loss with CL loss: for MLM, it masks 15% tokens; for CL, it substitutes 30% tokens first and then deletes 5 spans to generate augmented sentences.

Note that the hyperparameters we used might not be the most optimized ones. Yet, it is unknown whether optimized hyperparameters on a 1-CL-objective model perform consistently on a 2-CL-objective model. Additionally, it is also unclear whether the optimized hyperparameters for WikiText-103 are still the optimized ones on BookCorpus and English Wikipedia datasets. However, it is hard to tune every possible hyperparameter due to the extensive computation resource requirement for pre-training. We will leave these questions to explore in the future.

## 4.2 GLUE Results

We mainly evaluate all the models by the General Language Understanding Evaluation (GLUE) benchmark development set (Wang et al., 2018). GLUE is a benchmark containing several different types of NLP tasks: natural language inference task (MNLI, QNLI, and RTE), similarity task (QQP, MRPC, STS), sentiment analysis task (SST), and linguistic acceptability task (CoLA). It provides a comprehensive evaluation for pre-trained language models.

To fit the different downstream tasks’ requirements, we follow the RoBERTa’s hyperparameters to finetune our model for various tasks. Specifically, we add an extra fully connected layer and then finetune the whole model on different training sets.

The primary baselines we include are BERT-base and RoBERTa-base. The results for BERT-base are from huggingface’s reimplementation<sup>1</sup>. A more fair comparison comes from RoBERTa-base since we use the same hyperparameters RoBERTa-base used for MLM loss. Note that our models are all combining two-loss, it is still unfair to compare a MLM-only model with a MLM+CL model. To answer this question, we set two other baselines in Section 5.1 to make a more strict comparison: one combines two MLM losses, the other adopts a double batch size.

<sup>1</sup><https://huggingface.co/transformers/v1.1.0/examples.html>

Table 2: Performance of competing methods evaluated on SentEval. All results are pre-trained on BookCorpus and English Wikipedia datasets for 500k steps.

Method	SICK-R	STS-B	STS12	STS13	STS14	STS15	STS16	Avg
Baselines								
RoBERTa-base-mean	74.1	65.6	47.2	38.3	46.7	55.0	49.5	53.8
RoBERTa-base-[CLS]	75.9	71.9	47.4	37.5	47.9	55.1	57.6	56.1
MLM+1-CL-objective								
<b>MLM+ del-word-mean</b>	75.9	69.0	<b>50.6</b>	40.0	50.2	58.9	52.4	56.7
<b>MLM+ del-span-mean</b>	71.0	62.6	49.3	41.7	48.9	58.1	52.3	54.8
<b>MLM+ del-word-[CLS]</b>	<b>77.1</b>	71.6	<b>50.6</b>	44.5	48.3	58.4	56.1	58.1
<b>MLM+ del-span-[CLS]</b>	62.7	57.4	34.4	20.4	24.3	32.0	31.5	37.5
MLM+2-CL-objective								
<b>MLM+ del-word+ reorder-mean</b>	75.8	66.2	51.1	45.7	51.8	61.3	57.0	58.4
<b>MLM+ del-span+ reorder-mean</b>	75.4	67.8	48.3	50.3	54.9	60.4	56.8	59.1
<b>MLM+ subs+ del-word-mean</b>	73.6	63.4	44.6	39.8	50.1	55.5	49.6	53.8
<b>MLM+ subs+ del-span-mean</b>	75.5	67.0	48.3	45.0	54.6	60.9	58.5	58.5
<b>MLM+ del-word+ reorder-[CLS]</b>	71.9	63.8	41.9	30.9	37.4	48.9	52.1	49.6
<b>MLM+ del-span+ reorder-[CLS]</b>	75.0	68.7	49.4	<b>54.3</b>	<b>57.6</b>	<b>64.0</b>	61.4	61.5
<b>MLM+ subs+ del-word-[CLS]</b>	73.6	62.9	44.5	35.8	47.6	55.8	59.6	54.3
<b>MLM+ subs+ del-span-[CLS]</b>	75.6	<b>72.5</b>	49.0	48.9	57.4	63.6	<b>65.6</b>	<b>61.8</b>

As we can see in Table 1, our proposed several models outperform the baselines on GLUE. Note that different tasks adopt different evaluation matrices, our two best models MLM+del-word and MLM+del-span+reorder both improve the best baseline RoBERTa-base by 2.2% on average score. Besides, a more important observation is that all best performance for each task comes from our proposed model. On CoLA and RTE, our best model exceeds the baseline by 7.0% and 8.0% correspondingly. Further, we also find that different downstream tasks benefit from different augmentations. We will make a more specific analysis in Section 5.2.

One notable thing is that we don't show the result of MLM+subs, MLM+reorder, and MLM+subs+reorder in Table 1. We observe that the pre-training for these three models either converges quickly or suffers from a gradient explosion problem, which indicates that these three augmentations are too easy to distinguish.

### 4.3 SentEval Results for Semantic Textual Similarity Tasks

SentEval is a popular benchmark for evaluating general-purpose sentence representations (Conneau and Kiela, 2018). The specialty for this benchmark is that it doesn't do the

fine-tuning like in GLUE. We evaluate the performance of our proposed methods for common Semantic Textual Similarity (STS) tasks on SentEval. Note that some previous models (e.g., Sentence-BERT (Reimers and Gurevych, 2019)) on the SentEval leaderboard trains on the specific datasets such as Stanford NLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2017), which makes it hard for a direct comparison. To make it easier, we compare one of our proposed models with RoBERTa-base directly on SentEval. According to Sentence-BERT, using the mean of all output vectors in the last layer is more effective than using the CLS-token output. We test both pooling strategies for each model.

From Table 2, we observe that mean-pooling strategy does not show much advantages. In many of the cases, CLS-pooling is better than the mean-pooling for our proposed models. The underlying reason is that the contrastive learning directly updates the representation of [CLS] token. Besides that, we find adding the CL loss makes the model especially good at the Semantic Textual Similarity (STS) task, beating the best baseline by a large margin (+5.7%). We think it is because the pre-training of contrastive learning is to find the similar sentence pairs, which aligns with STS task.

Table 3: Ablation study for several methods evaluated on GLUE dev set. All results are pre-trained on wiki-103 data for 500 epochs.

Method	MNLI-m	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS	Avg
RoBERTa-base	80.4	87.5	87.4	61.4	91.4	<b>82.4</b>	38.9	81.9	76.4
MLM-variant									
Double-batch RoBERTa-base	80.3	88.0	87.1	59.9	91.9	82.1	43.0	82.0	76.8
Double MLM RoBERTa-base	80.5	87.6	87.3	57.4	90.4	77.7	42.2	83.0	75.8
MLM+CL-objective									
<b>MLM+ del-span</b>	80.6	<b>88.8</b>	87.3	<b>62.1</b>	<b>92.1</b>	77.8	44.1	81.4	76.8
<b>MLM+ del-span + reorder</b>	<b>81.1</b>	88.7	<b>87.5</b>	58.1	90.0	80.4	43.3	<b>87.4</b>	77.1
<b>MLM+ subs + del-word + reorder</b>	80.5	87.7	87.3	59.6	90.4	80.2	<b>45.1</b>	87.1	<b>77.2</b>

This could explain why our proposed models show such large improvements on STS.

## 5 Discussion

This section discusses an ablation study to compare the CL loss and MLM loss and shows some observations about what different augmentation learns.

### 5.1 Ablation Study

Our proposed CL-based models outperforms MLM-based models, one remaining question is, where does our proposed model benefit from? Does it come from the CL loss, or is it from the larger batch size (since to calculate CL loss, one needs to store extra information per batch)? To answer this question, we set up two extra baselines: Double MLM RoBERTa-base adopts the MLM+MLM loss, each MLM is performed on different mask for the same original sentence; the other Double-batch RoBERTa-base uses single MLM loss with a double-size batch.

Due to the limitation of computational resource, we conduct the ablation study on a smaller pre-training corpus, i.e., WikiText-103 dataset (Merity et al., 2016). All the models listed in Table 3 are pre-trained for 500 epochs on 64 NVIDIA Tesla V100 32GB GPUs. Three of our proposed models are reported in the table. The general performance for the variants doesn't show much difference compared with the original RoBERTa-base, with a +0.4% increase on the average score on Double-batch RoBERTa-base, which confirms the idea that a larger batch benefits the representation training as proposed by previous work (Liu et al., 2019). Yet, the best-performed baseline is still not as good as our best-proposed

model. It tells us the proposed model does not solely benefit from a larger batch; CL loss also helps.

### 5.2 Different Augmentation Learns Different Features

In Table 1, we find an interesting phenomenon: different proposed models are good at specific tasks.

One example is MLM+subs+del-span helps the model be good at dealing with similarity and paraphrase tasks. On QQP and STS, it achieves the highest score; on MRPC, it ranks second. We infer the outperformance of MLM+subs+del-span in this kind of task is because synonym substitution helps translate the original sentence to similar meaning sentences while deleting different spans makes more variety of similar sentences visible. Combining them enhances the model's capacity to deal with many unseen sentence pairs.

We also notice that MLM+del-span achieves good performance on inference tasks (MNLI, QNLI, RTE). The underlying reason is, with a span deletion, the model has already been pre-trained well to infer the other similar sentences. The ability to identify similar sentence pairs helps to recognize the contradiction. Therefore, the gap between the pre-trained task and this downstream task narrows.

Overall, we observe that different augmentation learns different features. Some specific augmentations are especially good at some certain downstream tasks. Designing a task-specific augmentation or exploring meta-learning to adaptively select different CL objectives is a promising future direction.



## 6 Conclusion

In this work, we presented an instantiation for contrastive sentence representation learning. By carefully designing and testing different data augmentations and combinations, we prove the proposed methods' effectiveness on GLUE and SentEval benchmark under the diverse pre-training corpus.

The experiment results indicate that the pre-trained model would be more robust when leveraging adequate sentence-level supervision. More importantly, we reveal that different augmentation learns different features for the model. Finally, we demonstrate that the performance improvement comes from both the larger batch size and the contrastive loss.

## References

- Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.
- Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.
- Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.
- 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.
- 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*.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.
- Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pages 1681–1691.
- Paras Jain, Ajay Jain, Tianjun Zhang, Pieter Abbeel, Joseph E Gonzalez, and Ion Stoica. 2020. Contrastive code representation learning. *arXiv preprint arXiv:2007.04973*.
- Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. *arXiv preprint arXiv:1909.00986*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.
- Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717.

- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. URL [https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/language\\_understanding\\_paper.pdf](https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/language_understanding_paper.pdf).
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. *arXiv preprint arXiv:1805.09843*.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.
- Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*.
- 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*, pages 5998–6008.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.
- Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.
- Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. 2019. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6002–6012.