# ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer

**Yuanmeng Yan**[1]*, **Rumei Li**[2]*, **Sirui Wang**[2], **Fuzheng Zhang**[2], **Wei Wu**[2], **Weiran Xu**[1]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan Inc., Beijing, China
{yanyuanmeng,xuweiran}@bupt.edu.cn
{lirumei,wangsirui,zhangfuzheng,wuwei30}@meituan.com

## Abstract

Learning high-quality sentence representations benefits a wide range of natural language processing tasks. Though BERT-based pre-trained language models achieve high performance on many downstream tasks, the native derived sentence representations are proved to be collapsed and thus produce a poor performance on the semantic textual similarity (STS) tasks. In this paper, we present ConSERT, a **Con**trastive Framework for Self-Supervised **SE**ntence **R**epresentation **T**ransfer, that adopts contrastive learning to fine-tune BERT in an unsupervised and effective way. By making use of unlabeled texts, ConSERT solves the collapse issue of BERT-derived sentence representations and make them more applicable for downstream tasks. Experiments on STS datasets demonstrate that ConSERT achieves an 8% relative improvement over the previous state-of-the-art, even comparable to the supervised SBERT-NLI. And when further incorporating NLI supervision, we achieve new state-of-the-art performance on STS tasks. Moreover, ConSERT obtains comparable results with only 1000 samples available, showing its robustness in data scarcity scenarios.

## 1 Introduction

Sentence representation learning plays a vital role in natural language processing tasks (Kiros et al., 2015; Hill et al., 2016; Conneau et al., 2017; Cer et al., 2018). Good sentence representations benefit a wide range of downstream tasks, especially for computationally expensive ones, including large-scale semantic similarity comparison and information retrieval.

Recently, BERT-based pre-trained language models have achieved high performance on many downstream tasks with additional supervision. However, the native sentence representations derived from BERT[1] are proved to be of low-quality (Reimers and Gurevych, 2019; Li et al., 2020). As shown in Figure 1a, when directly adopt BERT-based sentence representations to semantic textual similarity (STS) tasks, almost all pairs of sentences achieved a similarity score between 0.6 to 1.0 , even if some pairs are regarded as completely unrelated by the human annotators. In other words, the BERT-derived native sentence representations are somehow collapsed (Chen and He, 2020), which means almost all sentences are mapped into a small area and therefore produce high similarity.

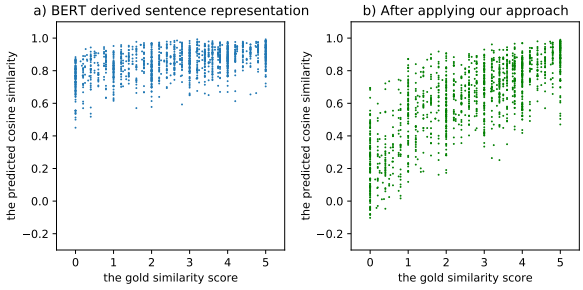


Figure 1: The correlation diagram between the gold similarity score (x-axis) and the model predicted cosine similarity score (y-axis) on the STS benchmark dataset.

Such phenomenon is also observed in several previous works (Gao et al., 2019; Wang et al., 2019; Li et al., 2020). They find the word representation space of BERT is anisotropic, the high-frequency words are clustered and close to the origin, while low-frequency words disperse sparsely. When averaging token embeddings, those high-frequency words dominate the sentence representations, inducing biases against their real semantics [2]. As a

*Work done during internship at Meituan Inc. The first two authors contribute equally. Weiran Xu is the corresponding author.

[1]Typically, we take the output of the [CLS] token or average token embeddings at the last few layers as the sentence representations.

[2]We also empirically prove this hypothesis, please refer to Section 5.1 for more details.

result, it is inappropriate to directly apply BERT's native sentence representations for semantic matching or text retrieval. Traditional methods usually fine-tune BERT with additional supervision. However, human annotation is costly and often unavailable in real-world scenarios.

To alleviate the collapse issue of BERT as well as reduce the requirement for labeled data, we propose a novel sentence-level training objective based on contrastive learning (He et al., 2020; Chen et al., 2020a,b). By encouraging two augmented views from the same sentence to be closer while keeping views from other sentences away, we reshape the BERT-derived sentence representation space and successfully solve the collapse issue (shown in Figure 1b). Moreover, we propose multiple data augmentation strategies for contrastive learning, including adversarial attack (Goodfellow et al., 2014; Kurakin et al., 2016), token shuffling, cutoff (Shen et al., 2020) and dropout (Hinton et al., 2012), that effectively transfer the sentence representations to downstream tasks. We name our approach ConSERT, a **Con**trastive Framework for **SE**ntence **R**epresentation **T**ransfer.

ConSERT has several advantages over previous approaches. Firstly, it introduces no extra structure or specialized implementation during inference. The parameter size of ConSERT keeps the same as BERT, making it easy to use. Secondly, compared with pre-training approaches, ConSERT is more efficient. With only 1,000 unlabeled texts drawn from the target distribution (which is easy to collect in real-world applications), we achieve 35% relative performance gain over BERT, and the training stage takes only a few minutes (1-2k steps) on a single V100 GPU. Finally, it includes several effective and convenient data augmentation methods with minimal semantic impact. Their effects are validated and analyzed in the ablation studies.

Our contributions can be summarized as follows: 1) We propose a simple but effective sentence-level training objective based on contrastive learning. It mitigates the collapse of BERT-derived representations and transfers them to downstream tasks. 2) We explore various effective text augmentation strategies to generate views for contrastive learning and analyze their effects on unsupervised sentence representation transfer. 3) With only fine-tuning on unsupervised target datasets, our approach achieves significant improvement on STS tasks. When further incorporating with NLI supervision, our approach achieves new state-of-the-art performance. We also show the robustness of our approach in data scarcity scenarios and intuitive analysis of the transferred representations.[3]

## 2 Related Work

### 2.1 Sentence Representation Learning

**Supervised Approaches** Several works use supervised datasets for sentence representation learning. Conneau et al. (2017) finds the supervised Natural Language Inference (NLI) task is useful to train good sentence representations. They use a BiLSTM-based encoder and train it on two NLI datasets, Stanford NLI (SNLI) (Bowman et al., 2015) and Multi-Genre NLI (MNLI) (Williams et al., 2018). Universal Sentence Encoder (Cer et al., 2018) adopts a Transformer-based architecture and uses the SNLI dataset to augment the unsupervised training. SBERT (Reimers and Gurevych, 2019) proposes a siamese architecture with a shared BERT encoder and is also trained on SNLI and MNLI datasets.

**Self-supervised Objectives for Pre-training** BERT (Devlin et al., 2019) proposes a bi-directional Transformer encoder for language model pre-training. It includes a sentence-level training objective, namely next sentence prediction (NSP), which predicts whether two sentences are adjacent or not. However, NSP is proved to be weak and has little contribution to the final performance (Liu et al., 2019). After that, various self-supervised objectives are proposed for pre-training BERT-like sentence encoders. Cross-Thought (Wang et al., 2020) and CMLM (Yang et al., 2020) are two similar objectives that recover masked tokens in one sentence conditioned on the representations of its contextual sentences. SLM (Lee et al., 2020) proposes an objective that reconstructs the correct sentence ordering given the shuffled sentences as the input. However, all these objectives need document-level corpus and are thus not applicable to downstream tasks with only short texts.

**Unsupervised Approaches** BERT-flow (Li et al., 2020) proposes a flow-based approach that maps BERT embeddings to a standard Gaussian latent space, where embeddings are more suitable for comparison. However, this approach introduces

[3]Our code is available at https://github.com/yym6472/ConSERT.

extra model structures and need specialized implementation, which may limit its application.

### 2.2 Contrastive Learning

**Contrastive Learning for Visual Representation Learning** Recently, contrastive learning has become a very popular technique in unsupervised visual representation learning with solid performance (Chen et al., 2020a; He et al., 2020; Chen et al., 2020b). They believe that good representation should be able to identify the same object while distinguishing itself from other objects. Based on this intuition, they apply image transformations (e.g. cropping, rotation, cutout, etc.) to randomly generate two augmented versions for each image and make them close in the representation space. Such approaches can be regarded as the invariance modeling to the input samples. Chen et al. (2020a) proposes SimCLR, a simple framework for contrastive learning. They use the normalized temperature-scaled cross-entropy loss (NT-Xent) as the training loss, which is also called InfoNCE in the previous literature (Hjelm et al., 2018).

**Contrastive Learning for Textual Representation Learning** Recently, contrastive learning has been widely applied in NLP tasks. Many works use it for language model pre-training. IS-BERT (Zhang et al., 2020) proposes to add 1-D convolutional neural network (CNN) layers on top of BERT and train the CNNs by maximizing the mutual information (MI) between the global sentence embedding and its corresponding local contexts embeddings. CERT (Fang and Xie, 2020) adopts a similar structure as MoCo (He et al., 2020) and uses back-translation for data augmentation. However, the momentum encoder needs extra memory and back-translation may produce false positives. BERT-CT (Carlsson et al., 2021) uses two individual encoders for contrastive learning, which also needs extra memory. Besides, they only sample 7 negatives, resulting in low training efficiency. DeCLUTR (Giorgi et al., 2020) adopts the architecture of SimCLR and jointly trains the model with contrastive objective and masked language model objective. However, they only use spans for contrastive learning, which is fragmented in semantics. CLEAR (Wu et al., 2020) uses the same architecture and objectives as DeCLUTR. Both of them are used to pre-train the language model, which needs a large corpus and takes a lot of resources.

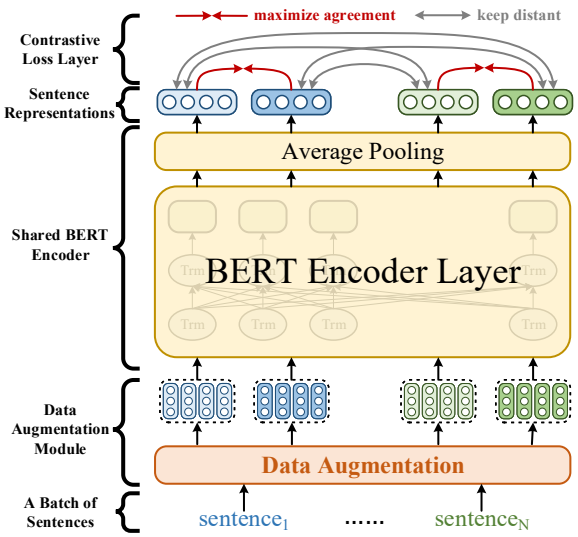


Figure 2: The general framework of our proposed approach.

## 3 Approach

In this section, we present ConSERT for sentence representation transfer. Given a BERT-like pre-trained language model $\mathbf{M}$ and an unsupervised dataset $\mathcal{D}$ drawn from the target distribution, we aim at fine-tuning $\mathbf{M}$ on $\mathcal{D}$ to make the sentence representation more task-relevant and applicable to downstream tasks. We first present the general framework of our approach, then we introduce several data augmentation strategies for contrastive learning. Finally, we talk about three ways to further incorporate supervision signals.

### 3.1 General Framework

Our approach is mainly inspired by SimCLR (Chen et al., 2020a). As shown in Figure 2, there are three major components in our framework:

- A data augmentation module that generates different views for input samples at the token embedding layer.
- A shared BERT encoder that computes sentence representations for each input text. During training, we use the average pooling of the token embeddings at the last layer to obtain sentence representations.
- A contrastive loss layer on top of the BERT encoder. It maximizes the agreement between one representation and its corresponding version that is augmented from the same sentence while keeping it distant from other sentence representations in the same batch.

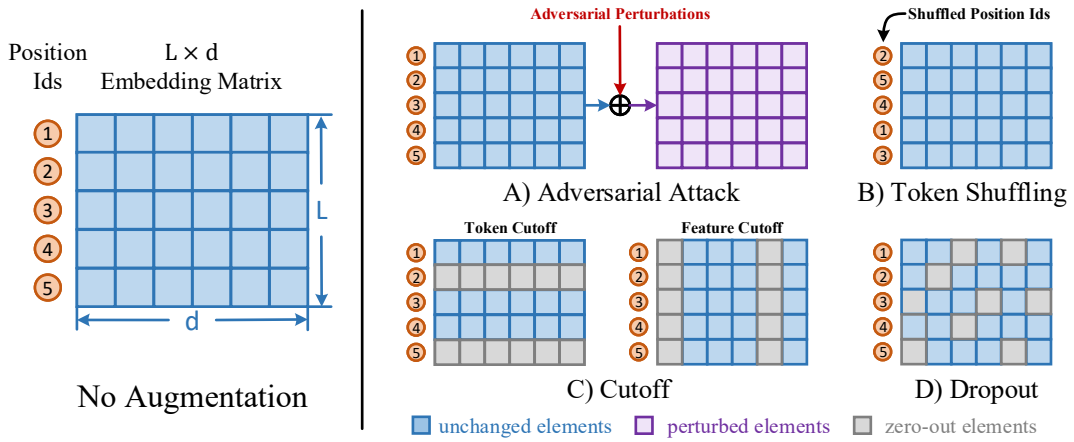


Figure 3: The four data augmentation strategies used in our experiments.

For each input text $x$, we first pass it to the data augmentation module, in which two transformations $T_1$ and $T_2$ are applied to generate two versions of token embeddings: $\boldsymbol{e}_i = T_1(x), \boldsymbol{e}_j = T_2(x)$, where $\boldsymbol{e}_i, \boldsymbol{e}_j \in \mathbb{R}^{L \times d}$, $L$ is the sequence length and $d$ is the hidden dimension. After that, both $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ will be encoded by multi-layer transformer blocks in BERT and produce the sentence representations $r_i$ and $r_j$ through average pooling.

Following Chen et al. (2020a), we adopt the normalized temperature-scaled cross-entropy loss (NT-Xent) as the contrastive objective. During each training step, we randomly sample $N$ texts from $\mathcal{D}$ to construct a mini-batch, resulting in $2N$ representations after augmentation. Each data point is trained to find out its counterpart among $2(N-1)$ in-batch negative samples:

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(r_i, r_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(r_i, r_k)/\tau)} \quad (1)$$

, where $\text{sim}(\cdot)$ indicates the cosine similarity function, $\tau$ controls the temperature and $\mathbb{1}$ is the indicator. Finally, we average all $2N$ in-batch classification losses to obtain the final contrastive loss $\mathcal{L}_{\text{con}}$.

### 3.2 Data Augmentation Strategies

We explore four different data augmentation strategies to generate views for contrastive learning, including adversarial attack (Goodfellow et al., 2014; Kurakin et al., 2016), token shuffling, cutoff (Shen et al., 2020) and dropout (Hinton et al., 2012), as illustrated in Figure 3.

**Adversarial Attack** Adversarial training is generally used to improve the model's robustness. They generate adversarial samples by adding a worst-case perturbation to the input sample. We implement this strategy with Fast Gradient Value (FGV) (Rozsa et al., 2016), which directly uses the gradient to compute the perturbation and thus is faster than two-step alternative methods. Note that this strategy is only applicable when jointly training with supervision since it relies on supervised loss to compute adversarial perturbations.

**Token Shuffling** In this strategy, we aim to randomly shuffle the order of the tokens in the input sequences. Since the bag-of-words nature in the transformer architecture, the position encoding is the only factor about the sequential information. Thus, similar to Lee et al. (2020), we implement this strategy by passing the shuffled position ids to the embedding layer while keeping the order of the token ids unchanged.

**Cutoff** Shen et al. (2020) proposes a simple and efficient data augmentation strategy called cutoff. They randomly erase some tokens (for token cutoff), feature dimensions (for feature cutoff), or token spans (for span cutoff) in the $L \times d$ feature matrix. In our experiments, we only use token cutoff and feature cutoff and apply them to the token embeddings for view generation.

**Dropout** Dropout is a widely used regularization method that avoids overfitting. However, in our experiments, we also show its effectiveness as an augmentation strategy for contrastive learning. For this setting, we randomly drop elements in the token embedding layer by a specific probability and set their values to zero. Note that this strategy is different from Cutoff since each element is considered individually.

| | STS12 | STS13 | STS14 | STS15 | STS16 | STSb | SICK-R | Total |
|---|---|---|---|---|---|---|---|---|
| Number of train samples | 0 | 0 | 0 | 0 | 0 | 5749 | 4500 | - |
| Number of valid samples | 0 | 0 | 0 | 0 | 0 | 1500 | 500 | - |
| Number of test samples | 3108 | 1500 | 3750 | 3000 | 1186 | 1379 | 4927 | - |
| Number of Unlabeled Texts | 6216 | 3000 | 7500 | 17000 | 18366 | 17256 | 19854 | 89192 |

Table 1: The statistics of STS datasets.

### 3.3 Incorporating Supervision Signals

Besides unsupervised transfer, our approach can also be incorporated with supervised learning. We take the NLI supervision as an example. It is a sentence pair classification task, where the model are trained to distinguish the relation between two sentences among *contradiction*, *entailment* and *neutral*. The classification objective can be expressed as following:

$$\begin{aligned} f &= \text{Concat}(r_1, r_2, |r_1 - r_2|) \\ \mathcal{L}_{\text{ce}} &= \text{CrossEntropy}(Wf + b, y) \end{aligned} \tag{2}$$

, where $r_1$ and $r_2$ denote two sentence representations.

We propose three ways for incorporating additional supervised signals:

- **Joint training (joint)** We jointly train the model with the supervised and unsupervised objectives $\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{ce}} + \alpha\mathcal{L}_{\text{con}}$ on NLI dataset. $\alpha$ is a hyper-parameter to balance two objectives.

- **Supervised training then unsupervised transfer (sup-unsup)** We first train the model with $\mathcal{L}_{\text{ce}}$ on NLI dataset, then use $\mathcal{L}_{\text{con}}$ to fine-tune it on the target dataset.

- **Joint training then unsupervised transfer (joint-unsup)** We first train the model with the $\mathcal{L}_{\text{joint}}$ on NLI dataset, then use $\mathcal{L}_{\text{con}}$ to fine-tune it on the target dataset.

## 4 Experiments

To verify the effectiveness of our proposed approach, we conduct experiments on Semantic Textual Similarity (STS) tasks under the unsupervised and supervised settings.

### 4.1 Setups

**Dataset** Following previous works(Reimers and Gurevych, 2019; Li et al., 2020; Zhang et al., 2020), we evaluate our approach on multiple STS datasets, including STS tasks 2012 - 2016 (STS12 - STS16) (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS benchmark (STSb) (Cer et al., 2017) and SICK-Relatedness (SICK-R) (Marelli et al.). Each sample in these datasets contains a pair of sentences as well as a gold score between 0 and 5 to indicate their semantic similarity. For our unsupervised experiments, we mix the unlabeled texts from these datasets to fine-tune our model. We obtain all 7 datasets through the SentEval toolkit (Conneau and Kiela, 2018). The statistics is shown in Table 1.

For supervised experiments, we use the combination of SNLI (570k samples) (Bowman et al., 2015) and MNLI (430k samples) (Williams et al., 2018) to train our model. In the *joint training* setting, the NLI texts are also used for contrastive objectives.

**Baselines** To show our effectiveness on unsupervised sentence representation transfer, we mainly select BERT-flow (Li et al., 2020) for comparison, since it shares the same setting as our approach. For unsupervised comparison, we use the average of GloVe embeddings, the BERT-derived native embeddings, CLEAR (Wu et al., 2020) (trained on BookCorpus and English Wikipedia corpus), IS-BERT (Zhang et al., 2020) (trained on unlabeled texts from NLI datasets), BERT-CT (Carlsson et al., 2021) (trained on English Wikipedia corpus). For comparison with supervised methods, we select InferSent (Conneau et al., 2017), Universal Sentence Encoder (Cer et al., 2018), SBERT (Reimers and Gurevych, 2019) and BERT-CT (Carlsson et al., 2021) as baselines. They are all trained with NLI supervision.

**Evaluation** When evaluating the trained model, we first obtain the representation of sentences by averaging the token embeddings at the last two layers[4], then we report the spearman correlation between the cosine similarity scores of sentence representations and the human-annotated gold scores. When calculating spearman correlation, we merge all sentences together (even if some STS datasets have multiple splits) and calculate spearman correlation for only once[5].

[4]As shown in Li et al. (2020), averaging the last two layers of BERT achieves slightly better results than averaging the last one layer.

[5]Note that such evaluation procedure is different from

| Method | STS12 | STS13 | STS14 | STS15 | STS16 | STSb | SICK-R | Avg. |
|---|---|---|---|---|---|---|---|---|
| *Unsupervised baselines* | | | | | | | | |
| Avg. GloVe embeddings† | 55.14 | 70.66 | 59.73 | 68.25 | 63.66 | 58.02 | 53.76 | 61.32 |
| $\text{BERT}_{\text{base}}$‡ | 35.20 | 59.53 | 49.37 | 63.39 | 62.73 | 48.18 | 58.60 | 53.86 |
| $\text{BERT}_{\text{large}}$‡ | 33.06 | 57.64 | 47.95 | 55.83 | 62.42 | 49.66 | 53.87 | 51.49 |
| $\text{CLEAR}_{\text{base}}$† | 49.0 | 48.9 | 57.4 | 63.6 | 65.6 | 75.6 | 72.5 | 61.8 |
| $\text{IS-BERT}_{\text{base}}$-NLI† | 56.77 | 69.24 | 61.21 | 75.23 | 70.16 | 69.21 | 64.25 | 66.58 |
| $\text{BERT}_{\text{base}}$-CT† | 66.86 | 70.91 | 72.37 | 78.55 | 77.78 | - | - | - |
| $\text{BERT}_{\text{large}}$-CT† | 69.50 | 75.97 | 74.22 | 78.83 | 78.92 | - | - | - |
| *Using STS unlabeled texts* | | | | | | | | |
| $\text{BERT}_{\text{base}}$-flow† | 63.48 | 72.14 | 68.42 | 73.77 | 75.37 | 70.72 | 63.11 | 69.57 |
| $\text{BERT}_{\text{large}}$-flow† | 65.20 | 73.39 | 69.42 | 74.92 | **77.63** | 72.26 | 62.50 | 70.76 |
| $\text{ConSERT}_{\text{base}}$‡ | 64.64 | 78.49 | 69.07 | 79.72 | 75.95 | 73.97 | 67.31 | 72.74 |
| $\text{ConSERT}_{\text{large}}$‡ | **70.69** | **82.96** | **74.13** | **82.78** | 76.66 | **77.53** | **70.37** | **76.45** |

Table 2: The performance comparison of ConSERT with other methods in an *unsupervised* setting. We report the spearman correlation $\rho \times 100$ on 7 STS datasets. Methods with † indicate that we directly report the scores from the corresponding paper, while methods with ‡ indicate our implementation.

**Implementation Details** Our implementation is based on the Sentence-BERT[6] (Reimers and Gurevych, 2019). We use both the BERT-base and BERT-large for our experiments. The max sequence length is set to 64 and we remove the default dropout layer in BERT architecture considering the *cutoff* and *dropout* data augmentation strategies used in our framework. The ratio of token cutoff and feature cutoff is set to 0.15 and 0.2 respectively, as suggested in Shen et al. (2020). The ratio of dropout is set to 0.2. The temperature $\tau$ of NT-Xent loss is set to 0.1, and the $\alpha$ is set to 0.15 for the joint training setting. We adopt Adam optimizer and set the learning rate to 5e-7. We use a linear learning rate warm-up over 10% of the training steps. The batch size is set to 96 in most of our experiments. We use the dev set of STSb to tune the hyperparameters (including the augmentation strategies) and evaluate the model every 200 steps during training. The best checkpoint on the dev set of STSb is saved for test. We further discuss the influence of the batch size and the temperature in the subsequent section.

### 4.2 Unsupervised Results

For unsupervised evaluation, we load the pre-trained BERT to initialize the BERT encoder in our framework. Then we randomly mix the unlabeled texts from 7 STS datasets and use them to fine-tune our model.

The results are shown in Table 2. We can observe that both BERT-flow and ConSERT can improve the representation space and outperform the GloVe and BERT baselines with unlabeled texts from target datasets. However, $\text{ConSERT}_{\text{large}}$ achieves the best performance among 6 STS datasets, significantly outperforming $\text{BERT}_{\text{large}}$-flow with an 8% relative performance gain on average (from 70.76 to 76.45). Moreover, it is worth noting that $\text{ConSERT}_{\text{large}}$ even outperforms several supervised baselines (see Figure 3) like InferSent (65.01) and Universal Sentence Encoder (71.72), and keeps comparable to the strong supervised method $\text{SBERT}_{\text{large}}$-NLI (76.55). For the $\text{BERT}_{\text{base}}$ architecture, our approach $\text{ConSERT}_{\text{base}}$ also outperforms $\text{BERT}_{\text{base}}$-flow with an improvement of 3.17 (from 69.57 to 72.74).

### 4.3 Supervised Results

For supervised evaluation, we consider the three settings described in Section 3.3. Note that in the *joint* setting, only NLI texts are used for contrastive learning, making it comparable to SBERT-NLI. We use the model trained under the *joint* setting as the initial checkpoint in the *joint-unsup* setting. We also re-implement the SBERT-NLI baselines and use them as the initial checkpoint in the *sup-unsup* setting.

The results are illustrated in Table 3. For the models trained with NLI supervision, we find that ConSERT *joint* consistently performs better than SBERT, revealing the effectiveness of our proposed contrastive objective as well as the data augmentation strategies. On average, $\text{ConSERT}_{\text{base}}$ *joint*

SentEval toolkit, which calculates spearman correlation for each split and reports the mean or weighted mean scores.

[6] https://github.com/UKPLab/sentence-transformers

| Method | STS12 | STS13 | STS14 | STS15 | STS16 | STSb | SICK-R | Avg. |
|---|---|---|---|---|---|---|---|---|
| *Using NLI supervision* | | | | | | | | |
| InferSent - GloVe† | 52.86 | 66.75 | 62.15 | 72.77 | 66.87 | 68.03 | 65.65 | 65.01 |
| Universal Sentence Encoder† | 64.49 | 67.80 | 64.61 | 76.83 | 73.18 | 74.92 | 76.69 | 71.22 |
| $\text{SBERT}_{\text{base}}$-NLI† | 70.97 | 76.53 | 73.19 | 79.09 | 74.30 | 77.03 | 72.91 | 74.89 |
| $\text{SBERT}_{\text{large}}$-NLI† | 72.27 | 78.46 | 74.90 | 80.99 | 76.25 | 79.23 | 73.75 | 76.55 |
| $\text{SBERT}_{\text{base}}$-NLI (re-impl.)‡ | 69.89 | 75.77 | 72.36 | 78.51 | 73.67 | 76.75 | 72.76 | 74.24 |
| $\text{SBERT}_{\text{large}}$-NLI (re-impl.)‡ | 72.69 | 78.77 | 75.13 | 80.95 | 76.89 | 79.53 | 73.25 | 76.74 |
| $\text{BERT}_{\text{base}}$-CT† | 68.80 | 74.58 | 76.62 | 79.72 | 77.14 | - | - | - |
| $\text{BERT}_{\text{large}}$-CT† | 69.80 | 75.45 | 76.47 | 81.34 | 78.11 | - | - | - |
| $\text{ConSERT}_{\text{base}}$ *joint*‡ | 70.53 | 79.96 | 74.85 | 81.45 | 76.72 | 78.82 | 77.53 | 77.12 |
| $\text{ConSERT}_{\text{large}}$ *joint*‡ | **73.26** | **82.36** | **77.73** | **83.84** | **78.75** | **81.54** | **78.64** | **79.44** |
| *Using NLI supervision and STS unlabeled texts* | | | | | | | | |
| $\text{BERT}_{\text{base}}$-flow† | 68.95 | 78.48 | 77.62 | 81.95 | 78.94 | 81.03 | 74.97 | 77.42 |
| $\text{BERT}_{\text{large}}$-flow† | 70.19 | 80.27 | 78.85 | 82.97 | **80.57** | 81.18 | 74.52 | 78.36 |
| $\text{ConSERT}_{\text{base}}$ *sup-unsup*‡ | 73.51 | 84.86 | 77.44 | 83.11 | 77.98 | 81.80 | 74.29 | 79.00 |
| $\text{ConSERT}_{\text{large}}$ *sup-unsup*‡ | 75.26 | **86.01** | 79.00 | 83.88 | 79.45 | 82.95 | 76.54 | 80.44 |
| $\text{ConSERT}_{\text{base}}$ *joint-unsup*‡ | 74.07 | 83.93 | 77.05 | 83.66 | 78.76 | 81.36 | 76.77 | 79.37 |
| $\text{ConSERT}_{\text{large}}$ *joint-unsup*‡ | **77.47** | 85.45 | **79.41** | **85.59** | 80.39 | **83.42** | **77.26** | **81.28** |

Table 3: The performance comparison of ConSERT with other methods in a *supervised* setting. We report the spearman correlation $\rho \times 100$ on 7 STS datasets. Methods with † indicate that we directly report the scores from the corresponding paper, while methods with ‡ indicate our implementation.

achieves a performance gain of 2.88 over the re-implemented $\text{SBERT}_{\text{base}}$-NLI, and $\text{ConSERT}_{\text{large}}$ *joint* achieves a performance gain of 2.70.

When further performing representation transfer with STS unlabeled texts, our approach achieves even better performance. On average, $\text{ConSERT}_{\text{large}}$ *joint-unsup* outperforms the initial checkpoint $\text{ConSERT}_{\text{large}}$ *joint* with 1.84 performance gain, and outperforms the previous state-of-the-art $\text{BERT}_{\text{large}}$-flow with 2.92 performance gain. The results demonstrate that even for the models trained under supervision, there is still a huge potential of unsupervised representation transfer for improvement.

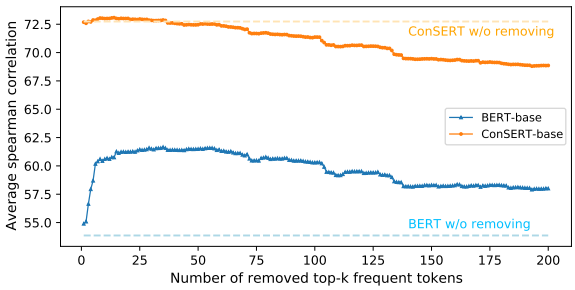


Figure 4: The average spearman correlation on STS tasks w.r.t. the number of removed top-k frequent tokens. Note that we also considered the [CLS] and [SEP] tokens and they are the 2 most frequent tokens. The frequency of each token is calculated through the test split of the STS Benchmark dataset.

## 5 Qualitative Analysis

### 5.1 Analysis of BERT Embedding Space

To prove the hypothesis that the collapse issue is mainly due to the anisotropic space that is sensitive to the token frequency, we conduct experiments that mask the embeddings of several most frequent tokens when applying average pooling to calculate the sentence representations. The relation between the number of removed top-k frequent tokens and the average spearman correlation is shown in Figure 4.

We can observe that when removing a few top frequent tokens, the performance of BERT improves sharply on STS tasks. When removing 34 most frequent tokens, the best performance is achieved (61.66), and there is an improvement of 7.8 from the original performance (53.86). For ConSERT, we find that removing a few most frequent tokens only results in a small improvement of less than 0.3. The results show that our approach reshapes the BERT's original embedding space, reducing the influence of common tokens on sentence representations.

### 5.2 Effect of Data Augmentation Strategy

In this section, we study the effect of data augmentation strategies for contrastive learning. We consider 5 options for each transformation, including None (i.e. doing nothing), Shuffle, Token Cutoff,

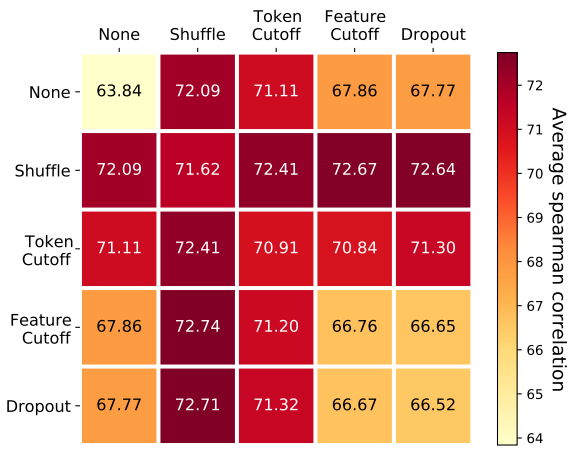


Figure 5: The performance visualization with different combinations of data augmentation strategies. The row indicates the 1st data augmentation strategy while the column indicates the 2nd data augmentation strategy.

Feature Cutoff, and Dropout, resulting in 5×5 combinations. Note that the Adversarial Attack strategy is not considered here, since it needs additional supervision to generate adversarial samples. All these experiments follow the unsupervised setting and use the $\text{BERT}_{\text{base}}$ architecture.

The results can be found in Figure 5. We can make the following observations. First, Shuffle and Token Cutoff are the two most effective strategies (where Shuffle is slightly better than Token Cutoff), significantly outperforming Feature Cutoff and Dropout. This is probably because Shuffle and Token Cutoff are more related to the downstream STS tasks since they are directly operated on the token level and change the structure of the sentence to produce hard examples.

Secondly, Feature Cutoff and Dropout also improve performance by roughly 4 points when compared with the None-None baseline. Moreover, we find they work well as a complementary strategy. Combining with another strategy like Shuffle may further improve the performance. When combined Shuffle with Feature Cutoff, we achieve the best result. We argue that Feature Cutoff and Dropout are useful in modeling the invariance of the internal noise for the sentence encoder, and thus improve the model's robustness.

Finally, we also observe that even without any data augmentation (the None-None combination), our contrastive framework can improve BERT's performance on STS tasks (from 53.86 to 63.84). This None-None combination has no effect on maximizing agreement between views since the representations of augmented views are exactly the same. On the contrary, it tunes the representation space by pushing each representation away from others. We believe that the improvement is mainly due to the collapse phenomenon of BERT's native representation space. To some extent, it also explains why our method works.

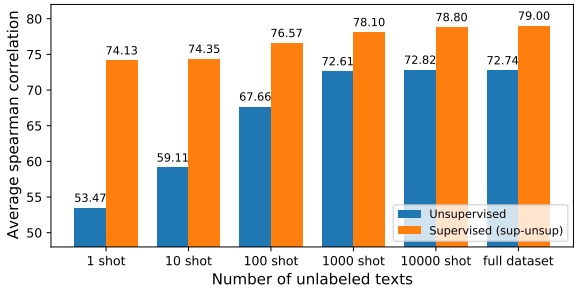


Figure 6: The few-shot experiments under the unsupervised and supervised settings. We report the average spearman correlation on STS datasets with 1, 10, 100, 1,000, and 10,000 unlabeled texts available, respectively. The full dataset indicates all 89192 unlabeled texts from 7 STS datasets.

### 5.3 Performance under Few-shot Settings

To validate the reliability and the robustness of ConSERT under the data scarcity scenarios, we conduct the few-shot experiments. We limit the number of unlabeled texts to 1, 10, 100, 1000, and 10000 respectively, and compare their performance with the full dataset.

Figure 6 presents the results. For both the unsupervised and the supervised settings, our approach can make a huge improvement over the baseline with only 100 samples available. When the training samples increase to 1000, our approach can basically achieve comparable results with the models trained on the full dataset. The results reveal the robustness and effectiveness of our approach under the data scarcity scenarios, which is common in reality. With only a small amount of unlabeled texts drawn from the target data distribution, our approach can also tune the representation space and benefit the downstream tasks.

### 5.4 Influence of Temperature

The temperature $\tau$ in NT-Xent loss (Equation 1) is used to control the smoothness of the distribution normalized by softmax operation and thus influences the gradients when backpropagation. A large temperature smooths the distribution while a small temperature sharpens the distribution. In our experiments, we explore the influence of temperature

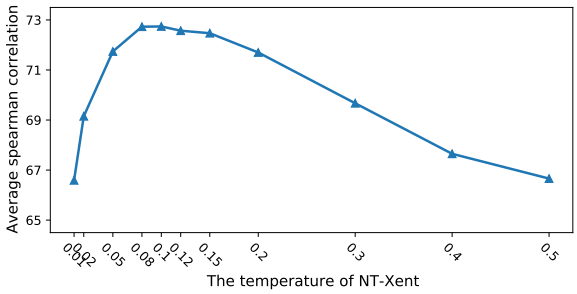


Figure 7: The influence of different temperatures in NT-Xent. The best performance is achieved when the temperature is set to 0.1.

| Batch Size | 16 | 48 | 96 | 192 | 288 |
|---|---|---|---|---|---|
| Avg. Spearman | 72.63 | 72.60 | 72.74 | 72.86 | 72.98 |
| Number of Steps | 6175 | 2459 | 1530 | 930 | 620 |

Table 4: The average spearman correlation as well as the training steps of our unsupervised approach with different batch sizes.

and present the result in Figure 7.

As shown in the figure, we find the performance is extremely sensitive to the temperature. Either too small or too large temperature will make our model perform badly. And the optimal temperature is obtained within a small range (from about 0.08 to 0.12). This phenomenon again demonstrates the collapse issue of BERT embeddings, as most sentences are close to each other, a large temperature may make this task too hard to learn. We select 0.1 as the temperature in most of our experiments.

### 5.5 Influence of Batch Size

In some previous works of contrastive learning, it is reported that a large batch size benefits the final performance and accelerates the convergence of the model since it provides more in-batch negative samples for contrastive learning (Chen et al., 2020a). Those in-batch negative samples improve the training efficiency. We also analyze the influence of the batch size for unsupervised sentence representation transfer.

The results are illustrated in Table 4. We show both the spearman correlation and the corresponding training steps. We find that a larger batch size does achieve better performance. However, the improvement is not so significant. Meanwhile, a larger batch size does speed up the training process, but it also needs more GPU memories at the same time.

## 6 Conclusion

In this paper, we propose ConSERT, a self-supervised contrastive learning framework for transferring sentence representations to downstream tasks. The framework does not need extra structure and is easy to implement for any encoder. We demonstrate the effectiveness of our framework on various STS datasets, both our unsupervised and supervised methods achieve new state-of-the-art performance. Furthermore, few-shot experiments suggest that our framework is robust in the data scarcity scenarios. We also compare multiple combinations of data augmentation strategies and provide fine-grained analysis for interpreting how our approach works. We hope our work will provide a new perspective for future researches on sentence representation transfer.

## Acknowledgements

We thank Keqing He, Hongzhi Zhang and all anonymous reviewers for their helpful comments and suggestions. This work was partially supported by National Key R&D Program of China No. 2019YFF0303300 and Subject II No. 2019YFF0303302, DOCOMO Beijing Communications Laboratories Co., Ltd, MoE-CMCC "Artifical Intelligence" Project No. MCM20190701.

## Broader Impact

Sentence representation learning is a basic task in natural language processing and benefits many downstream tasks. This work proposes a contrastive learning based framework to solve the collapse issue of BERT and transfer BERT sentence representations to target data distribution. Our approach not only provides a new perspective about BERT's representation space, but is also useful in practical applications, especially for data scarcity scenarios. When applying our approach, the user should collect a few unlabeled texts from target data distribution and use our framework to fine-tune BERT encoder in a self-supervised manner. Since our approach is self-supervised, no bias will be introduced from human annotations. Moreover, our data augmentation strategies also have little probability to introduce extra biases since they are all based on random sampling. However, it is still possible to introduce data biases from the unlabeled texts. Therefore, users should pay special attention to ensure that the training data is ethical, unbiased, and closely related to downstream tasks.

## References


Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Fredrik Carlsson, Magnus Sahlgren, Evangelia Gogoulou, Amaru Cuba Gyllensten, and Erik Ylipää Hellqvist. 2021. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020a. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. 2020b. Big self-supervised models are strong semi-supervised learners.

Xinlei Chen and Kaiming He. 2020. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

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*.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*.

John M Giorgi, Osvald Nitski, Gary D. Bader, and Bo Wang. 2020. Declutr: Deep contrastive learning for unsupervised textual representations. *ArXiv*, abs/2006.03659.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

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.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. *Advances in neural information processing systems*, 28:3294–3302.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Haejun Lee, Drew A Hudson, Kangwook Lee, and Christopher D Manning. 2020. Slm: Learning a discourse language representation with sentence unshuffling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1551–1562.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. 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. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. A sick cure for the evaluation of compositional distributional semantic models.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983.

Andras Rozsa, Ethan M Rudd, and Terrance E Boult. 2016. Adversarial diversity and hard positive generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–32.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2019. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

Shuohang Wang, Yuwei Fang, Siqi Sun, Zhe Gan, Yu Cheng, Jingjing Liu, and Jing Jiang. 2020. Cross-thought for sentence encoder pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 412–421.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

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*.

Ziyi Yang, Yinfei Yang, Daniel Cer, Jax Law, and Eric Darve. 2020. Universal sentence representation learning with conditional masked language model. *arXiv preprint arXiv:2012.14388*.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610.