RANKCSE : UNSUPERVISED SENTENCE REPRESENTATION LEARNING VIA LEARNING TO RANK

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised sentence representation learning is one of the fundamental problems in natural language processing with various downstream applications. Recently, contrastive learning has been widely adopted which derives high-quality sentence representations by pulling similar semantics closer and pushing dissimilar ones away. However, these methods fail to capture the fine-grained ranking information among the sentences, where each sentence is only treated as either positive or negative. In many real-world scenarios, one needs to distinguish and rank the sentences based on their similarities to a query sentence, e.g., very relevant, moderate relevant, less relevant, irrelevant, etc. In this paper, we propose a novel approach, RankCSE, for unsupervised sentence representation learning, which incorporates ranking consistency and ranking distillation with contrastive learning into a unified framework. In particular, we learn semantically discriminative sentence representations by simultaneously ensuring ranking consistency between two representations with different dropout masks, and distilling listwise ranking knowledge from the teacher. An extensive set of experiments are conducted on both semantic textual similarity (STS) and transfer (TR) tasks. Experimental results demonstrate the superior performance of our approach over several state-of-the-art baselines.

1 INTRODUCTION

Sentence representation learning refers to the task of encoding sentences into fixed-dimensional embeddings. The sentence embeddings can be leveraged in various applications, including information retrieval (Le & Mikolov, 2014), text clustering (Ma et al., 2016) and semantic textual similarity comparison (Agirre et al., 2012). With the recent success of pre-trained language models (PLMs), such as BERT/RoBERTa (Devlin et al., 2019; Liu et al., 2019), a straightforward way to generate sentence representations is to directly use the [CLS] token embedding or the average token embeddings from the last layer of PLMs (Reimers & Gurevych, 2019). However, several studies (Ethayarajh, 2019; Li et al., 2020) have found that the native sentence representations derived by PLMs occupy a narrow cone in the vector space, and thus severely limits their representation capabilities, which is known as the anisotropy problem.

Supervised methods like SBERT (Reimers & Gurevych, 2019) usually generate better sentence representations, but require finetuning on a large amount of labeled data. Recent unsupervised models (Carlsson et al., 2021; Zhang et al., 2021; Giorgi et al., 2021; Yan et al., 2021; Gao et al., 2021) adopt contrastive learning framework without any labels, which pulls similar semantics closer and pushes dissimilar ones away. These methods usually design different augmentation algorithms for generating positive examples, such as back-translation (Zhang et al., 2021), dropout (Gao et al., 2021) and token shuffling or cutoff (Yan et al., 2021). In-batch negatives are further combined with the positives. Despite achieving promising results, they treat positives/negatives equally without capturing the fine-grained semantic ranking information, resulting less effective sentence representations which fail to distinguish between very similar and less similar sentences. For example, Table 1 shows an example of a query sentence and five target sentences from a semantic textual similarity dataset. It is clear that the similarity scores produced by the contrastive learning method SimCSE are not optimized, where the sentence rankings are not preserved in the learned representations. On the other hand, our RankCSE generates effective sentence representations with consistent rankings to the ground-truth labels. More examples are presented in Appendix A.7. The fine-grained ranking information is crucial in various real-world applications including search and recommendation.

Table 1: An example of an input sentence and five other sentences from the STS datasets, with their similarity scores and rankings. The label scores are from human annotations. The SimCSE (Gao et al., 2021) and RankCSE similarity scores are from the model predictions respectively, with the corresponding ranking positions. It can be seen that sentence rankings based on SimCSE are incorrect, while RankCSE generates more effective scores with accurate rankings.

Sentences	Label	SimCSE	RankCSE
• because by measuring voltage, you find the gap where there's a difference in electrical states.	3.80 (1)	0.86 (1)	0.90 (1)
• it allows you to measure electrical states between terminals	3.20 (2)	0.64 (3)	0.84 (2)
• it checks the electrical state between two terminals.	2.60 (3)	0.65 (2)	0.78 (3)
• find where there are different electrical states	2.60 (3)	0.55 (5)	0.78 (3)
• you can see where the gap is	2.20 (5)	0.62 (4)	0.69 (5)
Input Sentence: measuring voltage indicates the place where the electrical state changes due to a gap.			

Therefore, it is an important research problem to learn ranking preserving sentence representations from unsupervised data.

To obtain semantically discriminative sentence representations, we propose a novel approach, RankCSE , which incorporates ranking consistency and ranking distillation with contrastive learning into a unified framework. Specifically, our model ensures ranking consistency between two representations with different dropout masks, and minimize the Jensen-Shannon (JS) divergence as the learning objective. In the meanwhile, our model also distills listwise ranking knowledge from the teacher model to the learned sentence representations. In our work, we explore two listwise ranking methods, ListNet (Cao et al., 2007) and ListMLE (Xia et al., 2008), and utilize the pre-trained SimCSE (Gao et al., 2021) models with coarse-grained semantic ranking information as the teachers to provide pseudo ranking labels. Our RankCSE is able to generalize fine-grained ranking information from the weak ranking knowledge learned by SimCSE. We conduct an extensive set of experiments on several semantic textual similarity (STS) and transfer (TR) tasks. Experimental results show that RankCSE outperforms the existing state-of-the-art baselines.

2 Related Work

Unsupervised Sentence Representation Learning Early works typically augment the idea of word2vec (Mikolov et al., 2013) to learn sentence representations, including Skip-Thought (Kiros et al., 2015), FastSent (Hill et al., 2016) and Quick-Thought (Logeswaran & Lee, 2018). With the great success of PLMs, various attempts focus on generating sentence representations by leveraging the embedding of [CLS] token or applying mean pooling on the last layer of BERT (Reimers & Gurevych, 2019). However, Ethayarajh (2019) identifies the anisotropy problem in language representations, which means the native learned embeddings from PLMs occupy a narrow cone in the vector space. BERT-flow (Li et al., 2020) and BERT-whitening (Su et al., 2021) propose to resolve the anisotropy problem through post-processing.

Recently, contrastive learning has been adopted to learn sentence representations by designing different augmentation methods, including IS-BERT (Zhang et al., 2020), CT-BERT (Carlsson et al., 2021), DeCLUTR (Giorgi et al., 2021), ConSERT (Yan et al., 2021), Self-Guided Contrastive Learning (Kim et al., 2021), and SimCSE (Gao et al., 2021). SimCSE is a simple but extremely effective method which uses dropout as data augmentation strategy and is also the foundation of many following works. ArcCSE (Zhang et al., 2022) proposes ArcCon loss to enhance the pairwise discriminative power and a new task to capture the entailment relation among triplet sentences. Trans-Encoder (Liu et al., 2021) combines bi-encoders and cross-encoders learning paradigms into an iterative joint framework. DCLR (Zhou et al., 2022) generates noise-based negatives to guarantee the uniformity of the presentation space and punish false negatives. DiffCSE (Chuang et al., 2022) learns representations that are insensitive to certain types of augmentations and sensitive to others. Although achieving promising results, these methods fail to capture the fine-grained ranking knowledge among the sentences.

Learning to Rank Given a query example, learning to rank aims to rank a list of examples according to their similarities with the query. Learning to rank methods can be divided into three

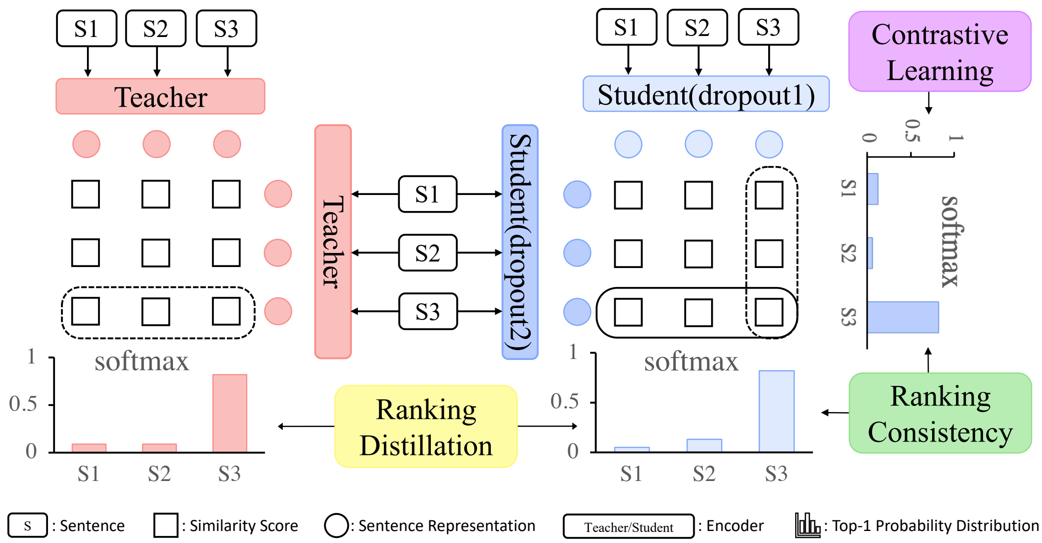


Figure 1: The framework of RankCSE which consists of three components: (1) standard contrastive learning object; (2) ranking consistency loss which ensures ranking consistency between two representations with different dropout masks; (3) ranking distillation loss which distills listwise ranking knowledge from the teacher.

categories: pointwise (Li et al., 2007), pairwise (Burges et al., 2005; 2006) and listwise (Cao et al., 2007; Xia et al., 2008; Volkovs & Zemel, 2009; Pobrotyn & Białobrzeski, 2021). Pointwise methods optimize the similarity between the query and each example, while pairwise approaches learn to correctly model the preference between two examples. Listwise methods directly evaluate the ranking of a list of examples based on the ground truth. In our framework, we leverage listwise ranking objectives for learning effective sentence representations, which have shown better performance compared to pointwise and pairwise methods.

3 PRELIMINARY

We provide some conceptual explanations and definitions in learning to rank.

Top One Probability Given the scores of all objects $S = \{s_i\}_{i=1}^n$, the top one probability of an object is the probability of its being ranked at top-1: $\widetilde{s_i} = \frac{\exp(s_i/\tau)}{\sum_{j=1}^n \exp(s_j/\tau)}$ where τ is a hyperparameter, usually utilized to smooth the distribution. We simply denote the formulation for calculating the top one distribution based on the scores S as: $\widetilde{S_\tau} = \text{softmax}(S/\tau)$.

Permutation Probability We use $\pi = \{\pi(i)\}_{i=1}^n$ to denote a permutation of the object indexes, which represents that the $\pi(i)$-th sample is ranked i-th. The probability of a specific permutation π is given as: $P(\pi|S,\tau) = \prod_{i=1}^n \frac{\exp(s_{\pi(i)}/\tau)}{\sum_{j=i}^n \exp(s_{\pi(j)}/\tau)}$.

4 METHODOLOGY

4.1 PROBLEM FORMULATION

Our goal is to learn sentence representations such that semantic similar sentences stay close while dissimilar ones should be far away in an unsupervised manner. Specifically, We aim to find an optimal function f that maps an sentence $s \in p_{\text{s}}$ to a d-dimensional vector $f(s) \in p_{\text{e}} \subseteq \mathcal{R}^d$, where p_{s} and p_{e} denote the distributions of sentences and sentence representations, respectively. Supposing s_1 and s_2 are more semantic similar than s_1 and s_3 ($s_1, s_2, s_3 \in p_{\text{s}}$), a good mapping function f

should satisfy that the distance between $f(s_1)$ and $f(s_2)$ is smaller than that between $f(s_1)$ and $f(s_3)$, i.e., $d(f(s_1), f(s_2)) < d(f(s_1), f(s_3))$, where d is the distance metric such as Euclidean distance and cosine similarity. In this way, the similarities among the sentences are preserved in the learned sentence representations.

The general idea of RankCSE is to learn semantically discriminative sentence representations by capturing the ranking information among the sentences. As shown in Figure 1, our model consists of three components: (1) standard contrastive learning objective (§4.2); (2) ranking consistency loss which ensures ranking consistency between two representations with different dropout masks (§4.3); (3) ranking distillation loss which distills listwise ranking knowledge from the teacher (§4.4).

4.2 Contrastive Learning

Contrastive learning aims to learn effective representations by pulling similar semantics closer and pushing away dissimilar ones. SimCSE (Gao et al., 2021) creates positive examples by applying different dropout masks and takes a cross-entropy object with in-batch negatives (Chen et al., 2017). More specifically, for any sentence x_i in a min-batch, we send it to the encoder $f(\cdot)$ twice and obtain two representations with different dropout masks $f(x_i)$, $f(x_i)'$. SimCSE use the standard InfoNCE loss (Oord et al., 2018) as the training objective:

$$\mathcal{L}_{\mathrm{infoNCE}} = -\sum_{i=1}^{N} \log \frac{\exp(d(f(x_i), f(x_i)')/\tau_1}{\sum_{j=1}^{N} \exp(d(f(x_i), f(x_j)')/\tau_1)}, \tag{1}$$

where N is the batch size, τ_1 is a temperature hyperparameter and $d(f(x_i), f(x_j)') = \frac{f(x_i)^\top f(x_j)'}{\|f(x_i)\| \cdot \|f(x_j)'\|}$ is the cosine similarity used in this work. Essentially, the contrastive learning objective is equivalent to maximizing the top one probability of the positive sample.

Although contrastive learning is effective in separating positive sentences with negative ones, it ignores the continuity modeling of the similarity. In other words, it is not effective in distinguishing highly similar sentences with moderate similar ones. To address this issue, we propose to directly model the ranking information among the sentences, which could enhance the discrimination of semantic similarity in the learned sentence representations.

4.3 Ranking Consistency

The main drawback of contrastive learning is that the distinction between the in-batch negatives is not modeled, resulting in less effective sentence representations in capturing the fine-grained sentence similarity. Therefore, instead of treating the negatives equivalently, we propose to explicitly model the ranking information within the sentences by ensuring the ranking consistency between the two similarity sets (circled by the solid and dashed curves respectively in the right part of Figure 1).

Concretely, by taking a close look at the contrastive modeling in section §4.2, there are two sets of sentence representations, $f(x_i)$ and $f(x_i)'$, derived from different dropout masks. For each sentence x_i, two lists of similarities with other sentences can be naturally obtained from the two representations, i.e., $S(x_i) = \{d(f(x_i), f(x_j)')\}_{j=1}^{N}$ and $S(x_i)' = \{d(f(x_i)', f(x_j))\}_{j=1}^{N}$. We then enforce the ranking consistency between these two similarity lists in our modeling. Intuitively, all corresponding elements in $S(x_i)$ and $S(x_i)'$ should have the same ranking positions.

Given two similarity lists $S(x_i)$ and $S(x_i)'$, we can obtain their top one probability distributions $\widetilde{S}_{\tau_1}(x_i) = \mathrm{softmax(S(x_i)}/\tau_1)$, $\widetilde{S}_{\tau_1}(x_i)' = \mathrm{softmax(S(x_i)'}/\tau_1)$. The ranking consistency can be ensured by minimizing the Jensen-Shannon (JS) divergence between the two top one probability distributions:

$$\begin{aligned} \mathcal{L}_{\mathrm{consistency}} &= \sum_{i=1}^{N} \mathrm{JS}(\widetilde{\mathrm{S}}_{\tau_1}(\mathrm{x_i}) || \widetilde{\mathrm{S}}_{\tau_1}(\mathrm{x_i})') = \\ &\sum_{i=1}^{N} (\widetilde{S}_{\tau_1}(x_i) \cdot \log(\frac{2\widetilde{S}_{\tau_1}(x_i)}{\widetilde{S}_{\tau_1}(x_i) + S_{\tau_1}(x_i)'}) + \widetilde{S}_{\tau_1}(x_i)' \cdot \log(\frac{2\widetilde{S}_{\tau_1}(x_i)'}{\widetilde{S}_{\tau_1}(x_i) + \widetilde{S}_{\tau_1}(x_i)'})). \end{aligned} \tag{2}$$

The reason we choose JS divergence instead of Kullback-Leibler (KL) divergence is that the two distributions are symmetric rather than one side being the ground truth.

4.4 Ranking Distillation

Contrastive learning based methods like SimCSE learn effective sentence representations with coarse-grained semantic ranking information (shown in Appendix A.6 and Appendix A.7), which have demonstrated their effectiveness in various downstream tasks. Orthogonal to ranking consistency, we further introduce ranking distillation by distilling the ranking knowledge from pre-trained teacher models into our learned sentence representations, to generalize effective ranking information from the weak ranking knowledge learned by SimCSE. More specifically, for each sentence in a min-batch, we obtain the similarity score list from the teacher model, which is then served as pseudo ranking labels in the ranking distillation. The intuitive idea is to transfer the ranking knowledge from the teacher to the student as guidance for learning ranking preserved sentence representations. In the ranking distillation, ListNet (Cao et al., 2007) and ListMLE (Xia et al., 2008) methods are utilized. Formally they are defined as:

$$\mathcal{L}_{\mathrm{rank}} = \sum_{i=1}^{N} \mathrm{rank}(\mathrm{S(x_i)}, \mathrm{S^T(x_i)}) \tag{3}$$

where $S(x_i)$ and $S^T(x_i)$ are the similarity score lists obtained from the student and the teacher, respectively, $\mathrm{rank}(\cdot,\cdot)$ is the listwise method.

ListNet The original ListNet minimizes the cross entropy between the permutation probability distribution and the ground truth as the training objective. However, the computations will be intractable when the number of examples n is large, since the number of permutations is $n!$. To reduce the computation complexity, the top one probability distribution is usually adopted as a substitute:

$$\mathcal{L}_{\mathrm{ListNet}} = -\sum_{i=1}^{N} \mathrm{softmax}(\mathrm{S^T(x_i)}/\tau_3) \cdot \log(\mathrm{softmax}(\mathrm{S(x_i)}/\tau_2)) \tag{4}$$

where τ_2 and τ_3 are temperature hyperparameters.[1]

ListMLE Different from ListNet, ListMLE aims to maximize the likelihood of the ground truth permutation π_i^T which represents the sorted indexes of the similarity scores calculated by the teacher model. The training objective of ListMLE can be defined as:

$$\mathcal{L}_{\mathrm{ListMLE}} = -\sum_{i=1}^{N} \log P(\pi_i^T | S(x_i), \tau_2) \tag{5}$$

In this work, we propose to use a multi-teacher model from which more listwise ranking knowledge can be transferred and preserved. In our experiments, we utilize the weighted average similarity scores of two teachers as pseudo ranking labels: $S^T(x_i) = \alpha S_1^T(x_i) + (1-\alpha) S_2^T(x_i)$ where α is a hyperparameter to balance the weight of the teachers.

The contrastive learning loss $\mathcal{L}_{\mathrm{infoNCE}}$ pushes apart the representations of different sentences to maximize the representation space, while the ranking consistency loss $\mathcal{L}_{\mathrm{consistency}}$ and the ranking distillation loss $\mathcal{L}_{\mathrm{rank}}$ pull similar negatives closer, thus capturing fine-grained semantic ranking information. Combining the above three loss functions, we can obtain the overall objective:

$$\mathcal{L}_{\mathrm{final}} = \mathcal{L}_{\mathrm{infoNCE}} + \beta \mathcal{L}_{\mathrm{consistency}} + \gamma \mathcal{L}_{\mathrm{rank}} \tag{6}$$

where β and γ are hyperparameters to balance losses.

5 Experiment

5.1 Setup

We evaluate our approach on two sentence related tasks, Semantic Textual Similarity (STS) and Transfer (TR) tasks. The SentEval toolkit (Conneau & Kiela, 2018) is used in our experiments. For

[1]In practice, we exclude the score of the positive pair from the list to calculate the top one distribution used in Eq.(4), to enhance the ranking information of negatives, because the score of the positive pair occupy most in the full top one distribution calculated by the teacher SimCSE.

Table 2: Sentence representations performance on STS tasks (Spearman's correlation). We employ our method to BERT and RoBERTa in both base and large versions. We directly import the results from the original papers and mark the best (bold) and second-best (underlined) results among models with the same PLMs. Results are statistically significant with p-value < 0.005.

PLMs	Methods	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	avg.
Non-BERT	GloVe(avg.)	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
	USE	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
$\text{BERT}_{\text{base}}$	first-last avg.	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
	+flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
	+whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
	+IS	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
	+ConSERT	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
	+SimCSE	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
	+DCLR	70.81	83.73	75.11	82.56	78.44	78.31	71.59	77.22
	+ArcCSE	72.08	84.27	76.25	82.32	79.54	79.92	72.39	78.11
	+DiffCSE	72.28	84.43	76.47	83.90	80.54	80.59	71.23	78.49
	+RankCSE$_{\text{listNet}}$	<u>74.38</u>	<u>85.97</u>	<u>77.51</u>	<u>84.46</u>	**81.31**	<u>81.46</u>	**75.26**	<u>80.05</u>
	+RankCSE$_{\text{listMLE}}$	**75.66**	**86.27**	**77.81**	**84.74**	<u>81.10</u>	**81.80**	<u>75.13</u>	**80.36**
$\text{BERT}_{\text{large}}$	+SimCSE	70.88	84.16	76.43	84.50	79.76	79.26	73.88	78.41
	+DCLR	71.87	84.83	77.37	84.70	79.81	79.55	74.19	78.90
	+ArcCSE	73.17	86.19	77.90	84.97	79.43	80.45	73.50	79.37
	+RankCSE$_{\text{listNet}}$	<u>74.75</u>	<u>86.46</u>	<u>78.52</u>	<u>85.41</u>	<u>80.62</u>	<u>81.40</u>	**76.12**	<u>80.47</u>
	+RankCSE$_{\text{listMLE}}$	**75.48**	**86.50**	**78.60**	**85.45**	**81.09**	**81.58**	<u>75.53</u>	**80.60**
$\text{RoBERTa}_{\text{base}}$	+SimCSE	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
	+DCLR	70.01	83.08	75.09	83.66	81.06	81.86	70.33	77.87
	+DiffCSE	70.05	83.43	75.49	82.81	82.12	82.38	71.19	78.21
	+RankCSE$_{\text{listNet}}$	**72.88**	**84.50**	**76.46**	<u>84.67</u>	**83.00**	**83.24**	<u>71.67</u>	**79.49**
	+RankCSE$_{\text{listMLE}}$	<u>72.74</u>	<u>84.24</u>	<u>75.99</u>	**84.68**	<u>82.88</u>	<u>83.16</u>	**71.77**	<u>79.35</u>
$\text{RoBERTa}_{\text{large}}$	+SimCSE	72.86	83.99	75.62	84.77	81.80	81.98	71.26	78.90
	+DCLR	73.09	84.57	76.13	85.15	81.99	82.35	71.80	79.30
	+RankCSE$_{\text{listNet}}$	<u>73.23</u>	<u>85.08</u>	**77.50**	**85.67**	**82.99**	**84.20**	**72.98**	**80.24**
	+RankCSE$_{\text{listMLE}}$	**73.40**	**85.34**	<u>77.25</u>	<u>85.45</u>	<u>82.64</u>	<u>84.14</u>	<u>72.92</u>	<u>80.16</u>

STS tasks, we evaluate on seven datasets: STS12-16 (Agirre et al., 2012; 2013; 2014; 2015; 2016), STS Benchmark (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014). These datasets contain pairs of sentences with similarity score labels from 0 to 5. Following SimCSE, we directly compute the cosine similarity between the sentence representations which means all the STS experiments are fully unsupervised, and report the Spearman's correlation. For TR tasks, we evaluate on seven datasets with the default configurations from SentEval: MR (Pang & Lee, 2005), CR (Hu & Liu, 2004), SUBJ (Pang & Lee, 2004), MPQA (Wiebe et al., 2005), SST-2 (Socher et al., 2013), TREC (Voorhees & Tice, 2000) and MRPC (Dolan & Brockett, 2005). We use a logistic regression classifier trained on top of the frozen sentence representations, and report the classification accuracy.

For fair comparison, we use the same 10^6 randomly sampled sentences from English Wikipedia provided by SimCSE. Following previous works, we start from pre-trained checkpoints of BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), and utilize [CLS] representation with MLP during training and [CLS] representation without MLP for evaluation. First we train SimCSE models including four variants: SimCSE-$\text{BERT}_{\text{base}}$, SimCSE-$\text{BERT}_{\text{large}}$, SimCSE-$\text{RoBERTa}_{\text{base}}$ and SimCSE-$\text{RoBERTa}_{\text{large}}$. We use the weighted average similarity scores of the first two as pseudo ranking labels for RankCSE-$\text{BERT}_{\text{base}}$ and RankCSE-$\text{BERT}_{\text{large}}$, while the last two for RankCSE-$\text{RoBERTa}_{\text{base}}$ and RankCSE-$\text{RoBERTa}_{\text{large}}$. We evaluate our model every 125 training steps on the dev set of STS-B and keep the best checkpoint for the evaluation on test sets of all STS and TR tasks. More training details can be found in Appendix A.1.

We compare RankCSE with several strong unsupervised sentence representation learning baselines, including average GloVe embeddings (Pennington et al., 2014), USE (Cer et al., 2018) and Skip-thought (Kiros et al., 2015), average BERT embeddings from the last layer, post-processing methods such as BERT-flow (Li et al., 2020) and BERT-whitening (Su et al., 2021), and contrastive learning methods such as IS-BERT, (Zhang et al., 2020), ConSERT (Yan et al., 2021) and SimCSE (Gao et al., 2021). We also include the recently proposed methods based on SimCSE such as DCLR

Table 3: Sentence representations performance on transfer tasks (accuracy). We employ our method to BERT and RoBERTa in both base and large versions. The results of DiffCSE† are obtained by its public available code and checkpoints for STS tasks, while others are imported from the original papers. We mark the best (bold) and second-best (underlined) results among models with the same PLMs. Results are statistically significant with p-value < 0.005.

PLMs	Methods	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	avg.
Non-BERT	GloVe(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
$\text{BERT}_{\text{base}}$	last avg.	78.66	86.25	94.37	88.66	84.40	**92.80**	69.54	84.94
	+IS	81.09	87.18	94.96	88.75	85.96	88.64	74.24	85.83
	+SimCSE	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.81
	+ArcCSE	79.91	85.25	**99.58**	89.21	84.90	89.20	74.78	86.12
	+DiffCSE†	81.76	86.20	94.76	89.21	86.00	87.60	75.54	85.87
	+RankCSE$_{\text{listNet}}$	**83.21**	88.08	95.25	**90.00**	**88.58**	90.00	76.17	**87.33**
	+RankCSE$_{\text{listMLE}}$	83.07	**88.27**	95.06	89.90	87.70	89.40	**76.23**	87.09
$\text{BERT}_{\text{large}}$	+SimCSE	**85.36**	89.38	95.39	89.63	90.44	91.80	76.41	88.34
	+ArcCSE	84.34	88.82	**99.58**	89.79	90.50	92.00	74.78	88.54
	+RankCSE$_{\text{listNet}}$	85.11	**89.56**	95.39	**90.30**	**90.77**	**93.20**	**77.16**	**88.78**
	+RankCSE$_{\text{listMLE}}$	84.63	89.51	95.50	90.08	90.61	**93.20**	76.99	88.65
$\text{RoBERTa}_{\text{base}}$	+SimCSE	81.04	87.74	93.28	86.94	86.60	84.60	73.68	84.84
	+DiffCSE†	82.42	88.34	93.51	87.28	87.70	86.60	76.35	86.03
	+RankCSE$_{\text{listNet}}$	**83.24**	**88.71**	93.93	**88.97**	**89.24**	90.20	**76.64**	**87.28**
	+RankCSE$_{\text{listMLE}}$	82.91	88.37	**93.97**	88.70	88.63	**90.40**	76.52	87.07
$\text{RoBERTa}_{\text{large}}$	+SimCSE	82.74	87.87	93.66	88.22	88.58	92.00	69.68	86.11
	+RankCSE$_{\text{listNet}}$	**84.30**	**89.06**	**94.60**	89.53	**89.46**	92.60	73.91	**87.64**
	+RankCSE$_{\text{listMLE}}$	83.48	88.64	94.20	**89.74**	88.63	**93.00**	**74.61**	87.47

(Zhou et al., 2022), ArcCSE (Zhang et al., 2022) and DiffCSE (Chuang et al., 2022). We don't compare with TRANS-ENCODER (Liu et al., 2021), because it uses pairs of sentences within STS datasets which are not general for unsupervised sentence representation learning.

5.2 MAIN RESULTS

Results on STS Tasks As shown in Table 2, it is clear that RankCSE significantly outperforms the previous methods on all datasets and PLMs, which demonstrates the effectiveness of our approach. For example, compared with SimCSE, RankCSE has brought noticeable improvements: 4.11% on $\text{BERT}_{\text{base}}$, 2.19% on $\text{BERT}_{\text{large}}$, 2.92% on $\text{RoBERTa}_{\text{base}}$ and 1.34% on $\text{RoBERTa}_{\text{large}}$. RankCSE-$\text{BERT}_{\text{base}}$ even outperforms SimCSE-$\text{BERT}_{\text{large}}$ by nearly 2%. Compared with the previous state-of-the-art methods, RankCSE still achieves consistent improvements, which validates that RankCSE is able to obtain more semantically discriminative representations by incorporating ranking consistency and ranking distillation. We also observe that the performances of $\text{RankCSE}_{\text{listNet}}$ and $\text{RankCSE}_{\text{listMLE}}$ are very consistent across all datasets, which demonstrates the effectiveness of both listwise ranking methods.

Results on TR Tasks It can be seen in Table 3 that RankCSE achieves the best performance among all the compared baselines on all PLMs. Note that for DiffCSE, we obtain the results by its public available code and checkpoints for STS tasks[2] instead of directly importing the results from its original paper. DiffCSE uses different dev sets to find the best hyperparameters for the two tasks (STS-B dev set for STS tasks, dev sets of 7 TR tasks for TR tasks), while other methods only use the STS-B dev set, which is a not fair comparison. To make a comprehensive comparison with DiffCSE, we also conduct experiments using dev sets of 7 TR tasks to find best hyperparameters for TR tasks. More detailed results are provided in Appendix A.2. Another observation is that the performance of the $\text{RankCSE}_{\text{listNet}}$ is slightly better than that of the $\text{RankCSE}_{\text{listMLE}}$. Our hypothesis is that the inaccurate pseudo ranking labels introduce more errors in the calculation of the permutation probability than the top one probability. Nevertheless, both listwise methods achieve better results than the baselines, which is consistent with the results in Table 2.

[2] `https://github.com/voidism/DiffCSE`

Table 4: Ablation studies of different loss functions based on $\text{BERT}_{\text{base}}$

Models	**STS(avg.)**	**TR(avg.)**
SimCSE	76.25	85.81
$\text{RankCSE}_{\text{listNet}}$	80.05	87.33
w/o $\mathcal{L}_{\text{consistency}}$	79.56	86.80
w/o $\mathcal{L}_{\text{infoNCE}}$	79.72	86.91
w/o $\mathcal{L}_{\text{consistency}}$,$\mathcal{L}_{\text{infoNCE}}$	79.41	86.76
$\text{RankCSE}_{\text{listMLE}}$	80.36	87.09
w/o $\mathcal{L}_{\text{consistency}}$	79.88	86.65
w/o $\mathcal{L}_{\text{infoNCE}}$	79.95	86.73
w/o $\mathcal{L}_{\text{consistency}}$,$\mathcal{L}_{\text{infoNCE}}$	79.73	86.24
RankCSE w/o $\mathcal{L}_{\text{rank}}$	76.93	85.97
RankCSE w/o $\mathcal{L}_{\text{infoNCE}}$, $\mathcal{L}_{\text{rank}}$	73.74	85.56

Table 5: Comparisons of different teachers. Results are average STS performance using $\text{BERT}_{\text{base}}$.

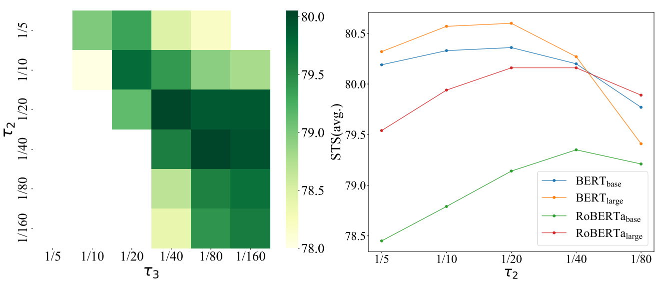
Teacher	$\textbf{RankCSE}_{\text{listNet}}$	$\textbf{RankCSE}_{\text{listMLE}}$
$\text{SimCSE-BERT}_{\text{base}}$	77.48	77.75
$\text{DiffCSE-BERT}_{\text{base}}$	78.87	79.06
$\text{SimCSE-BERT}_{\text{large}}$	79.66	79.81
$\text{SimCSE-BERT}_{\text{base}}$+$\text{DiffCSE-BERT}_{\text{base}}$	79.10	79.28
$\text{SimCSE-BERT}_{\text{base}}$+$\text{SimCSE-BERT}_{\text{large}}$	80.05	80.36
$\text{DiffCSE-BERT}_{\text{base}}$+$\text{SimCSE-BERT}_{\text{large}}$	80.20	80.47

5.3 Analysis and Discussion

Ablation Study To investigate the impact of different losses in our approach, we conduct a set of ablation studies by removing $\mathcal{L}_{\text{infoNCE}}$, $\mathcal{L}_{\text{consistency}}$ and $\mathcal{L}_{\text{rank}}$ from Eq.(6). The average results on STS and TR tasks are reported in Table 4. There are several observations from the results. First, when $\mathcal{L}_{\text{rank}}$ is removed, the performance significantly drops in both STS and TR tasks, which indicates the effectiveness of $\mathcal{L}_{\text{rank}}$ in our modeling. Second, it is also clear that without $\mathcal{L}_{\text{infoNCE}}$ or $\mathcal{L}_{\text{consistency}}$, the model performance also decreases, especially on TR tasks. Third, it is worth mentioning that RankCSE with only $\mathcal{L}_{\text{rank}}$ can also outperform the teachers on STS tasks. The reason is that RankCSE is able to preserve ranking knowledge from multiple teachers, and generalize fine-grained ranking information from multiple coarse-grained representations. Fourth, since $\mathcal{L}_{\text{consistency}}$ does not explicitly distingish the positives from negatives, RankCSE with only $\mathcal{L}_{\text{consistency}}$ will preserve inaccurate rankings leading to significant performance drop. Finally, the RankCSE with all components achieves the best performance on both STS and TR tasks.

Comparisons of Different Teachers We conduct experiments to explore the impact of different teachers on the performance of RankCSE. As shown in Table 5, RankCSE outperforms the teacher model which indicates that incorporating ranking consistency and ranking distillation leads to more semantically discriminative sentence representations. Comparing the performance of RankCSE using different teachers, we observe that better teacher leads to better RankCSE , which is consistent with our expectation since accurate ranking labels yield more effective ranking knowledge transfer. Another observation is that the performance of RankCSE with multi-teacher is better than that with single teacher, which verifies that RankCSE is able to preserve listwise ranking knowledge from more than one teacher. It is also interesting to see that using $\text{DiffCSE-BERT}_{\text{base}}$ and $\text{SimCSE-BERT}_{\text{large}}$ as multi-teacher leads to even higher performance than the results in Table 2. We plan to conduct more investigation along this direction to explore the upper bound of improvements.

Effect of Hyperparameters To study the effect of temperature hyperparameters, we conduct experiments by setting different τ_2 and τ_3. As shown in Figure 2a, we find that large discrepancy between τ_2 and τ_3 leads to significant drop in the performance of $\text{RankCSE}_{\text{ListNet}}$. The best temperature setting for $\text{RankCSE}_{\text{ListNet}}$ is $\tau_2 : \tau_3 = 2 : 1$. The performance of $\text{RankCSE}_{\text{ListMLE}}$ has similar trends based on different PLMs, as shown in Figure 2b. For both $\text{RankCSE}_{\text{ListNet}}$ and $\text{RankCSE}_{\text{ListMLE}}$, the temperature should be set moderate.



(a) RankCSE$_{\text{listNet}}$ (b) RankCSE$_{\text{listMLE}}$

Figure 2: Effect of the temperatures τ_2 and τ_3. Results are average STS performance, and RankCSE$_{\text{listNet}}$ is based on BERT$_{\text{base}}$ while RankCSE$_{\text{listMLE}}$ is based on different PLMs. We do not demonstrate results below 78 to make the variation obvious.

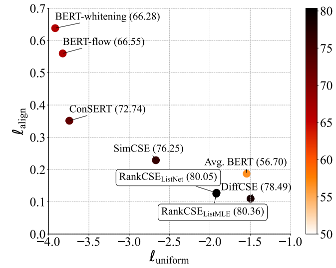


Figure 3: ℓ_{align}-ℓ_{uniform} plot for different sentence representation methods based on BERT$_{\text{base}}$, which are all measured on the STS-B dev set. Color of points represent average STS performance.

Table 6: Mean and standard deviation across five different runs of RankCSE and SimCSE.

PLMs	**RankCSE$_{\text{listNet}}$**		**RankCSE$_{\text{listMLE}}$**		**SimCSE**	
	STS(avg.)	TR(avg.)	STS(avg.)	TR(avg.)	STS(avg.)	TR(avg.)
BERT$_{\text{base}}$	80.00±0.13	87.28±0.19	80.39±0.04	87.05±0.06	75.52±0.70	85.44±0.47
BERT$_{\text{large}}$	80.41±0.10	88.74±0.14	80.59±0.05	88.63±0.06	77.79±0.64	88.10 ±0.36
RoBERTa$_{\text{base}}$	79.42±0.15	87.26± 0.20	79.36±0.03	87.06±0.04	76.45±0.56	84.74±0.38
RoBERTa$_{\text{large}}$	80.18±0.13	87.50±0.18	80.07±0.12	87.46±0.11	78.53±0.49	86.29±0.33

Robustness of RankCSE We conduct 5 runs of models training with the hyperparameter settings which can be referred to Appendix A.1 with different random seeds, and then calculate the mean and standard deviation values. The results provided in Table 6 demonstrate both the superior performance and the robustness of our model. It can also be seen that RankCSE$_{\text{listMLE}}$ achieves similar performance but more stable results compared with RankCSE$_{\text{listNet}}$.

Alignment and Uniformity Following previous works (Wang & Isola, 2020), we use alignment and uniformity to measure the quality of representation space. Alignment measures the distance between similar instances, while uniformity measures how well the representations are uniformly distributed (detailed in Appendix A.4). For both measures, the smaller value indicates the better result. We plot the distribution of ℓ_{align}-ℓ_{uniform} for different models using BERT$_{\text{base}}$ which are measured on the STS-B dev set. As shown in Figure 3, RankCSE effectively improves both alignment and uniformity compared with average BERT embeddings, while SimCSE and DiffCSE only improves uniformity and alignment respectively. Since RankCSE pulls similar negatives closer during incorporating ranking consistency and ranking distillation, RankCSE has smaller alignment and bigger uniformity than SimCSE. When compared with DiffCSE, RankCSE has smaller uniformity whereas similar alignment. We consider that RankCSE achieves a better trade-off than SimCSE.

6 Conclusion

In this work, we propose RankCSE, an unsupervised approach to learn more semantically discriminative sentence representations. The core idea of RankCSE is incorporating ranking consistency and ranking distillation with contrastive learning into a unified framework. When simultaneously ensuring ranking consistency and distilling listwise ranking knowledge from the teacher, RankCSE can learn how to make fine-grained distinctions in semantics, leading to more semantically discriminative sentence representations. Experimental results on STS and TR tasks demonstrate that RankCSE outperforms previous state-of-the-art methods. We also conduct thorough ablation study and analysis to demonstrate the effectiveness of each component and justify the inner workings of our approach. We leave what is the upper bound of improvements of the teacher for future work.

REFERENCES

Hervé Abdi. The kendall rank correlation coefficient. Encyclopedia of measurement and statistics, 2:508–510, 2007.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 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), pp. 385–393, 2012.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. * 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, pp. 32–43, 2013.

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

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 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), pp. 252–263, 2015.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor González-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pp. 497–511, 2016.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In Proceedings of the 22nd international conference on Machine learning, pp. 89–96, 2005.

Christopher Burges, Robert Ragno, and Quoc Le. Learning to rank with nonsmooth cost functions. Advances in neural information processing systems, 19, 2006.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In Proceedings of the 24th international conference on Machine learning, pp. 129–136, 2007.

Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. Semantic re-tuning with contrastive tension. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=Ov_sMNau-PF.

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

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder for english. In Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations, pp. 169–174, 2018.

Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. On sampling strategies for neural network-based collaborative filtering. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 767–776, 2017.

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

Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 659–666, 2008.

Alexis Conneau and Douwe Kiela. Senteval: An evaluation toolkit for universal sentence representations. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.

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

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

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 55–65, 2019.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6894–6910, 2021.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. Declutr: Deep contrastive learning for unsupervised textual representations. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 879–895, 2021.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1367–1377, 2016.

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

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. Self-guided contrastive learning for bert sentence representations. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2528–2540, 2021.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. Advances in neural information processing systems, 28, 2015.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In International conference on machine learning, pp. 1188–1196. PMLR, 2014.

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

Ping Li, Qiang Wu, and Christopher Burges. Mcrank: Learning to rank using multiple classification and gradient boosting. Advances in neural information processing systems, 20, 2007.

Fangyu Liu, Yunlong Jiao, Jordan Massiah, Emine Yilmaz, and Serhii Havrylov. Trans-encoder: Unsupervised sentence-pair modelling through self-and mutual-distillations. In International Conference on Learning Representations, 2021.

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

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

Shutian Ma, Chengzhi Zhang, and Daqing He. Document representation methods for clustering bilingual documents. Proceedings of the Association for Information Science and Technology, 53(1):1–10, 2016.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 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), pp. 216–223, 2014.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26, 2013.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), pp. 271–278, 2004.

Bo Pang and Lillian Lee. 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), pp. 115–124, 2005.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543, 2014.

Przemysław Pobrotyn and Radosław Białobrzeski. Neuralndcg: Direct optimisation of a ranking metric via differentiable relaxation of sorting. arXiv preprint arXiv:2102.07831, 2021.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992, 2019.

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

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. arXiv preprint arXiv:2103.15316, 2021.

Maksims N Volkovs and Richard S Zemel. Boltzrank: learning to maximize expected ranking gain. In Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1089–1096, 2009.

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

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

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

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In Proceedings of the 25th international conference on Machine learning, pp. 1192–1199, 2008.

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

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

Yan Zhang, Ruidan He, Zuozhu Liu, Lidong Bing, and Haizhou Li. Bootstrapped unsupervised sentence representation learning. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 5168–5180, 2021.

Yuhao Zhang, Hongji Zhu, Yongliang Wang, Nan Xu, Xiaobo Li, and Binqiang Zhao. A contrastive framework for learning sentence representations from pairwise and triple-wise perspective in angular space. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 4892–4903, 2022.

Kun Zhou, Beichen Zhang, Wayne Xin Zhao, and Ji-Rong Wen. Debiased contrastive learning of unsupervised sentence representations. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6120–6130, 2022.

Table 7: The Hyperparameters for RankCSE Training.

	RankCSE-BERT				**RankCSE-RoBERTa**			
	base		large		base		large	
	listNet	listMLE	listNet	listMLE	listNet	listMLE	listNet	listMLE
Batch size	128	128	128	128	128	128	128	128
Learning rate	3e-5	2e-5	3e-5	2e-5	3e-5	3e-5	2e-5	3e-5
τ_1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
τ_2	0.025	0.05	0.05	0.05	0.025	0.025	0.025	0.025
τ_3	0.0125	-	0.025	-	0.0125	-	0.0125	-
α	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3
β	1	1	1	1	1	1	1	1
γ	1	1	1	1	1	1	1	1

A Appendix

A.1 Training Details

We implement all experiments with the deep learning framework PyTorch on a single NVIDIA Tesla A100 GPU (40GB memory). We carry out grid-search of learning rate $\in \{$2e-5, 3e-5$\}$ and temperatures $\tau_2, \tau_3 \in \{0.0125, 0.025, 0.05\}$, while setting batch size to 128, temperature τ_1 to 0.05, α to 1/3 ,β to 1, γ to 1 and the rate of linear scheduling warm-up to 0.05 for all the experiments. We train our models for 4 epochs, and evaluate the model every 125 steps on the dev set of STS-B and keep the best checkpoint for the final evaluation on test sets of all STS and TR tasks. The hyperparameter settings we adopt are shown in Table 7. Following SimCSE, we utilize [CLS] representation with MLP during training and [CLS] representation without MLP for evaluation. We utilize the weighted average similarity scores of SimCSE-BERT$_{\text{base}}$ and SimCSE-BERT$_{\text{large}}$ as pseudo ranking labels for RankCSE-BERT$_{\text{base}}$ and RankCSE-BERT$_{\text{large}}$, while the weighted average similarity scores of SimCSE-RoBERTa$_{\text{base}}$ and SimCSE-RoBERTa$_{\text{large}}$ as pseudo ranking labels for RankCSE-RoBERTa$_{\text{base}}$ and RankCSE-RoBERTa$_{\text{large}}$.

A.2 Transfer Tasks

For a more comprehensive comparison with DiffCSE on TR tasks, we also use dev sets of 7 TR tasks to find the best hyperparameters and checkpoints. As shown in Table 8, RankCSE still outperforms DiffCSE in this setting.

A.3 Training Efficiency

We list the training time of SimCSE and RankCSE in Table 9, which are all tested on a single NVIDIA Tesla A100 GPU (40GB memory). All RankCSE base models can be trained within 2 hours and large models can be trained within 3.7 hours. Since RankCSE need to calculate pseudo ranking labels of the teacher, it has longer training time per epoch than SimCSE.

A.4 Alignment and Uniformity

Wang & Isola (2020) propose to use two properties related to contrastive learning, alignment and uniformity, to measure the quality of representation space. Alignment calculates expected distance between normalized representations of positive pairs p_{pos}:

$$\ell_{\text{align}} \triangleq \mathop{\mathbb{E}}_{(x,x^+)\sim p_{\text{pos}}} \|f(x) - f(x^+)\|^2, \tag{7}$$

while uniformity measures how well the normalized representations are uniformly distributed:

$$\ell_{\text{uniform}} \triangleq \log \mathop{\mathbb{E}}_{x,y \overset{i.i.d.}{\sim} p_{\text{data}}} e^{-2\|f(x)-f(y)\|^2}, \tag{8}$$

Table 8: Sentence representations performance on TR tasks (accuracy) using the dev sets of 7 TR tasks to find the best hyperparameters. The results of DiffCSE are from its original paper. We mark the best (bold) and second-best (underlined) results among models with the same PLMs.

PLMs	Methods	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	avg.
$\text{BERT}_{\text{base}}$	+DiffCSE	82.69	87.23	95.23	89.28	86.60	90.40	76.58	86.86
	+RankCSE$_{\text{listNet}}$	**83.64**	**88.32**	**95.26**	89.99	**89.02**	**90.80**	**77.10**	**87.73**
	+RankCSE$_{\text{listMLE}}$	83.05	88.03	95.13	**90.00**	88.41	90.60	76.81	87.43
$\text{RoBERTa}_{\text{base}}$	+DiffCSE	82.82	88.61	**94.32**	87.71	88.63	90.40	76.81	87.04
	+RankCSE$_{\text{listNet}}$	83.09	88.72	94.26	**89.04**	**89.79**	**91.20**	**78.32**	**87.77**
	+RankCSE$_{\text{listMLE}}$	**83.16**	**88.74**	94.13	89.01	89.73	90.60	77.22	87.51

Table 9: Training efficiency of SimCSE and RankCSE. $\text{SimCSE}_{\text{base}}$ and $\text{SimCSE}_{\text{large}}$ provide pseudo ranking labels for every RankCSE model.

	SimCSE				**RankCSE**			
	BERT		RoBERTa		BERT		RoBERTa	
	base	large	base	large	base	large	base	large
Batch size	64	64	128	128	128	128	128	128
Epoch	1	1	1	1	4	4	4	4
Time	32min	65min	20min	45min	120min	220min	120min	220min
Time per epoch	32min	65min	20min	45min	30min	55min	30min	55min

where p_{data} denotes the distribution of sentence pairs. Smaller alignment means positive instances have been pulled closer, while smaller uniformity means random instances scatter on the hypersphere. These two measures are smaller the better, and well aligned with the object of contrastive learning.

A.5 Cosine Similarity Distribution

We demonstrate the distribution of cosine similarities for sentence pairs of STS-B dev set in Figure 4. We can observe that cosine similarity distributions from all models are consistent with human ratings. However, the cosine similarities of RankCSE are slightly higher than that of SimCSE under the same human rating, as RankCSE pulls similar negatives closer during incorporating ranking consistency and ranking distillation, and shows lower variance. Compared with DiffCSE, RankCSE shows a more scattered distribution. This observation further validates that RankCSE can achieve a better alignment-uniformity balance.

A.6 Ranking Tasks

We build the ranking task based on each STS dataset to verify that RankCSE can capture fine-grained semantic ranking information. For one sentence x_i, if there are more than three sentence pairs

(a) SimCSE (b) DiffCSE (c) RankCSE

Figure 4: The distribution of cosine similarities for sentence pairs of STS-B dev set. Along the y-axis are 5 groups of pairs split based on ground truth ratings, and x-axis is the cosine similarity.

Table 10: Sentence representations performance on ranking tasks (KCC and NDCG) using $\text{BERT}_{\text{base}}$. The results of SimCSE and DiffCSE are obtained by their public available codes and checkpoints. We mark the best (bold) and second-best (underlined) results.

Metrics	Methods	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	avg.
KCC	+SimCSE	36.08	36.60	<u>44.14</u>	49.02	54.66	58.44	<u>54.65</u>	47.66
	+DiffCSE	<u>38.59</u>	<u>41.89</u>	42.37	<u>51.19</u>	**58.90**	<u>59.21</u>	53.42	<u>49.37</u>
	+RankCSE	**42.79**	**46.26**	**44.53**	**52.00**	<u>57.21</u>	**63.64**	**57.40**	**51.98**
NDCG	+SimCSE	97.80	89.33	92.71	<u>96.93</u>	94.28	96.49	<u>98.44</u>	95.14
	+DiffCSE	**98.35**	<u>90.22</u>	<u>93.05</u>	96.91	<u>94.79</u>	<u>97.05</u>	98.34	<u>95.53</u>
	+RankCSE	<u>98.20</u>	**92.27**	**93.46**	**97.21**	**95.24**	**97.45**	**98.67**	**96.07**

Table 11: Three examples of an input sentence and other sentences from the STS datasets, with their similarity scores and rankings. The label scores are from human annotations. The SimCSE and RankCSE similarity scores are from the model predictions respectively, with the corresponding ranking positions. It can be seen that sentence rankings based on SimCSE are incorrect, while RankCSE generates more effective scores with accurate rankings.

Sentences	Label	SimCSE	RankCSE
• Broccoli are being cut by a woman	4.80 (1)	0.82 (2)	0.95 (1)
• A woman is slicing vegetables	4.20 (2)	0.83 (1)	0.91 (2)
• A woman is cutting some plants	3.50 (3)	0.74 (5)	0.87 (3)
• There is no woman cutting broccoli	3.40 (4)	0.76 (3)	0.85 (4)
• A woman is cutting some flowers	2.87 (5)	0.71 (7)	0.81 (5)
• A man is slicing tomatoes	2.60 (6)	0.75 (4)	0.79 (6)
• A man is cutting tomatoes	2.40 (7)	0.73 (6)	0.76 (7)
Input Sentence: A woman is cutting broccoli			
• A woman is breaking eggs	4.80 (1)	0.93 (2)	0.97 (1)
• A man is cracking eggs	3.60 (2)	0.94 (1)	0.91 (2)
• A woman is talking to a man	1.60 (3)	0.45 (5)	0.65 (3)
• A man and a woman are speaking	1.40 (4)	0.47 (3)	0.61 (4)
• A man is talking to a boy	1.00 (5)	0.46 (4)	0.56 (5)
Input Sentence: A woman is cracking eggs			
• a and c are on the same closed path with the battery	3.60 (1)	0.81 (1)	0.90 (1)
• bulb a and bulb c affect each other.	2.80 (2)	0.58 (3)	0.75 (2)
• the are on the same wire	1.60 (3)	0.60 (2)	0.68 (3)
• becuase breaking one bulb then affects the ability of the others to light up.	1.20 (4)	0.37 (5)	0.59 (4)
• if one bulb is removed , the others stop working	0.60 (5)	0.38 (4)	0.54 (5)
Input Sentence: a and c are in the same closed path			

(x_i, x_i^j) containing x_i with similarity score label y_i^j in the dataset, we view $\{x_i, x_i^j, y_i^j\}_{j=1}^k (k > 3)$ as a sample of the ranking task, as shown in Table 11. We adopt KCC (Kendall's correlation coefficient (Abdi, 2007)) and NDCG (normalized discounted cumulative gain (Clarke et al., 2008)) as evaluation metrics for ranking tasks, and demonstrate the results in Table 10. RankCSE outperforms SimCSE and DiffCSE on both KCC and NDCG, which validates that RankCSE can capture fine-grained semantic ranking information by incorporating ranking consistency and ranking distillation. Another observation is that SimCSE and DiffCSE also achieve moderate results, which shows they can distinguish coarse-grained semantic differences via contrastive learning.

A.7 Case Study

We present another three examples of an input sentence and other sentences from the STS datasets, with their similarity scores and rankings in Table 11. It is obvious that the similarity scores produced by RankCSE are more effective than SimCSE, with consistent rankings to the ground-truth labels. It further demonstrates that SimCSE only captures coarse-grained semantic ranking information via contrastive learning, while RankCSE can capture fine-grained semantic ranking information.

Table 12: A listing of train/dev/test stats of STS datasets.

Dataset	Train	Dev	Test
STS12	-	-	3108
STS13	-	-	1500
STS14	-	-	3750
STS15	-	-	3000
STS16	-	-	1186
STS-B	5749	1500	1379
SICK-R	4500	500	4927

Table 13: A listing of train/dev/test stats of TR datasets.

Dataset	Train	Dev	Test
MR	10662	-	-
CR	3775	-	-
SUBJ	10000	-	-
MPQA	10606	-	-
SST	67349	872	1821
TREC	5452	-	500
MRPC	4076	-	1725

For example, SimCSE can distinguish between similar and dissimilar sentences, while it can not distinguish between very similar and less similar sentences as RankCSE.

A.8 Data Statistics

The complete listings of train/dev/test stats of STS and TR datasets can be found in Table 12 and 13, respectively. Note that for STS tasks, we only use test sets for the final evaluation and dev set of STS-B to find best hyperparameters and checkpoints. The train sets of all STS datasets are not used in our experiments. For TR tasks, we follow the default settings of SentEval toolkit (Conneau & Kiela, 2018) to use 10-fold evaluation for all TR datasets except SST. We can directly use the already split datasets to evaluate on SST.