arXiv:2104.08821v1 [cs.CL] 18 Apr 2021

SimCSE: Simple Contrastive Learning of Sentence Embeddings

Tianyu Gao[†*] **Xingcheng Yao**[‡*] **Danqi Chen**[†]
[†]Department of Computer Science, Princeton University
[‡]Institute for Interdisciplinary Information Sciences, Tsinghua University
{tianyug,danqic}@cs.princeton.edu
yxc18@mails.tsinghua.edu.cn

Abstract

This paper presents SimCSE, a simple contrastive learning framework that greatly advances the state-of-the-art sentence embeddings. We first describe an unsupervised approach, which takes an input sentence and predicts *itself* in a contrastive objective, with only standard dropout used as noise. This simple method works surprisingly well, performing on par with previous supervised counterparts. We hypothesize that dropout acts as minimal data augmentation and removing it leads to a representation collapse. Then, we draw inspiration from the recent success of learning sentence embeddings from natural language inference (NLI) datasets and incorporate annotated pairs from NLI datasets into contrastive learning by using "entailment" pairs as positives and "contradiction" pairs as hard negatives. We evaluate SimCSE on standard semantic textual similarity (STS) tasks, and our unsupervised and supervised models using $\text{BERT}_{\text{base}}$ achieve an average of 74.5% and 81.6% Spearman's correlation respectively, a 7.9 and 4.6 points improvement compared to previous best results. We also show that contrastive learning theoretically regularizes pre-trained embeddings' anisotropic space to be more uniform, and it better aligns positive pairs when supervised signals are available.[1]

1 Introduction

Learning universal sentence embeddings is a fundamental problem in natural language processing and has been studied extensively in the literature (Kiros et al., 2015; Hill et al., 2016; Conneau et al., 2017; Logeswaran and Lee, 2018; Cer et al., 2018; Reimers and Gurevych, 2019, *inter alia*). In this work, we advance state-of-the-art sentence embedding methods and demonstrate that a contrastive objective can be extremely effective in learning sentence embeddings, coupled with pre-trained language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). We present SimCSE, a simple contrastive sentence embedding framework, which can be used to produce superior sentence embeddings, from either unlabeled or labeled data.

	$\text{BERT}_{\text{base}}$
Unsupervised	
Avg. embeddings	56.7
IS-BERT (prev. SoTA)	66.6
SimCSE	**74.5** (+7.9%)
Supervised	
SBERT	74.9
SBERT-whitening (prev. SoTA)	77.0
SimCSE	**81.6** (+4.6%)

Table 1: Comparison between SimCSE and previous state-of-the-art (unsupervised and supervised). The reported numbers are the average of seven STS tasks (Spearman's correlation), see Table 6 for details. IS-BERT, SBERT, SBERT-whitening: Zhang et al. (2020), Reimers and Gurevych (2019) and Su et al. (2021).

Our *unsupervised* SimCSE simply predicts the input sentence itself, with only *dropout* (Srivastava et al., 2014) used as noise (Figure 1(a)). In other words, we pass the same input sentence to the pre-trained encoder *twice* and obtain two embeddings as "positive pairs", by applying independently sampled dropout masks. Although it may appear strikingly simple, we find that this approach largely outperforms training objectives such as predicting next sentences (Kiros et al., 2015; Logeswaran and Lee, 2018) and common data augmentation techniques, e.g., word deletion and replacement. More surprisingly, this unsupervised embedding method already matches all the previous supervised approaches. Through careful analysis, we find that dropout essentially acts as minimal data augmentation, while removing it leads to a representation collapse.

[*]The first two authors contributed equally (listed in alphabetical order). This work was done when Xingcheng visited the Princeton NLP group remotely.

[1]Our code and pre-trained models are publicly available at https://github.com/princeton-nlp/SimCSE.

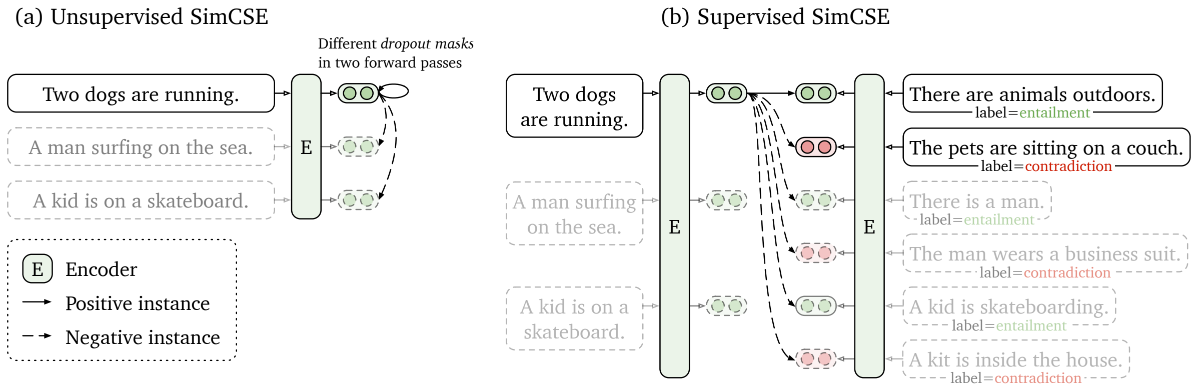


Figure 1: (a) Unsupervised SimCSE predicts the input sentence itself from in-batch negatives, with different dropout masks applied. (b) Supervised SimCSE leverages the NLI datasets and takes the entailment (premise-hypothesis) pairs as positives, and contradiction pairs as well as other in-batch instances as negatives.

In our *supervised* SimCSE, we build upon the recent success of leveraging natural language inference (NLI) datasets for sentence embeddings (Conneau et al., 2017; Reimers and Gurevych, 2019) and incorporate supervised sentence pairs in contrastive learning (Figure 1(b)). Unlike previous work that casts it as a 3-way classification task (entailment/neutral/contradiction), we take advantage of the fact that entailment pairs can be naturally used as positive instances. We also find that adding corresponding contradiction pairs as hard negatives further improves performance. This simple use of NLI datasets achieves a greater performance compared to prior methods using the same datasets. We also compare to other (annotated) sentence-pair datasets and find that NLI datasets are especially effective for learning sentence embeddings.

To better understand the superior performance of SimCSE, we borrow the analysis tool from Wang and Isola (2020), which takes *alignment* between semantically-related positive pairs and *uniformity* of the whole representation space to measure the quality of learned embeddings. We prove that theoretically the contrastive learning objective "flattens" the singular value distribution of the sentence embedding space, hence improving the uniformity. We also draw a connection to the recent findings that pre-trained word embeddings suffer from anisotropy (Ethayarajh, 2019; Li et al., 2020). We find that our unsupervised SimCSE essentially improves uniformity while avoiding degenerated alignment via dropout noise, thus greatly improves the expressiveness of the representations. We also demonstrate that the NLI training signal can further improve alignment between positive pairs and hence produce better sentence embeddings.

We conduct a comprehensive evaluation of SimCSE, along with previous state-of-the-art models on 7 semantic textual similarity (STS) tasks and 7 transfer tasks. On STS tasks, we show that our unsupervised and supervised models achieve a 74.5% and 81.6% averaged Spearman's correlation respectively using $\text{BERT}_{\texttt{base}}$, largely outperforming previous best (Table 1). We also achieve competitive performance on the transfer tasks. Additionally, we identify an incoherent evaluation issue in existing work and consolidate results of different evaluation settings for future research.

2 Background: Contrastive Learning

Contrastive learning aims to learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors (Hadsell et al., 2006). It assumes a set of paired examples $\mathcal{D} = \{(x_i, x_i^+)\}_{i=1}^m$, where x_i and x_i^+ are semantically related. We follow the contrastive framework in Chen et al. (2020) and take a cross-entropy objective with in-batch negatives (Chen et al., 2017; Henderson et al., 2017): let $\mathbf{h}_i$ and $\mathbf{h}_i^+$ denote the representations of x_i and x_i^+, for a mini-batch with N pairs, the training objective for (x_i, x_i^+) is:

$$\ell_i = \log \frac{e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^N e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}}, \tag{1}$$

where τ is a temperature hyperparameter and $\mathrm{sim}(\mathbf{h}_1, \mathbf{h}_2)$ is the cosine similarity $\frac{\mathbf{h}_1^\top \mathbf{h}_2}{\|\mathbf{h}_1\| \cdot \|\mathbf{h}_2\|}$. In this work, we encode input sentences using a pre-trained language model such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019): $\mathbf{h} = f_\theta(x)$, and then fine-tune all the parameters using the contrastive learning objective (Eq. 1).

Positive instances One critical question in contrastive learning is how to construct (x_i, x_i^+) pairs. In visual representations, an effective solution is to take two random transformations of the *same* image (e.g., cropping, flipping, distortion and rotation) as x_i and x_i^+ (Dosovitskiy et al., 2014). A similar approach has been recently adopted in language representations (Wu et al., 2020; Meng et al., 2021), by applying augmentation techniques such as word deletion, reordering, and substitution. However, data augmentation in NLP is inherently difficult because of its discrete nature. As we will see in §3, using standard dropout on intermediate representations outperforms these discrete operators.

In NLP, a similar contrastive learning objective has been also explored in different contexts (Henderson et al., 2017; Gillick et al., 2019; Karpukhin et al., 2020; Lee et al., 2020). In these cases, (x_i, x_i^+) are collected from supervised datasets such as mention-entity, or question-passage pairs. Because of the distinct nature of x_i and x_i^+ by definition, these approaches always use a *dual*-encoder framework, i.e., using two independent encoders f_{θ_1} and f_{θ_2} for x_i and x_i^+. For sentence embeddings, Logeswaran and Lee (2018) also use contrastive learning with a dual-encoder approach, by forming (current sentence, next sentence) as (x_i, x_i^+). Zhang et al. (2020) consider global sentence representations and local token representations of the same sentence as positive instances.

Alignment and uniformity Recently, Wang and Isola (2020) identify two key properties related to contrastive learning: *alignment* and *uniformity* and propose metrics to measure the quality of representations. Given a distribution of positive pairs p_{pos}, *alignment* calculates expected distance between embeddings of the paired instances (assuming representations are already normalized),

$$\ell_{\text{align}} \triangleq \mathop{\mathbb{E}}_{(x,x^+)\sim p_{\text{pos}}} \|f(x) - f(x^+)\|^2. \qquad (2)$$

On the other hand, *uniformity* measures how well the embeddings are uniformly distributed:

$$\ell_{\text{uniform}} \triangleq \log \mathop{\mathbb{E}}_{x,y \overset{i.i.d.}{\sim} p_{\text{data}}} e^{-2\|f(x)-f(y)\|^2}, \qquad (3)$$

where p_{data} denotes the data distribution. These two metrics are well aligned with the objective of contrastive learning: positive instances should stay close and embeddings for random instances should scatter on the hypersphere. In the following sections, we will also use the two metrics to justify the inner workings of our approaches.

Data augmentation				STS-B
None				**79.1**
Crop	*10%*	*20%*	*30%*	
	75.4	70.1	63.7	
Word deletion	*10%*	*20%*	*30%*	
	74.7	71.2	70.2	
Delete one word				74.8
w/o dropout				71.4
MLM 15%				66.8
Crop 10% + MLM 15%				70.8

Table 2: Comparison of different data augmentations on STS-B development set (Spearman's correlation). *Crop* $k\%$: randomly crop and keep a continuous span with 100-$k\%$ of the length; *word deletion* $k\%$: randomly delete $k\%$ words; *delete one word*: randomly delete one word; *MLM* $k\%$: use BERT$_{\texttt{base}}$ to replace $k\%$ of words. All of them include the standard 10% dropout (except "w/o dropout").

3 Unsupervised SimCSE

In this section, we describe our unsupervised SimCSE model. The idea is extremely simple: we take a collection of sentences $\{x_i\}_{i=1}^m$ and use $x_i^+ = x_i$. The key ingredient to get this to work with identical positive pairs is through the use of independently sampled *dropout masks*. In standard training of Transformers (Vaswani et al., 2017), there is a dropout mask placed on fully-connected layers as well as attention probabilities (default $p = 0.1$). We denote $\mathbf{h}_i^z = f_\theta(x_i, z)$ where z is a random mask for dropout. We simply feed the same input to the encoder *twice* by applying different dropout masks z, z' and the training objective becomes:

$$\ell_i = -\log \frac{e^{\text{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z'_i})/\tau}}{\sum_{j=1}^N e^{\text{sim}(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z'_j})/\tau}}, \qquad (4)$$

for a mini-batch with N sentences. Note that z is just the *standard* dropout mask in Transformers and we do not add any additional dropout.

Dropout noise as data augmentation We view this approach as a minimal form of data augmentation: the positive pair takes exactly the same sentence, and their embeddings only differ in dropout masks. We compare this approach to common augmentation techniques and other training objectives on the STS-B development set (Cer et al., 2017).

Training objective	f_θ	$(f_{\theta_1}, f_{\theta_2})$
Next sentence	66.8	67.7
Next 3 sentences	68.7	69.7
Delete one word	74.8	70.4
Unsupervised SimCSE	**79.1**	70.7

Table 3: Comparison of different unsupervised objectives. Results are Spearman's correlation on the STS-B development set using $\text{BERT}_{\texttt{base}}$, trained on 1-million pairs from Wikipedia. The two columns denote whether we use one encoder f_θ or two independent encoders f_{θ_1} and f_{θ_2} ("dual-encoder"). *Next 3 sentences*: randomly sample one from the next 3 sentences. *Delete one word*: delete one word randomly (see Table 2).

p	*0.0*	*0.01*	*0.05*	*0.1*
STS-B	64.9	69.5	78.0	**79.1**
p	*0.15*	*0.2*	*0.5*	*Fixed 0.1*
STS-B	78.6	78.2	67.4	45.2

Table 4: Effects of different dropout probabilities p on the STS-B development set (Spearman's correlation, $\text{BERT}_{\texttt{base}}$). *Fixed 0.1*: use the default 0.1 dropout rate but apply the same dropout mask on both x_i and x_i^+.

We use $N = 512$ and $m = 10^6$ sentences randomly drawn from English Wikipedia in these experiments. Table 2 compares our approach to common data augmentation techniques such as crop, word deletion and replacement, which can be viewed as $\mathbf{h} = f_\theta(g(x), z)$ and g is a (random) discrete operator on x. We find that even deleting one word would hurt performance and none of the discrete augmentations outperforms basic dropout noise.

We also compare this self-prediction training objective to next-sentence objective used in Logeswaran and Lee (2018), taking either one encoder or two independent encoders. As shown in Table 3, we find that SimCSE performs much better than the next-sentence objectives (79.1 vs 69.7 on STS-B) and using one encoder instead of two makes a significant difference in our approach.

Why does it work? To further understand the role of dropout noise in unsupervised SimCSE, we try out different dropout rates in Table 4 and observe that all the variants underperform the default dropout probability $p = 0.1$ from Transformers. We find two extreme cases particularly interesting: "no dropout" ($p = 0$) and "fixed 0.1" (using default dropout $p = 0.1$ but the same dropout masks for the pair). In both cases, the resulting embeddings

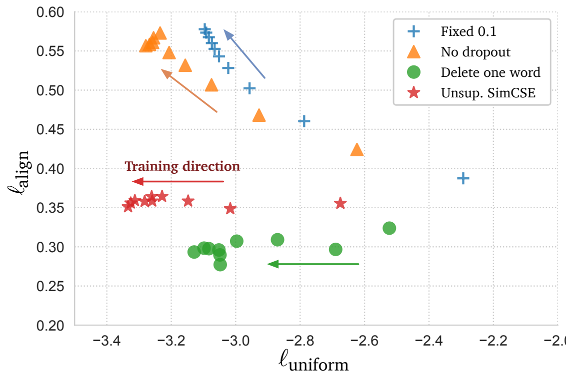


Figure 2: ℓ_{align}-ℓ_{uniform} plot for unsupervised SimCSE, "no dropout", "fixed 0.1" (same dropout mask for x_i and x_i^+ with $p = 0.1$), and "delete one word". We visualize checkpoints every 10 training steps and the arrows indicate the training direction. For both ℓ_{align} and ℓ_{uniform}, *lower numbers are better*.

for the pair are exactly the same, and it leads to a dramatic performance degradation. We take the checkpoints of these models every 10 steps during training and visualize the alignment and uniformity metrics[2] in Figure 2, along with a simple data augmentation model "delete one word". As is clearly shown, all models largely improve the uniformity. However, the alignment of the two special variants also degrades drastically, while our unsupervised SimCSE keeps a steady alignment, thanks to the use of dropout noise. On the other hand, although "delete one word" slightly improves the alignment, it has a smaller gain on the uniformity, and eventually underperforms unsupervised SimCSE.

4 Supervised SimCSE

We have demonstrated that adding dropout noise is able to learn a good alignment for positive pairs $(x, x^+) \sim p_{\text{pos}}$. In this section, we study whether we can leverage supervised datasets to provide better training signals for improving alignment of our approach. Prior work (Conneau et al., 2017; Reimers and Gurevych, 2019) has demonstrated that supervised natural language inference (NLI) datasets (Bowman et al., 2015; Williams et al., 2018) are effective for learning sentence embeddings, by predicting whether the relationship between two sentences is *entailment*, *neutral* or *contradiction*. In our contrastive learning framework, we instead directly take (x_i, x_i^+) pairs from supervised datasets and use them to optimize Eq. 1.

[2]We take STS-B pairs with a score higher than 4 as p_{pos} and all STS-B sentences as p_{data}.

Dataset	sample	full
Unsup. SimCSE (1m)	-	79.1
QQP (134k)	81.8	81.8
Flickr30k (318k)	81.5	81.4
ParaNMT (5m)	79.7	78.7
SNLI+MNLI		
entailment (314k)	**84.1**	**84.9**
neutral (314k)[3]	82.6	82.9
contradiction (314k)	77.5	77.6
SNLI+MNLI		
entailment + hard neg.	-	**86.2**
+ ANLI (52k)	-	85.0

Table 5: Comparisons of different supervised datasets as positive pairs. Results are Spearman's correlation on the STS-B development set using $\text{BERT}_{\texttt{base}}$. Numbers in brackets denote the # of pairs. *Sample*: subsampling 134k positive pairs for a fair comparison between datasets; *full*: using the full dataset. In the last block, we use entailment pairs as positives and contradiction pairs as hard negatives (our final model).

Exploiting supervised data We first explore which annotated datasets are especially suitable for constructing positive pairs (x_i, x_i^+). We experiment with a number of datasets with sentence-pair examples, including QQP[4]: Quora question pairs; Flickr30k (Young et al., 2014): each image is annotated with 5 human-written captions and we consider any two captions of the same image as a positive pair; ParaNMT (Wieting and Gimpel, 2018): a large-scale back-translation paraphrase dataset[5]; and finally NLI datasets: SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018).

We train the contrastive learning model (Eq. 1) with different datasets and compare the results in Table 5 (for a fair comparison, we also run experiments with the same # of training pairs). We find that most of these models using supervised datasets outperform our unsupervised approach, showing a clear benefit from supervised signals. Among all the options, using entailment pairs from the NLI (SNLI + MNLI) datasets perform the best. We think this is reasonable, as the NLI datasets consist of high-quality and crowd-sourced pairs, and human annotators are expected to write the hypotheses manually based on the premises, and hence two sentences tend to have less lexical overlap. For instance, we find that the lexical overlap (F1 measured between two bags of words) for the entailment pairs (SNLI + MNLI) is 39%, while they are 60% and 55% for QQP and ParaNMT.

Contradiction as hard negatives Finally, we further take the advantage of the NLI datasets by using its contradiction pairs as hard negatives[6]. In NLI datasets, given one premise, annotators are required to manually write one sentence that is absolutely true (*entailment*), one that might be true (*neutral*), and one that is definitely false (*contradiction*). Thus for each premise and its entailment hypothesis, there is an accompanying contradiction hypothesis[7] (see Figure 1 for an example).

Formally, we extend (x_i, x_i^+) to (x_i, x_i^+, x_i^-), where x_i is the premise, x_i^+ and x_i^- are entailment and contradiction hypotheses. The training objective ℓ_i is then defined by (N is the mini-batch size):

$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^+)/\tau} + e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^-)/\tau}\right)}. \tag{5}$$

As shown in Table 5, adding hard negatives can further improve performance (84.9 $\rightarrow$ 86.2) and this is our final supervised SimCSE. We also tried to add the ANLI dataset (Nie et al., 2020) or combine it with our unsupervised SimCSE approach, but didn't find a meaningful improvement. We also considered a dual encoder framework in supervised SimCSE and it hurt performance (86.2 $\rightarrow$ 84.2).

5 Connection to Anisotropy

Recent work identifies an *anisotropy* problem in language representations (Ethayarajh, 2019; Li et al., 2020), i.e., the learned embeddings occupy a narrow cone in the vector space, which largely limits their expressiveness. Gao et al. (2019) term it as a *representation degeneration* problem and demonstrate that language models trained with tied input/output embeddings lead to anisotropic word embeddings, and this is further observed by Ethayarajh (2019) in pre-trained contextual embeddings. Wang et al. (2020) show that the singular values of the word embedding matrix decay drastically. In other words, except for a few dominating singular values, all others are close to zero.

[3] Though our final model only takes entailment pairs as positives, here we also try neutral and contradiction pairs.

[4] https://www.quora.com/q/quoradata/

[5] ParaNMT is automatically constructed by machine translation systems and we should not call it a supervised dataset, although it even underperforms our unsupervised SimCSE.

[6] We do not use the neutral pairs for hard negatives.

[7] In fact, one premise can have multiple contradiction hypotheses. In our implementation, we only sample one as the hard negative and we did not find a difference by using more.

A simple way to alleviate the problem is post-processing, either to eliminate the dominant principal components (Arora et al., 2017; Mu and Viswanath, 2018), or to map embeddings to an isotropic distribution (Li et al., 2020; Su et al., 2021). Alternatively, one can add regularization during training (Gao et al., 2019; Wang et al., 2020). In this section, we show that the contrastive objective can inherently "flatten" the singular value distribution of the sentence-embedding matrix.

Following Wang and Isola (2020), the asymptotics of the contrastive learning objective can be expressed by the following equation when the number of negative instances approaches infinity (assuming $f(x)$ is normalized):

$$
\begin{aligned}
&-\frac{1}{\tau}\underset{(x,x^+)\sim p_{\text{pos}}}{\mathbb{E}}\left[f(x)^\top f(x^+)\right] \\
&+\underset{x\sim p_{\text{data}}}{\mathbb{E}}\left[\log \underset{x^-\sim p_{\text{data}}}{\mathbb{E}}\left[e^{f(x)^\top f(x^-)/\tau}\right]\right],
\end{aligned} \tag{6}
$$

where the first term keeps positive instances similar and the second pushes negative pairs apart. When p_{data} is uniform over finite samples $\{x_i\}_{i=1}^m$, with $\mathbf{h}_i = f(x_i)$, we can derive the following formula from the second term with Jensen's inequality:

$$
\begin{aligned}
&\underset{x\sim p_{\text{data}}}{\mathbb{E}}\left[\log \underset{x^-\sim p_{\text{data}}}{\mathbb{E}}\left[e^{f(x)^\top f(x^-)/\tau}\right]\right] \\
=&\frac{1}{m}\sum_{i=1}^{m}\log\left(\frac{1}{m}\sum_{j=1}^{m}e^{\mathbf{h}_i^\top\mathbf{h}_j/\tau}\right) \\
\geq&\frac{1}{\tau m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{h}_i^\top\mathbf{h}_j.
\end{aligned} \tag{7}
$$

Let $\mathbf{W}$ be the sentence embedding matrix corresponding to $\{x_i\}_{i=1}^m$, i.e., the i-th row of $\mathbf{W}$ is $\mathbf{h}_i$. Ignoring the constant terms, optimizing the second term in Eq. 6 essentially minimizes an upper bound of the summation of all elements in $\mathbf{W}\mathbf{W}^\top$, i.e., $\mathrm{Sum}(\mathbf{W}\mathbf{W}^\top) = \sum_{i=1}^m\sum_{j=1}^m \mathbf{h}_i^\top\mathbf{h}_j$.

Since we normalize $\mathbf{h}_i$, all elements on the diagonal of $\mathbf{W}\mathbf{W}^\top$ are 1 and then $\mathrm{tr}(\mathbf{W}\mathbf{W}^\top)$, also the sum of all eigenvalues, is a constant. According to Merikoski (1984), if all elements in $\mathbf{W}\mathbf{W}^\top$ are positive, which is the case in most times from Gao et al. (2019), then $\mathrm{Sum}(\mathbf{W}\mathbf{W}^\top)$ is an upper bound for the largest eigenvalue of $\mathbf{W}\mathbf{W}^\top$. Therefore, when minimizing the second term in Eq. 6, we are reducing the top eigenvalue of $\mathbf{W}\mathbf{W}^\top$ and inherently "flattening" the singular spectrum of the embedding space. Hence contrastive learning can potentially tackle the representation degeneration problem and improve the uniformity.

Compared to postprocessing methods in Li et al. (2020); Su et al. (2021), which only aim to encourage isotropic representations, contrastive learning also optimizes for aligning positive pairs by the first term in Eq. 6, which is the key to the success of SimCSE (a quantitative analysis is given in §7).

6 Experiment

6.1 Evaluation setup

We conduct our experiments on 7 standard semantic textual similarity (STS) tasks and also 7 transfer learning tasks. We use the SentEval toolkit (Conneau and Kiela, 2018) for evaluation. Note that we share a similar sentiment with Reimers and Gurevych (2019) that the main goal of sentence embeddings is to cluster semantically similar sentences. Hence, we take STS results as the main comparison of sentence embedding methods and provide transfer task results for reference.

Semantic textual similarity tasks We evaluate 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) and compute cosine similarity between sentence embeddings. When comparing to previous work, we identify invalid comparison patterns in published papers in the evaluation settings , including (a) whether to use an additional regressor, (b) Spearman's vs Pearson's correlation, (c) how the results are aggregated (Table B.1). We discuss the detailed differences in Appendix B and choose to follow the setting of Reimers and Gurevych (2019) in our evaluation. We also report our replicated study of previous work, as well as our results evaluated in a different setting in Table B.2 and Table B.3. We also call for unifying the setting in evaluating sentence embeddings for future research.

Transfer tasks We also evaluate on the following transfer tasks: MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), SST-2 (Socher et al., 2013), TREC (Voorhees and Tice, 2000) and MRPC (Dolan and Brockett, 2005). A logistic regression classifier is trained on top of (frozen) sentence embeddings produced by different methods. We follow default configurations from SentEval[8].

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

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
				Unsupervised models				
GloVe embeddings (avg.)♣	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
BERT$_{\texttt{base}}$ (first-last avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
BERT$_{\texttt{base}}$-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT$_{\texttt{base}}$-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-BERT$_{\texttt{base}}$♡	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
$*$ SimCSE-BERT$_{\texttt{base}}$	**66.68**	**81.43**	**71.38**	**78.43**	**78.47**	**75.49**	**69.92**	**74.54**
RoBERTa$_{\texttt{base}}$ (first-last avg.)	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
RoBERTa$_{\texttt{base}}$-whitening	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
$*$ SimCSE-RoBERTa$_{\texttt{base}}$	**68.68**	**82.62**	**73.56**	**81.49**	**80.82**	**80.48**	**67.87**	**76.50**
$*$ SimCSE-RoBERTa$_{\texttt{large}}$	**69.87**	**82.97**	**74.25**	**83.01**	**79.52**	**81.23**	**71.47**	**77.47**
				Supervised models				
InferSent-GloVe♣	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder♣	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
SBERT$_{\texttt{base}}$♣	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT$_{\texttt{base}}$-flow	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
SBERT$_{\texttt{base}}$-whitening	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
$*$ SimCSE-BERT$_{\texttt{base}}$	**75.30**	**84.67**	**80.19**	**85.40**	**80.82**	**84.25**	**80.39**	**81.57**
SRoBERTa$_{\texttt{base}}$♣	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
SRoBERTa$_{\texttt{base}}$-whitening	70.46	77.07	74.46	81.64	76.43	79.49	76.65	76.60
$*$ SimCSE-RoBERTa$_{\texttt{base}}$	**76.53**	**85.21**	**80.95**	**86.03**	**82.57**	**85.83**	**80.50**	**82.52**
$*$ SimCSE-RoBERTa$_{\texttt{large}}$	**77.46**	**87.27**	**82.36**	**86.66**	**83.93**	**86.70**	**81.95**	**83.76**

Table 6: Sentence embedding performance on STS tasks (Spearman's correlation, "all" setting). We highlight the highest numbers among models with the same pre-trained encoder. ♣: results from Reimers and Gurevych (2019); ♡: results from Zhang et al. (2020); all other results are reproduced or reevaluated by ourselves. For BERT-flow (Li et al., 2020) and whitening (Su et al., 2021), we only report the "NLI" setting (see Table D.3).

Training details We start from pre-trained checkpoints of BERT (Devlin et al., 2019) (uncased) or RoBERTa (Liu et al., 2019) (cased), and add an MLP layer on top of the `[CLS]` representation as the sentence embedding[9] (see §6.3 for comparison between different pooling methods). More training details can be found in Appendix A. Finally, we introduce one more optional variant which adds a masked language modeling (MLM) objective (Devlin et al., 2019) as an auxiliary loss to Eq. 1: $\ell + \lambda \cdot \ell^{\mathrm{mlm}}$ (λ is a hyperparameter). This helps SimCSE avoid catastrophic forgetting of token-level knowledge. As we will show in Table 9, we find that adding this term can help improve performance on transfer tasks (not on sentence-level STS tasks).

6.2 Main Results

We compare SimCSE to previous state-of-the-art unsupervised and supervised sentence embedding methods. Unsupervised methods include averaging GloVe embeddings (Pennington et al., 2014), Skip-thought (Kiros et al., 2015), and IS-BERT (Zhang et al., 2020). We also compare our models to average BERT or RoBERTa embeddings[10], and post-processing methods such as BERT-flow (Li et al., 2020) and BERT-whitening (Su et al., 2021). Supervised methods include InferSent (Conneau et al., 2017), Universal Sentence Encoder (Cer et al., 2018) and SBERT/SRoBERTa (Reimers and Gurevych, 2019) along with applying BERT-flow and whitening on them. More details about each baseline are provided in Appendix C.

Semantic textual similarity Table 6 shows the evaluation results on 7 STS tasks. SimCSE can substantially improve results on all the datasets in both supervised and unsupervised settings, largely outperforming the previous state-of-the-art. Specifically, our unsupervised SimCSE-BERT raises the previous best average Spearman's correlation from 66.58% to 74.54%, even comparable to supervised baselines. Using NLI datasets, SimCSE-BERT further pushes the state-of-the-art results from 77.00% to 81.57%. The gains are even larger for RoBERTa encoders, achieving 77.47% and 83.76% for unsupervised and supervised approaches respectively.

[9]There is an MLP pooler in BERT's original implementation and we just use the layer with random initialization.

[10]Following Su et al. (2021), we take the average of the first and the last layer, which is better than only taking the last.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Unsupervised models								
GloVe embeddings (avg.)♣	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Skip-thought♡	76.50	80.10	93.60	87.10	82.00	92.20	73.00	83.50
Avg. BERT embeddings♣	78.66	86.25	94.37	88.66	84.40	**92.80**	69.54	84.94
BERT-[CLS]embedding♣	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
IS-BERT$_{\texttt{base}}$♡	**81.09**	**87.18**	**94.96**	**88.75**	**85.96**	88.64	74.24	**85.83**
∗ SimCSE-BERT$_{\texttt{base}}$	80.41	85.30	94.46	88.43	85.39	87.60	71.13	84.67
w/ MLM	80.74	85.67	94.68	87.21	84.95	89.40	**74.38**	85.29
∗ SimCSE-RoBERTa$_{\texttt{base}}$	79.67	84.61	91.68	85.96	84.73	84.20	64.93	82.25
w/ MLM	**82.02**	**87.52**	**94.13**	**86.24**	**88.58**	**90.20**	**74.55**	**86.18**
∗ SimCSE-RoBERTa$_{\texttt{large}}$	80.83	85.30	91.68	86.10	85.06	89.20	**75.65**	84.83
w/ MLM	**83.30**	**87.50**	**95.27**	**86.82**	**87.86**	**94.00**	75.36	**87.16**
Supervised models								
InferSent-GloVe♣	81.57	86.54	92.50	90.38	84.18	88.20	75.77	85.59
Universal Sentence Encoder♣	80.09	85.19	93.98	86.70	86.38	93.20	70.14	85.10
SBERT$_{\texttt{base}}$♣	**83.64**	**89.43**	94.39	**89.86**	**88.96**	**89.60**	76.00	**87.41**
∗ SimCSE-BERT$_{\texttt{base}}$	82.69	89.25	**94.81**	89.59	87.31	88.40	73.51	86.51
w/ MLM	82.68	88.88	94.52	89.82	88.41	87.60	**76.12**	86.86
SRoBERTa$_{\texttt{base}}$	84.91	90.83	92.56	88.75	90.50	88.60	**78.14**	87.76
∗ SimCSE-RoBERTa$_{\texttt{base}}$	84.92	**92.00**	**94.11**	**89.82**	91.27	88.80	75.65	88.08
w/ MLM	**85.08**	91.76	94.02	89.72	**92.31**	**91.20**	76.52	**88.66**
∗ SimCSE-RoBERTa$_{\texttt{large}}$	88.12	92.37	95.11	90.49	92.75	91.80	76.64	89.61
w/ MLM	**88.45**	**92.53**	**95.19**	**90.58**	**93.30**	**93.80**	**77.74**	**90.23**

Table 7: Transfer task results of different sentence embedding models (measured as accuracy). ♣: results from Reimers and Gurevych (2019); ♡: results from Zhang et al. (2020). We highlight the highest numbers among models with the same pre-trained encoder. MLM: adding MLM as an auxiliary task (§ 6.1) with $\lambda = 0.1$.

Transfer tasks Table 7 shows the evaluation results on transfer tasks. We find that supervised SimCSE performs on par or better than previous approaches, although the trend of unsupervised models remains unclear. We find that adding this MLM term consistently improves performance on transfer tasks, confirming our intuition that sentence-level objective may not directly benefit transfer tasks. We also experiment with post-processing methods (BERT-flow/whitening) and find that they both hurt performance compared to their base models, showing that good uniformity of representations does not lead to better embeddings for transfer learning. As we argued earlier, we think that transfer tasks are not a major goal for sentence embeddings, and thus we take the STS results for main comparison.

6.3 Ablation Study

We investigate how different batch sizes, pooling methods and MLM auxiliary objectives affect our models' performance. All results are using our supervised SimCSE model, evaluated on the development set of STS-B or transfer tasks. A more detailed ablation study is provided in Appendix D.

Batch size	32	64	128	256	512	1024
STS-B	84.6	85.6	86.0	**86.2**	**86.2**	86.0

Table 8: Effect of different batch sizes (STS-B development set, Spearman's correlation, BERT$_{\texttt{base}}$).

Batch size We explore the impact of batch sizes (N in Eq. 5) in Table 8. We find that the performance increases as N increases but it will not further increase after 512. This is slightly divergent from the batch sizes used in visual representations (He et al., 2020; Chen et al., 2020), mostly caused by the smaller training data size we use.

Pooling methods Reimers and Gurevych (2019); Li et al. (2020) show that taking the average embeddings of pre-trained models, especially from both the first and last layers, leads to better performance than [CLS]. Table 9 shows the comparison between the two settings and we find that they do not make a significant difference in our approach. Thus we choose to use the [CLS] representation for simplicity and to be consistent with the common practice of using pre-trained embeddings.

Model	STS-B	Avg. transfer
`[CLS]`	**86.2**	85.8
First-last avg.	86.1	86.1
w/o MLM	**86.2**	85.8
w/ MLM		
$\lambda = 0.01$	85.7	86.1
$\lambda = 0.1$	85.7	**86.2**
$\lambda = 1$	85.1	85.8

Table 9: Ablation studies of different pooling methods and incorporating the MLM objective. The results are based on the development sets using BERT$_{\texttt{base}}$.

MLM auxiliary task Finally, we study the impact of the MLM auxiliary objective with different λ. As shown in Table 9, the token-level MLM objective improves the averaged performance on transfer tasks modestly, yet it brings a consistent drop in semantic textual similarity tasks.

7 Analysis

In this section, we further conduct analyses to understand the inner workings of SimCSE.

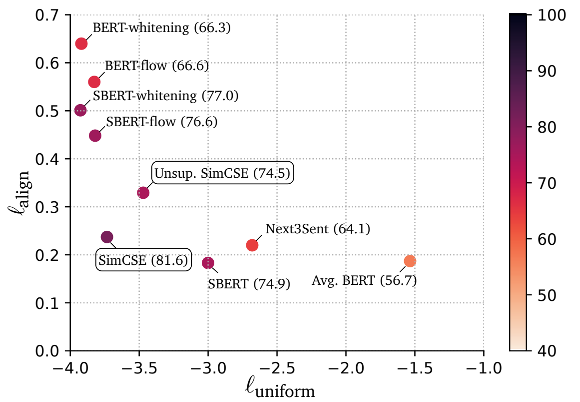


Figure 3: ℓ_{align}-ℓ_{uniform} plot of models based on BERT$_{\texttt{base}}$. Color of points and numbers in brackets represent average STS performance (Spearman's correlation). *Next3Sent*: "next 3 sentences" from Table 3.

Uniformity and alignment Figure 3 shows the uniformity and alignment of different sentence embeddings along with their averaged STS results. In general, models that attain both better alignment and uniformity will achieve better performance, confirming the findings in Wang and Isola (2020). We also observe that (1) though pre-trained embedding has good alignment, its uniformity is poor, i.e., it is highly anisotropic; (2) post-processing methods like BERT-flow and BERT-whitening largely improve the uniformity but also suffer a degeneration in alignment; (3) unsupervised SimCSE effectively improves the uniformity of pre-trained embeddings, while keeping a good alignment; (4) incorporating supervised data in SimCSE further amends the alignment. In Appendix E, we further show that SimCSE can effectively flatten singular value distribution of pre-trained embeddings.

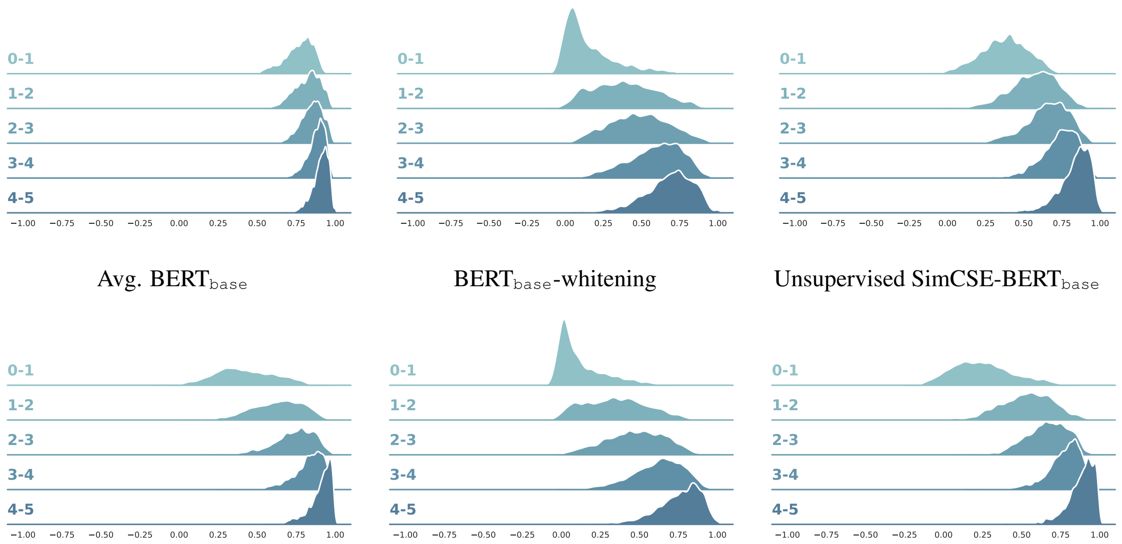
Cosine-similarity distribution To directly show the strengths of our approaches on STS tasks, we illustrate the cosine similarity distributions of STS-B pairs with different groups of human ratings in Figure 4. Compared to all the baseline models, both unsupervised and supervised SimCSE better distinguish sentence pairs with different levels of similarities, thus lead to a better performance on STS tasks. In addition, we observe that SimCSE generally shows a more scattered distribution than BERT or SBERT, but also preserves a lower variance on semantically similar sentence pairs compared to whitened distribution. This observation further validates that SimCSE can achieve a better alignment-uniformity balance.

Qualitative comparison We conduct a small-scale retrieval experiment using SBERT$_{\texttt{base}}$ and SimCSE-BERT$_{\texttt{base}}$. We use 150k captions from Flickr30k dataset and take any random sentence as query to retrieve similar sentences (based on cosine similarity). As several examples shown in Table 10, the retrieved instances by SimCSE have a higher quality compared to those retrieved by SBERT.

8 Related Work

Early work in sentence embeddings builds upon the distributional hypothesis by predicting surrounding sentences of a given sentence (Kiros et al., 2015; Hill et al., 2016; Logeswaran and Lee, 2018). Pagliardini et al. (2018) show that simply augmenting the idea of word2vec (Mikolov et al., 2013) with n-gram embeddings leads to strong results. Several recent models adopt contrastive objectives (Zhang et al., 2020; Wu et al., 2020; Meng et al., 2021) with unsupervised data by taking different views of the same sentence.

Compared to unsupervised approaches, supervised sentence embeddings demonstrate stronger performance. Conneau et al. (2017) propose to fine-tune a Siamese model on NLI datasets, which is further extended to other encoders or pre-trained models (Cer et al., 2018; Reimers and Gurevych,



Avg. $\text{BERT}_{\texttt{base}}$ | $\text{BERT}_{\texttt{base}}$-whitening | Unsupervised SimCSE-$\text{BERT}_{\texttt{base}}$

$\text{SBERT}_{\texttt{base}}$ | $\text{SBERT}_{\texttt{base}}$-whitening | Supervised SimCSE-$\text{BERT}_{\texttt{base}}$

Figure 4: Density plots of cosine similarities between sentence pairs in full STS-B. Pairs are divided into 5 groups based on ground truth ratings (higher means more similar) along the y-axis, and x-axis is the cosine similarity.

	$\textbf{SBERT}_{\texttt{base}}$	**Supervised SimCSE-**$\textbf{BERT}_{\texttt{base}}$
Query: A man riding a small boat in a harbor.		
#1	A group of men traveling over the ocean in a small boat.	A man on a moored blue and white boat.
#2	Two men sit on the bow of a colorful boat.	A man is riding in a boat on the water.
#3	A man wearing a life jacket is in a small boat on a lake.	A man in a blue boat on the water.
Query: A dog runs on the green grass near a wooden fence.		
#1	A dog runs on the green grass near a grove of trees.	The dog by the fence is running on the grass.
#2	A brown and white dog runs through the green grass.	Dog running through grass in fenced area.
#3	The dogs run in the green field.	A dog runs on the green grass near a grove of trees.

Table 10: Retrieved top-3 examples by SBERT and supervised SimCSE from Flickr30k (150k sentences).

2019). Furthermore, Wieting and Gimpel (2018); Wieting et al. (2020) demonstrate that bilingual and back-translation corpora provide useful supervision for learning semantic similarity. Another line of work focuses on regularizing embeddings (Li et al., 2020; Su et al., 2021; Huang et al., 2021) to alleviate the representation degeneration problem (as discussed in §5), and yields substantial improvement over pre-trained language models.

9 Conclusion

In this work, we propose SimCSE, a simple contrastive learning framework, which largely improves state-of-the-art sentence embedding performance on semantic textual similarity tasks. We present an unsupervised approach which predicts input sentence itself with dropout noise and a supervised approach utilizing NLI datasets. We further justify the inner workings of our approach by analyzing the alignment and uniformity of SimCSE along with other baseline models.

We believe that our contrastive training objective, especially the unsupervised approach, may have a broader application in NLP. It provides a new perspective on data augmentation with text input in contrastive learning, and may be extended to other continuous representations and integrated in language model pre-training.

Acknowledgements

We thank Tao Lei, Jason Lee, Zhengyan Zhang, Jinhyuk Lee, Alexander Wettig, Zexuan Zhong, and the members of the Princeton NLP group for helpful discussion and valuable feedback on our paper. TG is currently supported by a Graduate Fellowship at Princeton University.

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.

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

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, 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. Association for Computational Linguistics.

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

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations (ICLR)*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 632–642.

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

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607.

Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 767–776.

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

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

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

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. 2014. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 27.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations (ICLR)*.

Dan Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Computational Natural Language Learning (CoNLL)*, pages 528–537.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742. IEEE.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply.

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

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *ACM SIGKDD international conference on Knowledge discovery and data mining*.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. Whiteningbert: An easy unsupervised sentence embedding approach. *arXiv preprint arXiv:2104.01767*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3294–3302.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2020. Learning dense representations of phrases at scale.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

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 *International Conference on Language Resources and Evaluation (LREC)*, pages 216–223.

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

Jorma Kaarlo Merikoski. 1984. On the trace and the sum of elements of a matrix. *Linear Algebra and its Applications*, 60:177–185.

Tomas Mikolov, Ilya Sutskever, Kai Chen, G. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations (ICLR)*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Association for Computational Linguistics (ACL)*, pages 4885–4901.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 528–540.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Association for Computational Linguistics (ACL)*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Association for Computational Linguistics (ACL)*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Nils Reimers, Philip Beyer, and Iryna Gurevych. 2016. Task-oriented intrinsic evaluation of semantic textual similarity. In *International Conference on Computational Linguistics (COLING)*, pages 87–96.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6000–6010.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020. Improving neural language generation with spectrum control. In *International Conference on Learning Representations (ICLR)*.

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

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Association for Computational Linguistics (ACL)*, pages 451–462.

John Wieting, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A bilingual generative transformer for semantic sentence embedding. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1581–1594.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In *Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 38–45.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

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

A Training Details

We implement SimCSE based on Huggingface's `transformers` package (Wolf et al., 2020). For supervised SimCSE, we train our models for 3 epochs with a batch size of 512 and temperature $\tau = 0.05$ using an Adam optimizer (Kingma and Ba, 2015). The learning rate is set as 5e-5 for base models and 1e-5 for large models. We evaluate the model every 250 training steps on the development set of STS-B and keep the best checkpoint for the final evaluation on test sets. For unsupervised SimCSE, we take 5e-5 as the learning rate for both base and large models and only train for one epoch.

B Different Settings for STS Evaluation

We elaborate the differences in STS evaluation settings in previous work in terms of (a) whether to use additional regressors; (b) reported metrics; (c) different ways to aggregate results.

Additional regressors The default SentEval implementation applies a linear regressor on top of frozen sentence embeddings for STS-B and SICK-R, and train the regressor on the training sets of the two tasks, while most sentence representation papers take the raw embeddings and evaluate in an unsupervised way. In our experiments, *we do not apply any additional regressors and directly take cosine similarities for all STS tasks.*

Metrics Both Pearson's and Spearman's correlation coefficients are used in the literature. Reimers et al. (2016) argue that Spearman correlation, which measures the rankings instead of the actual scores, better suits the need of evaluating sentence embeddings. *For all of our experiments, we report Spearman's rank correlation.*

Aggregation methods Given that each year's STS challenge contains several subsets, there are different choices to gather results from them: one way is to concatenate all the topics and report the overall Spearman's correlation (denoted as "all"), and the other is to calculate results for different subsets separately and average them (denoted as "mean" if it is simple average or "wmean" if weighted by the subset sizes). However, most papers do not claim the method they take, making it challenging for a fair comparison. We take some of the most recent work: SBERT (Reimers and Gurevych, 2019), BERT-flow (Li et al., 2020) and BERT-whitening (Su et al., 2021)[11] as an example:

Paper	Reg.	Metric	Aggr.
Hill et al. (2016)		Both	all
Conneau et al. (2017)	✓	Pearson	mean
Conneau and Kiela (2018)	✓	Pearson	mean
Reimers and Gurevych (2019)		Spearman	all
Zhang et al. (2020)		Spearman	all
Li et al. (2020)		Spearman	wmean
Su et al. (2021)		Spearman	wmean
Wieting et al. (2020)		Pearson	mean
Ours		Spearman	all

Table B.1: STS evaluation protocols used in different papers. "Reg.": whether an additional regressor is used; "aggr.": methods to aggregate different subset results.

In Table B.2, we compare our reproduced results to reported results of SBERT and BERT-whitening, and find that Reimers and Gurevych (2019) take the "all" setting but Li et al. (2020); Su et al. (2021) take the "wmean" setting, even though Li et al. (2020) claim that they take the same setting as Reimers and Gurevych (2019). Since the "all" setting fuses data from different topics together, it makes the evaluation closer to real-world scenarios, and unless specified, *we take the "all" setting.*

We list evaluation settings for a number of previous work in Table B.1. Some of the settings are reported by the paper and some of them are inferred by comparing the results and checking their code. As we can see, the evaluation protocols are very incoherent across different papers. We call for unifying the setting in evaluating sentence embeddings for future research. We also release our evaluation code for better reproducibility. Since previous work uses different evaluation protocols from ours, we further evaluate our models in these settings to make a direct comparison to the published numbers. We evaluate SimCSE with "wmean" and Spearman's correlation to directly compare to Li et al. (2020) and Su et al. (2021) in Table B.3.

C Baseline Models

We elaborate on how we obtain different baselines for comparison:

- For average GloVe embedding (Pennington et al., 2014), InferSent (Conneau et al., 2017) and Universal Sentence Encoder (Cer et al., 2018), we directly report the results from Reimers and Gurevych (2019), since our evaluation setting is the same with theirs.

[11] Li et al. (2020) and Su et al. (2021) have consistent results, so we assume that they take the same evaluation and just take BERT-whitening in experiments here.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
SBERT (all)	70.97	76.53	73.19	79.09	74.30	76.98	72.91	74.85
SBERT (wmean)	66.35	73.76	73.88	77.33	73.62	76.98	72.91	73.55
SBERT♣	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
BERT-whitening (NLI, all)	57.83	66.90	60.89	75.08	71.30	68.23	63.73	66.28
BERT-whitening (NLI, wmean)	61.43	65.90	65.96	74.80	73.10	68.23	63.73	67.59
BERT-whitening (NLI)♠	61.69	65.70	66.02	75.11	73.11	68.19	63.60	67.63
BERT-whitening (target, all)	42.88	77.77	66.27	63.60	67.58	71.34	60.40	64.26
BERT-whitening (target, wmean)	63.38	73.01	69.13	74.48	72.56	71.34	60.40	69.19
BERT-whitening (target)♠	63.62	73.02	69.23	74.52	72.15	71.34	60.60	69.21

Table B.2: Comparisons of our reproduced results using different evaluation protocols and the original numbers. ♣: results from Reimers and Gurevych (2019); ♠: results from Su et al. (2021); Other results are reproduced by us. From the table we see that SBERT takes the "all" evaluation and BERT-whitening takes the "wmean" evaluation.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
$\text{BERT}_\texttt{base}$ (first-last avg.)♠	57.86	61.97	62.49	70.96	69.76	59.04	63.75	63.69
+ flow (NLI)♠	59.54	64.69	64.66	72.92	71.84	58.56	65.44	65.38
+ flow (target)♠	63.48	72.14	68.42	73.77	75.37	70.72	63.11	69.57
+ whitening (NLI)♠	61.69	65.70	66.02	75.11	73.11	68.19	63.60	67.63
+ whitening (target)♠	63.62	73.02	69.23	74.52	72.15	71.34	60.60	69.21
∗ Unsup. SimCSE-$\text{BERT}_\texttt{base}$	**68.92**	**78.70**	**73.35**	**79.72**	**79.42**	**75.49**	**69.92**	**75.07**
$\text{SBERT}_\texttt{base}$ (first-last avg.)♠	68.70	74.37	74.73	79.65	75.21	77.63	74.84	75.02
+ flow (NLI)♠	67.75	76.73	75.53	80.63	77.58	79.10	78.03	76.48
+ flow (target)♠	68.95	78.48	77.62	81.95	78.94	81.03	74.97	77.42
+ whitening (NLI)♠	69.11	75.79	75.76	82.31	79.61	78.66	76.33	76.80
+ whitening (target)♠	69.01	78.10	77.04	80.83	77.93	80.50	72.54	76.56
∗ Sup. SimCSE-$\text{BERT}_\texttt{base}$	**70.90**	**81.49**	**80.19**	**83.79**	**81.89**	**84.25**	**80.39**	**80.41**

Table B.3: STS results with "wmean" setting (Spearman). ♠: from Li et al. (2020); Su et al. (2021).

- For BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), we download the pre-trained model weights from HuggingFace's `Transformers`[12], and evaluate the models with our own scripts.

- For SBERT and SRoBERTa (Reimers and Gurevych, 2019), we reuse the results from the original paper. For results not reported by Reimers and Gurevych (2019), such as the performance of SRoBERTa on transfer tasks, we download the model weights from SentenceTransformers[13] and evaluate them.

- For BERT-flow (Li et al., 2020), since their original numbers take a different setting, we retrain their models using their code[14], and evaluate the models using our own script.

- For BERT-whitening (Su et al., 2021), we implemented our own version of whitening script following the same pooling method in Su et al. (2021), i.e. first-last average pooling. Our implementation can reproduce the results from the original paper (see Table B.2).

[12] https://github.com/huggingface/transformers
[13] https://www.sbert.net/
[14] https://github.com/bohanli/BERT-flow

D More Ablation Studies

τ	N/A	0.001	0.01	0.05	0.1	1
STS-B	85.9	84.9	85.4	**86.2**	82.0	64.0

Table D.1: STS-B development results (Spearman's correlation) with different temperatures. "N/A": Dot product instead of cosine similarity.

Hard neg	N/A	Contradiction			Contra.+ Neutral
α	-	0.5	1.0	2.0	1.0
STS-B	84.9	86.1	**86.2**	**86.2**	85.3

Table D.2: STS-B development results with different hard negative policies. "N/A": no hard negative.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
BERT-flow (NLI)	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
BERT-flow (target)	53.15	78.38	66.02	62.09	70.84	71.70	61.97	66.31
BERT-whitening (NLI)	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
BERT-whitening (target)	42.88	77.77	66.28	63.60	67.58	71.34	60.40	64.26
SBERT-flow (NLI)	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
SBERT-flow (target)	66.18	82.69	76.22	73.72	75.71	79.99	73.82	75.48
SBERT-whitening (NLI)	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
SBERT-whitening (target)	52.91	81.91	75.44	72.24	72.93	80.50	72.54	72.64

Table D.3: Comparison of using NLI or target data for postprocessing methods ("all", Spearman's correlation).

For both BERT-flow and BERT-whitening, they have two variants of postprocessing: one takes the NLI data ("NLI") and one directly learns the embedding distribution on the target sets ("target"). We find that in our evaluation setting, "target" is generally worse than "NLI" (Table D.3), so we only report the NLI variant in the main results.

Normalization and temperature We train SimCSE using both dot product and cosine similarity with different temperatures and evaluate them on the STS-B development set. As shown in Table D.1, with a carefully tuned temperature $\tau = 0.05$, cosine similarity is better than dot product.

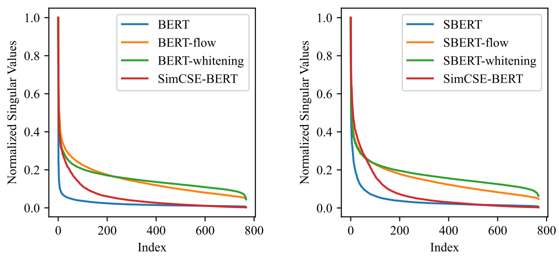
The use of hard negatives Intuitively, it may be not reasonable to use contradiction hypotheses equally with other in-batch negatives. Therefore, we extend the supervised training objective defined in Eq. 5 to a weighted one as follows:

$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^+)/\tau} + \alpha^{\mathbb{1}_i^j} e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^-)/\tau}\right)}, \quad (8)$$

where $\mathbb{1}_i^j \in \{0, 1\}$ is an indicator that equals 1 if and only if $i = j$. We train SimCSE with different α and evaluate the trained models on the development set of STS-B. Moreover, we also consider taking neutral hypotheses as hard negatives. As shown in Table D.2, $\alpha = 1$ performs the best, and neutral hypotheses do not bring further gains.

E Distribution of Singular Values

Figure E.1 shows the singular value distribution of SimCSE together with other baselines. For both unsupervised and supervised cases, singular value drops the fastest for vanilla BERT or SBERT embeddings, while SimCSE helps flatten the spectrum distribution. Postprocessing-based methods such as BERT-flow or BERT-whitening flatten the curve even more since they directly aim for the goal of mapping embeddings to an isotropic distribution.



Unsupervised models　　Supervised models

Figure E.1: Singular value distributions of sentence embedding matrix from sentences in STS-B. We normalize the singular values so that the largest one is 1.