ESimCSE: Enhanced Sample Building Method for Contrastive Learning of Unsupervised Sentence Embedding

Xing Wu\textsuperscript{1,2,3}, Chaochen Gao\textsuperscript{1,2}, Liangjun Zang\textsuperscript{1}, Jizhong Han\textsuperscript{1}, Zhongyuan Wang\textsuperscript{3}, Songlin Hu\textsuperscript{1,2}

\textsuperscript{1}Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China \textsuperscript{2}School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China \textsuperscript{3}Kuaishou Technology, Beijing, China

\{gaochaochen,zangliangjun,hanjizhong,husonglin\}@iie.ac.cn
\{wuxing,wangzhongyuan\}@kuaishou.com

Abstract

Contrastive learning has been attracting much attention for learning unsupervised sentence embeddings. The current state-of-the-art unsupervised method is the unsupervised SimCSE (unsup-SimCSE). Unsup-SimCSE takes dropout as a minimal data augmentation method, and passes the same input sentence to a pre-trained Transformer encoder (with dropout turned on) twice to obtain the two corresponding embeddings to build a positive pair. As the length information of a sentence will generally be encoded into the sentence embeddings due to the usage of position embedding in Transformer, each positive pair in unsup-SimCSE actually contains the same length information. And thus unsup-SimCSE trained with these positive pairs is probably biased, which would tend to consider that sentences of the same or similar length are more similar in semantics. Through statistical observations, we find that unsup-SimCSE does have such a problem. To alleviate it, we apply a simple repetition operation to modify the input sentence, and then pass the input sentence and its modified counterpart to the pre-trained Transformer encoder, respectively, to get the positive pair. Additionally, we draw inspiration from the community of computer vision and introduce a momentum contrast, enlarging the number of negative pairs without additional calculations. The proposed two modifications are applied on positive and negative pairs separately, and build a new sentence embedding method, termed Enhanced Unsup-SimCSE (ESimCSE). We evaluate the proposed ESimCSE on several benchmark datasets w.r.t the semantic text similarity (STS) task. Experimental results show that ESimCSE outperforms the state-of-the-art unsup-SimCSE by an average Spearman correlation of 2.02% on BERT-base.

1 Introduction

The large-scale pre-trained language model (Devlin et al., 2018; Liu et al., 2019), represented by BERT, benefits many downstream supervised tasks through finetuning methods. However, when applying BERT’s native sentence embeddings directly for semantic similarity tasks without labeled data, the performance is hardly satisfactory (Gao et al., 2021; Yan et al., 2021). Recently, researchers have proposed using contrastive learning to learn better unsupervised sentence embeddings. Contrastive learning aims to learn effective sentence embeddings based on the assumption that effective sentence embeddings should bring similar sentences closer while pushing away dissimilar ones. It generally uses various data augmentation methods to randomly generate different views for each sentence, and assumes a sentence is semantically more similar to its augmented counterpart than any other sentence. The current state-of-the-art method is unsup-SimCSE (Gao et al., 2021), which generates the state-of-the-art unsupervised sentence embeddings and performs on par with previously supervised counterparts. Unsup-SimCSE implicitly hypothesizes dropout acts as a minimal data augmentation method. Specifically, unsup-SimCSE composes \( N \) sentences in a batch and feeds each sentence to the pre-trained BERT twice with two independently sampled dropout masks. Then the embeddings derived from the same sentence constitute a “positive pair”, while those derived from two different sentences constitute a “negative pair”.

Using dropout as a minimal data augmentation method is simple and effective, but there is a weak point. Pretrained language models are built on Transformer blocks, which will encode the length information of a sentence through position embeddings. And thus a positive pair derived from the same sentence would contain the same length in-
Figure 1: The schematic diagram of the ESimCSE method. Unlike the unsup-SimCSE, ESimCSE performs word repetition operations on the batch so that the lengths of positive pairs vary without changing the semantics of sentences. This mechanism weakens the same-length hint for the model when predicting positive pairs. In addition, ESimCSE also maintains several preceding mini-batches’ model outputs in a queue, termed momentum contrast, which can expand the negative pairs involved in loss calculation. This mechanism allows pairs to be compared more sufficiently in contrastive learning.

Formulation, while a negative pair derived from two different sentences generally would contain different length information. Therefore, positive pairs and negative pairs are different in the length information they contained, which can act as a feature to distinguish them. Specifically, due to such a difference, the semantic similarity model trained with these pairs can be biased, which probably considers that two sentences of the same or similar lengths are more similar in semantics.

To confirm the impact of the length difference, we evaluate on standard semantic textual similarity (STS) tasks with the unsup-SimCSE-BERT\textsubscript{base} model published by (Gao et al., 2021). We partition STS task datasets into groups based on the sentence pairs’ length difference, and calculate the corresponding semantic similarity with spearman correlation separately. As shown in Table 1, as the length difference increases, the performance of unsup-SimCSE gets worse. The performance of unsup-SimCSE on sentences with similar length (≤ 3) far exceeds the performance on sentences with a larger difference in length (> 3).

To alleviate this problem, we propose a simple but effective enhancement method to unsup-SimCSE. For each positive pair, we expect to change the length of a sentence without changing its semantic meaning. Existing methods to change the length of a sentence generally use random insertion and random deletion. However, inserting randomly selected words into a sentence may introduce extra noise, which will probably distort the meaning of the sentence; deleting keywords from a sentence will also change its semantics substantially. Therefore, we propose a safer method, termed “word repetition”, which randomly duplicates some words in a sentence. For example, as shown in Table 2, the original sentence is “I like this apple because it looks so fresh and I think it

| Dataset | length diff ≤ 3 | length diff > 3 |
|---------|----------------|-----------------|
| STS12   | 0.7298         | 0.6035          |
| STS13   | 0.8508         | 0.8396          |
| STS14   | 0.7971         | 0.6676          |
| STS15   | 0.8374         | 0.7603          |
| STS16   | 0.8134         | 0.7677          |
| STS-B   | 0.8148         | 0.6924          |

Table 1: The spearman correlation of sentence pairs with a length difference of ≤ 3 and > 3.
should be delicious.” Random insertion may generate “I don’t like this apple because but it looks so not fresh and I think it should be dog delicious.”, and random deletion may generate “I this apple because it looks so and I think it should be.”. Both deviate far from the meaning of the original sentence. On the contrary, the method of “word repetition” may get “I like like this apple because it looks so so fresh and and I think it should be delicious.”, or “I I like this apple apple because it looks looks so fresh fresh and I think it should be delicious.”. Both keep the meaning of the original sentence quite well.

Apart from the optimization above for positive pairs construction, we further explore how to optimize the construction of negative pairs. Since contrastive learning is carried out between positive pairs and negative pairs, theoretically more negative pairs can lead to better comparison between the pairs (Chen et al., 2020). And thus a potential optimization direction is to leverage more negative pairs, encouraging the model towards more refined learning. However, according to (Gao et al., 2021), a larger batch size is not always a better choice. For example, as show in Figure 2, for the unsup-SimCSE-BERT\textsubscript{base} model, the optimal batch size is 64, and other settings of the batch size will lower the performance. Therefore, we tend to figure out how to expand the negative pairs more effectively. In the community of computer vision, to alleviate the GPU memory limitation when expanding the batch size, a feasible way is to introduce the momentum contrast (He et al., 2020), which is also applied to natural language understanding (Fang et al., 2020). Momentum contrast allows us to reuse the encoded embeddings from the immediate preceding mini-batches to expand the negative pairs, by maintaining a queue: which always enqueue the sentence embeddings of the current mini-batches and meanwhile dequeue the “oldest” ones. As the enqueued sentence embeddings come from the preceding mini-batches, we keep a momentum-updated model by taking the moving-average of its parameters and use the momentum model to generate enqueued sentence embeddings. Note that, we turn off dropout when using the momentum encoder, which can narrow the gap between training and prediction.

The above two optimizations are proposed separately for building positive and negative pairs. We finally combine both with unsup-SimCSE, which is termed Enhanced SimCSE (ESimCSE). We illustrate the schematic diagram of ESimCSE in Figure 1. The proposed ESimCSE is evaluated on the semantic text similarity (STS) task with 7 STS-B test sets. Experimental results show that ESimCSE can substantially improve the similarity measuring performance in different model settings over the previous state-of-the-art unsup-SimCSE. Specifically, ESimCSE gains an average increase of Spearman’s correlation over unsup-SimCSE by +2.02% on BERT\textsubscript{base}, +0.90% on BERT\textsubscript{large}, +0.87% on RoBERTa\textsubscript{base}, +0.55% on RoBERTa\textsubscript{large}, respectively.

Our contributions can be summarized as follows:

- We observe that unsup-SimCSE constructs each positive pair with two sentences of the same length, which can bias the learning process. We propose a simple but effective “word repetition” method to alleviate the problem.
- We propose to use the momentum contrast method to increase the number of negative pairs involved in the loss calculation, which encourages the model towards more refined learning.
- We conduct extensive experiments on several benchmark datasets w.r.t semantic text similarity task. The experimental results well demonstrate that both proposed optimizations bring substantial improvements to unsup-SimCSE.

2 Background: Unsup-SimCSE

Given a set of paired sentences \( \{x_i, x_i^+\}_{i=1}^m \), where \( x_i \) and \( x_i^+ \) are semantically related and will be re-
we take sub-word repetition as an example. Given a sentence \( s \), after processing by a sub-word tokenizer, we get a sub-word sequence \( x = \{x_1, x_2, ..., x_N\} \), \( N \) being the length of sequence. We define the number of repeated tokens as

\[
dup_{\text{len}} \in [0, \max(2, \text{int}(\text{dup}_{\text{rate}} \times N))] \quad (4)
\]

where \( \text{dup}_{\text{rate}} \) is the maximal repetition rate, which is a hyperparameter. Then \( \text{dup}_{\text{len}} \) is a randomly sampled number in the set defined above, which will introduce more diversity when extending the sequence length. After \( \text{dup}_{\text{len}} \) is determined, we use uniform distribution to randomly select \( \text{dup}_{\text{set}} \) sub-words that need to be repeated from the sequence, which composes the \( \text{dup}_{\text{set}} \) as follows,

\[
\text{dup}_{\text{set}} = \text{uniform}([\text{range} = [1, N], \text{num} = \text{dup}_{\text{len}}]) \quad (5)
\]

For example, if the 1th sub-word is in \( \text{dup}_{\text{set}} \), then sequence \( x \) becomes \( x^+ = \{x_1, x_1, x_2, ..., x_N\} \). And different from unsup-SimCSE which passes \( x \) to the pre-trained BERT twice, E-SimCSE passes \( x \) and \( x^+ \) independently.

### 3.2 Momentum Contrast

The momentum contrast allows us to reuse the encoded sentence embeddings from the immediate preceding mini-batches by maintaining a queue of a fixed size. Specifically, the embeddings in the queue are progressively replaced. When the output sentence embeddings of the current mini-batch is enqueued, the “oldest” ones in the queue are removed if the queue is full. Note that we use a momentum-updated encoder to encode the enqueued sentence embeddings. Formally, denoting

| Method                  | Text                                                                 | Similarity |
|-------------------------|----------------------------------------------------------------------|------------|
| original sentence       | I like this apple because it looks so fresh and I think it should be delicious. | 1.0        |
| random insertion        | I don’t like this apple because but it looks so not fresh and I think it should be dog delicious. | 0.76       |
| random deletion         | I like this apple because it looks so fresh and I think it should be delicious. | 0.77       |
| word repetition         | I like like this apple because it looks so fresh and and I think it should be delicious. | 1.0        |
| word repetition         | I I like this apple apple because it looks looks so fresh fresh and I think it should be delicious delicious. | 0.98       |

Table 2: An example of different methods to change the length of a sentence. The similarity scores are predicted by official released “unsup-simcse-bert-base-uncased” model.
Table 3: Data statistics of standard semantic textual similarity (STS) tasks.

|       | STS12 | STS13 | STS14 | SICK15 | STS16 | STS-B | SICK-R |
|-------|-------|-------|-------|--------|-------|-------|--------|
| train | 0     | 0     | 0     | 0      | 0     | 5,749 | 4,500  |
| dev   | 0     | 0     | 0     | 0      | 0     | 1,500 | 500    |
| test  | 3,108 | 1,500 | 3,750 | 3,000  | 1,186 | 1,379 | 4,927  |

the parameters of the encoder as $\theta_e$ and those of the momentum-updated encoder as $\theta_m$, we update $\theta_m$ in the following way,

$$\theta_m \leftarrow \lambda \theta_m + (1 - \lambda) \theta_e$$

(6)

where $\lambda \in [0, 1)$ is a momentum coefficient parameter. Note that only the parameters $\theta_e$ are updated by back-propagation. And here we introduce $\theta_m$ to generate sentence embeddings for the queue, because the momentum update can make $\theta_m$ evolve more smoothly than $\theta_e$. As a result, though the embeddings in the queue are encoded by different encoders (in different “steps” during training), the difference among these encoders can be made small.

With sentence embeddings in the queue, the loss function of ESimCSE is further modified as follows,

$$\ell_i = - \log \frac{e^{\text{sim}(h_i^+, h_i^-)/\tau}}{\sum_{j=1}^N e^{\text{sim}(h_i^+, h_j^+)/\tau} + \sum_{m=1}^M e^{\text{sim}(h_i^+, h_m^+)/\tau}}$$

(7)

where $h_m^+$ is denotes a sentence embedding in the momentum-updated queue, and $M$ is the size of the queue.

4 Experiment

4.1 Evaluation Setup

Following unsup-SimCSE, we use 1-million sentences randomly drawn from English Wikipedia for training. Then we conduct our experiments on 7 standard semantic textual similarity (STS) tasks. The detail statistics are shown in Table 3. STS12-STS16 datasets do not have train or development sets, and thus we evaluate the models on the development set of STS-B to search for better settings of the hyper-parameters. The SentEval toolkit is used for evaluation. For the compared baseline unsup-SimCSE, we download the officially published model checkpoints and reproduce evaluation results with the suggested hyper-parameters in dev/test mode. Experiments are conducted on Nvidia 3090 GPUs.

Semantic Textual Similarity Tasks

Semantic textual similarity measures the semantic similarity of any two sentences. STS 2012–2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016) and STS-B (Cer et al., 2017) are widely used semantic textual similarity benchmark datasets, which measure the semantic similarity of two sentences with the cosine similarity of the corresponding sentence embeddings. After deriving the semantic similarities of all pairs in the test set, we follow unsup-SimCSE to use Spearman correlation to measure the correlation between the ranks of predicted similarities and the ground-truth. For a set of size $n$, the $n$ raw scores $X_i, Y_i$ are converted to its corresponding ranks $\text{rg}_X_i, \text{rg}_Y_i$, then the Spearman correlation is defined as follows

$$r_s = \frac{\text{cov}(\text{rg}_X, \text{rg}_Y)}{\sigma_{\text{rg}_X} \sigma_{\text{rg}_Y}}$$

(8)

where $\text{cov}(\text{rg}_X, \text{rg}_Y)$ is the covariance of the rank variables, $\sigma_{\text{rg}_X}$ and $\sigma_{\text{rg}_Y}$ are the standard deviations of the rank variables. Spearman correlation has a value between -1 and 1, which will be high when the ranks of predicted similarities and the ground-truth are similar.

4.2 Training Details

We start from pre-trained checkpoints of BERT(uncased) or RoBERTa(cased) using both the base and the large versions, and we add an MLP layer on top of the [CLS] representation to get the sentence embedding. We implement ESimCSE based on Huggingface’s transformers package. And we train our models for one epoch.

---

1https://huggingface.co/datasets/princeton-nlp/datasets-for-simcse/resolve/main/wiki1m_for_simcse.txt
2https://github.com/facebookresearch/SentEval
3https://github.com/princeton-nlp/SimCSE
4https://github.com/huggingface/transformers, version 4.2.1.
by using the Adam optimizer with the batch size = 64 and the hyper-parameter temperature $\tau = 0.05$ in Eq. (3). The learning rate is set as $3e-5$ for ESimCSE-BERT$_{base}$ model and $1e-5$ for other models. The dropout rate is $p = 0.1$ for base models, $p = 0.15$ for large models. For the momentum contrast, we empirically choose a relatively large momentum $\lambda = 0.995$. In addition, we evaluate the model every 125 training steps on the development set of STS-B and keep the best checkpoint for the final evaluation on test sets. We use sub-word repetition instead of word repetition, which will be further discussed in the ablation study section.

### 4.3 Main Results

Table 4 shows the best results obtained on the STS-B development sets. We highlight the highest numbers among models with the same pre-trained encoder as bold. ♣ denotes the evaluation results from the official published model by (Gao et al., 2021). It can be seen that our proposed ESimCSE outperforms unsup-SimCSE by +2.40% on BERT$_{base}$, +2.19% on BERT$_{large}$, +1.19% on RoBERTa$_{base}$, +0.26% on RoBERTa$_{large}$, respectively.

We also explore how much improvement it can bring to unsup-SimCSE when only using word repetition or momentum contrast. As shown in table 6 and 7, either word repetition or momentum contrast can bring substantial improvements to unsup-SimCSE. It means that both proposed methods to enhance the positive pairs and negative pairs are effective. Better yet, these two modifications can be superimposed (ESimCSE) to get further improvements.

### 5 Ablation Study

This section investigates how different dropout rates, repetition rates, sentence-length-extension methods, and momentum contrast queue size affect ESimCSE’s performance. We only change one hyperparameter at a time. All results use our ESimCSE-BERT$_{base}$ model and are evaluated on the development set of STS-B.

#### 5.1 Effect of Dropout Rate

Dropout is the key ingredient to the unsup-SimCSE model, so different dropout rates $p$ are crucial to the model’s performance. According to (Gao et al., 2021), the optimal dropout rate for unsup-SimCSE-BERT$_{base}$ is $p = 0.1$. Considering that ESimCSE additionally introduces word repetition and momentum contrast mechanisms, we re-examine the impact of different dropouts on its performance. We experiment on three typical dropout rates, and the results are shown in the table 8. Specifically, when the dropout is 0.1, it achieves the best performance on the STS-B development set. When the dropout increases to 0.15, the performance is close to that of 0.1, with no significant drop. And even when the dropout reaches 0.2, the performance drops by nearly 1%, but it still outperforms unsup-SimCSE. The experimental results kind of show the robustness of the superiority of the proposed ESimCSE over unsup-SimCSE, in terms of dropout rate.

#### 5.2 Effect of Repetition Rate

Word repetition can bring improvement by diversifying the length difference of positive pairs in
| Model                      | STS12 | STS13 | STS14 | SICK15 | STS16 | STS-B | SICK-R | Avg.  |
|----------------------------|-------|-------|-------|--------|-------|-------|--------|-------|
| unsup-SimCSE-BERT<sub>base</sub> | 68.40 | 82.41 | 74.38 | 80.91  | 78.56 | 76.85 | 72.23  | 76.25 |
| ESimCSE-BERT<sub>base</sub>       | 73.40 | 83.27 | 77.25 | 82.66  | 78.81 | 80.17 | 72.30  | 78.27 (+2.02) |
| unsup-SimCSE-BERT<sub>large</sub> | 70.88 | 84.16 | 76.43 | 84.50  | 79.76 | 79.26 | 73.88  | 78.41 |
| ESimCSE-BERT<sub>large</sub>       | 73.21 | 85.37 | 77.73 | 84.30  | 78.92 | 80.73 | 74.89  | 79.31 (+0.90) |
| unsup-SimCSE-RoBERTa<sub>base</sub> | 70.16 | 81.77 | 73.24 | 81.36  | 80.65 | 80.22 | 68.56  | 76.57 |
| ESimCSE-RoBERTa<sub>base</sub>         | 73.20 | 84.93 | 76.88 | 84.86  | 81.21 | 82.79 | 72.27  | 79.45 (+0.55) |

Table 5: Sentence embedding performance on 7 semantic textual similarity (STS) test sets, in terms of Spearman’s correlation, with BERT<sub>base</sub>, BERT<sub>large</sub>, RoBERTa<sub>base</sub>, RoBERTa<sub>large</sub> as base models. ♣: results from official published model by (Gao et al., 2021).
reduce the effect. It is intuitive because the introduction of momentum contrast encourages more negative pairs to participate in the loss calculation so that the positive pairs can be compared more sufficiently. But a too large queue size also reduces the benefit. We guess that is because the negative pairs in the momentum contrast are generated by the past “steps” during training, and a larger queue will use the outputs of more outdated encoder models which are quite different from the current one. And thus that will reduce the reliability of the loss calculation.

| Queue Size       | STS-B   |
|------------------|---------|
| $1 \times \text{batch}\_\text{size}$ | 83.83   |
| $1.5 \times \text{batch}\_\text{size}$ | 83.81   |
| $2 \times \text{batch}\_\text{size}$ | 83.03   |
| $2.5 \times \text{batch}\_\text{size}$ | 84.85   |
| $3 \times \text{batch}\_\text{size}$ | 82.66   |

Table 11: Effects of queue size of momentum contrast on the STS-B development set in terms of Spearman’s correlation.

### 6 Related Work

Unsupervised sentence representation learning has been widely studied. (Socher et al., 2011; Hill et al., 2016; Le and Mikolov, 2014) propose to learn sentence representation according to the internal structure of each sentence. (Kiros et al., 2015; Logeswaran and Lee, 2018) predict the surrounding sentences of a given sentence based on the distribution hypothesis. (Pagliardini et al., 2017) propose Sent2Vec, a simple unsupervised model allowing to compose sentence embeddings using word vectors along with n-gram embeddings.

Recently, contrastive learning has been explored in unsupervised sentence representation learning and has become a promising trend (Zhang et al., 2020; Wu et al., 2020; Meng et al., 2021; Gao et al., 2021; Yan et al., 2021). Those contrastive learning
based methods for sentence embeddings are generally based on the assumption that a good semantic representation should be able to bring similar sentences closer while pushing away dissimilar ones. Therefore, those methods use various data augmentation methods to randomly generate two different views for each sentence and design an effective loss function to make them closer in the semantic representation space. Among these contrastive methods, the most related ones to our work are unsup-ConSERT and unsup-SimSCE. ConSERT explores various effective data augmentation strategies (e.g., adversarial attack, token shuffling, Cutoff, dropout) to generate different views for contrastive learning and analyze their effects on unsupervised sentence representation transfer. Unsup-SimSCE, the current state-of-the-art unsupervised method uses only standard dropout as minimal data augmentation, and feed an identical sentence to a pretrained model twice with independently sampled dropout masks to generate two distinct sentence embeddings as a positive pair. Unsup-SimSCE is very simple but works surprisingly well, performing on par with previously supervised counterparts. However, we find that unsup-SimCSE constructs each positive pair with two sentences of the same length, which can mislead the learning of sentence embeddings. So we propose a simple but effective method termed “word repetition” to alleviate it. We also propose to use the momentum contrast method to increase the number of negative pairs involved in the loss calculation, which encourages the model towards more refined learning.

7 Conclusion and Future Work

In this paper, we propose optimizations to construct positive and negative pairs for unsup-SimCSE and combine them with unsup-SimCSE, which is termed ESimCSE. Through extensive experiments, the proposed ESimCSE achieves considerable improvements on standard semantic text similarity tasks over unsup-SimCSE.

As unsup-SimCSE treats all negative pairs the same importance. Some negative pairs are quite different from positive pairs, while others are relatively close to positive pairs. This distinction will be helpful for embedding retrieval tasks but not reflected in the objective function of unsup-SimCSE. Therefore, in the future, we will focus on designing a more refined objective function to improve the discrimination between different negative pairs.

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 Janice 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 Janice 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.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. arXiv preprint arXiv:1708.00055.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR.

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

Hongchao Fang, Siccheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive
Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

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. *arXiv preprint arXiv:1602.03483*.

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.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

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*.

Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. Coco-Lm: Correcting and contrasting text sequences for language model pretraining. *arXiv preprint arXiv:2102.08473*.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.

Richard Socher, Eric Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Advances in neural information processing systems*, 24.

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*.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Conserv: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. *arXiv preprint arXiv:2009.12061*.