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

Xing Wu[1,2,3], **Chaochen Gao**[1,2]*, **Liangjun Zang**[1], **Jizhong Han**[1], **Zhongyuan Wang**[3], **Songlin Hu**[1,2]†

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Kuaishou Technology, Beijing, China
{wuxing,gaochaochen,zangliangjun,hanjizhong,husonglin}@iie.ac.cn
wangzhongyuan@kuaishou.com

Abstract

SimCSE [1] adopts *dropout* as data augmentation and encodes an input sentence *twice* into two corresponding embeddings to build a positive pair. Since SimCSE is a Transformer-based encoder that directly encodes the length information of sentences through positional embeddings, the two embeddings in a positive pair contain the same length information. Thus, a model trained with these positive pairs is biased, tending to consider that sentences of the same or similar length are more similar in semantics. To alleviate it, we apply a simple but effective repetition operation to modify the input sentence. Then we 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 computer vision community and introduce momentum contrast to enlarge the number of negative pairs without additional calculations. The proposed modifications are applied to positive and negative pairs separately, and build a new sentence embedding method, termed Enhanced 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 SimCSE by an average Spearman correlation of 2.02% on BERT-base. Our code are available at https://github.com/caskcsg/ESimCSE.

1 Introduction

Recently, researchers have proposed using contrastive learning to learn better unsupervised sentence embeddings (Wu et al.; Zhang et al., b; Liu et al., 2021; Gao et al., 2021; Yan et al., 2021). Contrastive learning aims to learn effective sentence embeddings based on the assumption that

Length Diff	Avg. Similarity Diff
> 3	16.34
≤ 3	**18.18** (+1.84)

Table 1: The average similarity difference between the model (SimCSE-BERT) predictions and the normalized ground truths.

effective sentence embeddings should bring similar sentences closer while pushing away dissimilar ones. It generally uses various data augmentation methods (Shleifer, 2019; Wei and Zou, 2019; Wu et al., 2019) to generate different views for each sentence randomly, and assumes a sentence is semantically more similar to its augmented counterpart than any other sentence. Among these methods, the most representative one is SimCSE (Gao et al., 2021), which performs on par with previously supervised counterparts. SimCSE implicitly hypothesizes *dropout* acts as a minimal data augmentation method. Specifically, 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. SimCSE models are built on Transformer blocks, which will encode a sentence's length information through positional embeddings. In a positive pair, two embeddings are derived from the same sentence to contain the same length information. In contrast, in a negative pair, two embeddings in a negative pair are derived from two different sentences and generally contain different length information. Therefore, positive and negative pairs are different in their length information, acting as a feature to distinguish them. The semantic simi-

---

Method	Text	Similarity
original sentence	I like this apple because it looks so fresh and it should be delicious.	1.0
random insertion	I **don't** like this apple because **but** it looks so **not** fresh and it should be **dog** delicious.	0.69
random deletion	I like this ~~apple~~ because it looks so ~~fresh~~ and it should be ~~delicious~~.	0.32
word repetition	I like **like** this apple because it looks so **so** fresh and **and** it should be delicious.	0.99
word repetition	I **I** like this apple **apple** because it looks **looks** so fresh **fresh** and it should be delicious **delicious**.	0.98

Table 2: An example of semantic similarity after different methods change a sentence's length.

larity 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 it, we evaluate on seven standard semantic textual similarity datasets with the SimCSE-BERT$_{base}$ model published by (Gao et al., 2021). We partition each STS test set into two groups based on whether the sentence pairs' length difference is ≤ 3. We calculate the similarity differences between the model predictions and the normalized ground truths for each group. As shown in Table 1, the average similarity difference of seven datasets is higher when the length difference is ≤ 3, which verifies our assumption. Comparison details on each dataset can refer to Table 7.

To alleviate this problem, we propose a simple but effective enhancement method to 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. Such operations are detrimental to SimCSE learning, which is also discussed in a contemporaneous work (Chuang et al., 2022). Therefore, we propose a safer method, termed "word repetition", which randomly duplicates some words in a sentence. For example, as shown in Table 2, either random insertion or random deletion may generate a sentence that deviates far from the meaning of the original sentence. On the contrary, the method of "word repetition" maintains 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 a better comparison between the pairs (Chen et al.). 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), larger batch size is not always a better choice. For example, for the SimCSE-BERT$_{base}$ 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.), which is also applied to natural language understanding (Fang et al.). Momentum contrast allows us to reuse the encoded embeddings from the immediate preceding mini-batches to expand the negative pairs by maintaining a queue. It always enqueues the sentence embeddings of the current mini-batches and meanwhile dequeues the "oldest" ones. As the enqueued sentence embeddings come from the preceding mini-batches, we keep a momentum updated encoder by taking the moving average of its parameters and use the momentum encoder 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 SimCSE, termed Enhanced SimCSE (ESimCSE). We illustrate the schematic diagram of ESimCSE in Figure 1. The proposed ESimCSE is evaluated on the semantic
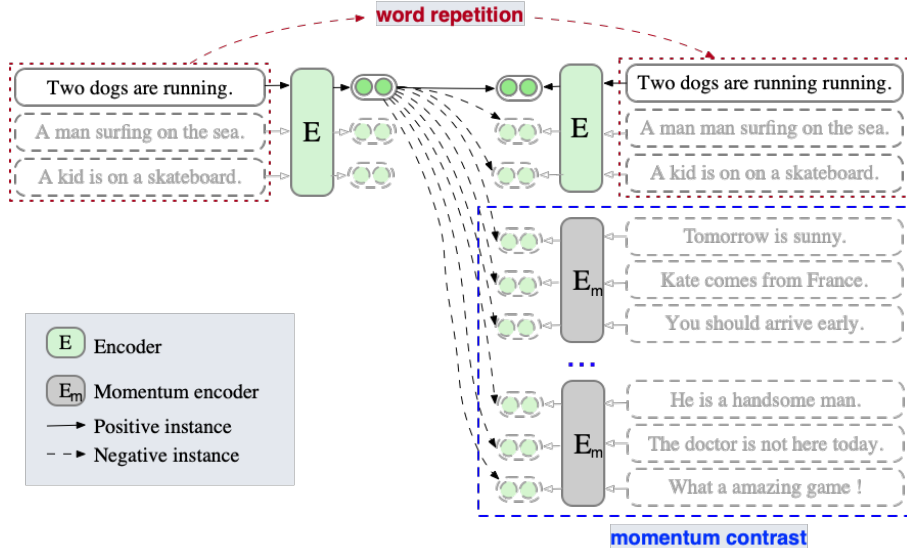
Figure 1: The schematic diagram of the ESimCSE method.

text similarity (STS) task with 7 STS-B test sets. Experimental results show that ESimCSE can improve the similarity measuring performance in different model settings over the previous state-of-the-art SimCSE. Specifically, ESimCSE gains an average increase of Spearman's correlation over SimCSE by +2.02% on BERT_{base}.

Our contributions can be summarized as follows:

- We observe that 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 improvements to SimCSE.

2 Background: SimCSE

Given a set of paired sentences $\left\{ x_i, x_i^+ \right\}_{i=1}^m$, where x_i and x_i^+ are semantically related and will be referred to positive pairs. The core idea of SimCSE is to use identical sentences to build the positive pairs, i.e., $x_i^+ = x_i$. Note that in Transformer, there is

a dropout mask placed on fully-connected layers and attention probabilities. And thus, the key ingredient is to feed the same input x_i to the encoder twice by applying different dropout masks z_i and z_i^+ and output two separate sentence embeddings to build a positive pair as follows:

$$\mathbf{h}_i = f_\theta \left(x_i, z_i \right), \mathbf{h}_i^+ = f_\theta \left(x_i, z_i^+ \right) \qquad (1)$$

With h_i and h_i^+ for each sentence in a mini-batch with batch size N, the contrastive learning objective w.r.t x_i is formulated as follows,

$$\ell_i = -\log \frac{e^{\text{sim}\left(\mathbf{h}_i, \mathbf{h}_i^+\right)/\tau}}{\sum_{j=1}^N e^{\text{sim}\left(\mathbf{h}_i, \mathbf{h}_j^+\right)/\tau}} \qquad (2)$$

where τ is a temperature hyperparameter and $\text{sim}\left(\mathbf{h}_i, \mathbf{h}_i'\right)$ is the similarity metric, which is typically the cosine similarity function.

3 Proposed Enhanced SimCSE

In this section, we first introduce the word repetition method to construct better positive pairs. Then we introduce the momentum contrast method to expand negative pairs.

3.1 Word Repetition

The word repetition mechanism randomly duplicates some words/sub-words in a sentence. Here 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

$$dup\_len \in [0, max(2, int(dup\_rate * N))] \quad (3)$$

where dup_rate is the maximal repetition rate, which is a hyperparameter. Then dup_len is a randomly sampled number in the set defined above, which will introduce more diversity when extending the sequence length. After dup_len is determined, we use uniform distribution to randomly select dup_len sub-words that need to be repeated from the sequence, which composes the dup_set as follows,

$$dup\_set = uniform([1, N], num = dup\_len) \quad (4)$$

For example, if the 1st sub-word is in dup_set, then sequence x becomes $x^+ = \{x_1, x_1, x_2, ..., x_N\}$. And different from 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 the parameters of the encoder as θ_e and those of the momentum-updated encoder as θ_m, we update θ_m in the following way,

$$\theta_m \leftarrow \lambda\theta_m + (1 - \lambda)\theta_e \quad (5)$$

where $\lambda \in [0, 1)$ is a momentum coefficient parameter. Note that only the parameters θ_e are updated by back-propagation. And here we introduce θ_m to generate sentence embeddings for the queue, because the momentum update can make θ_m evolve more smoothly than θ_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 modifed as follows,

$$\ell_i = -\log \frac{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau} + \sum_{m=1}^{M} e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_m^+)/\tau}} \quad (6)$$

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 Experiment Setup

Our experimental language is English. For a fair comparison, our experimental setup mainly follows SimCSE. We use 1-million sentences randomly drawn from English Wikipedia for training[2]. The semantic textual similarity task measures the capability of sentence embeddings, and we conduct our experiments on seven standard semantic textual similarity (STS) datasets. STS12-STS16 datasets (Agirre et al., d,e,b,a,c) do not have train or development sets, and thus we evaluate the models on the development set of STS-B (Cer et al.) to search for better settings of the hyper-parameters. The SentEval toolkit[3] is used for evaluation, and Spearman correlation coefficient [4] is used to report the model performance. All the experiments are conducted on Nvidia 3090 GPUs.

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[5]. We train our models for one epoch using the Adam optimizer with the batch size $= 64$ and the 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 λ

---

[2]https://huggingface.co/datasets/princeton-nlp/datasets-for-simcse/resolve/main/wiki1m_for_simcse.txt

[3]https://github.com/facebookresearch/SentEval

[4]https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

[5]https://github.com/huggingface/transformers,version 4.2.1.

Model	STS12	STS13	STS14	SICK15	STS16	STS-B	SICK-R	Avg.
IS-BERT$_{base}$ △	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CT-BERT$_{base}$ △	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
ConSERT$_{base}$ ♡	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
BERT$_{base}$-flow ◇	63.48	72.14	68.42	73.77	75.37	70.72	63.11	69.57
SG-OPT-BERT$_{base}$ ♠	66.84	80.13	71.23	81.56	77.17	77.23	68.16	74.62
Mirror-BERT$_{base}$ ♯	69.10	81.10	73.00	81.90	75.70	78.00	69.10	75.40
SimCSE-BERT$_{base}$ ♣	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
ESimCSE-BERT$_{base}$	**73.40**	**83.27**	**77.25**	**82.66**	**78.81**	**80.17**	**72.30**	**78.27**
ConSERT$_{large}$ ♡	70.69	82.96	74.13	82.78	76.66	77.53	70.37	76.45
BERT$_{large}$-flow ◇	65.20	73.39	69.42	74.92	77.63	72.26	62.50	70.76
SG-OPT-BERT$_{large}$ ♠	67.02	79.42	70.38	81.72	76.35	76.16	70.20	74.46
SimCSE-BERT$_{large}$ ♣	70.88	84.16	76.43	**84.50**	**79.76**	79.26	73.88	78.41
ESimCSE-BERT$_{large}$	**73.21**	**85.37**	**77.73**	84.30	78.92	**80.73**	**74.89**	**79.31**
Mirror-RoBERTa$_{base}$ ♯	66.60	82.70	74.00	82.40	79.70	79.60	69.70	76.40
SimCSE-RoBERTa$_{base}$ ♣	**70.16**	81.77	73.24	81.36	**80.65**	80.22	68.56	76.57
ESimCSE-RoBERTa$_{base}$	69.90	**82.50**	**74.68**	**83.19**	80.30	**80.99**	**70.54**	**77.44**
SimCSE-RoBERTa$_{large}$ ♣	72.86	83.99	75.62	84.77	**81.80**	81.98	71.26	78.90
ESimCSE-RoBERTa$_{large}$	**73.20**	**84.93**	**76.88**	**84.86**	81.21	**82.79**	**72.27**	**79.45**

Table 3: Sentence embedding performance on 7 semantic textual similarity (STS) test sets. ♣ : results from official published model by (Gao et al., 2021). ♡ : results from (Yan et al., 2021). ♠ : results from (Kim et al., 2021). ◇ : results from (Li et al., 2020). △ : results are reproduced and reevaluated by (Gao et al., 2021). ♯ : results from (Liu et al., 2021)

= 0.995. In addition, following SimCSE's code, 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, further discussed in the ablation study section.

4.3 Main Results

Table 3 shows the models' performance on seven semantic textual similarity (STS) test sets. We mainly select SimCSE for comparison, since it is the current state-of-the-art and shares the same setting as our approach. In addition, we also use IS-BERT (Zhang et al., a), CT-BERT (Carlsson et al., 2021), ConSERT (Yan et al., 2021), SG-OPT (Kim et al., 2021), BERT-flow (Li et al., 2020), Mirror-BERT (Liu et al., 2021) as baselines. It can be seen that ESimCSE improves the measurement of semantic textual similarity in different settings over SimCSE. Specifically, ESimCSE outperforms SimCSE by +2.02% on BERT$_{base}$, +0.90% on BERT$_{large}$, +0.87% on RoBERTa$_{base}$, +0.55% on RoBERTa$_{large}$, respectively.

Model	STS-B
SimCSE ♣	82.45
+ word repetition	84.09 (+1.64)
+ momentum contrast	83.98 (+1.53)
ESimCSE	**84.85** (+2.40)

Table 4: Improvement on STS-B development sets that word repetition or momentum contrast brings to SimCSE. ♣ : results from official published model by (Gao et al., 2021).

5 Ablation Study

This section investigates how different settings affect ESimCSE's performance. All results are compared on BERT$_{base}$ scale models and are evaluated on the development set of STS-B unless otherwise specified.

5.1 The Importance of Word Repetition and Momentum Contrast

We explore how much improvement it can bring to SimCSE when only using word repetition or momentum contrast. As shown in Table 4, either word

Length-extension Method	STS-B
+Inserting Stop-words	81.72
+Inserting [MASK]	83.08
+Inserting Masked Prediction	84.18
+Word Repetition	84.40
+Sub-word Repetition	**84.85**

Table 5: Effects of sentence-length-extension method.

repetition or momentum contrast can bring substantial improvements to 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.

5.2 Effect of Sentence-Length-Extension Method

In addition to sub-word repetition, we also explore three other methods to increase sentence length:

- **Word Repetition** is similar to sub-word repetition, except that the repetition operation occurs *before* tokenization. For example, given a word "microbiology", word repetition will produce "microbiology microbiology", while sub-word repetition will produce "micro micro ##biology" or "micro ##biology ##biology".

- **Inserting Stop-words** inserts a random stop-word after the selected word instead of repeating the selected word.

- **Inserting [MASK]** inserts a [MASK] token after the selected word. We can regard [MASK] as a dynamic context-compatible word placeholder.

- **Inserting Masked Prediction** inserts a [MASK] token after the selected word and uses the masked language model to predict the top-1 substitution. The substitution is used to replace the inserted [MASK] token.

As shown in Table 5, sub-word repetition achieves the best performance, and word repetition can also bring a good improvement, which shows that more fine-grained repetition can better alleviate the bias brought by the length difference of positive pairs. Inserting [MASK] can also improve slightly, but inserting stop words will decrease the effect.

Inserting masked prediction also brings a good improvement, but this method requires a pre-trained masked language model to predict replacements, bringing high additional computational overhead.

5.3 Batching Sentences of Similar Length in Training

Apart from sentence-length-extension methods, we explore whether batching sentences of similar length in training will alleviate the bias towards identical sequence length in inference. We divide training sentences into buckets by length and batch them within each bucket. We explore two different settings:

- We divide the training set into two coarse-grained buckets based on whether the sentence length is greater than buc_len, where $buc\_len \in [3, 8]$;

- We divide the training set by sentence length into 6 fine-grained buckets: $\{\leq 3, 4, 5, 6, 7, \geq 8\}$, which we use $buc\_len = 3 \sim 8$ for short.

We list the experimental results in Table 6. Dividing the training set into buckets does not bring significant improvements and even decreases in some settings. We believe that after being divided into buckets, shuffle can only be performed within a bucket, leading to an insufficient comparison in contrastive learning. In contrast, the effect of word repetition is much better.

5.4 The Relationship between The Similarity and Length Difference

We further explore the relationship between the similarity and length difference of sentence pairs on ESimcSE, compared with that of SimCSE in the Introduction. As STS12-STS16 datasets do not have train or development sets, and thus we evaluate the models on the test set of each dataset. We partition each STS test set into two groups based

buc_len	wr	3	4	5
STS-B	84.09	81.92	82.00	82.66
buc_len	6	7	8	$3 \sim 8$
STS-B	82.00	82.13	83.00	82.18

Table 6: Effects of different bucket lengths buc_len. "wr" means using word repetition method instead of bucketing sentences. "$3 \sim 8$" means fine-grained buckets setting: $\{\leq 3, 4, 5, 6, 7, \geq 8\}$.

Model	LD	STS12	STS13	STS14	SICK15	STS16	STS-B	SICK-R	Avg.
SimCSE	> 3	8.93	15.74	11.90	19.68	28.91	21.33	7.86	16.34
	≤ 3	9.29	22.81	19.53	19.92	24.08	22.12	9.53	18.18
ESimCSE	> 3	13.48	23.73	17.14	25.98	34.71	26.22	10.44	21.67
	≤ 3	12.52	28.56	24.13	24.17	29.32	25.63	12.35	22.38

Table 7: The difference between the model predicted cosine similarity and the true label on each dataset's test set. "LD" is short for length difference.

Model	Sim <q,s1 >	Sim <q,s2 >
SimCSE	26.39	27.07(**+0.68**)
ESimCSE	36.82	36.87(**+0.05**)

Table 8: Effect of repeated words on the average similarity of two sets

dup_rate	0.08	0.12	0.16	0.2
STS-B	83.5	83.62	82.01	83.01
dup_rate	0.24	0.28	0.32	0.36
STS-B	84.24	82.96	**84.85**	83.84

Table 9: Effects of repetition rate dup_rate.

on whether the sentence pairs' length difference is ≤ 3. Then we calculate the similarity differences between the model predictions and the normalized ground truths for each group. As listed in Table 7, ESimCSE significantly reduces the average similarity difference gap between > 3 and ≤ 3, from 1.84 to 0.71, alleviating the learning bias we mentioned in the Introduction.

5.5 Will Word Repetition Bring New Bias ?

We further explore whether word repetition will mislead the model to be more inclined to consider sentences with repeated overlaps are more similar. We conduct a detection experiment on wiki data with the following settings:

1. We randomly select a sentence as a query, such as q = "I like **this** apple because it **looks very** fresh"

2. We use the query to randomly recall a candidate sentence with 13%-17% overlap tokens, such as s1 = "**This** is a very tall tree and it **looks like** a giant"

3. We apply the word-repetition operation on the overlap tokens in the candidate sentence and produce a word-repeated sentence, such as s2 = "**This this** is a **very very** tall tree and it **looks looks** like a giant."

4. We calculate the similarity of <q, s1 >and <q, s2 >and compare them.

We experiment on 100 different query sentences and calculate their average similarity. As shown

in Table 8, compared to the 0.68 increase of the SimCSE, ESimCSE-BERT only increased by 0.05. Therefore, word repetition does not bring a new bias to the learning process.

5.6 Effect of Hyperparameters

Repetition Rate To quantitatively study the effect of repetition rate on the model performance, we slowly increase the repetition rate parameter dup_rate from 0.08 to 0.36, with each increase by 0.04. As shown in Table 9, when $dup\_rate = 0.32$, ESimCSE achieves the best performance, a larger or smaller dup_rate will cause performance degradation, which is consistent with our intuition.

Momentum Queue Size The size of the momentum contrast queue determines the number of negative pairs involved in the loss calculation. We experiment with the queue size equals to different multiples of the batch size. The experimental results are listed in Table 10. The optimal result is reached when the queue size was 2.5 times the batch size. A smaller queue size will reduce the effect. This is intuitive because more negative pairs participate in the loss calculation to compare positive pairs more fully. But a too large queue size also reduces the effect. 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
$1 \times batch\_size$	83.83
$1.5 \times batch\_size$	83.81
$2 \times batch\_size$	83.03
$2.5 \times batch\_size$	**84.85**
$3 \times batch\_size$	82.66

Table 10: Effects of queue size of momentum contrast.

5.7 Performance on Transfer Tasks

Following (Gao et al., 2021), we further evaluate ESimCSE on transfer tasks, to see the transferability of the sentence embeddings from ESimCSE. The transfer tasks include: MR (movie review) (Pang and Lee, 2005), CR (product review) (Hu and Liu, 2004), SUBJ (subjectivity status) (Pang and Lee, 2004) , MPQA (opinion-polarity) (Wiebe et al., 2005), SST-2 (binary sentiment analysis) (Socher et al., 2013), TREC (question-type classification) (Voorhees and Tice, 2000) and MRPC (paraphrase detection) (Dolan and Brockett, 2005). For more details, one can refer to SentEval[6]. As shown in Table 11, compared with the performance of SimCSE, ESimCSE slightly increases the transferability of embedding. As our optimizations are focused on semantic textual similarity tasks, the ability of ESimcse on transfer tasks remains stable relative to SimCSE.

6 Related Work

Unsupervised sentence representation learning has been widely studied. (Socher et al.; Hill et al.; Le and Mikolov) propose to learn sentence representation according to the internal structure of each sentence. (Kiros et al.; Logeswaran and Lee) predict the surrounding sentences of a given sentence based on the distribution hypothesis. (Pagliardini et al.) 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., b; Wu et al.; Meng et al.; Liu et al., 2021; Gao et al., 2021; Yan et al., 2021; Chuang et al., 2022). 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 (Yan et al., 2021) and unsup-SimSCE (Gao et al., 2021). 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 pre-trained 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 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 temed "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 SimCSE and combine them with SimCSE, which is termed ESim-CSE. Through extensive experiments, the proposed ESimCSE achieves considerable improvements on standard semantic text similarity tasks over Sim-CSE.

In the future, we will focus on designing a more refined objective function to improve the discrimination between different negative pairs. Also we will make attempt to optimize the performance on both semantic textual similarity tasks and transfer tasks.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada

---

[6]https://github.com/facebookresearch/SentEval

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
SimCSE ♣	81.18	**86.46**	94.45	**88.88**	85.50	89.80	74.43	85.81
ESimCSE	**81.32**	86.22	**94.74**	88.74	**85.50**	**91.00**	**74.90**	**86.06**

Table 11: Results on transfer tasks of different sentence embedding models, in terms of accuracy. ♣ : results from (Gao et al., 2021).

Mihalcea, et al. a. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. b. Semeval-2014 task 10: Multilingual semantic textual similarity.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. c. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. d. Semeval-2012 task 6: A pilot on semantic textual similarity.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. e. * sem 2013 shared task: Semantic textual similarity.

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

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 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. A simple framework for contrastive learning of visual representations. In International conference on machine learning.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James Glass. 2022. Diffcse: Difference-based contrastive learning for sentence embeddings. arXiv preprint arXiv:2204.10298.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005).

Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. Cert: Contrastive self-supervised learning for language understanding. arXiv preprint arXiv:2005.12766.

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. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. arXiv preprint arXiv:1602.03483.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 168–177.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for bert sentence representations. arXiv preprint arXiv:2106.07345.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In Advances in neural information processing systems.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In International conference on machine learning.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. arXiv preprint arXiv:2011.05864.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. arXiv preprint arXiv:2104.08027.

Lajanugen Logeswaran and Honglak Lee. 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. Coco-lm: Correcting and contrasting text sequences for language model pretraining. arXiv preprint arXiv:2102.08473.

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

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. arXiv preprint cs/0409058.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. arXiv preprint cs/0506075.

Sam Shleifer. 2019. Low resource text classification with ulmfit and backtranslation. arXiv preprint arXiv:1903.09244.

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing, pages 1631–1642.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pages 200–207.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. Language resources and evaluation, 39(2):165–210.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In International conference on computational science, pages 84–95. Springer.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 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. Consert: A contrastive framework for self-supervised sentence representation transfer. arXiv preprint arXiv:2105.11741.

Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. a. An unsupervised sentence embedding method by mutual information maximization. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.

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