A Sentence is Worth 128 Pseudo Tokens: A Semantic-Aware Contrastive Learning Framework for Sentence Embeddings

Anonymous ACL submission

Abstract

Contrastive learning has shown great potential in unsupervised sentence embedding tasks, e.g., SimCSE (Gao et al., 2021). However, these existing solutions are heavily affected by superficial features like the length of sentences or syntactic structures. In this paper, we propose a semantic-aware contrastive learning framework for sentence embeddings, termed Pseudo-Token BERT (PT-BERT), which is able to explore the pseudo-token space (i.e., latent semantic space) representation of a sentence while eliminating the impact of superficial features such as sentence length and syntax. Specifically, we introduce an additional pseudo token embedding layer independent of the BERT encoder to map each sentence into a sequence of pseudo tokens in a fixed length. Leveraging these pseudo sequences, we are able to construct same-length positive and negative pairs based on the attention mechanism to perform contrastive learning. In addition, we utilize both the gradient-updating and momentum-updating encoders to encode instances while dynamically maintaining an additional queue to store the representation of sentence embeddings, enhancing the encoder's learning performance for negative examples. Experiments show that our model outperforms the state-of-the-art baselines on six standard semantic textual similarity (STS) tasks. Furthermore, experiments on alignments and uniformity losses, as well as hard examples with different sentence lengths and syntax, consistently verify the effectiveness of our method.

1 Introduction

Sentence embedding serves as an essential technique in a wide range of applications, including semantic search, text clustering, text classification, etc. (Kiros et al., 2015; Logeswaran and Lee, 2018; Conneau et al., 2017; Cer et al., 2018; Reimers and Gurevych, 2019; Gao et al., 2021). Contrastive learning works on learning representations such that similar examples stay close whereas dissimilar ones are far apart, and thus is suitable for sentence embeddings due to its natural availability of similar examples. Incorporating contrastive learning in sentence embeddings improves the efficiency of semantic information learning in an unsupervised manner (He et al., 2020; Chen et al., 2020) and has been shown to be effective on a variety of tasks (Reimers and Gurevych, 2019; Gao et al., 2021; Zhang et al., 2020).

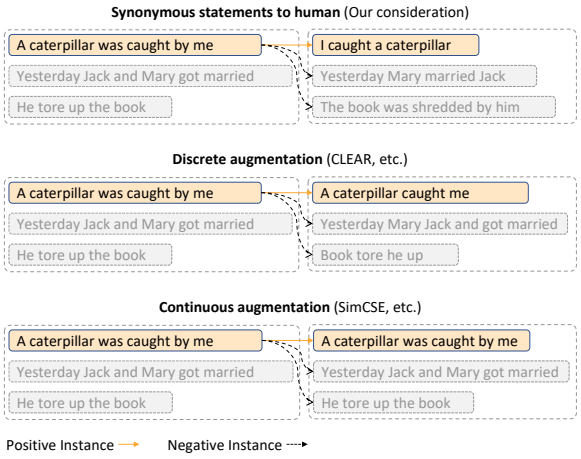


Figure 1: In a realistic scenario, negative examples have the same length and structure, while positive examples act in the opposite way. In comparison, discrete augmentation obtains positive instances with word deletion or reordering (Wu et al., 2020; Meng et al., 2021), which may misinterpret the meaning. The continuous method treats embeddings o the same original sentence as positive examples and augments sentences with the different encoding functions (Carlsson et al., 2021; Gao et al., 2021).

In contrastive learning for sentence embeddings, a key challenge is how to construct positive instances. Both discrete and continuous augmentation methods have been studied recently. Methods in (Wu et al., 2018; Meng et al., 2021) (e.g., CLEAR) perform discrete operations directly on the original sentences, such as word deletion and

sentence shuffling, to get positive samples. However, these methods may lead to unacceptable semantic distortions or even complete misinterpretations of the original statement. In contrast, the SimCSE method (Gao et al., 2021) obtains two different embeddings in the continuous embedding space as a positive pair for one sentence through different *dropout masks* (Srivastava et al., 2014) in the neural network for representation learning. Nonetheless, this method overly relies on superficial features existing in the dataset like sentence lengths and syntactic structures and may pay less reflection on meaningful semantic information. As an illustrative example, the sentence-pair in Fig. 1 "A caterpillar was caught by me." and "I caught a caterpillar." appear to organize differently in expression but convey exactly the same semantics.

To overcome these drawbacks, in this paper, we propose a semantic-aware contrastive learning framework for sentence embeddings, termed Pseudo-Token BERT (PT-BERT), that is able to capture the pseudo-token space (i.e., latent semantic space) representation while ignoring effects of superficial features like sentence lengths and syntactic structures. Inspired by previous work on prompt learning and sentence selection (Li and Liang, 2021; Liu et al., 2021; Humeau et al., 2020), which create a pseudo-sequence and have it serve the downstream tasks, we present PT-BERT to train pseudo token representations and then to map sentences into pseudo token spaces based on an attention mechanism.

In particular, we train additional 128 pseudo token embeddings, together with sentence embeddings extracted from the BERT model (i.e., gradient-encoder), and then use the attention mechanism (Devlin et al., 2019) to map the sentence embedding to the pseudo token space (i.e., semantic space). We use another BERT model (i.e., momentum-encoder) to encode the original sentence, adopt a similar attention mechanism with the pseudo token embeddings, and finally output a continuously augmented version of the sentence embedding. We treat the representations of original sentence encoded by the gradient-encoder and the momentum-encoder as a positive pair. In addition, the momentum-encoder also generates negative examples, dynamically maintains a queue to store these negative examples, and updates them overtime. By projecting all sentences onto the same pseudo sentence, the model greatly reduces the dependence on sentence length and syntax when making judgments and makes the model more focused on the semantic level information.

In our experiments, we compare our results with the previous state-of-the-art work. We train PT-BERT on 10^6 randomly sampled sentences from English Wikipedia and evaluate on seven standard semantic textual similarity (STS) tasks (Agirre et al., 2012, 2013, 2014, 2015, 2016) (Marelli et al., 2014). Besides, we also compare our approach with a framework based on an advanced discrete augmentation we proposed. We obtain a new state-of-the-art on standard semantic textual similarity tasks with our PT-BERT, which achieves 77.74% of Spearman's correlation. To show the effectiveness of pseudo tokens, we calculate the align-loss and uniformity loss (Wang and Isola, 2020) and verify our approach on a sub-dataset with hard examples sampled from STS-(2012-2016).

2 Related Work

In this section, we discuss related studies with repect to the contrastive learning framework and sentence embedding.

2.1 Contrastive Learning for Sentence Embedding

Contrastive learning and MoCo. Contrastive learning (Hadsell et al., 2006) has been used with much success in both natural language processing and computer vision (Yang et al., 2019; Klein and Nabi, 2020; Chen et al., 2020; He et al., 2020; Gao et al., 2021). In contrast to generative learning, contrastive learning requires learning to distinguish and match data at the abstract semantic level of the feature space. It focuses on learning common features between similar examples and distinguishing differences between non-similar examples. In order to compare the instances with more negative examples and less computation, memory bank (Wu et al., 2018) is proposed to enhance the performance under the contrastive learning framework. While with a large capacity to store more samples, the memory bank is not consistent enough, which could not update the "key" during comparison. Momentum-Contrast (MoCo) (He et al., 2020) uses a queue to maintain the dictionary of samples which allows the model to compare the query with more keys for each step and ensure the consistency of the framework. It updates the parameter of the dictionary in a momentum way.

Discrete and continuous augmentation. By equipping discrete augmentation that modifies sentences directly on token level with contrastive learning, significant success has been achieved in obtaining sentence embeddings. Such methods include word omission (Yang et al., 2019), entity replacement (Xiong et al., 2020), trigger words (Klein and Nabi, 2020) and traditional augmentations such as deletion, reorder and substitution (Wu et al., 2020; Meng et al., 2021). Examples with diverse expressions can be learned during training, making the model more robust to expressions of different sentence lengths and styles. However, these approaches are limited because there are huge difficulties in augmenting sentences precisely since a few changes can make the meaning completely different or even opposite.

Researchers have also explored the possibility of building sentences continuously, which instead applies operation in embedding space. CT-BERT (Carlsson et al., 2021) encodes same sentence with two different encoders. UnsupSimCSE (Gao et al., 2021) compares the representations of the same sentence with different dropout masks among the mini-batch. These approaches continuously augment sentences while retaining the original meaning. However, positive pairs seen by SimCSE always have the same length and structure, whereas negative samples are likely to act oppositely. As a result, sentence length and structure are highly correlated to the similarity score of examples. During training, the model has never seen positive samples with diverse expressions, so that in real test scenarios, the model would be more inclined to classify the synonymous pairs with different expressions as negatives, and those sentences with the same length and structures are more likely to be grouped as positive pairs. This may cause a biased encoder.

2.2 Pseudo Tokens

In the domain of prompt learning (Liu et al., 2021; Jiang et al., 2020; Li and Liang, 2021; Gao et al., 2020), the way to create prompt can be divided into two types, namely discrete and continuous ways. Discrete methods usually search the natural language template as the prompt (Davison et al., 2019; Petroni et al., 2019), while the continuous way always directly works on the embedding space with "pseudo tokens" (Liu et al., 2021; Li and Liang, 2021). In retrieval and dialogue tasks, the current approach adopts "pseudo tokens", namely "poly codes" (Humeau et al., 2020), to jointly encode the query and response precisely and ensure the inference time when compared with the Cross-Encoders and Bi-Encoders (Wolf et al., 2019; Mazaré et al., 2018; Dinan et al., 2019). The essence of these methods is to create a pseudo-sequence and have it serve the downstream tasks without the need for humans to understand the exact meaning. The parameters of these pseudo tokens are independent of the natural language embeddings, and can be tuned based on a specific downstream task. In the following sections, we will show the idea to weaken the model's consideration of sentence length and structures by introducing additional pseudo token embeddings on top of the BERT encoder.

	Sub-dataset	original
STS12	66.54	68.40
STS13	78.50	82.41
STS14	68.76	74.38
STS15	70.27	80.91
STS16	71.31	78.56

Table 1: SimCSE's results on sub-dataset from STS12-16, comparing with original results.

	SimCSE_{32}	SimCSE_{64}	SimCSE_{128}
Avg.	76.25	75.20	75.29

Table 2: Different acceptable sequence length of SimCSE would affect the result on STS tasks.

3 Methods

In this section, we introduce PT-BERT, which provides novel contributions on combining advantages of both discrete and continuous augmentations to advance the state-of-art of sentence embeddings. We first present the setup of problems with a thorough analysis on the bias introduced by the textual similarity theoretically and experimentally. Then we show the details of Pseudo-Token representation and our model's architecture.

3.1 Preliminary

Consider a sentence s, we say that the augmentation is continuous if s is augmented by different encoding functions, $f(\cdot)$ and $f'(\cdot)$. Sentence embeddings $\mathbf{h} = f(s)$ and $\mathbf{h}' = f'(s)$ are obtained by these two functions. With a slight change of the encoding function (e.g., encoders with different *dropout* masks), $\mathbf{h}'$ can be seen as a more precisely

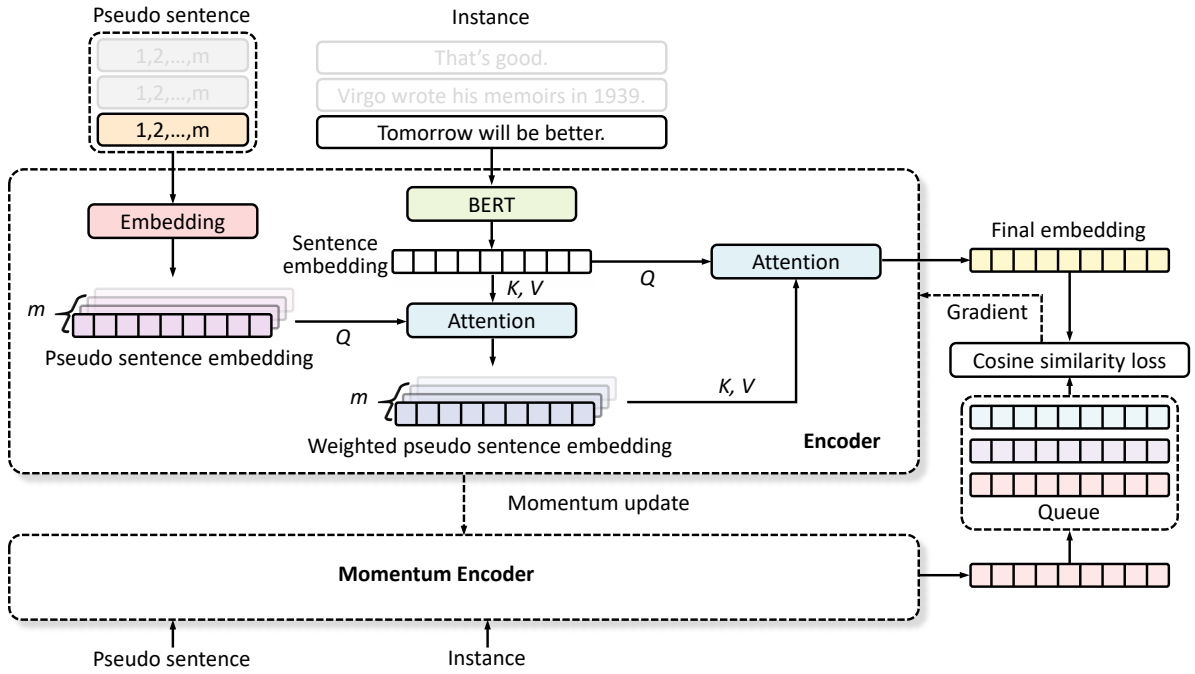


Figure 2: The model is divided into two parts, the upper part (Encoder) updates the learnable parameters with gradient, while the bottom (Momentum Encoder) inherits parameters from the upper part with momentum-updating. We repeatedly input the same sequence of pseudo tokens while processing the original sentences. An additional BERT attention mapping the pooler-output of BERT to pseudo sequence representation, extending the sentence embedding to a fixed length and mapping the syntactic structure to the style of the pseudo sentence. The two attentions in the figure are the same and with identical parameters.

augmented version of $\mathbf{h}$ compared with the discrete augmentation. Semantic information of $\mathbf{h}'$ could be the same as $\mathbf{h}$. Therefore, $\mathbf{h}$ and $\mathbf{h}'$ are a pair of positive examples and we could randomly sample a sentence to construct negative example pairs.

Previous state-of-the-art models (Gao et al., 2021) adopt the continuous strategy that augments sentences with *dropout* (Srivastava et al., 2014). Through careful observation, we find that all the positive examples in SimCSE have the same length and structure while negative examples act oppositely. In this way, SimCSE will inevitably take these two factors as hints during test. To further verify this conjecture, we sort out the positive pairs with a length difference of more than five words and negative pairs of less than two words from STS-(2012-2016).

Table 1 shows that the performance of SimCSE plummets on this dataset. Besides, we also find that SimCSE truncates all training corpus into 32 tokens, which shortens the discrepancy of the sentence's length. After we scale the max length that SimCSE could accept from 32 to 64 and 128, the performance degrades significantly during the test even though the model is supposed to learn more from the complete version of sentences. The reason for this result may lie in the fact that, without truncation, all positive pairs still have the same length, whereas the difference in length between the negative and positive ones is enlarged. Therefore, the encoder will rely more on sentence length and make the wrong decision.

3.2 Pseudo-Token BERT

We realize it is vital to train an unbiased encoder that captures the semantics and also would not introduce intermediate errors. This motivates us to propose the PT-BERT, as evidence shows that the encoder may fail to make predictions when trained on a biased dataset with same-length positive pairs, by learning the spurious correlations that work only well on the training dataset (Arjovsky et al., 2019; Nam et al., 2020).

Pseudo-Token representations. The idea of PT-BERT is to reduce the model's excessive dependence on textual similarity when making predictions. Discrete augmentation achieves this goal by providing both positive and negative examples with diverse expressions. Therefore the model does not

jump to conclusions based on sentence length and syntactic structure during the test.

Note that we achieve this same purpose in a seemingly opposite way: *mapping the representations of both positive and negative examples to a pseudo sentence with the same length and structure.* We take an additional embedding layer outside the BERT encoder to represent a pseudo sentence $\{0, 1, ..., m\}$ with fixed length m and unchangeable syntax. This embedding layer is fully independent of the BERT encoder, including the parameters and corresponding vocabulary. Random initialization is applied to this layer, and the parameter will be updated during training. The size of this layer depends on the length of pseudo sentences. Besides, adopting the attention mechanism (Vaswani et al., 2017; Bahdanau et al., 2015; Gehring et al., 2017), we take the pseudo sentence embeddings as the query while key and value are the sentence embeddings obtained from the BERT encoder. This allows the pseudo sentence to attend to the core part and ignore the redundant part while keeping the fixed length and pseudo syntactic structure.

Fig. 2 illustrates the framework of PT-BERT. Denoting the pseudo sentence embedding as $\mathbf{P}$ and the sentence embedding encoded by BERT as $\mathbf{Y}$, we obtain the weighted pseudo sentence embedding of each sentence by mapping the sentence embedding to the pseudo tokens with attention:

$$\mathbf{Z'_i} = \text{Attention}(\mathbf{P}\mathbf{W^Q}, \mathbf{Y_i}\mathbf{W^K}, \mathbf{Y_i}\mathbf{W^V}) \quad (1)$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K^T}}{\sqrt{\mathrm{d_k}}})\mathbf{V}, \quad (2)$$

where d_k is the dimension of the model, $\mathbf{W^Q}$, $\mathbf{W^K}$, $\mathbf{W^V}$ are the learnable parameters with $\mathbb{R}^{d_k \times d_k}$, i denotes the i-th sentence in the dataset. Then we obtain the final embedding $\mathbf{h_i}$ with the same attention layer by mapping pseudo sentences back to original sentence embeddings:

$$\mathbf{h_i} = \text{Attention}(\mathbf{Y_i}\mathbf{W^Q}, \mathbf{Z'_i}\mathbf{W^K}, \mathbf{Z'_i}\mathbf{W^V}). \quad (3)$$

Finally, we compare the cosine similarities between the obtained embeddings of $\mathbf{h}$ and $\mathbf{h}'$ using Eq. 4 , where $\mathbf{h}'$ are the samples encoded by the momentum-encoder and stored in a queue.

Model architecture. Instead of inputting the same sentence twice to the same encoder, we follow the architecture proposed in Momentum-Contrast (MoCo) (He et al., 2020) such that PT-BERT can efficiently learn from more negative examples. Samples in PT-BERT are encoded into vectors with two encoders: gradient-update encoder (the upper encoder in Fig. 2) and momentum-update encoder (the momentum encoder in Fig. 2). We dynamically maintain a queue to store the sentence representations from momentum-update encoder.

This mechanism allows us to store as much negative samples as possible without re-computation. Once the queue is full, we replace the "oldest" negative sample with a "fresh" one encoded by the momentum-encoder.

Similar to the works based on continuous augmentation, at the very beginning of the framework, PT-BERT takes input sentence s and obtained $\mathbf{h_i}$ and $\mathbf{h'_i}$ with two different encoder functions. We measure the loss function with:

$$\ell_i = -\log \frac{e^{sim(\mathbf{h_i}, \mathbf{h'_i})/\tau}}{\sum_{j=1}^{M} e^{sim(\mathbf{h_i}, \mathbf{h_{j'}})/\tau}}, \quad (4)$$

where $\mathbf{h_i}$ denotes the representations extracted from the gradient-update encoder, $\mathbf{h'_i}$ represents the sentence embedding in the queue, and M is the queue size. Our gradient-update and momentum-update encoder is based on the pre-trained language model with the same structure and dimensions as BERT-base-uncased (Devlin et al., 2019). The momentum encoder will update its parameters similar to MoCo:

$$\theta_k \leftarrow \lambda\theta_k + (1 - \lambda)\theta_q, \quad (5)$$

where θ_k is the parameter of the momentum-contrast encoder that maintains the dictionary, θ_q is the query encoder that updates the parameters with gradients, and λ is a hyperparameter used to control the updating process.

4 Experiments

In this section, we perform the standard semantic textual similarity (STS) (Agirre et al., 2012, 2013, 2014, 2015, 2016) tasks to test our model. For all tasks, we measure the Spearman's correlation to compare our performance with SimCSE (Gao et al., 2021). In the following, we will describe the training procedure in detail.

4.1 Training Data and Settings

Datasets. Following SimCSE, We train our model on 1-million sentences randomly sampled

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Discrete Augmentation								
CLEAR	49.00	48.90	57.40	63.60	65.60	72.50	**75.60**	61.80
MoCo	68.35	81.42	73.34	81.63	78.61	76.40	68.50	75.46
MoCo+reorder	66.14	80.06	73.14	81.35	76.01	73.99	65.76	73.78
MoCo+duplication	65.88	82.24	73.34	81.49	77.48	76.29	68.86	75.08
MoCo+deletion	67.86	81.43	72.8	81.48	77.84	76.91	69.46	75.40
MoCo+SRL	68.92	82.20	73.67	81.58	78.73	77.63	71.07	76.26
Continuous Augmentation								
CT-BERT	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
SimCSE-BERT$_{\text{base}}$	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
PT-BERT$_{\text{base}}$	**71.20**	**83.76**	**76.34**	**82.63**	**78.90**	**79.42**	71.94	**77.74**

Table 3: Sentence embedding performance on STS tasks with Spearman's correlation measured. We highlight the highest number for each methods. CLEAR (Wu et al., 2020) is trained on both English Wikipedia and Book Corpus with 500k steps with their own version of pre-trained models. Result of CT-BERT (Carlsson et al., 2021) is based on the settings of SimCSE (Gao et al., 2021)

Models	STS-B dev
SimCSE-BERT$_{base}$ + None	82.50
SimCSE-BERT$_{base}$ + Crop	77.80
SimCSE-BERT$_{base}$ + Deletion	75.90
MoCo-BERT$_{base}$ + None	82.03
MoCo-BERT$_{base}$ + Reorder	81.89
MoCo-BERT$_{base}$ + Duplication	81.82
MoCo-BERT$_{base}$ + Deletion	82.97
MoCo-BERT$_{base}$ + SRL	82.40
PT-BERT$_{base}$	**84.50**

Table 4: Results on STS-B development sets. Results of SimCSE (Gao et al., 2021) are reported from original paper.

from English Wikipedia, and evaluate the model every 125 steps to find the best checkpoints. Note that we do not fine-tune our model on any dataset, which indicates that our method is completely unsupervised.

Hardware and schedule. We train our model on the machine with one NVIDIA V100s GPU. Following the settings of SimCSE (Gao et al., 2021), it takes 50 minutes to run an epoch.

4.2 Implementations

In this subsection, we implement PT-BERT based on Huggingface transformers (Wolf et al., 2020) and initialize it with the released BERT$_{base}$ (Devlin et al., 2019). We initialize a new embedding for pseudo tokens with 128×768. During training, we create a pseudo sentence $\{0, 1, 2, ..., 127\}$ for every input and map the original sentence to this pseudo sentence by attention. With batches of 64 sentences and an additional dynamically maintained queue of 256 sentences, each sentence has one positive sample and 255 negative samples. Adam (Kingma and Ba, 2014) optimizer is used to update the model parameters. We also take the original dropout strategy of BERT with rate $p = 0.1$. We set the momentum for the momentum-encoder with $\lambda = 0.885$.

4.3 Evaluation Setup

We evaluate the fine-tuned BERT encoder on STS-B development sets every 125 steps to select the best checkpoints. We report all the checkpoints based on the evaluation results reported in Table 4. The training process is fully unsupervised since no training corpus from STS is used. During the evaluation, we also calculate the trends of alignment-loss and uniformity-loss. Losses were compared with SimCSE (Gao et al., 2021) under the same experimental settings. After training and evaluation, we test models on 7 STS tasks: STS 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014). We report the result of Spearman's correlation for all the experiments.

4.4 Main Results and Analysis

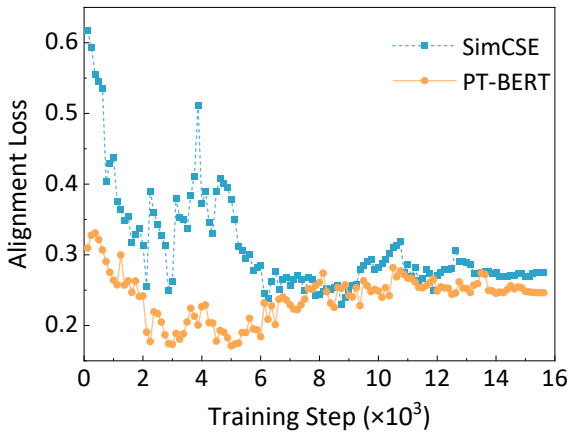
We first compare PT-BERT with our baseline: MoCo framework + BERT encoder (MoCo-BERT). MoCo-BERT could be seen as a version of PT-BERT without pseudo token embeddings. Then we apply traditional discrete augmentations such as re-

Method	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
(a) Ablation studies on pseudo sequence length								
L-64	67.04	82.04	73.65	81.12	78.64	77.35	71.33	75.88
L-90	68.94	82.08	74.53	81.22	79.06	78.01	71.49	76.48
L-128(Ours)	**71.20**	**83.76**	**76.34**	**82.63**	**78.90**	**79.42**	**71.94**	**77.74**
L-256	67.09	82.25	72.63	81.48	78.55	77.30	69.53	75.55
L-360	68.90	82.21	73.77	81.31	77.50	77.22	69.32	75.75
(b) Ablation studies on queue size								
Q-192	70.29	**83.78**	75.98	82.13	78.48	78.91	72.53	77.44
Q-256(Ours)	71.20	83.76	**76.34**	82.63	**78.90**	**79.42**	**71.94**	**77.74**
Q-320	**71.71**	83.36	75.00	**82.99**	78.76	79.17	**72.85**	77.69
(c) Evaluations on hard sentence pairs with different length								
SimCSE	66.54	78.50	68.76	70.27	71.31	-	-	71.08
PT-BERT	**72.02**	**80.24**	**72.92**	**74.50**	**72.50**	-	-	**74.44**

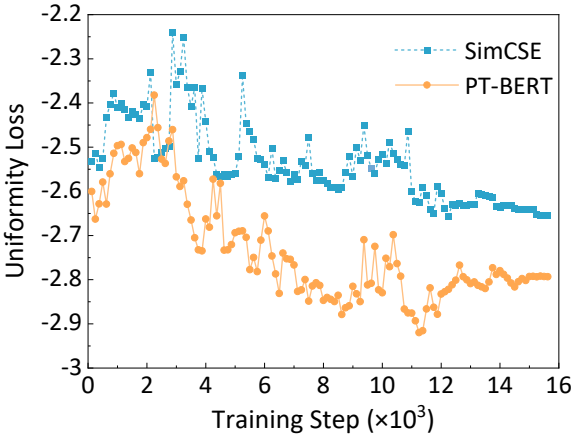
Table 5: Evaluation results of ablation studies and hard sentence pairs.

order, duplication, and deletion on this framework. We also compare our work with CLEAR (Wu et al., 2020) that substitutes and deletes the token spans. Besides, we argue that the performance of these methods is too weak. We additionally propose an advanced discrete augmentation approach that produces positive examples with the guidance of Semantic Role Labeling (SRL) (Gildea and Jurafsky, 2002) information, instead of random deletion and reordering. SRL-guided augmentation could compensate the errors caused by these factors, acting as a combination of deletion, duplication, and reordering with better accuracy. For the sentences with multiple predicates, we keep all the sets with order [ARG0, PRED, ARGM − NEG, ARG1] and concatenate them into a new sequence. For the sentences without recognized predicate-argument sets, we keep the original sentence as positive examples. In addition to the work based on discrete approaches, we also compare with SimCSE (Gao et al., 2021) which continuously augment sentences with *dropout*. In Table 3, PT-BERT with 128 pseudo tokens further pushed the state-of-the-art results to 77.74% and significantly outperformed SimCSE over six datasets.

In Fig 3, we observe that PT-BERT also achieves better alignment and uniformity against SimCSE, which indicates that pseudo tokens really help the learning of sentence representations. In detail, alignment and uniformity are proposed by (Wang and Isola, 2020) to evaluate the quality of representations in contrastive learning. The calculation of these two metrics are shown in the following



(a) Alignment loss comparison on STS-B



(b) Uniformity loss comparison on STS-B

Figure 3: Alignment and uniformity loss plot for PT-BERT and SimCSE. We visualize the checkpoints every 125 training steps. For both measurements, lower numbers are better.

formulas:

$$L_{alignment} = \underset{(x,x^+)\sim p_{pos}}{E} ||f(x) - f(x^+)||^2, \quad (6)$$

$$L_{uniformity} = \underset{(x,y)\sim p_{data}}{E} e^{-2||f(x)-f(y)||^2}, \quad (7)$$

where (x, x^+) is the positive pair, (x, y) is the pair consisting of any two different sentences in the whole sentence set, $f(x)$ is the normalized representation of x. We employ the final embedding **h** to calculate these scores.

According to the above formulas, lower alignment loss means a shorter distance between the positive samples, and low uniformity loss implies the diversity of embeddings of all sentences. Both are our expectations for the representations based on contrastive learning. To evaluate our model's performance on alignment and uniformity, we compare it with SimCSE on the STS-benchmark dataset (Cer et al., 2017), and the result is shown in Figure 3. The result demonstrates that PT-BERT outperforms SimCSE on these two metrics: our model has a lower alignment and uniformity than SimCSE in almost all the training steps, which indicates that the representations produced by our model are more in line with the goal of the contrastive learning.

5 Analysis

5.1 Ablation Studies

In this section, we first investigate the impact of different sizes of pseudo token embeddings. Then we would like to report the performance difference caused by queue size under the MoCo framework.

Pseudo Sentence Length Different lengths of pseudo tokens can affect the ability of the model to express the sentence representations. By mapping the original sentences to various lengths of pseudo tokens, the performance of PT-BERT could be different. In this section, we keep all the parts except the pseudo tokens and their embeddings unchanged. We scale the pseudo sequence length from 64 to 360. Table 5(a) shows a comparison between different lengths of pseudo sequence in PT-BERT. We find that during training, PT-BERT performs better when attending to pseudo sequences with 128 tokens. Too few pseudo tokens do not fully explain the semantics of the original sentence, while too many pseudo tokens increase the number of parameters and over-express the sentence.

Queue Size The introduction of more negative samples would make the model's training more reliable. By training with different queue sizes, we report the result of PT-BERT with different performances due to the number of negative samples. In Table 5(b), queue size $q = 4$ performs best. However, the difference in performance between the three sets of experiments is not large, suggesting that the model can learn well as long as it can see enough negative samples.

5.2 Exploration on Hard Examples with Different Length

To prove the effectiveness of PT-BERT that could weaken the hints caused by textual similarity, we further test PT-BERT on the sub-dataset introduced in Sec. 3.1. We sorted out the positive pairs with a length difference of more than five words and negative pairs of less than two words from STS-(2012-2016). PT-BERT significantly outperforms SimCSE with 3.36% Spearman's correlation, indicating that PT-BERT could handle these hard examples better than SimCSE. This further proves that PT-BERT could debias the spurious correlation introduced by sentence length and syntax, and focus more on the semantics.

6 Conclusion

In this paper, we propose a semantic-aware contrastive learning framework for sentence embeddings, termed PT-BERT. Our proposed PT-BERT approach is able to weaken textual similarity information, such as sentence length and syntactic structures, by mapping the original sentence to a fixed pseudo sentence embedding. We provide analysis of these factors on methods based on continuous and discrete augmentation, showing that PT-BERT augments sentences more accurately than discrete methods while considering more semantics instead of textual similarity than continuous approaches. Lower uniformity loss and alignment loss prove the effectiveness of PT-BERT and further experiments also show that PT-BERT could handle hard examples better than existing approaches.

Providing a new perspective to the continuous data augmentation in sentence embeddings, we believe our proposed PT-BERT has great potential to be applied in broader downstream applications, such as text classification, text clustering, and sentiment analysis.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 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, Denver, Colorado. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. 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, Montréal, Canada. Association for Computational Linguistics.

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, Atlanta, Georgia, USA. Association for Computational Linguistics.

Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *ArXiv*, abs/1907.02893.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

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

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, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. 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 1173–1178, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *CoRR*, abs/2012.15723.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Comput. Linguist.*, 28(3):245–288.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 3294–3302, Cambridge, MA, USA. MIT Press.

Tassilo Klein and Moin Nabi. 2020. Contrastive self-supervised learning for commonsense reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7517–7523, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv:2103.10385*.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 216–223. European Language Resources Association (ELRA).

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779, Brussels, Belgium. Association for Computational Linguistics.

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.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. 2020. Learning from failure: Training debiased classifier from biased classifier. In *Advances in Neural Information Processing Systems*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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*. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le

Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents.

Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance-level discrimination.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196, Florence, Italy. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.