Sentence Representation Learning with Generative Objective rather than Contrastive Objective

Bohong Wu[1,2], Hai Zhao[1,2,*]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University
chengzhipanpan@sjtu.edu.cn,zhaohai@cs.sjtu.edu.cn

Abstract

Though offering amazing contextualized token-level representations, current pre-trained language models take less attention on accurately acquiring sentence-level representation during their self-supervised pre-training. However, contrastive objectives which dominate the current sentence representation learning bring little linguistic interpretability and no performance guarantee on downstream semantic tasks. We instead propose a novel generative self-supervised learning objective based on phrase reconstruction. To overcome the drawbacks of previous generative methods, we carefully model intra-sentence structure by breaking down one sentence into pieces of important phrases. Empirical studies show that our generative learning achieves powerful enough performance improvement and outperforms the current state-of-the-art contrastive methods not only on the STS benchmarks, but also on downstream semantic retrieval and reranking tasks. Our code is available at https://github.com/chengzhipanpan/PaSeR.

1 Introduction

Sentence Representation Learning has long been a hot research topic (Conneau et al., 2017; Cer et al., 2018), for its effectiveness in a variety of downstream tasks like information retrieval and question answering (Yang et al., 2018).

Although pre-trained language models (PrLMs) like BERT (Devlin et al., 2019) have achieved overwhelming performance on various token-level tasks, they are also criticized for being unable to produce high-quality sentence-level representations. Research (Li et al., 2020) has shown that the native sentence representation produced by the "[CLS]" token of BERT shows extremely poor performance on sentence evaluation benchmarks like semantic textual similarity (STS) tasks.

The primary cause of these low-quality sentence representations is the lack of effective self-supervised sentence-level training objectives. As discussed in ConSERT (Yan et al., 2021), the original sentence-level pretraining objective Next Sentence Prediction (NSP) is too weak to provide high-quality sentence representation. Therefore, recent researchers are seeking other effective self-supervised sentence-level objectives (Carlsson et al., 2020; Yan et al., 2021; Gao et al., 2021). Generally, self-supervised methods include both (i) generative methods, like Masked Language Modelling (MLM), and (ii) contrastive methods, like Next Sentence Prediction (NSP).

By treating one sentence as a whole and contrasting sentence representations with each other in the same training batch, contrastive methods have been shown extremely effective in Sentence Representation Learning in recent years. Generally, contrastive methods often use various data augmentation techniques to create different views for one sentence, and align the representations of these views within the same batch. ConSERT (Yan et al., 2021) utilizes techniques including token shuffling, feature cutoff, etc., and provides a general contrastive learning framework for Sentence Representation Learning. SimCSE (Gao et al., 2021) suggests using different dropout masks is a simple yet more powerful augmentation technique for creating different views of one sentence.

Though effective, there also exist several drawbacks in contrastive methods. (i) The training procedure of contrastive methods lacks enough linguistic interpretability. What information is encoded into the sentence representation is kept unknown. (ii) As suggested in TSDAE (Wang et al., 2021), good performance on STS tasks by contrastive methods does not ensure good performance on downstream tasks like semantic retrieval and reranking because of the obvious inductive bias.

On the contrary, generative self-supervised learn-

ing techniques offer researchers good interpretability and controllable training by enabling the choice of what to generate. However, although generative methods like MLM have achieved overwhelming performance in token representation learning, little effort has been put into investigating the potential of generative methods in sentence representation learning. Cross-Thought (Wang et al., 2020) and CMLM (Yang et al., 2021) are the most representative generative methods, which both leverage the contextual sentence representations to recover the masked tokens in one sentence. Unfortunately, they highly depend on the contextual information, and mainly focus on the document-level corpus, thus performing unsatisfyingly in STS tasks where representations of short texts are valued. Latter, TSDAE (Wang et al., 2021) proposes to use a denoising auto-encoder to recover the original sentence from the corrupted version. Although TSDAE does not depend on document-level corpus anymore, it still suffers from inferior performance on the STS benchmarks.

We attribute the inferior performance of existing generative methods to the wrong modeling perspective. The existing generative methods still follow the training paradigm of contrastive methods, which considers one sentence as an inseparable whole, and learn sentence representations from the **inter-sentence perspective**. In this paper, we novelly propose to model sentence representation learning from the **intra-sentence perspective**. We especially emphasize the importance of the semantic components within the sentence, i.e. phrases. We therefore present **P**hrases-**a**ware **Se**ntence **R**epresentation (PaSeR), which explicitly encodes the representation of the most important phrases into sentence representations. In detail, we hypothesize a good sentence representation should be able to encode and reconstruct the important phrases in the sentence when given a suitable generation signal. Inspired by SimCSE (Gao et al., 2021), we provide such generation signals by our *duplication and masking* strategy. As shown in Figure 1, we mask out important phrases in the original sentences, and encode these masked sentences to provide signals for phrase reconstruction. Experiments show that our PaSeR achieves the SOTA performance on multiple single tasks in STS in both unsupervised and supervised settings, and especially better average STS performance than SimCSE in the supervised setting. Extensive ex-
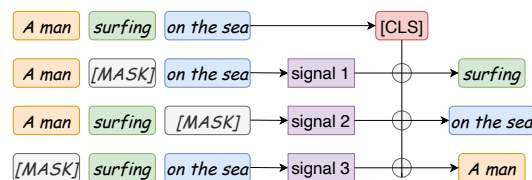

Figure 1: A description of the design intuition of our PaSeR.

periments further present that our PaSeR achieves better performances on downstream tasks including semantic retrieval and reranking.

Contributions (i) We propose an effective generative self-supervised objective of training sentence representations without leveraging document-level corpus. Based on such an objective we present PaSeR, a Phrase-aware Sentence Representation Learning model. (ii) Experiments show that our proposed PaSeR achieves SOTA performance on multiple single tasks in STS, and especially better average STS performance than previous best contrastive method, SimCSE, in the supervised setting. Our PaSeR provides an effective alternative for Sentence Representation Learning against the current trend of contrastive methods.

2 Related Work

2.1 Supervised Sentence Representations

Supervised sentence representations leverage the idea of transfer learning. Previous works (Conneau et al., 2017; Cer et al., 2018) have shown that utilizing labeled datasets from Natural Language Inference (NLI) is extremely helpful for Sentence Representation Learning. Based on these researches, Sentence-BERT (Reimers and Gurevych, 2019) introduces siamese BERT encoders with shared parameters and trains them on NLI datasets, achieving acceptable performance on STS tasks. Although these supervised methods can provide high-quality sentence representations, the labeling cost of sentence pairs still urges the researchers to search for a more effective unsupervised solution.

2.2 Post-processing of BERT Representations

Several post-processing methods are first proposed to improve the sentence representations produced by original BERT. Generally, these methods analyze the distorted sentence representation space, and propose changing the representation space to isotropic Gaussian ones via flow methods like BERT-flow (Li et al., 2020) or simple projection

methods like BERT-whitening (Su et al., 2021). However, their performance is very limited, as their sentence representations are not finetuned due to the lack of suitable sentence-level training objectives in the original BERT model.

2.3 Self-supervised Sentence-level Pre-training

Recently, researchers are seeking more effective sentence-level pre-training objectives, from the aspects of both generative ones and contrastive ones.

Generative Methods Little efforts have been paid into studying what generation methods can achieve in Sentence Representation Learning. Among these works, Cross-thought (Wang et al., 2020) and CMLM (Yang et al., 2021) are the most representative ones, which both propose to recover masked tokens of one sentence by the contextual-sentence representations. However, in both methods, document-level training data are needed, making it unsuitable for evaluating the similarity between short texts. Recently TSDAE (Wang et al., 2021) also present a generative method, which aims to recover the original sentence from a corrupted version. Although TSDAE doesn't need contextual texts anymore, it suffers from inferior performance on the STS benchmarks.

Contrastive Methods Recently, contrastive learning has presented its superiority in Sentence Representation Learning. Generally, existing contrastive methods are seeking effective ways of creating different views of one sentence, pushing their representations closer while pulling views of different sentences away. Contrastive Tension (CT) (Carlsson et al., 2020) introduces the Siamese network structure to create different views of one sentence, and treat different views from one sentence as positive pairs while others as negative pairs. ConSERT (Yan et al., 2021) creates different views of sentences by data augmentation techniques including token shuffling, feature cutoff, and adversarial attacks. After that, SimCSE (Gao et al., 2021) presents that the original dropout mask design is already a very effective data augmentation strategy, and has achieved the SOTA performance on the STS benchmarks.

3 Method

3.1 Data Pre-processing

3.1.1 Phrase Extraction

Phrase extraction is the core component of our PaSeR. Which phrase to generate directly determines what information we encode in the sentence representation. In this paper, we mainly use two off-the-shelf methods to extract the important phrases.
• A random sub-tree from the syntax parsing tree of one sentence. By using NLTK (Loper and Bird, 2002), we can easily extract components of one sentence including Subordinate Clauses (SBAR), Verb Phrases (VP), and Noun Phrases (NP).
• A statistical key-phrase extraction algorithm, RAKE (Rose et al., 2010). RAKE first tokenizes the original sentence or document into phrases using stopwords or punctuations. To acquire the importance of each phrase, RAKE first constructs the co-occurrence matrix of words, and computes the importance of word w via its degree $deg(w)$ in the co-occurrence matrix and its frequency $freq(w)$ in the document or sentence by

$$wordScore(w) = deg(w)/freq(w)$$

Finally, the importance of one phrase is calculated by summation of $wordScore$ on all words in the phrase.

Experimentally, the second method achieves much better performance on the STS tasks. Therefore, if not specified, we use the second method for phrase extraction. We will discuss the performance difference between these two extraction methods in Section 4.5.3.

3.1.2 Duplicate and Masking

For the motivation of recovering important phrases within one sentence, we propose to use difference modeling. In detail, for one given sentence s, which is composed of multiple phrases $\mathcal{P} = \{p_0, p_1, ..., p_n\}$, ordered by their importance calculated by RAKE (Rose et al., 2010). To recover the most important phrase like p_0, it is natural to come up with the following equality:

$$p_0 = \mathcal{P} - \mathcal{P}/\{p_0\} \qquad (1)$$

Therefore, we duplicate such s as $\tilde{s}$, but mask out the most important phrase p_0 that we need to generate, shown in the left part of Figure 2. Denote the sentence encoder as Enc, we can get the sentence representation of $E_s = f_{Enc}(s)$ and $E_{\tilde{s}} = f_{Enc}(\tilde{s})$.
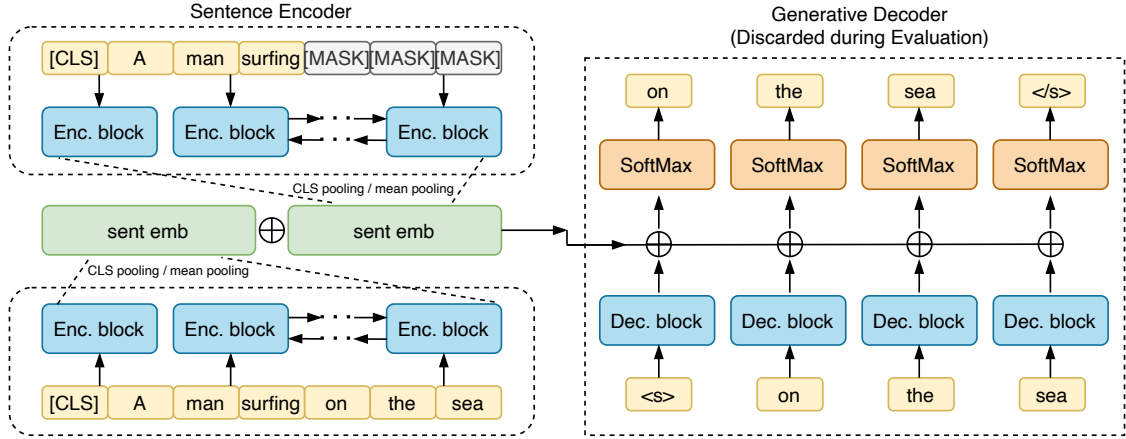
Figure 2: Overview of PaSeR. The left part presents the *Sentence Encoders* where parameters are shared. The right part presents the *Generative Decoder*, which will be discarded during evaluation. Therefore, the extra parameters are only used during the training stage, and do not affect the inference speed at the evaluation stage.

By combining the representations of both E_s and $E_{\tilde{s}}$, we can recover the masked phrases p_0 with a suitable transformer decoder.

3.1.3 Data Augmentation

To improve the robustness of the sentence representations produced by our PaSeR, following EDA (Wei and Zou, 2019), we introduce data augmentation on both s and $\tilde{s}$ before the paired sentences are fed into the sentence encoder. We mainly use three types of data augmentation strategies including *Synonym Replacement*, *Random Deletion* and *Token Reordering*.

We speculate that, (i) Using *Synonym Replacement* on both s and $\tilde{s}$ is an effective strategy to create semantic similar phrases with different tokens, which helps the model capture the semantic similarities instead of token similarities. (ii) *Random Deletion* strategy can well alleviate the effect brought by frequent words or phrases. (iii) *Token Reordering* strategy can make our sentence encoder less sensitive to token orders and changes in positional embeddings.

3.2 Unsupervised PaSeR

3.2.1 Sentence Encoder

Following previous works (Li et al., 2020; Su et al., 2021), our sentence encoders are based on the pre-trained language model, BERT (Devlin et al., 2019). The pooling methods include (i) directly using the "[CLS]" token representation, (ii) averaging the token representations in the last layer of BERT, (iii) using a weighted average of token representations from the intermediate layers of BERT, and

we choose the best pooling method based on its performance on the STS tasks.

3.2.2 Decoding Signal

After acquiring the sentence representation of both E_s and $E_{\tilde{s}}$, the way of combining these two representations also plays an important role in sentence representation learning. Following SBERT (Reimers and Gurevych, 2019) but a step further, we use a weighted combination of $E_s, E_{\tilde{s}}, |E_s - E_{\tilde{s}}|, E_s * E_{\tilde{s}}$ to create the decoding signal for the following generative decoder:

$$Signal_{Dec} = [E_s, E_{\tilde{s}}, m * |E_s - E_{\tilde{s}}|, n * |E_s * E_{\tilde{s}}|] \tag{2}$$

Here, m and n are scaling factors to normalize these four decoding signals, and both variables are selected by grid search. We will discuss the selection of m and n in Appendix A.

3.2.3 Generative Decoder

The generative decoder performs as a representation regularizer for training the sentence encoder, and can be discarded during the evaluation stage. Therefore, the decoder does not add up any additional hyperparameters for downstream tasks. In our experiments, we use variants of Transformer (Vaswani et al., 2017) decoders as our phrase reconstruction decoder Dec. Suppose now the masked phrase p_0 is composed of several tokens $\{t_1, t_2, ..., t_k\}$, and given the decoding signal $Signal_{Dec}$, the phrase reconstruction process is
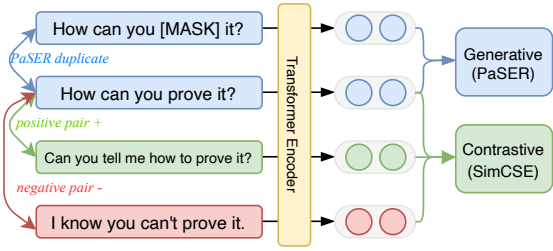
Figure 3: PaSeR baseline in the supervised setting.

formulated as:

$$L_{generative} = -\sum_{i=1}^{k} \log P_{f_{Dec}}\left(t_i \mid t_{<i}, Signal_{Dec}\right)$$
$$(3)$$

3.2.4 Combined with MLM

To preserve the quality of token-level representation, we also incorporate the MLM objective with our reconstruction objective. The final training loss is a combination:

$$L_{total} = L_{MLM} + L_{generative} \qquad (4)$$

3.3 Supervised PaSeR

For supervised settings, our PaSeR loss design can be easily incorporated with the frontier supervised methods. Moreover, our unsupervised PaSeR can provide a good initializing checkpoint for training the supervised sentence encoder.

Generally, methods of incorporating supervised signals in sentence representation learning can be divided into two types. (i) A sequence classification training objective following SBERT (Reimers and Gurevych, 2019). (ii) A contrastive learning objective following SimCSE (Gao et al., 2021). Given the prominent performance of the latter approach, we combined the contrastive loss introduced in Sim-CSE with our generative PaSeR loss. The training process is shown in Figure 3. We initialize the sentence encoder from the best checkpoint of our unsupervised PaSeR. The final loss function is formulated as:

$$L_{supervise} = L_{contrastive} + \alpha L_{generative} \quad (5)$$

where α is an adjustable hyper-parameter that is searched in our experiments.

4 Experiment

4.1 Evaluation Datasets

Semantic Textual Similarity (STS) For the similarity evaluation of sentence representations, following previous works (Su et al., 2021; Gao et al.,

2021; Yan et al., 2021), we use the Semantic Textual Similarity (STS) datasets as our evaluation benchmark, including STS tasks 2012-2016 (STS12-STS16) (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (STS-B) (Cer et al., 2017) and SICK-Relatedness (SICK-R) (Marelli et al., 2014). Samples in datasets are paired sentences with human-labeled relatedness scores from 0 to 5. We use the spearman correlation x 100 on these 7 STS datasets to evaluate and compare the performance between all baselines and frontier researches.

Quora Question Pair Quora Question Pair dataset [1] (QQP) consists of over 400,000 lines of potential question duplicate pairs, denoted as (q_1, q_2). We collect all the q_2s as the question corpus, and all q_1s that have at least a positive paired q_2 as the query set. We then use the query set to retrieve similar questions from the question corpus. The evaluation metrics contain Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR).

AskUbuntu Question AskUbuntu Question dataset is a semantic reranking dataset, which contains a pre-processed collection of questions taken from AskUbuntu.com[2] 2014 corpus dump. Different from QQP, the question corpus for each query is given with the size of 20, and models are required to re-rank these 20 given questions according to the similarity measurement. We also use MAP and MRR as evaluation metrics.

4.2 Training Details

We use BERT-base (Devlin et al., 2019) as the sentence encoder for all experiments. Following Sim-CSE (Gao et al., 2021), we restrict the maximum sequence length to 32 with an initial learning rate of 3e-5. The batch size is selected from [32, 64, 96]. For training in the unsupervised setting, following ConSERT (Yan et al., 2021), we mix all the unlabeled texts from the seven STS datasets as the training data. For training in the supervised setting, given the superior performance of SimCSE (Gao et al., 2021), we also use a combination of SNLI dataset (Bowman et al., 2015) and MNLI dataset (Williams et al., 2018), and combine the contrastive learning framework of SimCSE with our PaSeR to train the supervised sentence encoders.

Following previous works (Yan et al., 2021; Gao et al., 2021), we use the development set of STS-

---

[1] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs
[2] https://askubuntu.com/

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Original BERT/Glove								
GloVe embeddings	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
$BERT_{base}$-[CLS]	21.54	32.11	21.28	37.89	44.24	20.29	42.42	31.40
$BERT_{base}$-mean	30.87	59.89	47.73	60.29	63.73	47.29	58.22	52.57
$BERT_{base}$-first-last avg.	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
Post-Processing Methods								
BERT-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
Contrastive Methods								
CT-$BERT_{base}$	66.86	70.91	72.37	78.55	<u>77.78</u>	-	-	-
$ConSERT_{base}$	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE-$BERT_{base}$ + MLM	44.40	66.60	51.27	67.48	67.95	52.44	59.86	58.57
SimCSE-$BERT_{base}$	68.40	<u>82.41</u>	**74.38**	<u>80.91</u>	**78.56**	76.85	72.23	**76.25**
Generative Methods								
CMLM-$BERT_{base}$	58.20	61.07	61.67	73.32	74.88	76.60	64.80	67.22
PaSeR-$BERT_{base}$ (Our Method)	**70.21**	**83.88**	<u>73.06</u>	**83.87**	77.60	**79.19**	65.31	<u>76.16</u>

Table 1: Unsupervised sentence representation performance on STS tasks. Bold statistics represent the best performance while underlined ones represent the second-best performance.

B to choose the best-performing model. If not specified, we take the "[CLS]" representation as the sentence representation for most of the experiments, and discuss different effects when different pooling methods are adopted in Appendix B.

For the generative decoder, any type of transformer (Vaswani et al., 2017) decoder is acceptable. If not specified, we use a 6-layer transformer decoder as the generative decoder, and the word embedding layer is shared between the sentence encoder and the generative decoder. We also present a discussion on how the complexity of the decoder affects the performance of the sentence encoder in Section 4.5.1. The weights of the generative decoder are randomly initialized.

4.3 Results on Semantic Textual Similarity

Unsupervised Settings The performance of our PaSeR and other frontier researches are presented in Table 1. Here, we separate all these frontier researches into four categories. (i) Original baselines including different pooling methods of BERT and Glove embeddings. (ii) Post-processing baselines including BERT-flow (Li et al., 2020) and BERT-whitening (Su et al., 2021). (iii) Contrastive methods including CT-$BERT_{base}$ (Carlsson et al., 2020), ConSERT (Yan et al., 2021) and SimCSE (Gao et al., 2021). (iv) Generative methods including CMLM (Yang et al., 2021) and our PaSeR.

From Table 1, (i) Under the unsupervised setting, our PaSeR achieves the SOTA performance on several benchmark datasets like STS12, STS13, STS15, STS-B, and is also the second-best model considering the average performance. (ii) Unlike our PaSeR which is naturally combined with MLM, SimCSE suffers from significant performance degradation on the STS benchmarks when combined with the MLM objective. (iii) Compared to the previous best generative method CMLM, PaSeR achieves an average of 8.94 absolute performance gain. Such improvement especially presents the superiority of the intra-sentence modeling perspective within the scope of generative methods.

Supervised Settings The results of supervised sentence encoders are shown in Table 2. By initiating the sentence encoder from a previous best unsupervised checkpoint, our PaSER can achieve an average of 1.04 performance gain on the STS benchmark, compared with the SimCSE baseline. The results demonstrate that using our PaSeR design is an effective self-supervised learning objective that could provide high-quality sentence representation during the pre-training stage.

4.4 Results on Semantic Retrieval/Reranking

Because sentence representation learning has broad application scenarios, the performance on the STS tasks only is not enough to present the quality of

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Universal Sentence Encoder	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
SBERT$_{base}$	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT$_{base}$-flow	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
SBERT$_{base}$-whitening	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
CT-SBERT$_{base}$	74.84	83.20	78.07	83.84	77.93	81.46	76.42	79.39
ConSERT-BERT$_{base}$	74.07	83.93	77.05	83.66	78.76	81.36	76.77	79.37
SimCSE-BERT$_{base}$	75.30	84.67	80.19	85.40	80.82	**84.25**	**80.39**	81.57
PaSeR-BERT$_{base}$ (Our Method)	**78.40**	**85.80**	**80.79**	**86.36**	**82.95**	83.67	80.31	**82.61**

Table 2: Supervised sentence representation performance on STS tasks. Bold statistics represent the best performance among all baselines.

Method	QQP		AskUbuntu	
	MAP	MRR$_{@10}$	MAP	MRR$_{@10}$
Unsupervised Baselines				
BERT$_{base}$	72.15	32.50	41.51	53.80
BERT-whitening$_{base}$	74.67	61.43	46.15	60.11
SimCSE-BERT$_{base}$	75.42	62.69	**50.63**	63.49
PaSeR-BERT$_{base}$	**76.30**	**65.07**	50.51	**64.56**
Supervised Baselines				
SimCSE-BERT$_{base}$	76.17	63.94	51.50	66.16
PaSeR-BERT$_{base}$	**76.22**	**65.13**	**52.28**	**67.31**

Table 3: Downstream performance on semantic retrieval and reranking datasets, including QQP and AskUbuntu.

Figure 4: Ablations on the effects of the complexity of the generative decoder.

sentence representations. According to TSDAE (Wang et al., 2021), good STS performance does not necessarily correlate with good performance on downstream semantic retrieval or reranking task, as there exists obvious inductive bias. Therefore, in this section, we conduct extensive experiments on semantic retrieval on the Quora Question Pairs dataset, and semantic reranking on the AskUbuntu dataset. We compare the performance between our PaSeR and other frontier works including SimCSE and BERT-whitening.

Table 3 presents the performance of all models on both datasets. Compared to the previous best contrastive method SimCSE, our PaSeR achieves better performance in both supervised and unsupervised settings. Better results on both semantic retrieval and reranking indicate that our PaSeR is better at ranking sentences with similar meanings, which is a core feature that can not be present by STS benchmarks, but is extremely valued in semantic retrieval and reranking.
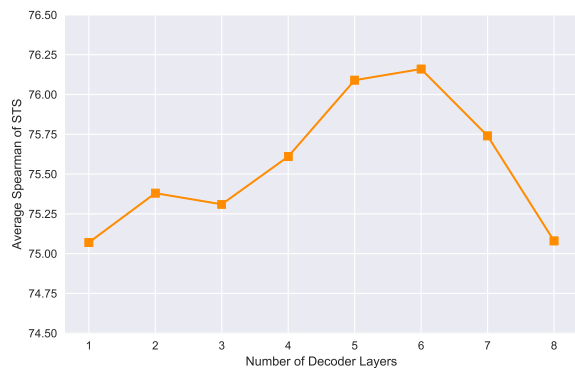
4.5 Ablation Study

4.5.1 Complexity of Generative Decoder

Inspired by the design of Electra (Clark et al., 2020), we further conduct extensive experiments to study the effects brought by the complexity of generative decoders. We vary the layer number of the generative decoder from 1 to 8, and train all versions for 5 epochs on the dataset. The evaluation metric is the average Spearman's correlation of the whole 7 STS tasks.

The results are shown in Figure 4. From the figure, we can see that the complexity of generative decoders largely affects the performance of the sentence encoder. The sentence encoder achieves its best performance when the layer number is set close to 6. We speculate this is because too small generators lack enough model capacity, while too large generators tend to cause overfitting on the training data.

4.5.2 Effectiveness of Data Augmentation

In this section, we compare the effects of different data augmentation strategies, shown in Table 4.

Model	STS12-16	STS-B	SICK-R
BERT-[CLS]	31.4	20.3	42.4
PaSeR$_{base}$-no aug	73.8	74.5	62.5
+ RS	74.2	75.0	63.5
+ RD	74.2	75.5	63.8
+ SR	**77.7**	**79.3**	**65.4**
+ SR + RD	77.2	77.5	64.5
+ SR + RS	77.3	77.6	63.1

Table 4: Ablation study of data augmentation strategies, "STS12-16" represents the average Spearman score of STS12-16 datasets.

Phrase to Mask	STS12-16	STS-B	SICK-R
None (BERT-[CLS])	31.4	20.3	42.4
NP only	68.5	68.2	62.0
VP only	63.8	64.1	58.3
NP+VP	69.5	70.4	64.3
RAKE (top1)	73.5	73.9	62.4
RAKE (top2)	76.8	77.6	63.6
RAKE (top3)	**77.8**	**79.4**	64.4
RAKE (top4)	77.6	79.3	64.6
RAKE (top5)	77.7	79.3	**65.4**

Table 5: Ablation study of different phrases to mask in the unsupervised setting. "STS12-16" represents the average Spearman score of STS12-16 datasets.

From the experiments, (i) Compared to the original BERT baseline, our PaSeR can already achieve remarkable performance gain without any data augmentation techniques. (ii) Synonymy Replacement (SR) is extremely effective in boosting downstream performance. (iii) Random Swapping (RS) and Random Deletion (RD) also help, but with a much smaller effect. (iv) We do not observe performance gain when different data augmentation techniques are combined. We speculate this is because the semantic meaning of one sentence is more likely to change when different augmentation methods are combined, which influences the alignment of the input sentence pairs.

4.5.3 Choices of Phrases to Mask

In this section, we will discuss the effect brought by the masking choices of important phrases. As we have discussed previously, we conduct experiments to examine the two masking strategies we proposed. For masking using NLTK toolkit, we specifically conduct three experiments, including (i) masking out Noun Phrases (NP) only, (ii) masking out Verb Phrases (VP) only, (iii) both NP and VP. For masking using RAKE, we vary the number (from 1 to 5) of the most important phrases we choose to mask in the sentence.

Table 5 presents the results of each masking strategy. (i) Apparently, using RAKE as the phrase extraction method achieves significantly better performance than using NLTK toolkit. In fact, the syntax parsing based method views all the phrases in one sentence with equal importance, while it is obvious that different phrases contribute differently to the semantic meaning of one sentence. (ii) For masking using NLTK toolkit, *masking out Noun Phrases only* or *masking out Verb Phrases only* perform worse than *masking both phrases*. We

speculate this is because neither Verb Phrases nor Noun Phrases can fully cover the semantic meaning of one sentence. Our PaSeR model needs to encode information of both phrases into the sentence representation. (iii) For masking using RAKE, when compared to the BERT-[CLS] baseline, the top three phrases contribute the most in facilitating the modeling of sentence representation, as adding each one will result in remarkable performance improvement. However, adding more phrases (top 4 or top 5) results in trivial improvement. We speculate this is because the top 4 or 5 phrase contributes little important information to the semantic meaning of one sentence, especially when sentences are often short in the STS benchmark.

5 Qualitative Analysis

5.1 Sentence Retrieval

In this section, we present the qualitative analysis of the retrieval results on Quora Question Pair dataset. We showcase two examples in Table 6, where PaSeR retrieves generally better quality sentences.

In the first case, PaSeR successfully captures the semantic similarity between the phrase *What can one do* and phrase *What has worked for you* where SimCSE fails. In the second case, PaSeR captures the correlation between *emotional* and *happy/angry*, while SimCSE captures only phrase *happy* in the Top3 prediction. Both cases have demonstrated the superiority of our PaSeR in capturing semantic similar phrases between sentences.

SimCSE-BERT$_{base}$	PaSeR-BERT$_{base}$
Query: What can one do to relieve severe chronic pain?	
Is it possible to come to terms with a life of chronic pain?	*What has worked for you to help relieve chronic pain?*
What is the best way to avoid pain?	*How can I get rid of chronic and acute back pain?*
How would you/do you cope with chronic pain?	*What is the best way to ease period pain?*
Query: Is there a biological reason that people cry when they are emotional?	
Why do people cry when they feel happy?	*Why do some people cry when they get angry?*
Why do some people cry more than others?	*Why do people cry when they feel happy?*
Why do some people like crying so much?	*Why do some people like crying so much?*

Table 6: Qualitative Analysis of semantic retrieval on the QQP Dataset.

Raw Sentence	Masked version	Generate result
Work *with a tool.*	*_ with a tool*	*Try*
*At **least 89 dead** in china earthquakes*	*At _ _ _ dead in china earthquakes*	*least 7 dead.*
*Do I need a transit visa for a stop in **London**?*	*Do I need a transit visa for a stop in _?*	*UK*
*There are **two options** for you*	*There are _ _ for you*	*options options*

Table 7: Qualitative Analysis of the phrases that PaSeR decoder reconstructs.

5.2 Phrases Reconstructed by Decoder

In this section, we present what the decoder in our PaSeR can do to better illustrate the linguistic interpretability provided by our PaSeR. We sample several sentences, mask one phrase in each of them, and let the decoder reconstruct the missing part. We especially list the cases that our decoder reconstructs differently from the original text in Table 7 (most of the cases generate the same phrases as original texts).

From the table, we can see that our PaSeR decoder can learn approximately what information is missing from the given sentence representations. It learns phrase similarity between *Work* and *Try*, and the semantic connection between *London* and *UK*. Although it fails to learn the exact arithmetic number *89* and *two*, it is still able to produce a wrong arithmetic number and learn the plurality (*s* in *options*).

6 Conclusion

As most pre-trained language models fail to attach enough importance to sentence-level representation learning, it usually leads to unsatisfactory performance in downstream tasks when good sentence representation is right indispensable. Based on investigating the intra-sentence relationship between components of sentences (important phrases)

and the whole sentence representations, we propose a generative objective to align these phrases with their corresponding sentence representations. This idea leads to PaSeR, a **P**hrase-**a**ware **S**entence **R**epresentation model. As an effective alternative in Sentence Representation Learning, our PaSeR achieves comparable performance with strong contrastive learning baselines on STS tasks, and better performance on the downstream semantic retrieval and reranking tasks on datasets including QQP and AskUbuntu.

7 Limitations

We think our PaSeR has the following limitations, and leave them for future work.
• The combination of decoding signals is empirically designed. Hyperparameters m and n are selected by grid search and lack technical analysis.
• From the experiments, what phrases to mask and what augmentations on the sentences are taken can cause significant performance differences. Better masking strategies can be explored.

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.

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

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. 2020. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*.

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

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

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.

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

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *ArXiv preprint*, abs/2103.15316.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688.

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

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

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

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Ziyi Yang, Yinfei Yang, Daniel Cer, Jax Law, and Eric Darve. 2021. Universal sentence representation learning with conditional masked language model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6216–6228.

Parameter Setting	STS12-16	STS-B	SICK-R
$m = 0.1, n = 1.0$	77.2	78.5	64.6
$m = 1.0, n = 1.0$	77.6	78.7	64.8
$m = 10 , n = 1.0$	**77.7**	79.0	65.1
$m = 10 , n = 0.1$	77.6	79.0	65.1
$m = 10 , n = 10$	**77.7**	**79.3**	**65.4**

Table 8: Ablation for finding best m and n in the unsupervised setting.

A Hyperparameters for Combined Decoding Signals

According to SBERT (Reimers and Gurevych, 2019), the way of combining sentence representations for decoding signal also plays an important role in the downstream performance. In this paper, given the original sentence representation E_s and masked sentence representation $E_{\tilde{s}}$ we create the decoding signal by the following formula:

$$Signal_{Dec} = [E_s, E_{\tilde{s}}, m * |E_s - E_{\tilde{s}}|, n * |E_s * E_{\tilde{s}}|] \quad (6)$$

In this section, we use grid search to find the best performing m and n that maximize the downstream performance on the STS benchmarks. We first select m from [0.1, 1, 10], and then fix best m to select best n from [0.1, 1, 10]

The results are shown in Table 8. All experiments use a 6-layer transformer decoder and share the same hyperparameters except for m and n. Experimentally, we find that increasing m and n can lead to better performance within the range of [0.1, 10]. Best unsupervised checkpoint is acquired when both m and n are set to 10.

B Choices of Pooling Method

Previous studies (Li et al., 2020; Su et al., 2021; Gao et al., 2021) have verified that different pooling methods might lead to very different results, and different models may prefer different types of pooling methods. Therefore, we also investigate what pooling method is preferred by our PaSeR. We mainly investigate four types of pooling methods. (i) Average representation over the last layer of BERT, denoted as Top1 avg. (ii) Average representation from the last two layers of BERT, denoted as Top2 avg. (iii) Average representation from the combination of the first layer and the last layer of BERT, denoted as First-last avg. (iv) Directly using the "[CLS]" token as the sentence representation.

Pooling	STS12-16	STS-B	SICK-R
Top1 avg.	76.2	78.3	65.7
Top2 avg.	75.6	78.0	**66.1**
First-last avg.	76.2	**78.4**	65.8
[CLS]	**77.7**	**79.3**	65.4

Table 9: Ablation study of different pooling method on PaSeR.

level	BERT-[CLS]	SimCSE$_{base}$	PaSeR$_{base}$
0-1	0.86 / 0.006	0.50 / 0.013	0.55 / 0.015
1-2	0.88 / 0.006	0.67 / 0.014	0.74 / 0.011
2-3	0.89 / 0.006	0.74 / 0.010	0.81 / 0.008
3-4	0.90 / 0.004	0.81 / 0.008	0.86 / 0.005
4-5	0.90 / 0.005	0.87 / 0.006	0.92 / 0.003

Table 10: Mean/variance of cosine similarity on STS-Benchmark.

Table 9 present the results when different pooling methods are applied on PaSeR. Experimentally, we found that directly using "[CLS]" token as the final sentence representation performs the best among all the pooling methods, with nearly 1 point increase on the average performance of all STS tasks.

C Cosine Similarity Density Plots

Following SimCSE, we visualize the cosine density plots on the STS-Benchmark dataset in Figure 5. Concretely, we split the STS-B dataset into five similarity levels according to their labeled scores, and count all similarity scores in each sentence level. From Figure 5, BERT-[CLS] shows similar cosine distribution in all similarity levels, while SimCSE and PaSeR present good performance in distinguishing samples from different levels.

Theoretically, high-quality sentence representation should present two characteristics. (1) Significant mean value difference between each similarity level, which represents inter-class distance. (2) Lower variance in each similarity level, which represents smaller intra-class distance.

Table 10 presents the exact mean/var values of different models in each similarity level. We can see that both SimCSE and PaSeR achieve good inter-class distance compared to the original BERT-[CLS]. As for intra-class distance, when compared to SimCSE, PaSeR shows generally better performance on almost all similarity levels.
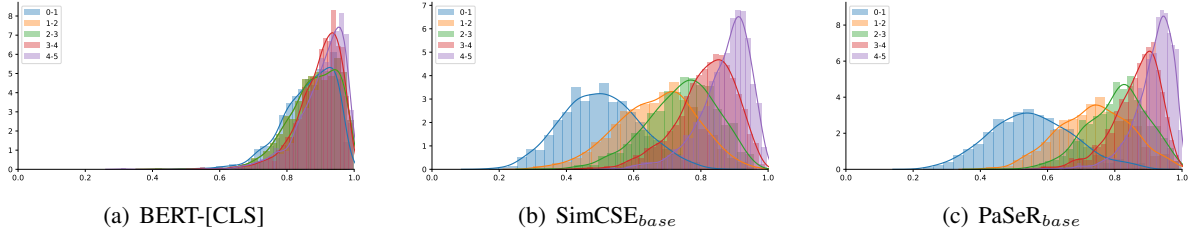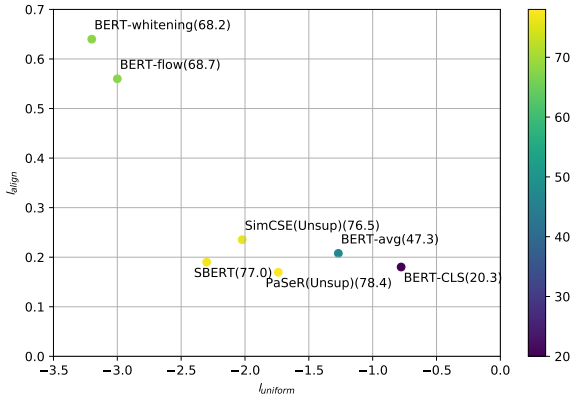
| (a) BERT-[CLS] | (b) SimCSE$_{base}$ | (c) PaSeR$_{base}$ |

Figure 5: Cosine Similarity Density Plots of different models on different similarity levels.

Figure 6: Visualization of uniformity and alignment for sentence representations produced by different methods. All models are trained on BERT$_{base}$. Color of points and numbers in brackets represent Spearman's correlation on the test set of STS-Benchmark.

D Uniformity and Alignment

Following SimCSE (Gao et al., 2021), we analyze the uniformity and alignment (Wang and Isola, 2020) of our PaSeR along with other frontier works. For "uniformity", the representations of all sentences should be approximately uniformly distributed on a unit hypersphere, in order to preserve as much information as possible. While for "alignment", similar sentences should have similar representations. We use the STS-Benchmark dataset as the evaluation corpus, and also list the corresponding Spearman correlation for each method for comparison.

From Figure 6, we can see that our PaSeR achieves the best alignment loss among all the listed models (0.17), which is even better than supervised baseline SBERT (0.19) or the SOTA unsupervised method SimCSE (0.24). For the uniformity measurement, previous works (Li et al., 2020; Su et al., 2021) have pointed out that original BERT sentence representation space *collapse*, which presents high similarity between representations of any sentence pairs. Therefore, both BERT-

avg and BERT-[CLS] suffer from high uniformity loss. When compared to BERT-[CLS] or BERT-avg, our PaSeR also achieves much better uniformity, meaning that our proposed self-supervised sentence-level training objective naturally eases the *collapse*.