Unsupervised Sentence Representation via Contrastive Learning with Mixing Negatives

Yanzhao Zhang[1,2], Richong Zhang[1,2]*, Samuel Mensah[3], Xudong Liu[1,2], Yongyi Mao[4]

[1]Beijing Advanced Institution for Big Data and Brain Computing, Beihang University, Beijing, China
[2]SKLSDE, School of Computer Science and Engineering, Beihang University, Beijing, China
[3]Department of Computer Science, University of Sheffield, UK
[4]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
zhangyanzhao00@gmail.com,zhangrc@act.buaa.edu.cn, s.mensah@sheffield.ac.uk,
liuxd@act.buaa.edu.cn,ymao@uottawa.ca

Abstract

Unsupervised sentence representation learning is a fundamental problem in natural language processing. Recently, contrastive learning has made great success on this task. Existing constrastive learning based models usually apply random sampling to select negative examples for training. Previous work in computer vision has shown that hard negative examples help contrastive learning to achieve faster convergency and better optimization for representation learning. However, the importance of hard negatives in contrastive learning for sentence representation is yet to be explored. In this study, we prove that hard negatives are essential for maintaining strong gradient signals in the training process while random sampling negative examples is ineffective for sentence representation. Accordingly, we present a contrastive model, MixCSE, that extends the current state-of-the-art SimCSE by continually constructing hard negatives via mixing both positive and negative features. The superior performance of the proposed approach is demonstrated via empirical studies on Semantic Textual Similarity datasets and Transfer task datasets.

Introduction

Sentence representation learning is a basic task of natural language processing (NLP). Briefly, an embedding model learns to map a sentence to a single d dimensional vector. This area has found strong applications in several NLP tasks including semantic textual similarity (Zhang et al. 2020), information retrieval (Cer et al. 2018) and text classification (Pang and Lee 2005, 2004).

Early work (Conneau et al. 2017) considered learning sentence representations in a supervised way. However, obtaining ample training data is expensive in practice. This has raised the desire to use little supervision as possible. Recently, several works (Gao, Yao, and Chen 2021; Yan et al. 2021) have opted to learn sentence representations in an unsupervised fashion, taking advantage of large availability of unlabeled data. Among proposed works, the common consensus is that the semantic information of the sentence should be preserved when learning a good representation.

*Corresponding author: zhangrc@act.buaa.edu.cn

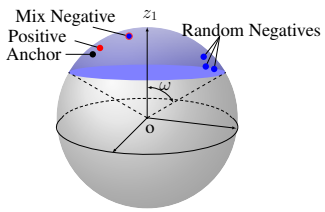


Figure 1: An illustration of the embedding distribution induced by our method. The mix negative features is closer to the anchor as compared to the random negatives.

Elsewhere, Wang and Isola (2020) identified two properties for good computer vision representations (Chen et al. 2020; He et al. 2020). That is, (1) *alignment*: representations of similar examples should be close to each other; (2) *uniformity*: the embeddings should be distributed uniformly in the representation space. We find that these two properties are also suitable for learning good sentence representations.

Learning representations such that the embeddings of similar examples are close to each other while dissimilar ones are far apart can be considered as an instance of contrastive learning (Wang and Liu 2021). Although it has received much attention in the computer vision community, only a few works (Gao, Yao, and Chen 2021; Yan et al. 2021) have employed contrastive learning coupled with pre-trained language models such as BERT (Devlin et al. 2018) for sentence representations. Since BERT suffers from anistropy (Li et al. 2020) (i.e., the learned embeddings are distributed into a narrow cone), contrastive learning alleviates this problem by distributing the embeddings uniformly, leading to significant improvement in current sentence representation methods (Gao, Yao, and Chen 2021).

In the context of unsupervised sentence representation learning, the core idea of contrastive learning is to train a neural network (i.e., typically BERT) for all sentence inputs by constructing similar (positive) pairs and dissimilar (negative) pairs. The goal is to learn a good sentence embedding where positive pairs are pulled together while negative pairs are pushed apart in space. Thus, the network uses a contrastive loss to minimize the distance of a positive pair and maximize the distance of a negative pair. Given an an-

chor sentence, a positive pair consist of augmented features of the anchor. A negative pair is typically constructed by pairing an anchor's features with the features of a randomly sampled sentence in a corpus. The current state-of-the-art SimCSE (Gao, Yao, and Chen 2021) uses dropout masks to generate positive pairs while random sampling is used to generate negative pairs. Very recently, several works (Wang and Liu 2021; Xuan et al. 2020; Kalantidis et al. 2020) have indicated the importance of hard negatives for contrastive learning. Hard negatives are those negatives that are hard to distinguish from the anchor in the embedding space. Hard negatives has aided in learning computer vision representations (Xuan et al. 2020). However, integrating hard negatives into contrastive learning is yet to be explored for sentence representation.

In this paper, we provide positive results that hard negatives is important for sentence representation under the framework of contrastive learning. Specifically, we prove that without hard negatives the gradient signals of contrastive learning loss becomes increasingly small, hindering effective learning. That is, the underlying mechanisms of how hard negatives affect the sentence representation learning via contrastive learning is theoretically formulated and proved. We then present theoretical support to justify that the strategy of randomly sampling negatives is incapable of generating strong gradient signals, particularly when the initial distribution of the features is highly anisotropic such as those produced by BERT. Finally, we present a novel model based on the framework of SimCSE (Gao, Yao, and Chen 2021), called MixCSE, which continuously injects artificial hard negative features via mixing both positive and negative samples in the training process in order to maintain strong gradient signals throughout training. Hence, we refer to these hard negatives as mix negatives. An illustration of the embedding distribution induced by our method is shown in Fig 1.

Briefly, our main contributions are to:

- prove that hard negatives are important for sentence representation learning and random sampling is not effective for choosing hard negatives even with many repeats of random sampling.
- propose the contrastive model MixCSE, an extension of SimCSE, that constructs hard negatives by mixing the positive features and random negative features for sentence representation.
- demonstrate through extensive experiments that MixCSE achieves state-of-the-art results on semantic textual similarity (STS) and transfer tasks (TR).

Related Work

Unsupervised Sentence Representation

Unsupervised sentence representation learning has gained traction recently. Traditional methods generate a sentence embedding by a weighted average of word embeddings (Arora, Liang, and Ma 2017; Ethayarajh 2018). SkipThought (Kiros et al. 2015) adapts the skip-gram model to the sentence level, where an encoded sentence is used to predict sentences around it. Other works (Qiao et al. 2019) use the output of pretrained language models (e.g. BERT) for sentence embedding. However, Ethayarajh (2019) revealed that the direct use of BERT does not perform well. Indeed, Li et al. (2020) found that BERT induces an anisotropic space of sentences embeddings, which is detrimental to the performance on STS tasks. For this reason, Li et al. (2020) proposed BERT-flow, a method that transforms the anisotropic sentence embedding distribution to a smooth and isotropic Gaussian distribution using a flow-based method (Dinh, Krueger, and Bengio 2014), improving performance on STS tasks. BERT-whitening (Su et al. 2021) uses the traditional whitening method to obtain a smooth sentence embedding distribution, achieving a performance equivalent to BERT-flow while reducing the dimensionality of the sentence embedding.

Contrastive Learning for Sentence Representation

Recently, contrastive learning for sentence representation has made great success. ConsBERT (Yan et al. 2021) combine multiple data augmentation strategies like token shuffling and cutoff for contrastive learning. Kim, Yoo, and Lee (2021) construct positive pairs using the hidden representations of BERT as well as its final sentence embedding. SimCSE (Gao, Yao, and Chen 2021) pass the same sentence to the pre-trained language twice using different dropout masks to construct positive pairs. This simple method has proved to be effective than other data augmentation strategies.

Current methods mainly focus on using different data augmentation strategies to generate positive pairs while negative pairs are generated through random sampling. Several works (Wang and Isola 2020; Xuan et al. 2020) in the computer vision community show that hard examples is essential for contrastive learning. Robinson et al. (2020) use an importance sampling method to select hard examples. MoCo (He et al. 2020) keeps a queue with features of the last few batches as a memory bank to obtain more negatives. Kalantidis et al. (2020) improve MoCo (He et al. 2020) by generating hard negative examples through mixing positive and negative examples in the memory bank. However, hard negatives is yet to be explored for unsupervised sentence representation.

Model

In this section, we first analyze the gradient of the contrastive loss and discuss the important role of hard negative examples in contrastive learning. We then show that it is difficult to obtain hard negative examples through random sampling. Finally, we introduce our method MixCSE, an extension of SimCSE that constructs hard negatives to learn sentence representation.

Contrastive Learning Framework

Let D denote a corpus of sentences, where x_i is a sentence in D. As shown in Fig 3 (b), the framework typically consist of an encoder module and a data augmentation module. The encoder maps $x_i \in D$ to a feature vector h_i in $\mathbb{R}^d$. The data

augmentation module augments the original data or its feature vector to obtain two different views of the data. Accordingly, we obtain two d-dimensional feature vectors h_i and h'_i of example $x_i \in D$ to form a *positive pair* (h_i, h'_i). Note that these feature vectors are l_2 normalized, so their vectors distribute on the $(d-1)$-sphere of unit radius centered at the origin of $\mathbb{R}^d$, a sphere we will denote by S^{d-1}. For the positive pair (h_i, h'_i), we will refer to one of the feature vectors as the "anchor feature" and the other as the "positive feature" of the anchor. With respect to the anchor h_i (or h'_i), the feature vectors h_j (or h'_j) of any other sentence x_j will be referred to as a "negative feature". For the clarity of presentation, we will always use h_i as opposed to h'_i to refer to the anchor, thereby reserving h'_i for referring to the positive feature of the anchor. However, in our implementation both h_i and h'_i may serve as the anchor.

Using these terminologies, contrastive learning can be summarized concisely as follows. For each anchor h_i in the pair (h_i, h'_i), N negative features of the anchor h_i are randomly sampled, and the following contrastive loss is minimized.

$$L_{\text{cl}} = -\log \frac{\exp(h_i^T h'_i/\tau)}{\exp(h_i^T h'_i/\tau) + \sum_j^N \exp(h_i^T h'_j)/\tau)}$$
$$= -h_i^T h'_i/\tau + \log\left(\exp(h_i^T h'_i/\tau) + \sum_j^N \exp(h_i^T h'_j/\tau)\right)$$

where τ is a scalar temperature hyperparameter.

It is insightful to inspect the derivative of L_{cl} with respect to h_i is:

$$-\left(\frac{h'_i}{\tau} + \frac{\exp(h_i^T h'_i/\tau)h'_i + \sum_j^N \exp(h_i^T h'_j/\tau)h'_j}{(\exp(h_i^T h'_i/\tau)/\tau + \sum_j^M \exp(h_i^T h'_j/\tau))/\tau}\right)$$
$$= -\frac{1}{C\tau}\sum_j^N \exp(h_i^T h'_j/\tau)(h'_i - h'_j) \quad (1)$$

where $C = \exp(h_i^T h'_i/\tau) + \sum_j^N \exp(h_i^T h'_j/\tau)$.

When training is driven by such a gradient signal, we see that for each negative feature h'_j, h_i is updated in the direction of $h'_i - h'_j$. But $h'_i - h'_j = (h'_i - h_i) - (h'_j - h_i)$. Such an update direction can be seen to have a net effect of pushing h_i in the direction of $h'_i - h_i$ and in the opposite direction of $h'_j - h_i$. In other words, training pushes h_i towards h'_i (i.e., aligning the positive pair) while moving it away from every h'_j (i.e., makes the sentence embeddings well separated, or uniformly distributed). Secondly, the j^{th} term in the gradient of Equation (1) depends on $\exp(h_i^T h'_j/\tau)$ and therefore it increases exponentially with the inner product $h_i^T h'_j$. This widely spreads the gradient values that correspond to different negative features h'_j. As a consequence, less distinguishable negative features h'_j of the anchor (namely those with larger inner products $h_i^T h'_j$) receive much larger gradient signals, consequently, pushing them away from the anchor.

Playing an important role in contrastive learning, the second aspect above also results in a limitation of contrastive learning. To see this, note that $\exp(h_i^T h'_j) \ll \exp(h_i^T h'_i)$, making $\sum_j \exp(h_i^T h'_j)$ rather insignificant relative to $\exp(h_i^T h'_i)$, particularly as training proceeds and the former continuously decreases and the latter increases to approach e. Then the gradient signal given in (1) continuously decreases, which slows down training and even halts it.

At this point, we see that the existence of negative features near the anchor are critical for maintaining a strong gradient signal. We will refer to such hard-to-distinguish negative features as "hard negative features". The key development of this work is to continuously inject artificial hard negative features to the training process as the originally hard negatives are being pushed away and becoming "easier".

Distribution of BERT Embeddings

As BERT will be adopted as the encoder to create sentence embedding (or features), we now recall an issue with such embeddings, as observed in (Gao et al. 2019). Specifically, the work of Gao et al. (2019) shows that the embeddings obtained from BERT adopts an anisotropic distribution. This leads the original sentence embeddings to only occupy a narrow cone in the feature space. When these embeddings are l_2 normalized, they will be distributed only in a spherical cap of S^{d-1}. We now analyze the consequence of such anisotropic distribution of BERT embeddings in contrastive learning, where we will assume that the embeddings are distributed uniformly over the sphere cap.

We consider the sphere cap centered at the "north pole" (see Fig 1). The sphere cap is the set all points $z = (z_1, z_2, \ldots, z_d) \in \mathbb{R}^d$ with $\|z\| = 1$ and $z_2^2 + z_3^2 + \ldots + z_d^2 \leq R$ for a positive value $R < 1$. A point z in the sphere cap can be represented alternatively using a spherical coordinate system. Specifically, any z on the sphere cap can be represented by a set of angles $(\phi_1, \phi_2, \ldots, \phi_{d-1})$, where $\phi_1 \leq \omega$ for some angle $\omega \in (0, \pi)$ and $\phi_2, \ldots, \phi_{d-1}$ are unconstrained[1]. We denote this sphere cap by $\mathcal{O}_\omega$. Note that here ω specifies the maximum angle between a vector on $\mathcal{O}_\omega$ and the "north pole", i.e. the point $(1, 0, 0, \ldots, 0)$ in the Cartesian coordinate system. For the ease of reference, we denote the "north pole" by μ.

Let $\mathcal{S}'$ denote the projection of S^{d-1} on the "equator plane", namely, $\mathcal{S}' = \{\text{proj}(z) : z \in S^{d-1}\}$, where $\text{proj}(z_1, z_2, \ldots, z_d) = (0, z_2, \ldots, z_d)$.

For any two points $z, z' \in \mathbb{R}^d$, let $\measuredangle(z, z')$ denote the angle between vectors z and z'. That is, $\measuredangle(z, z') = \arccos(\frac{z^T z'}{\|z\| \cdot \|z'\|})$. We have the following result (See proof in appendix).

Lemma 1 *Suppose that h and h' are two points on the sphere cap $\mathcal{O}_\omega$ and $\measuredangle(h, \mu) = \phi_1$, $\measuredangle(h', \mu) = \phi'_1$, $\measuredangle(\text{proj}(h), \text{proj}(h')) = \beta$ and $\measuredangle(h, h') = \theta$. Then*

$$\cos\theta = \cos\phi_1 \cos\phi'_1 + sin\phi_1 \sin\phi'_1 \cos\beta$$

Lemma 2 *If in Lemma 1, h is fixed and h' is distributed uniformly in the spherical cap $\mathcal{O}_\omega$, the probability density*

[1] $\phi_{d-1} \in [0, 2\pi)$ and others in $[0, \pi)$

functions of ϕ_1' and β are:

$$P_\phi(\phi_1') = \frac{(\sin\phi_1')^{d-2}}{\int_0^\omega (\sin\phi)^{d-2} d\phi}, \phi_1 \in [0, \omega]$$

$$P_\beta(\beta) = \frac{(\sin\beta)^{d-2}}{\int_0^\pi (\sin\phi)^{d-2} d\phi}, \beta \in [0, \pi)$$

Using these two lemmas, it is then possible to compute the mean and variance of $\cos\theta$ in the setting of Lemma 2 as shown in Figure 2.

In Fig 2, it can be seen that as d increases, the variance of $\cos\theta$ declines toward zero. Thus when d is large enough, the value $\cos\theta$ is concentrated at its mean. The mean of $\cos\theta$ on the other hand depends on ω: when $\omega \leq \pi/2$, the mean of $\cos\theta$ approaches $\cos(\phi)cos(\omega)$ with increasing d; when $\omega \geq \pi/2$, the mean of $\cos\theta$ approaches 0. Note that this latter phenomenon is in fact independent of the value of ϕ_1.

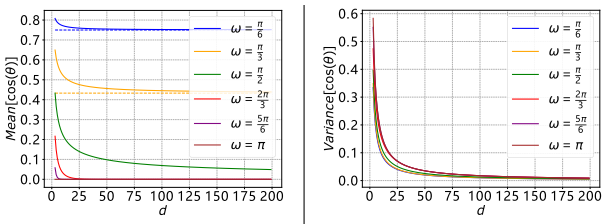
In this analysis, note that h represents an anchor, and h' represents a random negative feature of the anchor, distributed uniformly on the sphere cap $\mathcal{O}_\omega$, ϕ_1' is the angle between the anchor vector and the north-pole direction, and θ is the angle between the random negative feature vector and the anchor vector. When the feature embedding is initialized with pretrained BERT embeddings, recall that the features obtained from BERT are distributed in a sphere cap $\mathcal{O}_\omega$ with a small ω. As contrastive training proceeds, dissimilar feature are pushed away from each other, thereby enlarging the sphere cap $\mathcal{O}_\omega$, namely, increasing the value of ω. When the value of ω exceeds $\pi/2$, $\cos\theta$ all becomes close to 0 namely, all negative feature vectors are nearly orthogonal to the anchor vector[2]. That is, there hardly exists any "hard" negative features that are close to the anchor. As explained in the previous subsection, this results in very weak gradient signal for further training.

Cancellation effect among negative features One way to obtain hard negative features is to increase the number N of sampled negatives. However, we show that such an approach is highly inefficient, due to a cancellation effect among the hard negatives.

To see this, consider the large N limit. When N is sufficiently large, for every negative feature h' of anchor h with $\measuredangle(h, h') \in [0, \omega - \phi]$, there is an h'' located symmetrically to h' with respect to h, namely, $(h' + h'')/2$ lies in the same direction as h (or $(h' + h'')/2 = \rho h$ for some real value $\rho < 1$. The contribution to the gradient in Equation (1) by h'' then cancels the contribution by h'. But using large N inevitably increases the denominator $C = \exp(h_i^T h_i'/\tau) + \sum_j^N \exp(h_i^T h_j'/\tau)$. The combination of these effects reduces the gradient signal.

At this end, we have shown that the strategy of randomly sampling the negative features is incapable of generating strong gradient signals for contrastive training, particularly when the initial distribution of the features is highly anisotropic, such as that obtained from BERT.

[2]Note from Figure 2, this phenomenon does not require a very large embedding dimension d.



(a) Mean of $\cos\theta$ (b) Variance of $\cos\theta$

Figure 2: Mean and variance of $\cos\theta$ in the setting of Lemma 2, plotted with different ω and embedding dimension d and $\phi_1 = \pi/4$. The dotted line represents $\cos(\phi_1)\cos(\omega)$, plotted as a (constant) function of embedding dimension d for several ω values lower than $\pi/2$.

MixCSE

In this subsection, we propose a method, MixCSE, which continuously injects artificial hard negative features into the training process so as to maintain a strong gradient signal throughout training. This resolves the limitation of the standard contrastive training and the issue of BERT resulting from its anisotropic embedding distributions. We now describe MixCSE.

For an anchor feature h_i, we construct a negative feature $\widetilde{h'}_{i,j}$ by mixing the positive feature h_i' and a random negative feature h_j':

$$\widetilde{h'}_{i,j} = \frac{\lambda h_i' + (1-\lambda) h_j'}{||\lambda h_i' + (1-\lambda) h_j'||_2}$$

where λ is an hyperparameter to control the degree of mixing. By including these mixed negatives, the contrastive loss becomes

$$L_{\text{mix}} = -\log \frac{\exp\left(\frac{h_i^T h_i'}{\tau}\right)}{C + \sum_j^N \exp\left(\frac{h_i^T \text{SG}(\widetilde{h'}_{i,j})}{\tau}\right)}.$$

Here $\text{SG}(\cdot)$ denotes a "stop gradient" operator (Paszke et al. 2019) which ensures that back-propagation does not go through the mixed negative $\widetilde{h}_{i,j}$.

Mixed negatives help maintain strong gradients As discussed earlier, the contribution to the gradient signal from a negative feature h' of an anchor h is an exponentially increasing function of the inner product $h^T h'$. The inner product $h_i^T \widetilde{h'}_{i,j}$ of the anchor feature and the mixed negative is

$$h_i^T \widetilde{h'}_{i,j} = \frac{\lambda(h_i^T h_i') + (1-\lambda)(h_i^T h_j')}{||\lambda h_i' + (1-\lambda) h_j'||}$$

As discussed earlier, at some stage of training, $h_i^T h_j' \approx 0$. Additionally, when alignment is achieved, we have $h_i^T h_i' \approx 1$. It then follows that

$$h_i^T \widetilde{h'}_{i,j} \approx \frac{\lambda}{\sqrt{\lambda^2 + (1-\lambda)^2}} \quad (2)$$

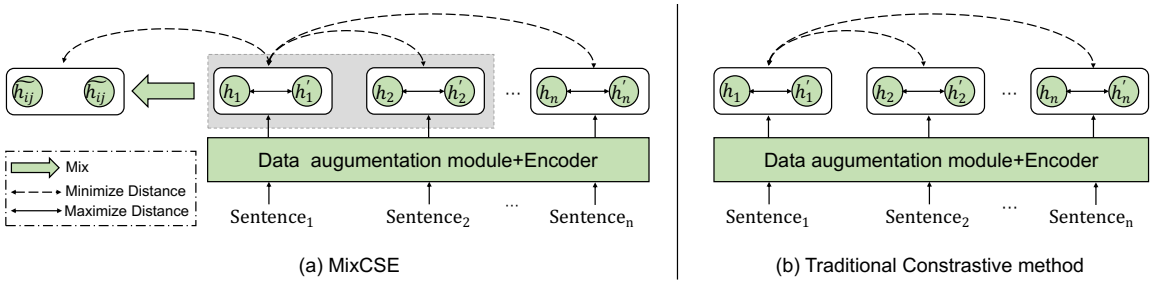


Figure 3: An illusion of our MixCSE method (a) and the traditional contrastive learning (b).

Thus unlike the standard negatives h'_j which gives rise to $h_i^T h'_j \approx 0$, the mixed negatives ensures the inner product value is consistently above zero. Such negative then serve to maintain a stronger gradient signal.

The choice of λ From Equation (2), it might appear that a larger λ is more beneficial. But this is only true when h_i and h'_i have been nearly perfectly aligned.

Consider the case where we have two positive features h'_i, h''_i for h and $\measuredangle(h_i, h'_i) = 0$ and $\measuredangle(h_i, h'_i) = \gamma$. The mixed negatives $\widetilde{h'}_{i,j}$ is construsted by mixing h'_i and a random negative h'_j. The angle between h_i and $\widetilde{h'}_{i,j}$ is:

$$arccos(h_i^T \widetilde{h'}_{i,j}) = arccos(\frac{\lambda + (1-\lambda)h_i^T h'_j}{||\lambda h'_i + (1-\lambda)h'_j||})$$

Although we want to construct hard negatives which are close to the anchor feature, we should still ensure that the mixed negatives are farther away from the anchor as compared to the corresponding positive features, namely, $\measuredangle(h_i, \widetilde{h'}_{i,j}) \geq \gamma$. Or this should be regarded as a "mixed positive feature". Thus, λ should have an upper bound:

$$\lambda < \frac{||\lambda h'_i + (1-\lambda)h'_j||cos(\gamma) - cos(h_i^T h'_j)}{1 - cos(h_i^T h'_j)}$$

In practice, we do not know the value of γ, so we choose a small λ to avoid generating the wrong hard negative.

The necessity of stop gradient for the mixed negatives Let $L_{\text{mix}}^{\text{no-sg}}$ represent the contrastive loss with a mixed negative without a stop-gradient. The derivative of $L_{\text{mix}}^{\text{no-sg}}$ with respect to h'_i is given by

$$\frac{\partial L_{\text{mix}}^{\text{no-sg}}}{\partial h'_i} = -\frac{1}{C'\tau}((\sum_j^N \exp(h_i^T h'_j/\tau) + \exp(h_i^T \widetilde{h'}_{i,j}/\tau))h_i^T - \frac{\partial \widetilde{h'}_{i,j}}{\partial h'_i}\exp(h_i^T \widetilde{h'}_{i,j}/\tau)h_i^T)$$

Notice that if $\widetilde{h'}_{i,j}$ participates in the gradient update process, the derivation has a direction $-\frac{\partial \widetilde{h'}_{i,j}}{\partial h'_i} exp(h_i^T \widetilde{h'}_{i,j}/\tau)h_i^T$, which will push h'_i away from h_i. In other words, if $\widetilde{h'}_{i,j}$ participates in the gradient update, the net effect is that the encoder pushes the positive feature close to the anchor feature. So we stop the gradient of $\widetilde{h'}_{i,j}$.

In practice, both h_i and h'_i can be regarded as the anchor feature. So we also construct $\widetilde{h}_{i,j}$ for h'_i by mixing h_i and h_j in the same way.

Experiment

Evaluation Setup

Following the standard evaluation protocol established in (Gao, Yao, and Chen 2021), we use the SentEval toolkit (Conneau and Kiela 2018) for evaluation purposes.

For Semantic Textual Similarity (STS), we evaluate on seven datasets: STS12-16 (Agirre et al. 2012; Lee et al. 2013; Agirre et al. 2014, 2015, 2016), STS-B (Cer et al. 2017) and SICK-R (Marelli et al. 2014). We use the Spearman's correlation coefficient as the performance metric.

For Transfer task (TR), we evaluate on seven datasets with the default configurations from SentEval.: MR (Pang and Lee 2005), CR (Kifer, Ben-David, and Gehrke 2004), SUBJ (Pang and Lee 2004), MPQA (Wiebe, Wilson, and Cardie 2005), SST-2 (Socher et al. 2013), TREC (Voorhees and Tice 2000) and MRPC (Dolan and Brockett 2005). Specifically, for each sentence representation method, SentEval uses the sentence representation it generates to train a classifier on downstream tasks, and verifies the quality of the sentence representation by the classification accuracy.

Implementation Details

We use the same training data and protocol in the work of Gao, Yao, and Chen (2021). Training data contains one million sentences crawled from Wikipedia. For each sentence, we extract a sentence embedding using a fine-tuned BERT model (Devlin et al. 2018) and use two independent dropout masks to obtain augmented versions of the sentence embedding. We set $\tau = 0.05, \lambda = 0.2$ and use the Adam optimizer (Kingma and Ba 2014) for optimization. We experiment with the $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$ models using the respective learning rates $3e-5$ and $1e-5$. For both models, we train for one epoch with batch size 64. We use early stopping to avoid overfitting. Our code is implemented in Python 3.6, using Pytorch 1.60 (Paszke et al. 2019), and the experiments are run on a single 32G NVIDIA A100 GPU.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg
Avg.Glove	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
$\text{BERT}_{\text{base}}$	35.20	59.53	49.37	63.39	62.73	48.18	58.60	53.86
$\text{BERT}_{\text{base}}$-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
$\text{BERT}_{\text{base}}$-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-$\text{BERT}_{\text{base}}$	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
ConSBERT$_{\text{base}}$	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE-$\text{BERT}_{\text{base}}$	67.17±9.61	79.79±2.72	71.96±3.97	80.21±1.42	77.65±1.24	76.46±1.44	70.57±1.25	74.83±2.32
MixCSE-$\text{BERT}_{\text{base}}$	**71.71±4.04**	**83.14±0.72**	**75.49±1.25**	**83.64±2.32**	**79.00±0.16**	**78.48±0.82**	**72.19±0.46**	**77.66±0.61**
$\text{BERT}_{\text{large}}$	33.06	57.64	47.95	55.83	62.42	49.66	53.87	51.49
$\text{BERT}_{\text{large}}$-flow	65.20	73.39	69.42	74.92	77.63	72.26	62.50	70.76
$\text{BERT}_{\text{large}}$-whitening	64.35	74.60	69.64	74.68	75.94	60.81	72.47	70.35
ConSBERT$_{\text{large}}$	70.69	82.96	74.13	82.78	76.66	77.53	70.37	76.45
SimCSE-$\text{BERT}_{\text{large}}$	70.21±1.49	83.97±1.18	75.92±0.56	83.9±0.49	78.87±0.75	79.0±1.0	73.89±1.08	77.97±0.7
MixCSE-$\text{BERT}_{\text{large}}$	**72.55±0.49**	**84.32±0.53**	**76.69±0.76**	**84.31±0.10**	**79.67±0.28**	**79.90±0.18**	**74.07±0.13**	**78.80±0.09**

Table 1: Results on the STS datasets. We implement and reproduce results of SimCSE, and report the average and standard variance of its results. The performances of other comparing models are from their original papers.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg
Glov.Avg	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
$\text{BERT}_{\text{base}}$	78.66	86.25	94.37	88.66	84.40	92.80	69.54	84.94
IS-$\text{BERT}_{\text{base}}$	81.09	**87.18**	94.96	88.75	85.96	**88.64**	74.24	85.83
SimCSE-$\text{BERT}_{\text{base}}$	71.12±5.94	85.92±0.41	98.56±2.05	88.61±0.18	85.34±0.37	88.4±0.59	73.48±1.26	84.49±0.59
MixCSE-$\text{BERT}_{\text{base}}$	**81.3±1.75**	86.77±0.4	**99.64±0.01**	**89.71±0.17**	85.87±0.49	84.91±0.24	**76.08±0.68**	**86.33±0.26**
$\text{BERT}_{\text{large}}$	60.89	90.15	99.62	86.04	89.95	93.00	69.86	84.22
SimCSE-$\text{BERT}_{\text{large}}$	73.93±2.79	88.87±0.75	99.6±0	89.49±0.16	90.59±0.96	**91.72**±0.95	75.49±0.88	86.8±0.42
MixCSE-$\text{BERT}_{\text{large}}$	**82.95±0.52**	**89.57±0.05**	**99.67±0.01**	**90.14±0.02**	**89.17±0.84**	86.13±0.58	**76.74±0.16**	**87.77±0.11**

Table 2: Results on the TR datasets. Bold numbers indicate best performance based on the same pretrained model.

Experiment Results

Our results are reported in Tables 1 and 2 on the STS and TR tasks respectively. Results are the mean and standard deviation computed over five runs for each dataset. We compare our model with BERT-flow (Li et al. 2020), BERT-whitening (Su et al. 2021), IS-BERT (Zhang et al. 2020), ConSBERT (Yan et al. 2021), SimCSE (Gao, Yao, and Chen 2021). As a naive baseline, we include Avg.Glove and BERT, which generate a sentence embedding by a weighted average of word embeddings.

STS task Experimental results on the STS datasets are shown in Table 1. We find that Avg.Glove outperforms $\text{BERT}_{\text{base}}$, showing the negative impact of the anisotropy of BERT embeddings. We also observe that $\text{BERT}_{\text{base}}$-flow/$\text{BERT}_{\text{large}}$-flow and $\text{BERT}_{\text{base}}$-whitening/$\text{BERT}_{\text{large}}$-whitening outperforms $\text{BERT}_{\text{base}}$/$\text{BERT}_{\text{large}}$ by alleviating the anisotropy. Interestingly, we find that models based on contrasting learning, including ConSBERT$_{\text{base}}$/ConSBERT$_{\text{large}}$ and SimCSE-$\text{BERT}_{\text{base}}$/SimCSE-$\text{BERT}_{large}$ show a substantial boost in model performance when compared to previous methods. However, our contrastive model Mix-$\text{BERT}_{\text{base}}$/Mix-$\text{BERT}_{\text{large}}$ not only show the best performance but also produces more stable results than the current state-of-the-art SimCSE-$\text{BERT}_{\text{base}}$/SimCSE-$\text{BERT}_{\text{large}}$.

TR task Experimental results on the TR datasets are shown in Table 2. We observe that $\text{BERT}_{\text{base}}$ performs better than Glove.Avg. On the TR task, we train a linear classifier on fixed embeddings. Hence, BERT embeddings contain rich semantic information as compared to Glove. That might explain why we have such results. Meanwhile, IS-$\text{BERT}_{\text{base}}$ shows competitive performance with the contrastive learning models SimCSE-$\text{BERT}_{\text{base}}$ and our own model Mix-$\text{BERT}_{\text{base}}$ by taking into account local word-level features which may be useful for these transfer classification tasks. However, our modMix-$\text{BERT}_{\text{large}}$ achieves the best performance on six out of seven datasets and has a relatively low variance, suggesting its effectiveness and stability.

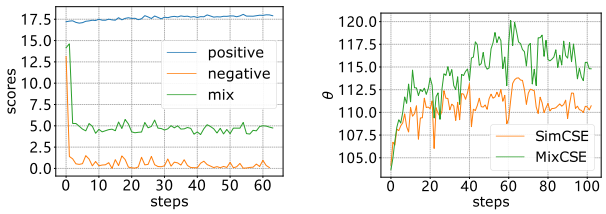
Model	STS(Avg)	TR(Avg)
MixCSE-$\text{BERT}_{\text{base}}$	**77.66±0.61**	**86.33±0.26**
$\text{MixCSE}_{\text{single}}$-$\text{BERT}_{\text{base}}$	76.43±0.11	85.86±0.04
$\text{MixCSE}_{\text{wo sg}}$-$\text{BERT}_{\text{base}}$	75.74±0.87	84.55 ± 0.27
MixCSE-$\text{BERT}_{\text{large}}$	**78.80±0.09**	**87.77±0.11**
$\text{MixCSE}_{\text{single}}$-$\text{BERT}_{\text{large}}$	78.11±0.20	87.21±0.23
$\text{MixCSE}_{\text{wo sg}}$-$\text{BERT}_{\text{large}}$	77.61±0.45	86.02±0.27

Table 3: Results of the Ablation Study.

Ablation Study

To analyze the impact of each model component, we conduct ablation experiments. Specifically, we design two variants of our model $\text{Mix}_{\text{single}}$ and $\text{Mix}_{\text{w grad}}$ and apply them to our datasets using the same experimental setup and hyperparameters as in the main experiment. Brief descriptions of these model variants are as follows:

$\text{MixCSE}_{\text{single}}$ Recall, in our model description we construct a mixing hard negative for h_i and h_i'. For



(a) (b)

Figure 4: (a) The change for positive, negative and mix scores. (b) The changes of biggest angle θ (y-axis) between any two embedding in STS-B in every 125 steps (x-axis).

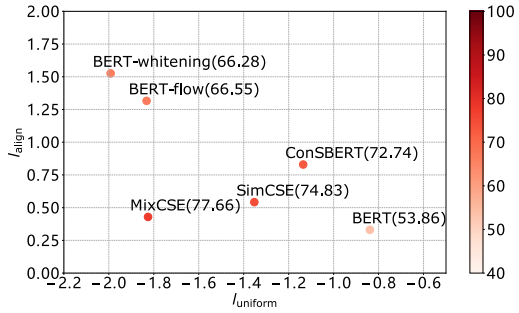


Figure 5: Alignment and uniformity for different sentence embedding methods measured on the STS-B dev set. Color of points represent average STS performance.

MixCSE$_{\text{single}}$, we only construct a hard negative for h_i to see the effect of the parallel mix method.

MixCSE$_{\text{wo sg}}$ Recall, in our model description we noted that if we update the gradient of $\widetilde{h'}_{i,j}$, the gradient direction of h'_i changes. We remove the stop-gradient operation of $\widetilde{h'}_{i,j}$ to see its effect.

Ablation results are shown in Table 3. We find that MixCSE$_{\text{single}}$ generally underperforms MixCSE, suggesting that our hard negatives construction method is effective. Then, MixCSE$_{\text{wo sg}}$ shows a further drop in performance, revealing the importance of our stop gradient operation.

Analysis

In this section, we conduct further analysis to understand the inner workings of MixCSE.

Analysis of the mix score In this part, we analyze whether the mix negative feature $\widetilde{h'}_{i,j}$ is more close to the anchor feature h_i. First, we define $h_i^T h_j/\tau$ as the positive score, $h_i^T h'_j/\tau$ as the negative score, and $h_i^T \widetilde{h'}_{i,j}/\tau$ as the mix score with $\tau = 0.05$. These scores indicate the distance between the two features. For each type, we log the corresponding average score in a batch at every 10 training steps.

The results are shown in Fig 4 (a). In the beginning, we observe that the scores for positive, negative, and mix are high, which goes on to support the anisotropic property of sentence embeddings from BERT. However, during the training process, we observe that the positive scores consistently remain high while both the negative and mix scores decline. Specifically, as the negative scores reduce and approach zero, the angle between h_i and h'_j approximates $\pi/2$. This is consistent with our previous derivation. On the other, the mix scores decrease but it is significantly higher than the negative scores, indicating that our method can indeed get hard negative features.

Analysis of the change of embedding distributed Recall, we assume that the original sentence embedding is distributed within a spherical cap O_ω. One aim of contrastive learning is to make the sentence embedding distributed uniformly in the whole embedding space. So the size of the spherical cap is enlarged during training, namely, ω keeps increasing during training.

We observe the change of O_ω in this part. For every 125 steps, we log the change of the biggest angle θ between any two sentences' embedding in the STS-B dataset, and regard ω as θ. The result of SimCSE and our method are shown in Fig 4 (b). We find that our method expands ω quickly and converges at a high value as compared to SimCSE. This indicates that our method makes the training more effective.

Alignment and Uniformity Wang and Isola (2020) proposed two widely used metric in contrastive learning to evaluate the quality of the computer embedding: alignment and uniformity. Alignment measures the expected distance between positive features:

$$L_{\text{align}} \triangleq \mathbb{E}_{(x,y)\ P_pos(x,y)}[||f(x) - f(y)||_2^2]$$

Uniformity on the other hand measures the expected distances between embeddings of two random examples:

$$L_{\text{uniform}} \triangleq \log \mathbb{E}_{(x,y)\ P_data(x,y)}[e^{-2||f(x)-f(y)||_2^2}]$$

We plot the distribution of the "uniformity-alignment" map for different sentence embedding models based on BERT$_{\text{base}}$ in Fig 5. The uniformity and alignment are calculated on the STS-B dataset. For both uniformity and alignment, lower values represent better performance. We observe that MixCSE achieves a better trade-off compared with the original BERT$_{\text{base}}$ model and the post-processing method like BERT-whitening and BERT-flow. Comparing with the contrastive learning methods such as SimCSE and ConSBERT, our method shows a better uniformity with a close alignment. This result indicates that MixCSE mainly helps improve the performance by learning better uniform features.

Conclusion

In this study, we prove that hard negatives play an important role in contrastive learning to maintain a strong gradient signal while randomly sampling negative features is incapable of generating strong gradient signals for contrastive training. We propose a contrastive model, MixCSE, that constructs hard negatives for sentence representation. Empirical studies on Semantic Textual Similarity datasets and Transfer task datasets confirm the effectiveness of the proposed model.

Acknowledgments

This work is supported partly by the National Key R&D Program of China under Grant 2021ZD0110700, in part by the National Natural Science Foundation of China under Grant 61772059, in part by the Fundamental Research Funds for the Central Universities, in part by the State Key Laboratory of Software Development Environment and by a Leverhulme Trust Research Project Grant.

References

Agirre, E.; Banea, C.; Cardie, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W.; Lopez-Gazpio, I.; Maritxalar, M.; Mihalcea, R.; Rigau, G.; Uria, L.; and Wiebe, J. 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)*, 252–263. Denver, Colorado: Association for Computational Linguistics.

Agirre, E.; Banea, C.; Cardie, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Guo, W.; Mihalcea, R.; Rigau, G.; and Wiebe, J. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 81–91. Dublin, Ireland: Association for Computational Linguistics.

Agirre, E.; Banea, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Mihalcea, R.; Rigau, G.; and Wiebe, J. 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)*, 497–511. San Diego, California: Association for Computational Linguistics.

Agirre, E.; Cer, D.; Diab, M.; and Gonzalez-Agirre, A. 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)*, 385–393. Montréal, Canada: Association for Computational Linguistics.

Arora, S.; Liang, Y.; and Ma, T. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *ICLR*.

Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 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)*, 1–14. Vancouver, Canada: Association for Computational Linguistics.

Cer, D.; Yang, Y.; Kong, S.-y.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Céspedes, M.; Yuan, S.; Tar, C.; et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Conneau, A.; and Kiela, D. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. *arXiv preprint arXiv:1803.05449*.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dinh, L.; Krueger, D.; and Bengio, Y. 2014. Nice: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*.

Dolan, W. B.; and Brockett, C. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ethayarajh, K. 2018. Unsupervised Random Walk Sentence Embeddings: A Strong but Simple Baseline. In *Proceedings of The Third Workshop on Representation Learning for NLP*, 91–100. Melbourne, Australia: Association for Computational Linguistics.

Ethayarajh, K. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Gao, J.; He, D.; Tan, X.; Qin, T.; Wang, L.; and Liu, T.-Y. 2019. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv preprint arXiv:2104.08821*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Kalantidis, Y.; Sariyildiz, M. B.; Pion, N.; Weinzaepfel, P.; and Larlus, D. 2020. Hard negative mixing for contrastive learning. *arXiv preprint arXiv:2010.01028*.

Kifer, D.; Ben-David, S.; and Gehrke, J. 2004. Detecting change in data streams. In *VLDB*, volume 4, 180–191. Toronto, Canada.

Kim, T.; Yoo, K. M.; and Lee, S.-g. 2021. Self-Guided Contrastive Learning for BERT Sentence Representations. *arXiv preprint arXiv:2106.07345*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, 3294–3302.

Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; and Li, L. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Marelli, M.; Menini, S.; Baroni, M.; Bentivogli, L.; Bernardi, R.; and Zamparelli, R. 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)*, 216–223. Reykjavik, Iceland: European Language Resources Association (ELRA).

Pang, B.; and Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*.

Pang, B.; and Lee, L. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 115–124. Ann Arbor, Michigan: Association for Computational Linguistics.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037.

Qiao, Y.; Xiong, C.; Liu, Z.; and Liu, Z. 2019. Understanding the Behaviors of BERT in Ranking. *arXiv preprint arXiv:1904.07531*.

Robinson, J.; Chuang, C.-Y.; Sra, S.; and Jegelka, S. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Su, J.; Cao, J.; Liu, W.; and Ou, Y. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

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

Wang, F.; and Liu, H. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2495–2504.

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

Wiebe, J.; Wilson, T.; and Cardie, C. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2): 165–210.

Xuan, H.; Stylianou, A.; Liu, X.; and Pless, R. 2020. Hard negative examples are hard, but useful. In *European Conference on Computer Vision*, 126–142. Springer.

Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; and Xu, W. 2021. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. *arXiv preprint arXiv:2105.11741*.

Zhang, Y.; He, R.; Liu, Z.; Lim, K. H.; and Bing, L. 2020. An Unsupervised Sentence Embedding Method by Mutual Information Maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1601–1610.