Refining Sentence Embedding Model through Ranking Sentences Generation with Large Language Models

Anonymous ACL submission

Abstract

Sentence embedding is essential for many NLP tasks, with contrastive learning methods achieving strong performance using annotated datasets like NLI. Yet, the reliance on manual labels limits scalability. Recent studies leverage large language models (LLMs) to generate sentence pairs, reducing annotation dependency. However, they overlook ranking information crucial for fine-grained semantic distinctions. To tackle this challenge, we propose a method for controlling the generation direction of LLMs in the latent space. Unlike unconstrained generation, the controlled approach ensures meaningful semantic divergence. Then, we refine exist sentence embedding model by integrating ranking information and semantic information. Experiments on multiple benchmarks demonstrate that our method achieves new SOTA performance with a modest cost in ranking sentence synthesis[1].

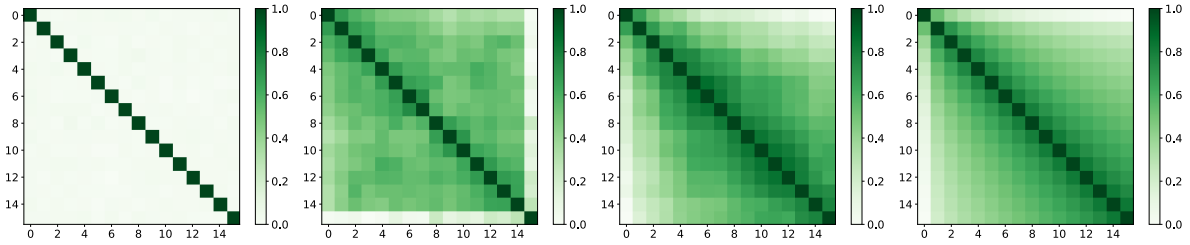
1 Introduction

Sentence embedding is a fundamental task in natural language processing. It provides effective semantic representations for various downstream applications, such as semantic search (He et al., 2023), text classification (Wang et al., 2022a), and question-answering systems (Nguyen et al., 2022). In recent years, significant progress has been made in the study of sentence embeddings, with methods based on contrastive learning standing out in particular. These approaches learn embeddings of sentences by bringing semantically similar sentences closer together and pushing dissimilar ones further apart. Current mainstream research relies on high-quality annotated data, especially natural language inference (NLI) datasets (Bowman et al., 2015; Williams et al., 2018). For instance, supervised contrastive learning methods based on NLI have demonstrated a remarkable ability to surpass the unsupervised approaches (Limkonchotiwat et al., 2022; Jiang et al., 2022). However, such annotated datasets are often unavailable in most real-world scenarios, and the manual construction of these datasets incurs extremely high costs.

To reduce reliance on manually annotated data, recent studies have begun to explore leveraging the powerful generative capabilities of large language models (LLMs) to construct high-quality sentence pairs automatically. For instance, SynCSE (Zhang et al., 2023) employs LLMs to generate semantically similar sentence pairs, enhancing the effectiveness of contrastive learning. MultiCSR (Wang et al., 2024) further evaluates the quality of LLM-generated outputs, filtering out erroneous results. GCSE (Lai et al., 2024) utilizes knowledge graphs to extract entities and quantities, enabling LLMs to generate more diverse and knowledge-enriched samples. These approaches significantly diminish the dependence on manual annotation.

However, current research focuses on generating sentence pairs, overlooking the critical role of ranking sentences. While sentence pairs can capture the similarity between sentences, they fail to effectively distinguish between "highly similar" sentences and "slightly different". Liu et al. (2023) point out that the limitation of sentence pairs lies in their inability to represent finer-grained semantic distinctions. Existing unsupervised methods, such as RankCSE (Liu et al., 2023) and RankEncoder (Seonwoo et al., 2023), attempt to construct ranking information using in-batch data to address this shortcoming. However, the ranking information in these methods is derived in an unsupervised manner, lacking explicit ranking supervision signals. As shown in Figure 1 (a), the heatmap illustrates the similarity calculations between sentences acquired from the in-batch data. We observe that the relationships among sentences within the in-batch data are treated as equivalent, failing to capture the

[1]Our code is available at https://anonymous.4open.science/r/RankingSentence-44EE



(a) (b) (c) (d)

Figure 1: Sentence similarity within the ranking sentences obtained through different methods. We randomly selected 1,000 ranking sentences generated by these methods and extracted the first 16 sentences from each ranking sentence. Then, we use a trained DiffCSE (Chuang et al., 2022) to obtain their embeddings and compute their average similarity. (a) Directly extracted from the trained batch. (b) Prompting the LLM to generate complete ranking sentences at once. (c) Prompting the LLM to generate ranking sentences step by step. (d) Generating the ranking sentences using our proposed directionally controlled generation method.

hierarchical semantic distinctions. Thus, we propose a new research question: **Can LLMs be used to generate ranking sentences to enhance the performance of sentence embedding models?**

A straightforward method for generating ranking sentences is to directly prompt LLMs to produce them. However, such an unconstrained generation process will result in ambiguous sentence semantic relationships. As illustrated in Figure 1 (b) and (c), neither prompting the LLM to generate complete ranking sentences at once nor guiding it to generate them step by step can ensure a gradual increase in semantic distance [2]. Thus, it fails to provide high-quality ranking information for sentence embedding models.

In this paper, we propose a latent space directional control method for ranking sentence generation and a post-training method for synthesized ranking sentences. Specifically, we design a directionally controlled generation method that LLMs to produce ranking sentences. By utilizing the generation probabilities of the preceding two sentences, we ensure that the resulting latent space remains in a consistent direction. As shown in Figure 1 (d), our generated ranking sentences exhibit a gradual increase in semantic divergence within the semantic space. Then, we integrate the ranking information and semantic information from the synthesized ranking sentences to refine existing sentence embedding models through post-training. The contributions of this paper can be summarized as follows:

- We are the first to use LLMs to generate ranking sentences. We have curated a dataset consisting of 16,063 ranking sentences and 530,079 sentences, opening new avenues for research in sentence embedding.
- We propose a post-training approach that incorporates both ranking and semantic information from the synthesized ranking sentences, substantially enhancing the performance of sentence embedding models on STS, reranking, and TR tasks.
- Extensive experiments on multiple benchmark datasets demonstrate the effectiveness of the proposed method. Even using merely 5% of the synthesized ranking sentences is sufficient to surpass the original sentence embedding model significantly.

[2]We provide a detailed description of their generation process in Appendix A.

2 Background

In unsupervised sentence embedding models, a series of works represented by SimCSE (Gao et al., 2021) employ contrastive learning to acquire effective embeddings by bringing semantically similar neighbours closer while pushing dissimilar ones apart. Assume there exists an unlabeled dataset $\mathcal{X}$. For each sentence $x \in \mathcal{X}$, SimCSE processes the same input through an encoder, such as BERT (Kenton and Toutanova, 2019) or RoBERTa (Liu, 2019), twice. It yields two embeddings $\boldsymbol{h}_i$ and $\boldsymbol{h}_i^+$ for the i-th sentence with different dropout masks. The objective for the pair $(\boldsymbol{h}_i, \boldsymbol{h}_i^+)$ within a mini-batch of B is:

$$\mathcal{L}_i = -\log \frac{e^{\mathrm{sim}(\boldsymbol{h_i}, \boldsymbol{h}_i^+)/\tau}}{\sum_{j=1}^{B} e^{\mathrm{sim}(\boldsymbol{h}_i, \boldsymbol{h}_j^+)/\tau}}, \quad (1)$$

where τ is a temperature hyperparameter and $\mathrm{sim}(\cdot, \cdot)$ is the cosine similarity between two embeddings. Follow-up methods such as CARDS

(Wang et al., 2022b), DiffCSE (Chuang et al., 2022), and RankCSE (Liu et al., 2023) have been proposed.

Data Generation with LLM. Unsupervised approaches often lag behind their supervised counterparts, which leverage labeled datasets such as natural language inference (NLI) corpora (Bowman et al., 2015; Williams et al., 2018). The NLI dataset provides each x with a positive sample x^+ and a hard negative sample x^- to construct the triplet (x, x^+, x^-) for the supervised contrastive loss. However, such annotated data are typically unavailable in the majority of scenarios. Thus, researchers began exploring the potential of LLMs for the triplet (x, x^+, x^-) generation for each $x \in \mathcal{X}$. A representative work is SynCSE (Zhang et al., 2023), which leverages ChatGPT (OpenAI, 2022) in a few-shot setting to generate positive samples and hard negative samples. MultiCSR (Wang et al., 2024) and GCSE (Lai et al., 2024) further refined the process of utilizing LLMs for data generation. These works fundamentally revolve around generating triplets.

Ranking Sentences Generation. We further advance this research by concentrating on the generation of ranking sentences. Formally, a ranking sentence is defined as a sequence of sentences $l = \{x^{(1)}, x^{(2)}, \cdots, x^{(n)}\}$ for each $x \in \mathcal{X}$, $x^{(1)}$ is equal to x itself. Let $\varphi(a, b)$ denote the semantic similarity between two sentences a and b, where a larger value indicates a closer semantic similarity. For any three sentences $(x^{(a)}, x^{(b)}, x^{(c)}) \in l$ with $a < b < c$, the condition should hold: $\varphi(x^{(a)}, x^{(b)}) > \varphi(x^{(a)}, x^{(c)})$. In other words, these sentences are arranged in order within the semantic space.

3 Methodology

3.1 Ranking Sentences Generation

A straightforward approach to generating ranking sentences is prompting LLM, either by directly producing ranking sentences at once or by generating them step by step. Taking the second case as an example, let the prompt be denoted as an instruction I. The LLM C_θ generates the i-th sentence $x^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_k^{(i)}]$ in the following form:

$$p_\theta(x^{(i)}|x^{(i-1)}, I) = \prod_{t=1}^{n} p_\theta(x_t^{(i)}|x_{<t}^{(i)}, x^{(i-1)}, I), \quad (2)$$

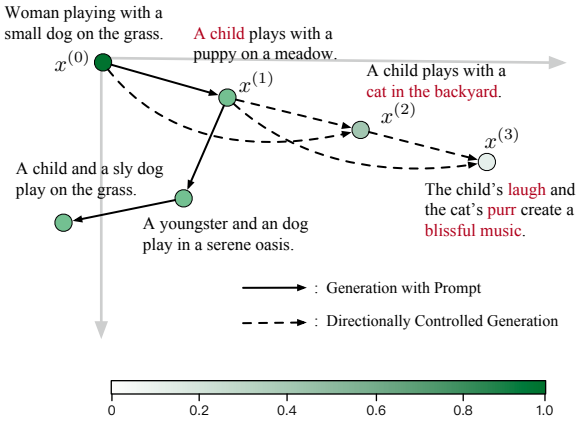


Figure 2: A 2D semantic space illustrating the generation of ranking sentences using prompts (solid line) and our directionally controlled method (dashed line). The color of each point represents its semantic similarity to the initial point $x^{(0)}$.

where $x_{<t}^{(i)}$ represents the tokens generated before the t-th step. Eq.(2) use the previously generated sentence $x^{(i-1)}$ and the instruction I to prompt the LLM to generate the next sentence $x^{(i)}$, thereby progressively constructing a sequence of ranking sentences. However, as mentioned before, this method of generation leads to ambiguous semantic relationships among the ranking sentences, as illustrated in Figure 1 (c).

Our core idea is to integrate directional control into the ranking sentence generation process. As illustrated in Figure 2, our generation process sequentially combines the directional tendencies of two sentences, ensuring that the subsequent generation maintains a consistent trajectory. For example, the generation direction of $x^{(3)}$ is controlled by the latent generation space of $x^{(1)}$ and $x^{(2)}$, ensuring maximal consistency in their generated directions within the semantic space. Specifically, we modified the sampling method of $x_t^{(i)}$ in the LLM as follows:

$$p_\theta(x_t^{(i)}|x_{<t}^{(i)}, c) \propto \frac{p_\theta(x_t^{(i)}|x_{<t}^{(i)}, x^{(i-1)}, I)^{1+\lambda}}{p_\theta(x_t^{(i)}|x_{<t}^{(i)}, x^{(i-2)}, I)^{\lambda}} \quad (3)$$

where $x_{<t}^{(i)}$ represents the tokens generated before the t-th step and c represents the generation condition based on $x^{(i-1)}$, $x^{(i-2)}$, and the instruction I. λ is a hyperparameter that assigns weights to the two generation probabilities. In other words, the generation of a new sentence depends on the directional tendencies of the generation probabilities of the preceding two sentences, ensuring that the resulting latent space remains aligned in a consistent

direction. Then, we can thus sample the next t-th token $x_t^{(i)}$ in the logits space:

$$\log p_\theta(x_t^{(i)}|x_{<t}^{(i)}, c) = (1+\lambda)\log p_\theta(x_t^{(i)}|x_{<t}^{(i)}, x^{(i-1)}, I) - \lambda \log p_\theta(x_t^{(i)}|x_{<t}^{(i)}, x^{(i-2)}, I). \quad (4)$$

According to Eq.(4), we concatenate the instruction I and the previously generated segment $x_{<t}^{(i)}$ with $x^{(i-1)}$ and $x^{(i-2)}$ separately. We then perform two decoding procedures to obtain their respective log probabilities. After computing a weighted sum of these log probabilities, we apply greedy sampling to generate $x_t^{(i)}$. When generating the first sentence $x^{(1)}$, since only $x^{(0)}$ is available, we set λ to 0.

Our method generalizes to Eq.(2) when we set $\lambda = 0$. However, when we use two sentences as conditions, the generative process undergoes a fundamental transformation. This can be likened to basic geometric theorems: "Infinitely many lines pass through a single point" and "The uniqueness of a line through two points." The presence of two preceding sentences ensures the directional consistency of our generation. Besides, our controlled generation process is formally similar to classifier-free guidance (Ho and Salimans, 2021), which employs a linear combination to integrate conditional and unconditional score estimations. However, our method differs fundamentally. We rely solely on conditional control, meaning that all terms depend on preceding sentences rather than an unconditional distribution. By doing so, we effectively guide the sentence generation process, ensuring that the generated text maintains a stable and coherent flow within the semantic space.

3.2 Model Post-training

After obtaining the ranking sentences l for each $x \in \mathcal{X}$, we aim to post-training the existing sentence embedding model to enhance its ability to distinguish fine-grained semantic differences. The ranking sentence l provides order information among sentences. However, the semantic gaps between these sentences are not evenly spaced. Thus, we propose a post-training method that considers both ranking and semantic information among the ranking sentences.

Let $\varphi_{j,k} = \varphi(x^{(j)}, x^{(k)})$ denote the semantic similarity between $x^{(j)}$ and $x^{(k)}$. For any $x^{(j)} \in l$, the following semantic ranking relationship should be satisfied according to the ordering within the ranking sentences:

$$\varphi_{j,1} < \cdots < \varphi_{j,j} > \varphi_{j,j+1} > \cdots > \varphi_{j,n}. \quad (5)$$

Let $r = \{r(i)\}_{i=0}^n$ denote a permutation of the object indices arranged in descending order of semantic similarity, where $r(i)$ represents the rank of the i-th index in the list $[\varphi_{j,1}, \varphi_{j,2}, \ldots, \varphi_{j,n}]$ based on its magnitude. Then, we process each x in l using an encoder model, specifically adopting the DiffCSE (Chuang et al., 2022) base series in our experiments. This model can be trained through a standard unsupervised contrastive learning approach. We obtain their corresponding embeddings $\{\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \ldots, \boldsymbol{h}^{(n)}\}$. Assuming $\phi_{j,k} = \phi(\boldsymbol{h}^{(j)}, \boldsymbol{h}^{(k)})$ represents the cosine similarity between $\boldsymbol{h}^{(j)}$ and $\boldsymbol{h}^{(k)}$. For $\boldsymbol{h}^{(j)}$, we can then derive its similarity relationships with other sentences in l, represented as $\phi^j = [\phi_{j,1}, \phi_{j,2}, \ldots, \phi_{j,n}]$.

Next, we integrate the ranking information r with the semantic information ϕ^j. Our fundamental idea is to adjust $\phi_{j,k}$ based on the ranking position $r(k)$. Let $\phi^j[i]$ represent the value at index i in ϕ^j, and let $\hat{r} = \{\hat{r}(i)\}_{i=0}^n$ denote the permutation of object indices based on the similarity relationships in ϕ^j. We modify each $\phi_{j,k}$ using the following approach:

$$\hat{\phi}_{j,k} = \begin{cases} \phi_{j,k} + m(j,k) & \text{if} \quad r(k) < \hat{r}(k), \\ \phi_{j,k} - m(j,k) & \text{if} \quad r(k) > \hat{r}(k). \end{cases} \quad (6)$$

$$m(j,k) = \log\left(\omega \cdot |\phi_{j,k} - \phi^j[r(k)]| + 1\right), \quad (7)$$

where ω is a hyperparameter to control the importance of ranking information. When the ranking order $\hat{r}$ reflected by semantic information differs from the ranking order r in the ranking information, the value of $\phi_{j,k}$ is adjusted based on the ranking discrepancy to bring $\hat{r}$ closer to r. Through Eq.(6), we obtain a score $\hat{\phi}^j = [\hat{\phi}_{j,1}, \hat{\phi}_{j,2}, \ldots, \hat{\phi}_{j,n}]$ that seamlessly integrates both ranking information and semantic information. Appendix D presents the detailed algorithm.

Finally, we post-training the sentence embedding model using the ListMLE (Xia et al., 2008) loss. Suppose the representation of $x^{(j)}$ obtained through the sentence embedding model is $\boldsymbol{e}^{(j)}$. Similarly, we can get a similarity relationships list $s^j = [s_{j,1}, s_{j,2}, \ldots, s_{j,n}]$. The objective of ListMLE for ranking sentence l is defined as:

$$\mathcal{L}_{\text{ListMLE}}(l) = -\sum_{j=1}^{n} \log P\left(\hat{\phi}^j \mid s^j\right). \quad (8)$$

This target ensures that the ranking results produced by the sentence embedding model learn to align with the ranking results obtained from the fusion of ranking information and semantic information in the ranking sentences.

4 Experiments

4.1 Dataset

Similar to SynCSE (Zhang et al., 2023) and MultiCSR (Wang et al., 2024), we utilize the premises of the NLI dataset (Bowman et al., 2015; Williams et al., 2018) as the initial unlabeled dataset, denoted as X_1. Unlike SynCSE and MultiCSR, which employ the full dataset, we sample only a subset for a generation. Specifically, to enhance data diversity, we first apply k-means clustering to X_1. We set the number of cluster centers to 1,000 and then performed random sampling, selecting 20 samples per cluster, resulting in the dataset X_2. Next, we generate ranking sentences for each sentence in X_2 using our method, where the generation step is set to 32, and the hyperparameter γ is set to 1.5. In contrast to SynCSE, which relies on ChatGPT with approximately 175B parameters for generation, we utilize the LLaMA3-8B-Instruct. This process ultimately produces the dataset X_3, consisting of 16,063 sentence ranking lists and a total of 530,079 sentences. Appendix A presents the detailed generation process of our method.

4.2 Experiment Setup

Baselines. We chose the following strong baselines, including SimCSE (Gao et al., 2021), DiffCSE (Chuang et al., 2022), PromptBERT (Jiang et al., 2022), PCL (Wu et al., 2022a), DebCSE (Miao et al., 2023), InfoCSE (Wu et al., 2022b), RankCSE (Liu et al., 2023), SynCSE (Zhang et al., 2023), and MultiCSR (Wang et al., 2024). Our model is built upon existing sentence embedding models as a post-training approach. In the following experiments, we primarily selected two SOTA models, SynCSE and MultiSCR, as our base models to evaluate whether integrating our ranked sentence data and post-training method can enhance performance[3]. We designate the post-trained SynCSE and MultiSCR as SynCSE-r and MultiSCR-r, respectively. The details of our training process are provided in Appendix C.

Evaluation Settings. We conduct evaluation tests across three tasks: Semantic Textual Similarity (STS), Reranking Task, and Transfer Task (TR). Specifically, for the STS task, we assess performance on seven STS benchmarks: STS 2012–2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al., 2017), and SICK-Relatedness (Marelli et al., 2014). These datasets consist of sentence pairs annotated with similarity scores ranging from 0 to 5. For the retrieval task, we conduct experiments on four datasets: AskUbuntuDupQuestions (Barzilay et al., 2016), MindSmallReranking (Wu et al., 2020), SciDocsRR (Wu et al., 2020), and StackOverflowDupQuestions (Liu et al., 2018). We followed the validation approach of SynCSE (Zhang et al., 2023), adopting the methodology of MTEB (Muennighoff et al., 2023) and employing Mean Average Precision (MAP) as the primary evaluation metric. For the TR task, we use SentEval (Conneau and Kiela, 2018) to evaluate the results, as detailed in Appendix F.

4.3 Main Results

STS Tasks. As shown in Table 1, the post-trained model obtained through our method significantly outperforms previous baselines. Compared to the standard unsupervised SimCSE, our SOTA results improve Spearman's correlation by an average of 7.11% on base models and 5.09% on large models. In comparison with ranking-aware models such as RankCSE, our method achieves improvements of 2.19% and 2.65%, respectively. Furthermore, compared to the underlying sentence embedding models we employ, such as MultiCSR and SynCSE, our approach enhances performance by 0.61% and 0.45% on base models and large models, respectively. These results demonstrate that our method has successfully achieved new SOTA models.

Reranking Tasks. Table 2 presents the performance on four reranking datasets. We followed the experimental setup of SynCSE (Zhang et al., 2023) without utilizing the training sets of reranking tasks. During model training, only the synthesized data was used. We compared the changes in MAP for SynCSE and MultiCSR after post-training with our ranking sentences. Overall, our synthesized data and method led to an average improvement of 1.35% and 1.11% for SynCSE and MultiCSR, respectively. Note that both SynCSE and MultiCSR

[3]By the time this work was completed, GCSE (Lai et al., 2024) was one of the most recent approaches utilizing synthetic data for sentence embedding training. However, as it has not yet publicly released its code and dataset, we did not consider it as a base model.

Model	Method	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	Avg.
BERT-base	SimCSE†	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
	DiffCSE†	72.28	84.43	76.47	83.90	80.54	80.59	71.23	78.49
	PromptBERT♣	71.56	84.58	76.98	84.47	80.60	81.60	69.87	78.54
	PCL♠	72.84	83.81	76.52	83.06	79.32	80.01	73.38	78.42
	DebCSE†	76.15	84.67	**78.91**	**85.41**	80.55	82.99	73.60	80.33
	InfoCSE††	70.53	84.59	76.40	85.10	81.95	82.00	71.37	78.85
	RankCSE♠	75.66	**86.27**	77.81	84.74	81.10	81.80	75.13	80.36
	SynCSE*	74.53	82.14	78.22	83.46	80.66	81.42	80.51	80.13
	MultiCSR*	75.88	82.39	78.80	84.42	80.54	82.23	80.03	80.61
	SynCSE-r	75.82	83.24	78.61	84.75	81.68	83.45	80.67	81.17
	MultiCSR-r	**76.37**	82.50	78.37	85.38	**82.15**	**84.01**	80.55	**81.33**
BERT-large	SimCSE†	70.88	84.16	76.43	84.50	79.76	79.26	73.88	78.41
	PCL♠	74.87	86.11	78.29	85.65	80.52	81.62	73.94	80.14
	DebCSE†	**76.82**	86.36	79.81	85.80	80.83	83.45	74.67	81.11
	InfoCSE††	71.89	86.17	77.72	**86.20**	81.29	83.16	74.84	80.18
	RankCSE♠	75.48	**86.50**	78.60	85.45	81.09	81.58	75.53	80.60
	SynCSE*	75.23	84.28	79.41	84.89	82.09	83.48	81.79	81.60
	MultiCSR*	75.56	85.19	**80.14**	85.91	82.40	84.19	81.65	82.15
	SynCSE-r	76.32	85.17	79.29	85.78	**82.76**	84.76	**82.51**	82.37
	MultiCSR-r	75.69	85.63	79.92	86.08	82.69	**84.88**	82.37	**82.47**
RoBERTa-base	SimCSE†	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
	DiffCSE†	70.05	83.43	75.49	82.81	82.12	82.38	71.19	78.21
	PromptRoBERTa♣	73.94	84.74	77.28	84.99	81.74	81.88	69.50	79.15
	PCL♠	71.13	82.38	75.40	83.07	81.98	81.63	69.72	77.90
	DebCSE†	74.29	85.54	79.46	85.68	81.20	83.96	74.04	80.60
	RankCSE♠	73.20	**85.95**	77.17	84.82	82.58	83.08	71.88	79.81
	SynCSE*	76.15	84.41	79.23	84.85	82.87	83.95	**81.41**	81.84
	MultiCSR*	**77.03**	84.72	79.71	85.80	82.68	84.24	80.64	82.12
	SynCSE-r	76.01	83.18	79.13	85.51	**83.03**	84.66	80.93	81.78
	MultiCSR-r	76.79	85.03	**80.00**	**86.05**	82.65	**84.79**	81.14	**82.35**
RoBERTa-large	SimCSE†	72.86	83.99	75.62	84.77	81.80	81.98	71.26	78.90
	PCL♠	74.08	84.36	76.42	85.49	81.76	82.79	71.51	79.49
	DebCSE†	**77.68**	**87.17**	**80.53**	**85.90**	83.57	85.36	73.89	82.01
	RankCSE♠	73.20	85.83	78.00	85.63	82.67	84.19	73.64	80.45
	SynCSE*	75.92	85.01	80.43	85.83	84.40	85.05	81.99	82.66
	MultiCSR*	74.42	84.46	79.17	84.76	83.67	84.23	81.50	81.74
	SynCSE-r	75.64	84.53	80.36	85.88	**84.47**	**85.82**	**83.24**	**82.85**
	MultiCSR-r	74.28	84.81	79.20	85.26	83.93	84.40	81.62	81.93

Table 1: Comparison of Spearman's correlation results on STS tasks, where the value highlighted in bold is the best value, and the value underlined is the second-best value. "†": results from (Miao et al., 2023), "♣": results from (Wang et al., 2024), "♠": results from (Liu et al., 2023), "††": results from (Wu et al., 2022b). "*": we reproduce the results with the officially released codes and corpus from (Zhang et al., 2023; Wang et al., 2024).

employ contrastive learning, which is originally a training paradigm for retrieval models (Izacard et al., 2021; Li et al., 2021). Our synthesized ranking sentences further enhance the reranking performance of SynCSE and MultiCSR, demonstrating their effectiveness in this context.

4.4 Ablation Study

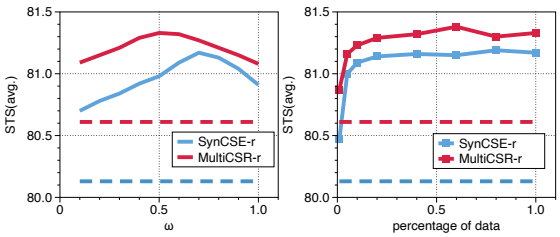
Since our method consists of both a data generation phase and a model post-training phase, we conduct two groups of ablation experiments. For data synthesis, we design the following three ablation settings: (a) Prompting the LLM to generate complete ranking sentences at once. (b) Prompting the LLM to generate ranking sentences step by step. (c) Using our method, we first generate ranking sentences. Then, we randomly shuffle them and reconstruct new ranking sentences. In this case, only semantic information is utilized since the ranking information is lost. For the post-training phase, we designed the following two ablation settings: (d) Only ranking information r is used. (e) Only

Dataset	SynCSE	SynCSE-r	MultiCSR	MultiCSR-r
BERT-base				
AskU.	51.79	52.34 (+1.05%)	51.04	51.51 (+0.92%)
Mind.	28.96	29.01 (+0.17%)	29.04	29.37 (+1.14%)
SciD.	69.49	70.73 (+1.79%)	69.32	70.61 (+1.87%)
StackO.	39.88	40.66 (+1.94%)	39.50	40.68 (+2.97%)
Avg.	47.53	48.19 (+1.37%)	47.22	48.04 (+1.73%)
BERT-large				
AskU.	51.36	50.73 (-1.22%)	51.62	50.49 (-2.19%)
Mind.	30.56	30.62 (+0.18%)	29.47	30.68 (+4.11%)
SciD.	71.33	72.22 (+1.25%)	71.31	71.71 (+0.56%)
StackO.	40.06	39.82 (-0.60%)	39.76	40.09 (+0.84%)
Avg.	48.33	48.35 (+0.04%)	48.04	48.24 (+0.43%)
RoBERTa-base				
AskU.	52.59	53.26 (+1.28%)	51.91	52.18 (+0.52%)
Mind.	27.58	28.70 (+4.06%)	27.97	28.37 (+1.45%)
SciD.	63.39	65.94 (+4.02%)	62.83	64.18 (+2.15%)
StackO.	38.81	38.84 (+0.07%)	39.35	39.95 (+1.53%)
Avg.	45.59	46.69 (+2.39%)	45.51	46.17 (+1.45%)
RoBERTa-large				
AskU.	55.22	54.92 (-0.54%)	54.01	54.66 (+1.21%)
Mind.	29.88	30.17 (+0.99%)	29.16	29.32 (+0.56%)
SciD.	69.33	70.99 (+2.39%)	69.73	70.08 (+0.49%)
StackO.	39.00	40.42 (+3.65%)	40.50	40.92 (+1.04%)
Avg.	48.36	49.13 (+1.59%)	48.35	48.75 (+0.82%)

Table 2: Comparison of Mean Average Precision (MAP) results on reranking tasks, illustrating the changes in SynCSE and MultiCSR before and after training with ranking sentence data.

Phase	Method	Spearman's	Δ
-	MultiCSR-r	81.33	0.0
Data Generation	(a)	80.85	-0.48
Data Generation	(b)	80.97	-0.36
Data Generation	(c)	80.78	-0.55
Post-training	(d)	80.65	-0.68
Post-training	(e)	81.18	-0.15
-	MultiCSR	80.61	-0.72

Table 3: Ablation studies on different data generation methods and components of post-training. We use MultiCSR-r based on BERT-base as the model, and the results are reported on average Spearman's correlation of STS Task.



(a) impact of ω (b) impact of data

Figure 3: The impact of hyperparameter ω on average STS test score for SynCSE-r and MultiCSR-r based on BERT-base as the model. The base model scores are shown in dashed lines

semantic similarity information $\phi^{(j)}$ is used.

Table 3 presents the average Spearman's correlation on the STS dataset. For data synthesis, comparing (a) shows that generating ranking sentences at once via prompts is limited, as LLMs struggle with semantic understanding in longer texts. Comparison with (b) suggests that even a step-by-step approach lacks effective directional control, leading to suboptimal results. The results of (c) highlight the importance of ranking information, confirming that our method's improvements are not solely due to semantic information. For post-training, comparing (d) indicates that ranking information alone is insufficient due to fine-grained semantic differences, emphasizing the need for semantic information. The results of (e) remain inferior to the full method, showing that incorporating ranking information helps refine semantic representations and improve model performance.

4.5 Analysis

In this section, we conduct a more in-depth analysis of the synthesized dataset and our post-training method. We employ SynCSR-r and MultiCSR-r with the BERT-base model. We report the results in terms of Spearman's correlation on the STS task.

The impact of the hyperparameter ω. Figure 3 (a) illustrates the impact of different hyperparameter ω on model performance in the STS task. The ω plays a crucial role in our post-training method, as defined in Eq. (7), where it controls the importance of ranking information. The results indicate that while the optimal ω varies across different models, it remains robust within a relatively broad range. Based on these findings, we set $\omega = 0.7$ for SynCSE-r and $\omega = 0.5$ for MultiCSR-r.

The impact of the amount of synthetic data amount. Figure 3 (b) illustrates the performance on the STS task when using different proportions of our synthesized ranking sentences dataset. We find that although approximately 16,000 sentence ranking lists were generated, utilizing only 10% of the data is sufficient to achieve a substantial improvement, while merely 5% is enough to surpass the original model significantly. This underscores the pivotal role of ranking sentences in enhancing sentence embedding models. On the other hand, we also observe that continuously increasing the number of ranking sentences does not lead to a

Method	STS-12(Δ)	STS-13(Δ)	STS-14(Δ)	STS-15(Δ)	STS-16(Δ)	STS-B(Δ)	SICK-R(Δ)	Avg.(Δ)
SimCSE	+4.04	+1.47	+1.61	+2.31	+0.49	+1.87	+0.66	+1.78
InfoCSE	+0.24	-0.42	+0.27	+0.36	+0.64	+0.14	+1.38	+0.37
PCL	+0.69	+0.69	+0.25	+2.49	+2.79	+2.27	+1.4	+1.51
RankCSE	-0.11	+0.6	+0.69	+2.00	+1.98	+0.15	-0.29	+0.71

Table 4: We compare the changes in Spearman's correlation on STS tasks across several sentence embedding models after post-training with ranking sentences. Their checkpoints based on BERT-base as the model are obtained from their official sources.

consistent improvement in STS performance. This may be attributed to an excessive number of ranking sentences, potentially reducing the model's generalization ability.

4.6 Post-training Experiment

We further analyze the impact of post-training our data on sentence embedding models other than SynCSE and MultiCSR. Table 4 presents the changes in Spearman's correlation for SimCSE (Gao et al., 2021), InfoCSE (Wu et al., 2022b), PCL (Wu et al., 2022a), and RankCSE (Liu et al., 2023) on STS tasks after applying our ranking sentences dataset and post-training approach. From these results, we can observe that employing ranking sentences along with our post-training method has led to improvements across most datasets in the STS task. This demonstrates the versatility and effectiveness of both ranking sentences and our post-training approach. The detailed results are presented in Figure 6 of the Appendix E.

5 Related Work

Unsupervised sentence embedding has been widely studied. Early methods extended the word2vec framework (Mikolov et al., 2013) to sentence-level embeddings, such as Skip-Thought (Kiros et al., 2015), FastSent (Hill et al., 2016), and Quick-Thought (Logeswaran and Lee, 2018). With the rise of PLMs, models like BERT (Kenton and Toutanova, 2019) and RoBERTa (Liu, 2019) have been explored for sentence representation. However, issues like anisotropy (Ethayarajh, 2019) have led to post-processing techniques such as BERT-flow (Li et al., 2020) and BERT-whitening (Su et al., 2021) to improve embedding quality.

With the rise of contrastive learning, the focus shifted toward deriving sentence embeddings by maximizing agreement between different views of the same sentence. Techniques like SimCSE (Gao et al., 2021) utilized dropout-based augmentation to create positive pairs, inspiring follow-up methods (Wang et al., 2022b; Chuang et al., 2022; Liu et al., 2023; Jiang et al., 2022; Wu et al., 2022a; Miao et al., 2023). These methods proved highly effective. However, unsupervised approaches often lag behind their supervised counterparts, which leverage labeled datasets such as natural language inference (NLI) corpora (Bowman et al., 2015; Williams et al., 2018). Yet, such datasets are not easily accessible due to the high annotation cost.

To address these limitations, researchers began exploring sentence generation for unlabeled data (Chen et al., 2022; Ye et al., 2022) using models like T5 (Chung et al., 2024). With the advent of large language models (LLMs), both data annotation and generation have seen significant improvements (Gilardi et al., 2023; Alizadeh et al., 2023, 2025). SynCSE (Zhang et al., 2023) leverages LLMs to generate semantically similar sentence pairs, enhancing the effectiveness of contrastive learning. MultiCSR (Wang et al., 2024) and GCSE (Lai et al., 2024) further refine the utilization of LLMs for data generation. This line of research builds upon the training paradigm of supervised SimCSE (Gao et al., 2021), where a triplet is generated for contrastive learning. In contrast to these works, our approach shifts the generation objective towards ranking sentence generation, introducing a novel refinement strategy for contrastive learning models.

6 Conclusion

In this paper, we investigate a method for synthesizing ranking sentences by leveraging LLMs to generate sentences progressively increasing semantic divergence, guided by a controlled direction in the latent space. Furthermore, we explore a post-training approach that integrates ranking information and semantic information. Experimental results demonstrate that our method achieves new SOTA performance with minimal cost in ranking sentence synthesis.

7 Limitations

Although our work has achieved a new SOTA performance for existing sentence embedding models, several promising directions still need to be explored, which we leave for future research. In the realm of data synthesis, this paper primarily concentrates on the process of data generation. However, the selection and refinement of synthesized data are equally crucial. For instance, the analysis in (An et al., 2024) highlights the issue that synthesized sentences tend to be longer than the original ones. The selection mechanism for synthesized ranking sentence datasets has yet to be explored. Besides, during the post-training process, our primary approach is to integrate ranking information with semantic information. This process involves a hyperparameter ω, whose magnitude influences the model's performance. Exploring an adaptive method to eliminate the dependence on ω is also a worthwhile consideration.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

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

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. in* sem 2012: The first joint conference on lexical and computational semantics–volume 1: Proceedings of the main conference and the shared task, and volume 2: Proceedings of the sixth international workshop on semantic evaluation (semeval 2012). *Association for Computational Linguistics. URL http://www. aclweb. org/anthology/S12-1051*.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43.

Meysam Alizadeh, Maël Kubli, Zeynab Samei, and Shirin Dehghani. 2023. Open-source large language models outperform crowd workers and approach chatgpt in text-annotation tasks. *arXiv preprint arXiv:2307.02179*, 101.

Meysam Alizadeh, Maël Kubli, Zeynab Samei, Shirin Dehghani, Mohammadmasiha Zahedivafa, Juan D Bermeo, Maria Korobeynikova, and Fabrizio Gilardi. 2025. Open-source llms for text annotation: a practical guide for model setting and fine-tuning. *Journal of Computational Social Science*, 8(1):1–25.

Na Min An, Sania Waheed, and James Thorne. 2024. Capturing the relationship between sentence triplets for llm and human-generated texts to enhance sentence embeddings. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 624–638.

Tao Lei Hrishikesh Joshi Regina Barzilay, Tommi Jaakkola, Katerina Tymoshenko, and Alessandro Moschitti Lluıs Marquez. 2016. Semi-supervised question retrieval with gated convolutions. In *Proceedings of NAACL-HLT*, pages 1279–1289.

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

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

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Yiming Chen, Yan Zhang, Bin Wang, Zuozhu Liu, and Haizhou Li. 2022. Generate, discriminate and contrast: A semi-supervised sentence representation learning framework. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8150–8161.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. Diffcse: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

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

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*.

Kawin Ethayarajh. 2019. 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)*, pages 55–65.

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

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Liyang He, Zhenya Huang, Enhong Chen, Qi Liu, Shiwei Tong, Hao Wang, Defu Lian, and Shijin Wang. 2023. An efficient and robust semantic hashing framework for similar text search. *ACM Transactions on Information Systems*, 41(4):1–31.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. 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*, pages 1367–1377.

Jonathan Ho and Tim Salimans. 2021. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

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

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. Promptbert: Improving bert sentence embeddings with prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1. Minneapolis, Minnesota.

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

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

Peichao Lai, Zhengfeng Zhang, Wentao Zhang, Fangcheng Fu, and Bin Cui. 2024. Enhancing unsupervised sentence embeddings via knowledge-driven data augmentation and gaussian-decayed contrastive learning. *arXiv preprint arXiv:2409.12887*.

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

Yizhi Li, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2021. More robust dense retrieval with contrastive dual learning. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 287–296.

Peerat Limkonchotiwat, Wuttikorn Ponwitayarat, Lalita Lowphansirikul, Can Udomcharoenchaikit, Ekapol Chuangsuwanich, and Sarana Nutanong. 2022. Congen: Unsupervised control and generalization distillation for sentence representation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6467–6480.

Jiduan Liu, Jiahao Liu, Qifan Wang, Jingang Wang, Wei Wu, Yunsen Xian, Dongyan Zhao, Kai Chen, and Rui Yan. 2023. Rankcse: Unsupervised sentence representations learning via learning to rank. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13785–13802.

Xueqing Liu, Chi Wang, Yue Leng, and ChengXiang Zhai. 2018. Linkso: a dataset for learning to retrieve

similar question answer pairs on software development forums. In *Proceedings of the 4th ACM SIGSOFT International Workshop on NLP for Software Engineering*, pages 2–5.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

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

Pu Miao, Zeyao Du, and Junlin Zhang. 2023. Debcse: Rethinking unsupervised contrastive sentence embedding learning in the debiasing perspective. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1847–1856.

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

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037.

Nhung Thi-Hong Nguyen, Phuong Phan-Dieu Ha, Luan Thanh Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. 2022. Spbertqa: A two-stage question answering system based on sentence transformers for medical texts. In *International Conference on Knowledge Science, Engineering and Management*, pages 371–382. Springer.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. In *OpenAI Blog*.

Bo Pang and Lillian Lee. 2004. 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)*, pages 271–278.

Bo Pang and Lillian Lee. 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)*, pages 115–124.

Yeon Seonwoo, Guoyin Wang, Changmin Seo, Sajal Choudhary, Jiwei Li, Xiang Li, Puyang Xu, Sunghyun Park, and Alice Oh. 2023. Ranking-enhanced unsupervised sentence representation learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15783–15798.

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

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

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

Huiming Wang, Zhaodonghui Li, Liying Cheng, Lidong Bing, et al. 2024. Large language models can contrastively refine their generation for better sentence representation learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7867–7884.

Suhe Wang, Xiaoyuan Liu, Bo Liu, and Diwen Dong. 2022a. Sentence-aware adversarial meta-learning for few-shot text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4844–4852.

Wei Wang, Liangzhu Ge, Jingqiao Zhang, and Cheng Yang. 2022b. Improving contrastive learning of sentence embeddings with case-augmented positives and retrieved negatives. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2159–2165.

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

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

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie,

Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3597–3606.

Qiyu Wu, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, and Daxin Jiang. 2022a. Pcl: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12052–12066.

Xing Wu, Chaochen Gao, Zijia Lin, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022b. Infocse: Information-aggregated contrastive learning of sentence embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3060–3070.

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

Jiacheng Ye, Jiahui Gao, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2022. Progen: Progressive zero-shot dataset generation via in-context feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3671–3683.

Junlei Zhang, Zhenzhong Lan, and Junxian He. 2023. Contrastive learning of sentence embeddings from scratch. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3916–3932.

A Ranking Sentences Generation Details

In this section, we present the detailed methodology of several ranking sentence generation methods involved in this paper and the detailed methodology of our method. We employ the premises from the NLI dataset (Bowman et al., 2015; Williams et al., 2018) as the initial unlabeled dataset for these methods.

1. **Single-step Generation.** Prompting the LLM to generate ranking sentences at once. Our preliminary experiments reveal that generating complete ranking sentences in a single step is too challenging for the LLaMA3-8B-Instruct model. Therefore, we employ the LLaMA3-70B-Instruct model and adopt a few-shot approach to guide the LLM in the generation, ensuring both the coherence and usability of the generated ranking sentences.

2. **Iterative Step-by-step Generation.** Prompting the LLM to generate ranking sentences step by step. Specifically, based on the result of the previous sentence, we prompt the LLaMA3-8B-Instruct to generate the next sentence. This process continues until 32 sentences have been generated.

3. **Our Method.** Generating the ranking sentences using our proposed directionally controlled generation method. We employ the LLaMA3-8B-Instruct model to generate ranking sentences. Specifically, for the first generation, we prompt the LLM to generate sentence $x^{(2)}$. Then, as designed in our method, each input consists of the previous two sentences along with an instruction. By adjusting the sampling strategy of the 8B LLaMA3 model, we progressively generate the final ranking sentences, setting γ to 1.5.

For all generation methods, we employ a rule-based verification process at the final stage to ensure that the generated results are as complete and non-redundant as possible. The first generation method is performed on a Linux server equipped with 8 NVIDIA A800 GPUs, while the second and third methods are conducted on a Linux server with 8 NVIDIA GeForce RTX 4090 GPUs.

Prompt for directly generating complete ranking sentences

Your task is to take an input sentence and generate a sequence of 32 sentences that gradually and progressively diverge in meaning from the original sentence. The final sentence should be completely unrelated to the original sentence.
Example Input: The cat is sleeping on the warm windowsill.
Example Output:
1. The cat is resting on the cozy windowsill.
2. The cat is lying on a soft cushion by the window.
3. A small animal is curled up near the window.
... [Omit the following sentence list here for conciseness.]
Here is the sentence: {sentence}
Each sentence should be similar in length to the original sentence. Do not explain yourself or output anything else.

Prompt for generating ranking sentences step by step

Rewrite the following sentence in a way that slightly changes the meaning while keeping it semantically close. The new sentence should not be an exact paraphrase but should introduce a subtle variation in meaning. Do not lose the core idea of the original sentence.
Here is the sentence: {sentence}
Your response should be similar in length to the original sentence. Do not explain yourself or output anything else.

Prompt for our method

Rewrite the following sentence or phrase using different words and sentence structure while preserving its original meaning. Directly answer with the rewritten sentence. Don't give any explanation or description other than the rewritten sentence.
Write a sentence that is entailment with:{sentence}.
Result:

B Case Study

In this section, we present a case study to illustrate the generated results of our approach, the single-step generation method, and the iterative step-by-step generation method. Table 5 presents the top 10 generated sentences produced by different methods for a given input sentence. We employ the BGE-m3(Chen et al., 2024) model to obtain their embeddings and compute the cosine similarity between the generated results and the original sentence. Similarity scores for results that are not ranked in descending order of semantic similarity are highlighted in red. We observe that, compared to the other two generation methods, our approach produces results that adhere more closely to a descending order in the semantic space. Moreover, as the generation progresses, the likelihood of producing results that deviate from the expected order increases. This underscores the importance of controlling the direction of generation.

C Model Training Details

All our experimental code is implemented using Python and the PyTorch library. The experiments were conducted on a Linux server equipped with eight NVIDIA GeForce RTX 4090 GPUs. We utilized the official implementations for SynCSE (Zhang et al., 2023) and MultiCSR (Wang et al., 2024). Specifically, SynCSE provides both model training code and a synthesized dataset. We used their code and dataset to train SynCSE models, including BERT-base, BERT-large, RoBERTa-base, and RoBERTa-large. MultiCSR offers both training and data synthesis code, which we employ to generate data before proceeding with MultiCSR model training. Additionally, in Section 4.6, we reproduce several models, including SimCSE (Gao et al., 2021), InfoCSE (Wu et al., 2022b), PCL (Wu et al., 2022a), and RankCSE (Liu et al., 2023). We downloaded their checkpoints from the official HuggingFace repositories and applied our post-training method. During post-training, each model receives a ranking sentence as input per training step. We use the Adam (Kingma and Ba, 2014) optimizer and set the learning rate to 3×10^{-6}. For SynCSE, the hyperparameter ω is set to 0.7, while for the other models, ω is set to 0.5.

D Algorithm to Get $\hat{\phi}^j$

We propose an Algorithm 1 for efficiently computing $\hat{\phi}^j$ for $j = 1, 2, \ldots, n$. This algorithm takes a similarity relation matrix $\Phi = [\phi^1, \phi^2, \ldots, \phi^n]$ as input and outputs a refined matrix $\hat{\Phi} = [\hat{\phi}^1, \hat{\phi}^2, \ldots, \hat{\phi}^n]$. For each ϕ^j, it is unnecessary to compute the full set of values. We only need to calculate the results from index i onward and then leverage the symmetry of the similarity matrix to complete the remaining entries. The complexity of the algorithm is $O(n^3)$. Since n represents the length of the ranking sentences and is a finite value, the computational complexity of this algorithm is significantly lower than that of the model's inference process.

Algorithm 1 Refined Semantic Similarity Computation

Input: Initial similarity matrix Φ, hyperparameter ω.
Output: Refined similarity matrix $\hat{\Phi}$.

```
Initialize A ← 0 ∈ ℝ^{n×n}.
for each row index i = 1 to n do
    Extract the subarray Φ[i, i : n].
    Sort the subarray in descending order and obtain sorted indices.
    for each column index j = i to n do
        Find the position index j' of Φ[i, j] in the sorted array.
        Compute A[i, j] = Φ[i, j] − Φ[i, j'].
    end for
end for
Fill A symmetrically: A[j, i] = A[i, j] for j > i.
Compute Φ̂ = Φ + sign(A) · log(ω · |A| + 1).
return Φ̂.
```

E Post-training Experiments

In the preceding experiments, we demonstrated the changes in Spearman's correlation for SimCSE (Gao et al., 2021), InfoCSE (Wu et al., 2022b), PCL (Wu et al., 2022a), and RankCSE (Liu et al., 2023) on STS tasks before and after training with our data and methodology. In this section, we present the comprehensive results, as illustrated in Figure 6.

F Transfer Task

For the TR tasks, we evaluate our method on seven datasets using the default configurations from SentEval: MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), SST-2 (Socher et al., 2013), TREC (Voorhees and Tice, 2000), and MRPC (Dolan and Brockett, 2005). Table 7 shows the results.

Original Sentence: A young man wearing a knit cap with the word PARIS on it and a blue jacket on the street.		
Our Method	**Iterative Step-by-step Generation**	**Single-step Generation**
A stylish young man, sporting a PARIS-emblazoned knit cap and a blue jacket, strolled down the street. [0.8016]	A young man, sporting a knit cap with the word PARIS emblazoned on it, walked down the street in a blue jacket. [0.8660]	A young man wearing a warm hat with a city name on it and a casual jacket outdoors. [0.6829]
As he walked, the stylish young man's PARIS-emblazoned knit cap and blue jacket made a stylish statement down the street. [0.7996]	A young man, sporting a knit cap with the word PARIS emblazoned on it, walked down the street in a blue jacket. [0.8342]	A person wearing a soft hat with a name tag and a casual coat in public. [0.5749]
His outfit, featuring a PARIS-emblazoned knit cap and blue jacket, turned heads as he strolled down the street. [0.7367]	A young man, sporting a knit cap with a bold, cursive "PARIS" emblazoned on it, strolled down the street in a navy blue jacket. [0.8016]	Someone wearing a hat with a label and a casual coat on the sidewalk. [0.5525]
As he walked down the street, people couldn't help but notice his outfit, which boasted a knit cap adorned with the iconic PARIS logo and a striking blue jacket. [0.7313]	A young man, sporting a knit cap adorned with a bold, cursive "PARIS" script, walked down the street in a navy blue coat. [0.8077]	A person wearing a hat with a logo and a casual coat on the pavement. [0.5861]
Noticing his outfit became second nature to passersby as he strolled down the street, the knit cap's PARIS logo and striking blue jacket standing out in bold relief. [0.7246]	A young man, sporting a knit cap bearing a bold, cursive "PARIS" script, strolled down the street in a navy blue overcoat. [0.7801]	A man wearing a hat with a brand name and a casual coat in the city. [0.6443]
As he walked down the street, it soon became automatic for people to notice his outfit, with the PARIS logo on his knit cap and the striking blue jacket making a bold statement. [0.7064]	A young man, sporting a knit cap adorned with a striking, cursive "PARIS" motif, ambled down the sidewalk in a navy blue overcoat. [0.7460]	A man wearing a hat with a logo and a casual coat on the corner. [0.5745]
The moment he strolled along the sidewalk, passersby couldn't help but be drawn to his attire, particularly the eye-catching PARIS emblem on his beanie and the vibrant blue coat that turned heads. [0.6063]	A young man, sporting a knit cap featuring a bold, cursive "PARIS" logo, strolled leisurely along the sidewalk in a navy blue coat. [0.7859]	A man wearing a hat with a brand name and a casual coat in the morning. [0.5850]
As soon as he ambled down the sidewalk, people walking by were irresistibly captivated by his outfit, specifically the striking PARIS logo on his beanie and the radiant blue coat that commanded attention. [0.5788]	A young man, sporting a knit cap adorned with a prominent, cursive "PARIS" logo, ambled along the sidewalk in a navy blue overcoat. [0.7683]	A man wearing a hat with a label and a casual coat in the evening. [0.5787]
The moment he strolled down the sidewalk, passersby couldn't help but be drawn in by his ensemble, particularly the eye-catching PARIS logo on his beanie and the dazzling blue coat that demanded notice. [0.5868]	A young man, sporting a knit cap bearing a large, cursive "PARIS" logo, strolled leisurely along the sidewalk, clad in a navy blue overcoat. [0.7792]	A person wearing a hat with a logo and a formal coat at a party. [0.5256]
As soon as he walked down the street, people couldn't resist being captivated by his outfit, specifically the striking PARIS emblem on his hat and the mesmerizing blue coat that commanded attention. [0.5829]	A young man, wearing a knit cap adorned with a prominent, cursive "PARIS" emblem, ambled along the sidewalk, wrapped in a navy blue overcoat. [0.7471]	Someone wearing a hat with a brand name and a formal dress at a wedding. [0.5064]

Table 5: A case study is conducted to compare our generation method with the Iterative Step-by-Step Generation and Single-Step Generation approaches. The similarity to the original sentence is indicated at the end of each sentence, highlighted in red if not ranked in descending order of semantic similarity.

Overall, we have achieved a new SOTA performance on RoBERTa-base and RoBERTa-large. On BERT-base and BERT-large, both SynCSE-r and MultiCSR-r have demonstrated improvements compared to the results after post-training. Furthermore, our enhancement on the MRPC task is particularly significant. This is because MRPC focuses on distinguishing the similarity between sentence pairs, and by incorporating ranking sentences in post-training, the model becomes more adept at capturing fine-grained semantic differences.

Method	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	Avg.
SimCSE	66.05	81.49	73.61	79.73	78.12	76.52	71.86	75.34
SimCSE-r	70.09	82.96	75.22	82.04	78.61	78.39	72.52	77.12
InfoCSE	70.23	84.05	75.98	84.78	81.72	81.75	71.09	78.51
InfoCSE-r	70.47	83.63	76.25	85.14	82.36	81.89	72.47	78.88
PCL	73.46	81.57	74.91	82.24	79.94	79.41	71.76	77.61
PCL-r	74.15	82.26	75.16	84.73	82.73	81.68	73.16	79.12
RankCSE	74.55	85.13	77.67	84.23	81.18	81.6	74.28	79.81
RankCSE-r	74.44	85.73	78.36	86.23	83.16	81.75	73.99	80.52

Table 6: We compare Spearman's correlation on STS tasks across several sentence embedding models after post-training with ranking sentences. Their checkpoints based on BERT-base as the model are obtained from their official sources.

Model	Method	MR	CR	SUBJ	MPQA	SST2	TREC	MRPC	Avg.
BERT-base	SimCSE♠	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.81
	DiffCSE♠	81.76	86.20	94.76	89.21	86.00	87.60	75.54	85.80
	PromptBERT♣	80.74	85.49	93.65	89.32	84.95	88.20	76.06	85.49
	PCL♠	80.11	85.25	94.22	89.15	85.12	87.40	76.12	85.34
	RankCSE♠	**83.07**	88.27	**95.06**	89.90	**87.70**	**89.40**	76.23	**87.09**
	SynCSE*	81.09	**88.29**	93.53	**90.02**	86.60	84.40	75.30	85.60
	MultiCSR*	81.64	87.79	93.83	89.91	87.15	80.20	75.25	85.11
	SynCSE-r	81.13	87.82	94.07	89.87	87.42	83.80	**77.86**	86.00
	MultiCSR-r	81.47	87.53	93.99	89.68	86.55	83.80	76.00	85.57
BERT-large	SimCSE♠	**85.36**	89.38	95.39	89.63	90.44	91.80	76.41	88.34
	PCL♠	82.47	87.87	95.04	89.59	87.75	93.00	76.00	87.39
	RankCSE♠	84.63	89.51	**95.50**	**90.08**	**90.61**	**93.20**	76.99	**88.65**
	SynCSE*	84.66	89.96	94.49	**90.08**	90.44	86.40	76.75	87.54
	MultiCSR*	84.95	89.86	94.42	89.88	90.33	84.60	76.52	87.22
	SynCSE-r	84.74	90.15	94.99	89.82	**90.61**	87.80	**77.57**	87.95
	MultiCSR-r	84.86	**90.17**	95.00	89.88	89.68	88.00	76.29	87.70
RoBERTa-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
	PromptBERT♣	83.82	88.72	93.19	**90.36**	88.08	90.60	76.75	87.36
	PCL♠	81.83	87.55	92.92	87.21	87.26	85.20	76.46	85.49
	RankCSE♠	83.32	88.61	**94.03**	88.88	89.07	**90.80**	76.46	87.31
	SynCSE*	84.82	**91.31**	93.18	89.70	90.28	84.80	76.70	87.26
	MultiCSR*	**84.99**	91.23	93.07	89.42	**91.10**	84.60	77.28	87.38
	SynCSE-r	83.78	91.15	92.98	89.50	89.95	85.80	77.33	87.21
	MultiCSR-r	84.89	90.70	93.62	89.50	90.06	85.40	**78.38**	**87.51**
RoBERTa-large	SimCSE♠	82.74	87.87	93.66	88.22	88.58	92.00	69.68	86.11
	PCL♠	84.47	89.06	94.60	89.26	89.02	94.20	74.96	87.94
	RankCSE♠	84.61	89.27	94.47	89.99	89.73	**92.60**	74.43	87.87
	SynCSE*	87.42	92.21	94.19	**90.82**	91.60	85.00	76.87	88.30
	MultiCSR*	87.05	91.87	94.07	90.53	91.60	88.80	78.26	88.88
	SynCSE-r	87.24	**92.29**	**94.65**	90.52	**92.37**	91.40	**79.01**	**89.64**
	MultiCSR-r	**87.45**	**92.29**	94.56	90.45	91.98	90.80	78.61	89.45

Table 7: Comparison of different sentence embedding models accuracy on transfer tasks. The value highlighted in bold is the best value, and the value underlined is the second-best value. "♠": results from (Liu et al., 2023). "♣": results from (Wang et al., 2024). "*": we reproduce the results with the officially released corpus from (Zhang et al., 2023) and (Wang et al., 2024).