

SemPA: Improving Sentence Embeddings of Large Language Models through Semantic Preference Alignment

Ziyang Chen^{*1}, Zhenxuan Huang^{*1}, Yile Wang^{✉1}, Weiqin Wang¹, Lu Yin², Hui Huang¹

¹College of Computer Science and Software Engineering, Shenzhen University

²School of Computer Science and Electronic Engineering, University of Surrey

Abstract

Traditional sentence embedding methods employ token-level contrastive learning on non-generative pre-trained models. Recently, there have emerged embedding methods based on generative large language models (LLMs). These methods either rely on fixed prompt templates (Jiang et al., 2024) or involve modifications to the model architecture (BehnamGhader et al., 2024). The former lacks further optimization of the model and results in limited performance, while the latter alters the internal computational mechanisms of the model, thereby compromising its generative capabilities. We propose SEMPA, a novel approach that boosts the sentence representations while preserving the generative ability of LLMs via semantic preference alignment. We leverage sentence-level Direct Preference Optimization (DPO) to efficiently optimize LLMs on a paraphrase generation task, where the model learns to discriminate semantically equivalent sentences while preserving inherent generative capacity. Theoretically, we establish a formal connection between DPO and contrastive learning under the Plackett-Luce model framework. Empirically, experimental results on both semantic textual similarity tasks and various benchmarks for LLMs show that SEMPA achieves better semantic representations without sacrificing the inherent generation capability of LLMs. We release the code at <https://github.com/szu-tera/SemPA>.

1 Introduction

Representing text as a semantically meaningful vector is a fundamental and essential step in natural language processing, supporting various downstream applications such as text classification, clustering, and information retrieval. Seminal works primarily leverage token-level contrastive learning (Gao et al.,

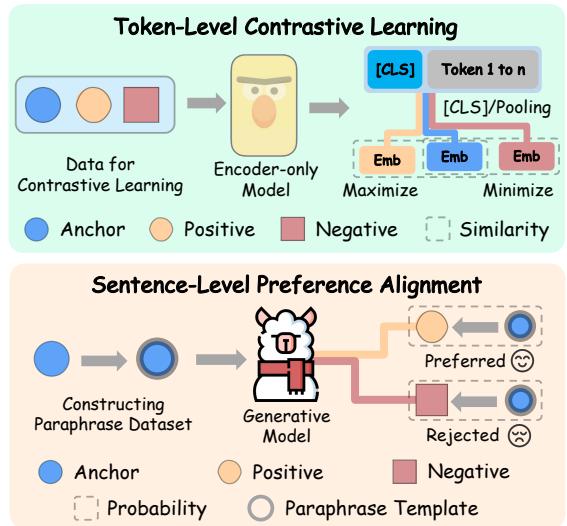


Figure 1: Comparison of sentence embedding methods. Top: contrastive learning for encoder-only models. Bottom: our semantic preference alignment for LLMs.

2021; Chuang et al., 2022) and derive sentence embeddings by extracting the hidden states of special tokens such as [CLS] or the pooled representation of all tokens. The success of such methods relies on powerful pre-trained encoder-only models like BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019), as shown in Figure 1(top).

Generative large language models (LLMs; Brown et al., 2020; OpenAI, 2022, 2023), with massive parameters and trained on extensive data, have demonstrated remarkable capabilities. This has led to the emergence of sentence embedding methods based on either prompting (Jiang et al., 2022; Lei et al., 2024) or modifying these LLMs (BehnamGhader et al., 2024; Fu et al., 2025). However, prompting-based approaches, which rely solely on prompt engineering, have inherent performance limitations. While model-modification methods can improve sentence representations, they often compromise LLMs' generative ability.

To achieve semantic representation with LLMs

^{*}Equal contribution.

[✉]Corresponding author: {wangyile@szu.edu.cn}.

Related Works	Models	Methods	Optimization Levels	Generation
SimCSE (Gao et al., 2021)	PTMs	Contrastive Learning	Token-Level	✗
LLM2Vec (BehnamGhader et al., 2024)	LLMs	Contrastive Learning	Token-Level	✗
NV-Embed (Lee et al., 2025a)	LEMs	Contrastive Learning	Token-Level	✗
Qwen3-embedding (Zhang et al., 2025b)	LEMs	Contrastive Learning	Token-Level	✗
PromptBERT (Jiang et al., 2022)	PTMs	Prompt Learning	Context-Level	✗
PromptEOL (Jiang et al., 2024)	LLMs	Prompt Learning	Context-Level	✓
MetaEOL (Lei et al., 2024)	LLMs	Prompt Learning	Context-Level	✓
Contrastive Prompting (Cheng et al., 2025)	LLMs	Prompt Learning	Context-Level	✗
Token Prepending (Fu et al., 2025)	LLMs	Internal Operations Intervention	Architecture-Level	✗
Hierarchical Token Prepending (Ding et al., 2025)	LLMs	Internal Operations Intervention	Architecture-Level	✗
INSTRUCTOR (Su et al., 2023)	PTMs	Instruction Tuning	Sentence-Level	✗
GRIT (Muennighoff et al., 2025)	LLMs	Instruction Tuning	Sentence-Level	✓
SEMPA (Ours)	LLMs	Preference Alignment	Sentence-Level	✓

Table 1: Comparison of typical sentence embedding related works and our SEMPAs. PTMs, LLMs, and LEMs denote pre-trained models, large language models, and large embedding models, respectively.

while preserving their inherent generative capabilities, we fine-tune LLMs using preference alignment (Ouyang et al., 2022). Preference alignment was originally designed to enhance the truthfulness and safety of LLMs by constructing preference data pairs, enabling sentence-level optimization in certain aspects. Inspired by this, we propose SEMPAs, a semantic preference alignment method that operates at the sentence level by applying Direct Preference Optimization (DPO; Rafailov et al., 2023) to a paraphrase generation task. This allows LLMs to learn fine-grained semantic distinctions by assigning higher relative probabilities to semantically accurate paraphrases and thus improves the underlying semantic representations with lightweight training, without sacrificing their generative proficiency, as shown in Figure 1(bottom).

Theoretically, we provide a derivation that connects contrastive learning and DPO in a unified form, offering a new perspective on why preference alignment serves as an effective paradigm for representation learning. Empirically, results on Semantic Textual Similarity (STS) tasks and various generative benchmarks (Cobbe et al., 2021b, Hendrycks et al., 2021, etc.) using LLaMA models (Touvron et al., 2023; Grattafiori et al., 2024) demonstrate that SEMPAs outperforms pre-trained and prompt-based baselines and stays competitive with specialized embedding models, while maintaining the core generation capability of LLMs across various tasks.

2 Related Work

Sentence Embedding. Sentence embedding aims to encode the semantic content of text into numeri-

cal vectors for semantic similarity assessment and downstream tasks. Current sentence embedding methods can be categorized into four types:

1) *Token-Level*. Token-level contrastive learning is widely used in traditional pre-trained models (Gao et al., 2021; Chuang et al., 2022) or generative language models (Neelakantan et al., 2022; BehnamGhader et al., 2024) for sentence embedding. It optimizes the token-level representations of models by distinguishing positive and negative samples, thereby extracting general semantic representations from special tokens such as [CLS] or the pooling result of all tokens. Contrastive learning is also extensively employed in commercial embedding models such as NV-Embed (Lee et al., 2025a), Qwen3-embedding (Zhang et al., 2025b), and Gemini embedding (Lee et al., 2025b), demonstrating strong performance and scalability.

2) *Sentence-Level*. To further adapt LLMs to various downstream tasks (e.g., classification, retrieval, and clustering), Su et al. (2023) and Muennighoff et al. (2025) employ sentence-level instruction tuning to train LLMs to acquire task-specific semantic representations. These types of methods often rely on large-scale and high-quality instruction datasets such as MEDI (Su et al., 2023) and E5 (Wang et al., 2022).

3) *Context-Level*. There have been some studies on context-based modification that extract overall semantic representations of given texts without fine-tuning the model parameters. Jiang et al. (2022) obtains semantic information based on the [MASK] token of BERT through specific templates. Jiang et al. (2024) and Lei et al. (2024) design prompts to extract semantic information from generative mod-

els such as LLaMA. Cheng et al. (2025) enhances semantic representation by leveraging multiple contexts with different meanings. The effectiveness of such context-based methods largely depends on the model’s inherent capabilities and the quality of specialized prompts.

4) *Architecture-Level*. Modifying the internal structure of LLMs has emerged recently for sentence embeddings (Fu et al., 2025; Ding et al., 2025). These approaches alter the computational operations within the Transformer decoder architecture to extract specific hidden states as semantic representations. However, such invasive modifications change the model’s structure, thus compromising the model’s original generative capabilities.

Preference Alignment of LLMs. Preference alignment has been shown to be an effective way to improve the safety and truthfulness of LLMs. It can be achieved by training LLMs on datasets of human or AI-generated preferences, using methods such as reinforcement learning from human feedback (RLHF; Ouyang et al., 2022) or direct preference optimization (DPO; Rafailov et al., 2023). Preference alignment optimizes the model beyond token-level prediction and improves the overall response quality, and it has also been applied to tasks such as combinatorial optimization (Pan et al., 2025), code generation (Zhang et al., 2025a), and mathematical reasoning (Jiao et al., 2025). To our knowledge, we are the first to explore preference alignment for improving the semantic representations of LLMs.

We summarize and compare the main related works with our SEMPA in Table 1. Note that the primary goal of our work is not to compete with state-of-the-art embedding models in terms of performance, but rather to offer insights for improving semantic representations of LLMs through the perspective of lightweight preference alignment, which has not been explored in existing work.

3 Preliminary

PromptEOL. Given a sentence $\mathcal{S} = s_1, \dots, s_n$, PromptEOL (Jiang et al., 2024) extracts the sentence embedding of \mathcal{S} using LLMs by inserting \mathcal{S} in a template \mathcal{T} defined as:

This sentence: “ \mathcal{S} ” means in one word:“

The template with filled sentence $\mathcal{T}(\mathcal{S})$ is used as the input for a generative model \mathcal{M} , and the corresponding sentence embedding of \mathcal{S} is the hidden state of the last token in the final layer L :

$$h_1^L, h_2^L, \dots, h_{\text{last}}^L = \mathcal{M}(\mathcal{T}(\mathcal{S})), \quad (1)$$

$$\text{Emb}(\mathcal{S}) = h_{\text{last}}^L,$$

where $\text{Emb}(\mathcal{S})$ serves as a training-free baseline embedding extracted from generative LLMs.

Direct Preference Optimization. To tailor the behavior of model \mathcal{M} more closely to human preferences, DPO (Rafailov et al., 2023) directly optimizes a policy by leveraging a closed-form mapping from reward functions to optimal policies, thereby avoiding the need for an explicit reward model. Formally, given dataset \mathcal{D} consisting of triplets (x, y_w, y_l) where y_w is preferred over y_l for input x , the loss function of DPO is defined as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right], \quad (2)$$

where π_θ is the current policy to be optimized and π_{ref} is the reference policy by the original model \mathcal{M} . After identity transformation, the above formula can be expressed as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \frac{e^{\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}}}{e^{\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}} + e^{\beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}}} \right], \quad (3)$$

which shares a unified framework with contrastive learning, as discussed in Section 4.4.

4 Our Method

4.1 Paraphrase Generation Task

We integrate the concept of “paraphrase” to improve the semantic representations of models. Traditional paraphrase-related tasks center on paraphrase identification (Yin and Schütze, 2015; Wang et al., 2021; Peng et al., 2022), where an auto-encoding model is trained with an explicit classification head to determine whether two given sentences convey the same meaning. To leverage paraphrase to enhance the semantic representations of generative LLMs, we propose a paraphrase generation task and construct paraphrase-based preference pairs for model optimization.

Preference alignment pairs for training LLMs are typically sourced from user-annotated (Ouyang et al., 2022) or model-generated (Taori et al., 2023) data, which entail significant annotation costs or computational overhead. In this work, we employ existing natural language inference (NLI; Williams et al., 2018) datasets as the source of semantic

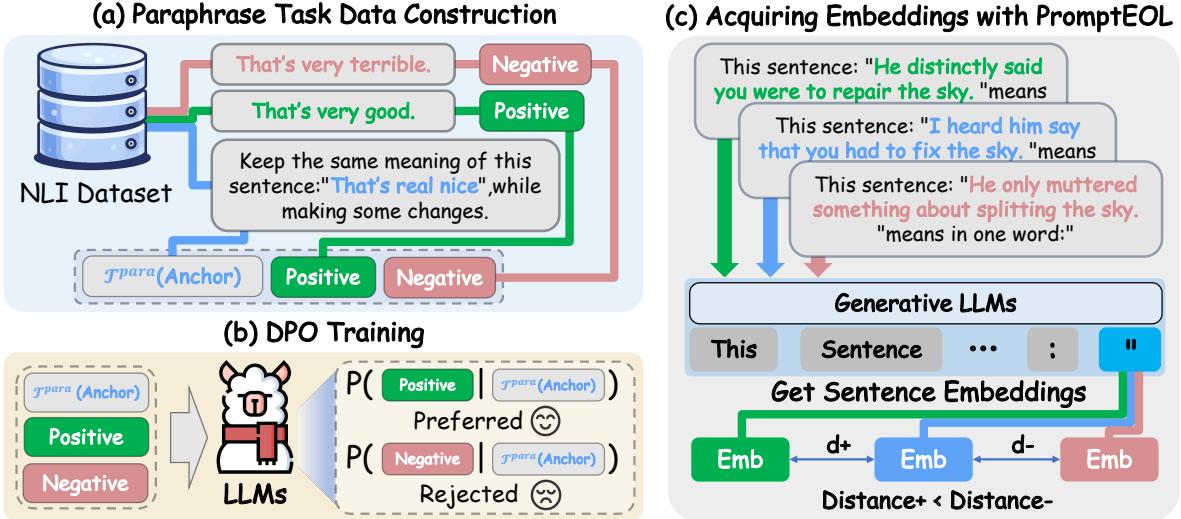


Figure 2: The overall pipeline of our proposed SEMPA method. (a) We first construct the paraphrase generation preference pairs using NLI datasets (§4.1). (b) Then we perform semantic DPO training on LLMs (§4.2). (c) The final sentence embedding is acquired using the PromptEOL template (§4.3).

	Contrastive Learning	DPO
Unified Form	$P(y_i \text{ is the best} \mid x) = \frac{\exp(s_\theta(x, y_i))}{\sum_{j=1}^N \exp(s_\theta(x, y_j))}$	
Candidate Set	y^+ and $(y_1^-, \dots, y_{N-1}^-)$	y_w and y_l
Ranking Event	y^+ is the best	y_w is the best
Target Model	auto-encoding model e_θ	auto-regressive policy π_θ, π_{ref}
Scoring Function s_θ	$\text{sim}(e_\theta(x), e_\theta(y))$	$\beta \log \frac{\pi_\theta(y x)}{\pi_{ref}(y x)}$
Objective Function	$\frac{\exp(s_\theta(x, y^+))}{\exp(s_\theta(x, y^+)) + \sum_{k=1}^{N-1} \exp(s_\theta(x, y_k^-))}$	$\frac{\exp(s_\theta(x, y_w))}{\exp(s_\theta(x, y_w)) + \exp(s_\theta(x, y_l))}$

Table 2: Comparison between contrastive learning and DPO under the unified Plackett–Luce model framework.

preference pairs, thereby enabling a lightweight and low-cost model fine-tuning approach.

For instance, in NLI datasets, given a premise, another sentence needs to be labeled with one of the following labels: entailment, neutral, or contradiction. Following Gao et al. (2021), we treat the premise as the anchor sentence x^* , regard the entailment-labeled sentence as the positive sample y^+ , and the contradiction-labeled sentence as the negative sample y^- , as shown in Figure 2(a). The paraphrase generation task encourages the model to generate y^+ given x^* , which requires the generative model to possess fine-grained semantic understanding of the sentences.

4.2 Semantic Preference Alignment via DPO

Given the triplet (x^*, y^+, y^-) , instead of directly performing sequence-to-sequence generation from input x^* to output y^+ , we align the model to prefer generating y^+ over y^- conditioned on x^* , thereby preventing the model from degenerating into a para-

phrasing model. We first insert the anchor x^* into a paraphrasing instruction template $\mathcal{T}^{\text{para}}$ as follows:

*Keep the same meaning of this sentence:
“ x^* ”, while making some changes.*

Then we apply the constructed data to DPO training, as shown in Figure 2(b). In particular, we fill the template $\mathcal{T}^{\text{para}}$ with the anchor sentence x^* to obtain $\mathcal{T}^{\text{para}}(x^*)$, which is used as the model input x in Eq. 2; the positive sample y^+ serves as the preferred response y_w , and the negative sample y^- serves as the rejected response y_l . We compare different paraphrasing templates $\mathcal{T}^{\text{para}}$ in Section 6.

We refer to the above alignment method as *semantic preference alignment* (i.e., SEMPA), which aims to: 1) retain the original architecture of the generative LLMs without modification; 2) enhance the semantic understanding capability of LLMs; and 3) preserve the foundational capabilities of the original LLMs through lightweight fine-tuning.

4.3 Acquisition of Sentence Embedding

As shown in Figure 2(c), after obtaining the semantic preference aligned model $\mathcal{M}^{\text{SEMPA}}$, we follow PromptEOL by using the template \mathcal{T} to extract our sentence embedding of \mathcal{S} from the last-layer hidden state of the LLM similar to Eq. 1:

$$h_1^L, h_2^L, \dots, h_{\text{last}}^L = \mathcal{M}^{\text{SEMPA}}(\mathcal{T}(\mathcal{S})), \quad (4)$$

$$\text{Emb}^{\text{SEMPA}}(\mathcal{S}) = h_{\text{last}}^L$$

We discuss different templates \mathcal{T} to obtain the final sentence embedding $\text{Emb}^{\text{SEMPA}}(\mathcal{S})$ in Section 6.

Methods	Params	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
BERT-avg (Devlin et al., 2019)	110M	30.87	59.89	47.73	60.29	63.73	47.29	58.22	52.57
Sentence-T5-avg (Ni et al., 2022)	4.8B	34.97	60.19	47.59	66.40	70.62	62.83	63.57	58.02
PromptBERT (Jiang et al., 2022)	110M	60.96	73.83	62.18	71.54	68.68	70.60	67.16	67.85
SBERT-NLI-base (Reimers and Gurevych, 2019)	110M	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.86
SBERT-NLI-large (Reimers and Gurevych, 2019)	340M	72.27	78.46	74.90	80.99	76.25	79.23	73.75	76.55
LLM2Vec-LLaMA2 (BehnamGhader et al., 2024)	7B	65.39	79.26	72.98	82.72	81.02	78.32	71.77	75.92
<i>LLaMA2-7B</i>									
Mean Pooling	7B	35.45	53.62	39.87	54.68	53.00	41.52	48.86	46.71
Echo Embedding (Springer et al., 2025)	7B	50.43	73.77	61.61	73.21	70.80	68.67	65.11	66.23
PromptEOL (Jiang et al., 2024)	7B	58.81	77.01	66.34	73.22	73.56	71.66	69.64	70.03
Contrastive Prompting (Cheng et al., 2025)	7B	63.34	82.15	71.73	79.68	77.23	78.71	74.04	75.27
Token Prepending (Fu et al., 2025)	7B	66.90	83.12	74.31	79.87	80.03	80.67	75.40	77.19
SEMPA (Ours)	7B	68.77	83.73	74.73	82.08	77.48	79.76	77.26	77.69
<i>LLaMA3-8B</i>									
Mean Pooling	8B	32.62	52.65	42.86	58.07	54.28	46.25	50.57	48.19
Echo Embedding (Springer et al., 2025)	8B	50.30	71.75	62.43	72.01	75.62	69.75	65.79	66.81
PromptEOL (Jiang et al., 2024)	8B	60.89	78.57	68.18	76.75	77.16	72.83	68.94	71.90
Contrastive Prompting (Cheng et al., 2025)	8B	60.80	80.67	70.06	78.34	77.46	75.05	70.28	73.24
Token Prepending (Fu et al., 2025)	8B	64.45	81.61	71.52	78.97	80.18	77.43	73.22	75.34
SEMPA (Ours)	8B	68.55	84.62	75.89	82.11	78.36	80.04	77.12	78.10

Table 3: Results on semantic textual similarity tasks, reported by Spearman correlation $\rho \times 100$.

4.4 Unifying Contrastive Learning and DPO

We offer a theoretical connection between contrastive learning and DPO in terms of their objective formulations, which can help us understand that DPO can also enhance semantic representations similar to contrastive learning. Specifically, they can both be formally regarded as specific instantiations of the Plackett-Luce model (Plackett, 1975). For N responses $\{y_1, \dots, y_N\}$, let the observed ranking be $\sigma = (y_{(1)} \succ y_{(2)} \succ \dots \succ y_{(N)})$, where $y_{(1)}$ is the optimal response and $y_{(N)}$ is the worst. The probability of the complete ranking is:

$$P(\sigma | x) = \prod_{k=1}^N \frac{\exp(s_\theta(x, y_{(k)}))}{\sum_{j=k}^N \exp(s_\theta(x, y_{(j)}))}, \quad (5)$$

where s_θ is a score function between the input and the candidate. The above formula can be interpreted as a sequential selection process: the top-ranked candidate is chosen first, followed by the second-ranked candidate from the remaining options, and this procedure continues iteratively.

Similarly, both contrastive learning and DPO focus solely on the selection probability of the top-ranked candidate. As a result, the formula can be simplified to:

$$P(y_1 \text{ is the best} | x) = \frac{\exp(s_\theta(x, y_1))}{\sum_{j=1}^N \exp(s_\theta(x, y_j))}, \quad (6)$$

where the main difference lies in the scoring function s_θ and the final objective, as shown in Table 2.

For the scoring function, contrastive learning uses a similarity function with representations (usually from the special token [CLS]) given by the encoder e_θ , while DPO uses the log ratio of the policy π_θ to a generative reference model π_{ref} . For the objective \mathcal{L} , contrastive learning uses the log-likelihood InfoNCE (Ord et al., 2018) similar to Eq. 3 in DPO.

5 Experiments

5.1 Settings

Datasets. We use the natural language inference dataset for training, which is also used in SimCSE¹. This dataset consists of 275K triplets, and we use a subset of 40~80K samples based on the performance on the development set.

For evaluation, we use seven standard Semantic Textual Similarity (STS) datasets, including STS 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS-B (Cer et al., 2017), and SICK-R (Marelli et al., 2014).

Baselines. Following Fu et al. (2025), we compare our SEMPA with the following baselines: **BERT-avg** (Devlin et al., 2019) averages the last-layer token embeddings from BERT to form a sentence embedding. **Sentence-T5-avg** (Ni et al., 2022) uses the mean pooling of token representations from the T5 encoder. **PromptBERT** (Jiang et al., 2022) represents sentences with the [MASK] to-

¹https://huggingface.co/datasets/princeton-nlp/datasets-for-simcse/blob/main/nli_for_simcse.csv

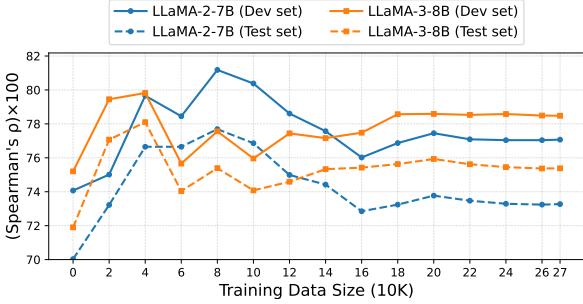


Figure 3: Performance of LLaMA models on the STS-B dev set and STS test sets with different training data sizes. STS test results are averaged over all STS tasks.

ken using pre-defined prompts. **SBERT** (Reimers and Gurevych, 2019) fine-tunes a BERT model with siamese networks and refines its semantic representations. **LLM2Vec** (BehnamGhader et al., 2024) converts decoder-only LLMs into bidirectional encoders by modifying the attention pattern and performs contrastive learning. **Echo Embedding** (Springer et al., 2025) forms sentence representations by repeating the input twice and pooling only the hidden states from the second occurrence. **PromptEOL** (Jiang et al., 2024) designs prompts and extracts the last-token hidden state as semantic representation from LLMs. **Contrastive Prompting** (Cheng et al., 2025) introduces an auxiliary prompt to contrastively intervene in the original prompt’s representations, thereby suppressing redundant information. **Token Prepending** (Fu et al., 2025) prepends the per-layer sentence embedding to the next layer’s input, allowing early tokens to attend to full-sentence context under causal attention. For the Contrastive Prompting and Token Prepending baselines, PromptEOL is also applied for embedding extraction and fair comparison.

Models and Implementation Details. Considering the reported results of the aforementioned baselines, we use LLaMA2-7B (Touvron et al., 2023) and LLaMA3-8B (Grattafiori et al., 2024) as base models. We apply LoRA (Hu et al., 2022) fine-tuning on four NVIDIA RTX 5090 GPUs to train our models. See Appendix A for details.

5.2 Main Results

The results are shown in Table 3. Overall, our method achieves the highest score on 5 out of 7 datasets with LLaMA2-7B, and on 6 out of 7 datasets when leveraging the LLaMA3-8B backbone, indicating that our proposed semantic preference alignment method can indeed enhance the

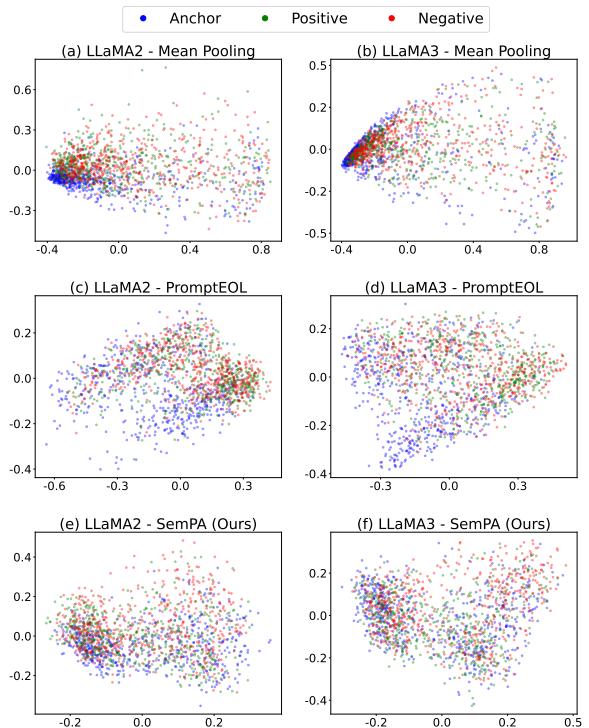


Figure 4: Qualitative visualization of embedding space using LLaMA2 (left) and LLaMA3 (right). Our method results in a more isotropic embedding space.

semantic representation capability of generative large language models.

Our method, based on LoRA fine-tuning (with only 0.3% of the 7B model parameters being trainable), outperforms smaller-scale encoder models such as SBERT. This indicates that, in terms of performance, generative semantic representation can effectively replace traditional representation schemes. On the other hand, compared with recent LLM-based approaches such as Token Prepending, our method achieves performance improvements of 0.5~2.8 points without requiring modifications to the model architecture or computation process.

Across different backbone LLMs, we find that our method performs better with the more advanced LLaMA3-8B than with LLaMA2-7B, showing more substantial improvements over the baselines. This indicates that our approach can further exploit the intrinsic semantic representation capability as the base model’s capacity increases.

6 Analyses

Impact of Data Size. Figure 3 shows the results obtained using different amounts of training data for a simple epoch. We find that both LLaMA2 and LLaMA3 achieve their highest Spearman corre-

Template $\mathcal{T}^{\text{para}}$ for Training		Template \mathcal{T} for Embedding Extraction	
		PromptEOL	Pretended CoT
<i>Keep the same meaning of this sentence: “x^*”, while making some changes.</i>		77.69/78.10/77.90	76.69/78.96/77.83
<i>Generate a paraphrase of this sentence: “x^*” that preserves its meaning, while making some changes.</i>		76.88/78.20/77.54	76.32/79.00/77.66
<i>Rewrite the sentence: “x^*” while preserving its main meaning, but the wording may be simplified or rephrased.</i>		76.75/75.77/76.26	75.21/77.90/76.56
<i>Keep the main meaning of this sentence: “x^*”, and rewrite it in a different way.</i>		73.62/78.84/76.23	71.95/79.78/75.87
<i>Generate a paraphrase of the sentence: “x^*”.</i>		74.62/76.73/75.68	72.27/77.84/75.06
Baseline w/o using $\mathcal{T}^{\text{para}}$ for SEMPA training		70.03/71.90/70.97	76.86/76.09/76.48

Table 4: Performance (LLaMA2/LLaMA3/average) using different paraphrasing templates $\mathcal{T}^{\text{para}}$ and CoT-enhanced embedding extraction template \mathcal{T} of pretended CoT (Zhang et al., 2024). Apart from a decline for LLaMA2 using CoT-based template, our method consistently achieves improvements across different templates and model settings.

Models	Methods	Uniformity↓	Isotropy Score↑
LLaMA2-7B	Mean Pooling	-0.7596	0.4284
	PromptEOL	-1.0066	0.4173
	SEMPA (Ours)	-1.9139	0.4882
LLaMA3-8B	Mean Pooling	-0.8112	0.4130
	PromptEOL	-1.3277	0.4448
	SEMPA (Ours)	-2.3298	0.5312

Table 5: Quantitative results on uniformity and isotropy scores. The results are aligned with qualitative ones.

tion scores when trained with 40K to 80K NLI samples, and the best scores for both exceed 77. When trained with 100K or more samples, performance declines slightly and then plateaus, indicating some degree of over-alignment in paraphrase generation. Nonetheless, performance remains mostly above 73, outperforming the PromptEOL baseline. Overall, these results demonstrate that our method effectively enhances the semantic representations of generative LLMs in a data-efficient manner, without relying on large-scale, high-quality data for contrastive learning or instruction tuning.

Visualization of Embedding Space. Representations of neural models often exhibits an anisotropic distribution, especially for generative LLMs, which limits their semantic expressiveness (Ethayarajh, 2019; Li et al., 2020). Figure 4 shows the PCA visualization of embeddings from randomly selected 0.2% instances from the NLI dataset.

Results show that sentence embeddings derived via mean pooling suffer from a severe anisotropy issue, with the majority of sentence vectors clustered in a specific region of the embedding space. In comparison, PromptEOL and our SEMPA mitigate this issue to a certain extent and generates embeddings of superior quality, where positive and negative samples are more clearly separated.

We further use two metrics that assess the representation quality for comparison: 1) *uniformity* introduced by Wang and Isola (2020), and 2) *isotropy*

score proposed by Mu and Viswanath (2018). Details of these two metrics are shown in Appendix B. Results in Table 5 demonstrate that SEMPA outperforms both the original model and the PromptEOL baseline across the two metrics, indicating that SEMPA can enhance semantic expressiveness by improving the model’s representation space.

Impact of Templates. We test the performance using different paraphrase generation templates $\mathcal{T}^{\text{para}}$ and the embedding extraction templates \mathcal{T} . To guide LLMs in generating paraphrases, we design multiple instructions that include phrases like “Generate a paraphrase ...”, “Keep the meaning ...”, or “Rewrite the sentence ...”, supplemented by constraints to promote syntactic diversity. For embedding extraction, besides the PromptEOL baseline, we also use a chain-of-thought enhanced template Pretended CoT (Zhang et al., 2024) as follows:

*After thinking step by step, this sentence:
“S” means in one word:“*

The averaged results on seven STS datasets using LLaMA2 and LLaMA3 are shown in Table 4. We observe that when using PromptEOL as the sentence embedding extraction method, all five paraphrase generation templates we designed lead to significant improvements in model performance, elevating the baseline scores from 70.03/71.90 to a minimum of 74.62/76.73 and a maximum of 77.69/78.10. This demonstrates that our semantic alignment approach is generally stable and effective across different templates.

When Pretended CoT is used as the sentence embedding extraction template, we observe varying degrees of performance decline when fine-tuning the LLaMA2 model. For instance, the impact is minimal with the first two templates, while the latter three templates show a drop of 1~4 points. On the more advanced LLaMA3 model, however, our method consistently improves performance across

Models	GSM8K	MMLU	HellaSwag	DROP	TruthfulQA	Avg.
	Acc.	Acc.	Acc_norm	F1	MC1	
LLaMA2-7B	14.03	45.91	76.13	40.23	24.85	40.23
w/ Contrastive Learning	3.79 ($\downarrow 10.24$)	46.12 ($\uparrow 0.21$)	73.50 ($\downarrow 2.63$)	15.92 ($\downarrow 24.31$)	23.87 ($\downarrow 0.98$)	32.64 ($\downarrow 7.59$)
w/ Semantic DPO (Ours)	11.68 ($\downarrow 2.35$)	47.43 ($\uparrow 1.52$)	79.99 ($\uparrow 3.86$)	31.02 ($\downarrow 9.21$)	35.62 ($\uparrow 10.77$)	41.15 ($\uparrow 0.92$)
LLaMA3-8B	55.80	64.96	79.22	58.99	27.29	57.25
w/ Contrastive Learning	9.86 ($\downarrow 45.94$)	62.68 ($\downarrow 2.28$)	75.44 ($\downarrow 3.78$)	25.02 ($\downarrow 33.97$)	24.72 ($\downarrow 2.57$)	39.54 ($\downarrow 17.71$)
w/ Semantic DPO (Ours)	52.01 ($\downarrow 3.79$)	64.71 ($\downarrow 0.25$)	82.56 ($\uparrow 3.34$)	60.93 ($\uparrow 1.94$)	36.47 ($\uparrow 9.18$)	59.34 ($\uparrow 2.09$)

Table 6: Comparison of general capabilities of LLMs. For the results of contrastive learning, we obtain the model by extracting sentence embeddings derived from the PromptEOL template and performing contrastive learning.

Input: Fire from the American ships did not actually stop at this point, and after clarifying the position of his ships, he ordered TF 64 to resume firing at 23:51.
PromptEOL: The <0x0A> _He Fire He _The After after _Fire ...
SEMPA: _After res After _res _firing _after after _Fire Fire _Res
Input: After the resumption of National College Entrance Examination, he earned his Master of Engineering degree from the University of the Chinese Academy of Sciences in August 1981.
PromptEOL: _He He _After After _he he <0x0A> <0xE5> <0xE6>
SEMPA: After _After _after after Master August _he _earned he master
Input: He has been a member of the Portfolio Committee on International Relations and Cooperation (National Assembly Committees) since 27th June 2019.
PromptEOL: _He He <0x0A> _he he H Member _ since I
SEMPA: Member _has _he _member _He _Member _a has member He
GAR Score of PromptEOL: 0.185
GAR Score of SEMPA: 0.324

Table 7: Top-10 aligned tokens in vocabulary according to Nie et al. (2025). Tokens in green strictly match the original input. Tokens in yellow match the input after stemming or lemmatization. Tokens in red are unrelated.

all paraphrase generation templates. This discrepancy may stem from differences in the semantic representations elicited by CoT prompting across different base models, which could be a direction worth further exploration in future work.

Top-10 Aligned Tokens in Vocabulary. Nie et al. (2025) finds that the tokens mapped by the embeddings from PromptEOL tend to align with key tokens from the original inputs, which helps explain the effectiveness of sentence embeddings from generative LLMs. Following their analysis, we show examples of the top-10 aligned tokens for three different texts using the LLaMA2-7B backbone in Table 7. These cases show that the embeddings of PromptEOL still align with some unrelated and meaningless tokens such as <0x0A>, which may be due to the lack of semantic refinement. In contrast, after lightweight semantic preference alignment, our method makes the aligned tokens more accurate. The GAR scores (see detailed definition in Appendix C) proposed by Nie et al. (2025) also show that our method indeed increases overall coverage of tokens at the dataset level.

Impact to Generation Capability of LLMs.

To verify that the model fine-tuned with our proposed method can enhance its semantic representation ability while preserving its inherent generative capability, we adopt a variety of benchmarks across multiple dimensions, including mathematical reasoning (**GSM8K**; Cobbe et al., 2021a), multi-task language understanding (**MMLU**; Hendrycks et al., 2021), commonsense reasoning (**HellaSwag**; Zellers et al., 2019), reading comprehension (**DROP**; Dua et al., 2019), and response reliability (**TruthfulQA**; Lin et al., 2022).

The results are shown in Table 6. We find that after token-level optimization based on contrastive learning, the model’s performance declines across almost all tasks (by 7.59 on LLaMA2-7B and 17.71 on LLaMA3-8B). This indicates that contrastive learning methods tend to cause the model to degenerate into a purely representational model, thereby sacrificing the original generative capabilities of the language model.

In contrast, our preference alignment approach remains a lightweight, sentence-level optimization method that does not significantly affect performance on these general tasks. Moreover, we observe that on the TruthfulQA dataset, our semantically aligned optimization improves the accuracy of the base model by 9.18~10.77 points. This reflects the inherent impact of semantic representation on certain downstream tasks.

7 Conclusion

We introduce SEMPA, a sentence-level semantic preference alignment method to obtain better sentence embeddings from LLMs. We establish a theoretical connection between contrastive learning and our preference alignment method for improving sentence representations. Empirically, we evaluate SEMPA and other approaches on a series of semantic textual similarity tasks and LLM-related

benchmarks. The results show that SEMPA enables better sentence embeddings compared with various baselines, without sacrificing the intrinsic generative capabilities of LLMs. Analysis of the embedding space and the aligned tokens of sentences shows that SEMPA can alleviate the embedding space anisotropy and reflect the information in sentences more accurately.

Limitations

This study preliminarily demonstrates through lightweight experiments that semantic preference optimization enhances the semantic representation of generative LLMs. Further improvements and exploration could be pursued in terms of the quantity and quality of the dataset, or automatic prompt optimization methods to enhance robustness. Additionally, our method employs pairwise DPO for preference optimization, which only considers binary comparisons between two responses. Recent advances such as LiPO (Liu et al., 2025) demonstrate that listwise preference optimization can better leverage the complete ranking information among multiple candidates through sophisticated weighting mechanisms. Future work could integrate listwise ranking objectives to capture more nuanced semantic relationships.

References

- Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uriar, and Janyce Wiebe. 2015. *SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability*. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.
- Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. *SemEval-2014 task 10: Multilingual semantic textual similarity*. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.
- Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. *SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation*. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.
- Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. *SemEval-2012 task 6: A pilot on semantic textual similarity*. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.
- Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. **SEM 2013 shared task: Semantic textual similarity*. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. *LLM2Vec: Large language models are secretly powerful text encoders*. In *First Conference on Language Modeling*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. *Language models are few-shot learners*. *Advances in neural information processing systems*, 33:1877–1901.
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. *SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation*. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.
- Zifeng Cheng, Zhonghui Wang, Yuchen Fu, Zhiwei Jiang, Yafeng Yin, Cong Wang, and Qing Gu. 2025. *Contrastive prompting enhances sentence embeddings in LLMs through inference-time steering*. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3475–3487, Vienna, Austria. Association for Computational Linguistics.
- Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, 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, Seattle, United States. Association for Computational Linguistics.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias

- Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021a. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021b. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Xueying Ding, Xingyue Huang, Mingxuan Ju, Liam Collins, Yozen Liu, Leman Akoglu, Neil Shah, and Tong Zhao. 2025. [Hierarchical token prepending: Enhancing information flow in decoder-based ILM embeddings](#). *arXiv preprint arXiv:2511.14868*.
- Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. [DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.
- 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, Hong Kong, China. Association for Computational Linguistics.
- Yuchen Fu, Zifeng Cheng, Zhiwei Jiang, Zhonghui Wang, Yafeng Yin, Zhengliang Li, and Qing Gu. 2025. [Token prepending: A training-free approach for eliciting better sentence embeddings from LLMs](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3168–3181, Vienna, Austria. Association for Computational Linguistics.
- 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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. [The Llama 3 herd of models](#). *arXiv preprint arXiv:2407.21783*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). In *International Conference on Learning Representations*.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024. [Scaling sentence embeddings with large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196, Miami, Florida, USA. Association for Computational Linguistics.
- Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. [Prompt-BERT: Improving BERT sentence embeddings with prompts](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8826–8837, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Fangkai Jiao, Geyang Guo, Xingxing Zhang, Nancy F Chen, Shafiq Joty, and Furu Wei. 2025. [Preference optimization for reasoning with pseudo feedback](#). In *International Conference on Representation Learning*, volume 2025, pages 19638–19665.
- Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025a. [NV-embed: Improved techniques for training LLMs as generalist embedding models](#). In *The Thirteenth International Conference on Learning Representations*.
- Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, et al. 2025b. [Gemini embedding: Generalizable embeddings from gemini](#). *arXiv preprint arXiv:2503.07891*.
- Yibin Lei, Di Wu, Tianyi Zhou, Tao Shen, Yu Cao, Chongyang Tao, and Andrew Yates. 2024. [Meta-task prompting elicits embeddings from large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10141–10157, Bangkok, Thailand. Association for Computational Linguistics.
- Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. [On the sentence embeddings from pre-trained language models](#). In

- Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. **TruthfulQA: Measuring how models mimic human falsehoods**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.
- Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, Peter J Liu, and Xuanhui Wang. 2025. **LiPO: Listwise preference optimization through learning-to-rank**. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2404–2420, Albuquerque, New Mexico. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. **RoBERTa: A robustly optimized bert pretraining approach**. *arXiv preprint arXiv:1907.11692*.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. **A SICK cure for the evaluation of compositional distributional semantic models**. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Jiaqi Mu and Pramod Viswanath. 2018. **All-but-the-top: Simple and effective postprocessing for word representations**. In *International Conference on Learning Representations*.
- Niklas Muennighoff, Hongjin SU, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. **Generative representational instruction tuning**. In *The Thirteenth International Conference on Learning Representations*.
- Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. **Text and code embeddings by contrastive pre-training**. *arXiv preprint arXiv:2201.10005*.
- Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. **Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models**. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.
- Zhijie Nie, Richong Zhang, and Zhanyu Wu. 2025. **A text is worth several tokens: Text embedding from LLMs secretly aligns well with the key tokens**. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7683–7694, Vienna, Austria. Association for Computational Linguistics.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. **Representation learning with contrastive predictive coding**. *arXiv preprint arXiv:1807.03748*.
- OpenAI. 2022. **GPT-3.5 turbo model**.
- OpenAI. 2023. **GPT-4 technical report**. *arXiv preprint arXiv:2303.08774*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. **Training language models to follow instructions with human feedback**. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Mingjun Pan, Guanquan Lin, You-Wei Luo, Bin Zhu, Zhiend Dai, Lijun Sun, and Chun Yuan. 2025. **Preferece optimization for combinatorial optimization problems**. In *Forty-second International Conference on Machine Learning*.
- Qiwei Peng, David Weir, Julie Weeds, and Yekun Chai. 2022. **Predicate-argument based bi-encoder for paraphrase identification**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5579–5589, Dublin, Ireland. Association for Computational Linguistics.
- R. L. Plackett. 1975. **The analysis of permutations**. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. **Direct preference optimization: Your language model is secretly a reward model**. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.
- Nils Reimers and Iryna Gurevych. 2019. **Sentence-BERT: Sentence embeddings using Siamese BERT-networks**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2025. **Repetition improves language model embeddings**. In *The Thirteenth International Conference on Learning Representations*.

- Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. **One embedder, any task: Instruction-finetuned text embeddings**. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. **Stanford alpaca: An instruction-following llama model**. https://github.com/tatsu-lab/stanford_alpaca.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, and other. 2023. **Llama 2: Open foundation and fine-tuned chat models**. *Preprint*, arXiv:2307.09288.
- Haoyu Wang, Fenglong Ma, Yaqing Wang, and Jing Gao. 2021. **Knowledge-guided paraphrase identification**. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 843–853, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Dixin Jiang, Rangan Majumder, and Furu Wei. 2022. **Text embeddings by weakly-supervised contrastive pre-training**. *arXiv preprint arXiv:2212.03533*.
- Tongzhou Wang and Phillip Isola. 2020. **Understanding contrastive representation learning through alignment and uniformity on the hypersphere**. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. **A broad-coverage challenge corpus for sentence understanding through inference**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.
- Wenpeng Yin and Hinrich Schütze. 2015. **Convolutional neural network for paraphrase identification**. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 901–911, Denver, Colorado. Association for Computational Linguistics.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. **HellaSwag: Can a machine really finish your sentence?** In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.
- Bowen Zhang, Kehua Chang, and Chunping Li. 2024. **Simple techniques for enhancing sentence embeddings in generative language models**. In *Advanced Intelligent Computing Technology and Applications*, pages 52–64, Singapore. Springer Nature Singapore.
- Kechi Zhang, Ge Li, Jia Li, Yihong Dong, Jia Li, and Zhi Jin. 2025a. **Focused-DPO: Enhancing code generation through focused preference optimization on error-prone points**. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9578–9591, Vienna, Austria. Association for Computational Linguistics.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. 2025b. **Qwen3 embedding: Advancing text embedding and reranking through foundation models**. *arXiv preprint arXiv:2506.05176*.
- Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. 2025. **SWIFT: A scalable lightweight infrastructure for fine-tuning**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28):29733–29735.

A More Implementation Details

We implement our DPO training using the SWIFT framework (Zhao et al., 2025). We use four NVIDIA RTX 5090 GPUs to train LLMs with LoRA fine-tuning, setting LoRA modules to all linear layers except the language model head, with rank $r = 8$ and scaling factor $\alpha = 32$. Additionally, we set the per-step batch size to 8 and the gradient accumulation steps to 8, resulting in an effective batch size of 256. For optimization, we employ the AdamW optimizer with a learning rate schedule that includes a linear warmup phase with a ratio of 0.05, peaks at 1×10^{-4} , and is followed by a cosine annealing schedule. A checkpoint is saved every 20K training samples to enable subsequent investigations into the relationship between data size and performance.

B Uniformity and Isotropy Score

B.1 Uniformity

The uniformity metric serves as a quantitative measure for evaluating embedding quality. It assesses the extent to which the embedding distribution is roughly uniform on the unit hypersphere. Preserving such uniformity is essential for maintaining maximal information and effectively utilizing the feature space, attributes that have been shown

to correlate strongly with performance on downstream tasks.

Theoretical Basis. Given the data distribution p_{data} , it is formally defined as:

$$\ell_{\text{uniform}} \triangleq \log \mathbb{E}_{\substack{x, y \sim p_{\text{data}} \\ \text{i.i.d.}}} e^{-2\|f(x) - f(y)\|^2}. \quad (7)$$

Interpretation. A lower value of ℓ_{uniform} indicates that the embeddings are more spread out and isotropic. This uniformity is desirable because it ensures that the feature space is utilized to its full capacity. When features are uniformly distributed, they are more likely to be linearly separable and contain maximal entropy, which corresponds to preserving the most information from the data.

B.2 Isotropy Score

The isotropy score provides a quantitative measure of how uniformly distributed word embeddings are in the high-dimensional space. It is rigorously defined based on the partition function $Z(c)$, which aggregates the contributions of all sentence vectors in a specific direction c .

Rigorous Definition. Given a set of sentence representations $\{v(s) : s \in V\}$, where $v(s) \in \mathbb{R}^d$, the partition function for a unit vector c is defined as:

$$Z(c) = \sum_{s \in V} \exp(c^\top v(s)). \quad (8)$$

The isotropy score $I(\{v(s)\})$ is then defined as the ratio of the minimum to the maximum value of this partition function over all possible unit vectors:

$$I(\{v(s)\}) = \frac{\min_{\|c\|=1} Z(c)}{\max_{\|c\|=1} Z(c)}. \quad (9)$$

Intuitively, if the embeddings are perfectly isotropic, $Z(c)$ should be a constant independent of the direction c , implying $I = 1$, while a score closer to 0 indicates a stronger anisotropy.

Empirical Verification via Eigenvectors. Since there is no closed-form solution for $\arg \max_{\|c\|=1} Z(c)$ or $\arg \min_{\|c\|=1} Z(c)$, we employ an empirical verification strategy based on the eigenvectors of the covariance matrix. We restrict the candidate directions c to the eigenvectors of $V^\top V$, where V is the matrix of all sentence vectors. Let $\{u_1, \dots, u_d\}$ be the eigenvectors of $V^\top V$. The isotropy score is then estimated by computing the partition function for each eigenvector, denoted as $Z(u_j) = \sum_{s \in V} \exp(u_j^\top v(s))$,

and taking the ratio of the minimum to maximum values observed among these eigenvectors:

$$I_{\text{approx}} = \frac{\min_j Z(u_j)}{\max_j Z(u_j)}. \quad (10)$$

C Comparison of GAR Score

We adopt the global alignment rate (GAR) score proposed by Nie et al. (2025) to measure, at the dataset level, how well the aligned tokens of the model’s embeddings cover the real tokens. Given an input sentence s_i , we first tokenize it using the tokenizer of the model and deduplicate the resulting token sequence to obtain the surface token set T_{s_i} . We then compute the top- K_i aligned tokens of the sentence embedding and form the aligned token set $\hat{T}_{s_i}^{K_i}$. The intersection $\hat{T}_{s_i}^{K_i} \cap T_{s_i}$ indicates which tokens in the top- K_i aligned tokens also appear in the surface token set. We perform the same procedure for every sentence in the dataset D and take the union of the resulting “hit token sets” $\bigcup_{i=1}^{|D|} (\hat{T}_{s_i}^{K_i} \cap T_{s_i})$. The cardinality of this union represents the number of distinct surface token types that are hit by alignment across the dataset. On the other hand, $\bigcup_{i=1}^{|D|} T_{s_i}$ gives the set of all surface token types appearing in the dataset. Therefore, GAR is defined as the ratio of the sizes (cardinalities) of the two sets and measures the global coverage of surface tokens by the aligned tokens at the dataset level:

$$\text{GAR} = \frac{\left| \bigcup_{i=1}^{|D|} (\hat{T}_{s_i}^{K_i} \cap T_{s_i}) \right|}{\left| \bigcup_{i=1}^{|D|} T_{s_i} \right|}. \quad (11)$$

We randomly sample 10K of the 1M Wikipedia texts provided by Gao et al. (2021) as the dataset D , set $K = 10$ for all sentences, and compute GAR for LLaMA2-7B before and after fine-tuning following the above procedure. The GAR score increases from 0.185 (pre-finetuning) to 0.324 (post-finetuning). This result indicates that our method substantially improves the dataset-level coverage of surface token types by the model’s top- K aligned tokens and thus increases the diversity of aligned surface tokens.

D Use of AI Assistants

We used AI tools solely to assist with translation and language polishing (e.g., wording and grammar suggestions). The tools were not used to generate research conclusions, experimental results, or core technical content.