Enhancing Unsupervised Sentence Embeddings via Knowledge-Driven Data Augmentation and Gaussian-Decayed Contrastive Learning

Peichao Lai[1], Zhengfeng Zhang[2], Wentao Zhang[1], Fangcheng Fu[1], Bin Cui[1*]
[1]School of Computer Science, Peking University,
[2]College of Computer and Data Science, Fuzhou University
bin.cui@pku.edu.cn

Abstract

Recently, using large language models (LLMs) for data augmentation has led to considerable improvements in unsupervised sentence embedding models. However, existing methods encounter two primary challenges: limited data diversity and high data noise. Current approaches often neglect fine-grained knowledge, such as entities and quantities, leading to insufficient diversity. Besides, unsupervised data frequently lacks discriminative information, and the generated synthetic samples may introduce noise. In this paper, we propose a pipeline-based data augmentation method via LLMs and introduce the Gaussian-decayed gradient-assisted Contrastive Sentence Embedding (GCSE) model[1] to enhance unsupervised sentence embeddings. To tackle the issue of low data diversity, our pipeline utilizes knowledge graphs (KGs) to extract entities and quantities, enabling LLMs to generate more diverse samples. To address high data noise, the GCSE model uses a Gaussian-decayed function to limit the impact of false hard negative samples, enhancing the model's discriminative capability. Experimental results show that our approach achieves state-of-the-art performance in semantic textual similarity (STS) tasks, using fewer data samples and smaller LLMs, demonstrating its efficiency and robustness across various models.

1 Introduction

Sentence representation learning, a fundamental task in natural language processing (NLP), aims to generate accurate sentence embeddings to enhance performance in downstream tasks such as semantic inference (Reimers and Gurevych, 2019), retrieval (Thakur et al., 2021; Wang et al., 2022a), and question answering (Sen et al., 2020). To improve computational efficiency and reduce labor costs, unsupervised sentence embedding methods based on contrastive learning (Gao et al., 2021; Wu et al., 2022c) have emerged as highly effective paradigms. Generally, contrastive learning operates on the principle that robust sentence embeddings should pull semantically similar sentences closer while pushing dissimilar ones further apart. The performance of unsupervised contrastive learning methods largely depends on the quantity and quality of training samples (Chen et al., 2022), highlighting the importance of strategies that effectively enhance both.

Previous studies mainly focused on increasing the number of samples using rule-based word modifications (Wang and Dou, 2023; Wu et al., 2022c) or feature sampling and perturbation techniques (Xu et al., 2023; Chuang et al., 2022a). Recent studies (Zhang et al., 2023; Wang et al., 2024a) use either few-shot manually constructed samples or zero-shot generalized refactoring instructions to create prompts that guide large language models (LLMs) in generating new samples from original sentences, increasing both the quantity and quality of the data. Although these methods have achieved commendable performance, two limitations remain:

Low Data Diversity. Diverse data samples in sentence representation learning should contain varied expressions of the same knowledge. However, existing approaches often struggle to distinguish fine-grained semantic knowledge like entities and quantities in the context. Traditional methods modify sentences using limited patterns without considering fine-grained knowledge, restricting their effectiveness in enhancing sample diversity. Recent LLM-based methods like Wang et al. (2024b), SynCSE (Zhang et al., 2023) and MultiCSR (Wang et al., 2024a), adjust topic and entailment categories in prompts to guide the model in generating varied samples. These methods focus on the global context but lack precise control over the knowledge in the samples. Consequently, the diversity of gen-

*indicates the corresponding author
[1]Code is available at: https://github.com/aleversn/GCSE

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4919–4940
July 27 - August 1, 2025

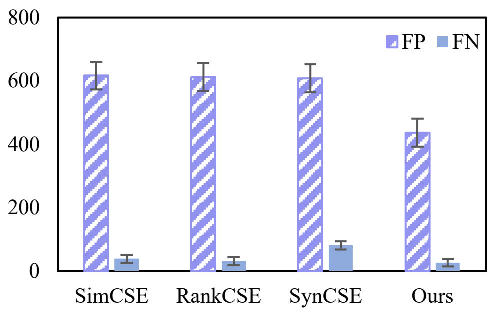


Figure 1: Comparison of false positives (FP) and negatives (FN). Both the predicted scores and labels are normalized (see details in Appendix J), where positives have a score greater than the label, while negatives lower than the label. False samples are identified when the root mean square error (RMSE) between the prediction and the label exceeds 0.2.

erated samples is constrained by the probability distributions of LLMs, resulting in unpredictable data quality.

High Data Noise. Unsupervised sentence representation learning often suffers from data noise caused by confusing negative samples, which mainly arise from two sources. First, traditional methods generate datasets by duplicating samples to create positive instances, leading to negatives with similar surface-level semantics that affect the model's understanding of fine-grained semantic information (Miao et al., 2023; Zhou et al., 2022). Second, in data synthesis, differences in semantic distributions can cause the LLM's criteria for distinguishing between positive and negative samples to misalign with the target domain, introducing additional noise (Huang et al., 2023; Poerner and Schütze, 2019). Existing method like MultiCSR attempts to remove noisy samples using linear programming, but this can eliminate potentially valuable samples and reduce data diversity. Figure 1 compares various baselines on the STS-Benchmark development set. The results show that the prediction of false positives outnumber false negatives, and data synthesis in SynCSE increases false negatives, further supporting the above analysis.

In this paper, we propose a pipeline-based data augmentation method using LLMs and introduce the Gaussian-decayed gradient-assisted Contrastive Sentence Embedding (GCSE) model to improve the performance of unsupervised sentence embedding methods. To address the issue of ***low data diversity***, we begin by extracting entities and quantities from the data samples and constructing a knowledge graph (KG). Next, we create a sentence construction prompt using the extracted knowledge to guide LLM in generating more diverse positive samples. To tackle ***high data noise***, we employ an evaluation model to annotate the synthesized data and initially filter out false positive samples. To further minimize the impact of false negatives while maintaining sample diversity, we align hard negatives with the evaluation model's distribution and reduce their gradient during the initial training step. Then, we leverage other in-batch negative samples to optimize the semantic space. Inspired by locally weighted linear regression (Atkeson et al., 1997), we propose the GCSE model, which utilizes a Gaussian-decayed function to adjust prediction discrepancies between the GCSE model and the evaluation model. Initially, it reduces the gradient impact of hard negatives, gradually restoring their gradient weights as training progresses if they deviate significantly from the evaluation model's distribution. This approach prevents false negatives from being pushed further in the semantic space, promoting a more uniform distribution.

Methods	**Synthesis Approach**	**Use Knowledge**	**Denoise**
SynCSE	Few-shot Synthesis	No	No
MultiCSR	Zero-shot Synthesis	No	Yes
Ours	Zero-shot Synthesis	Yes	Yes

Table 1: Comparison of our methods and related LLM-based methods.

We highlight the key innovations of our approach in Table 1: (i) We are the first to incorporate fine-grained knowledge for sample synthesis in LLM-based methods. (ii) Unlike MultiCSR's denoising approach, our method retains more false samples for training rather than discarding them. (iii) Our data selection strategy is particularly well-suited for leveraging a local LLM to synthesize domain-specific samples from a limited number of samples, leading to improved performance. Experimental results demonstrate the efficiency of our model, outperforming previous best methods in average scores for semantic textual similarity (STS) tasks by 1.05% with BERT-base, 1.89% with BERT-large, 0.50% with RoBERTa-base, and 1.50% with RoBERTa-large.

In summary, our contributions are as follows: (1) *New method.* We introduce a pipeline-based data augmentation method using LLM for few-shot domain data and propose a Gaussian-decayed gradient-assisted Contrastive Sentence Embed-

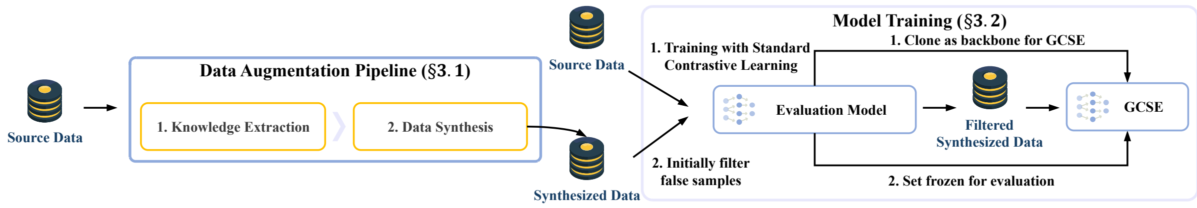


Figure 2: The overall workflow of our method.

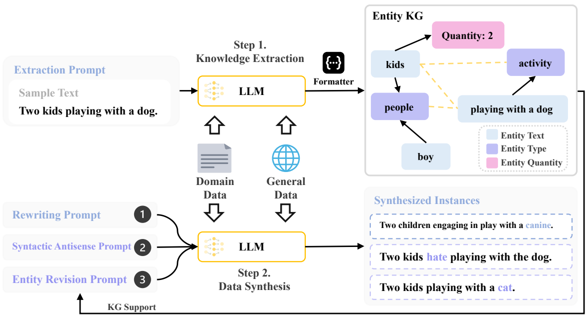


Figure 3: The pipeline of knowledge extraction and data synthesis, where the solid black arrows in the Entity KG are hard edges, and dotted yellow lines are soft edges.

ding (GCSE) model to reduce data noise. (2) *New perspective.* To the best of our knowledge, we are the first to explore combining knowledge graphs with LLM to synthesize data, enhancing fine-grained sentence representation learning by generating diverse positive and negative samples. (3) *State-of-the-art performance.* Experimental results demonstrate that our method achieves superior performance on STS tasks while using fewer samples for data synthesis with smaller LLM parameters.

2 Related Work

Early work on sentence embeddings builds on the distributional hypothesis, predicting surrounding sentences (Kiros et al., 2015; Logeswaran and Lee, 2018; Hill et al., 2016) or extending the word2vec framework (Mikolov et al., 2013) with n-gram embeddings (Pagliardini et al., 2018). Post-processing techniques like BERT-flow (Li et al., 2020) and BERT-whitening (Su et al., 2021) address the anisotropy issue in pre-trained language models (PLMs), and more recent methods focus on generative approaches (Wang et al., 2021; Wu and Zhao, 2022) and regularizing embeddings to prevent representation degeneration (Huang et al., 2021). Recently, contrastive learning approaches have become prominent, using various augmentation methods to derive different views of the same sentence (Zhang et al., 2020; Giorgi et al., 2021; Kim et al., 2021; Gao et al., 2021). Among these, SimCSE uses dropout as a simple augmentation and achieves strong results in unsupervised STS tasks, inspiring further approaches like ArcCSE (Zhang et al., 2022), DiffCSE (Chuang et al., 2022a), GS-InfoNCE (Wu et al., 2022b), and RankCSE (Liu et al., 2023).

With the advent of LLMs (OpenAI, 2023; Bai et al., 2023; Touvron et al., 2023), some works attempt to utilize LLM for sentence representation learning. For example, Ni et al. (2022) uses T5 with mean pooling to obtain a sentence embedding model by fine-tuning on a large-scale NLI corpus; Cheng et al. (2023) uses prompt learning to measure the semantic similarity of sentence pairs; Springer et al. (2024) employs sentence repetition to enhance the capacity for sentence representation; AoE (Li and Li, 2024a) optimize angle differences for improving supervised text embedding; and BeLLM (Li and Li, 2024b) designs a Siamese structure for learning sentence embeddings.

3 Methodology

In this section, we present the data augmentation pipeline via LLM and the specific structure of the GCSE. As shown in Figure 2, we start by using a data augmentation pipeline to synthesize new samples from the source data, and then train our model with the filtered synthetic data.

3.1 Data Augmentation

In the data augmentation pipeline, we utilize both domain data and partial general data to balance domain-specific relevance and general-domain applicability. We start by extracting knowledge from the source data and then synthesize new data for our model training. The detailed structure of the pipeline is shown in Figure 3.

Knowledge Extraction and Integration. The variety and relationships between samples directly impact model performance in sentence representation learning. A major challenge with existing LLM-based data synthesis methods is the limited diversity they generate for each short text. To trade off the low diversity of the generated samples with their relevance to the domain semantic space, we first design an extraction prompt to obtain entities and quantities from the given data. Formally, we denote the extraction prompt as $\mathcal{P}_e$, and LLM $\mathcal{L}$, suppose we finally extract instances with d sample number, the knowledge set $\mathcal{K}_i = \{k_{i1}, \ldots, k_{in}\}$ of each instance x_i is computed in Equation 1, where t_j, c_j and q_j represent the entity text, entity type, and quantity of k_i. n is the size of $\mathcal{K}_i$, and $\mathcal{F}(\cdot)$ is the formatting function that converts text to a triplet. Next, we integrate all knowledge by establishing an entity knowledge graph $\mathcal{G} = \langle V, E \rangle$, where the node set V contains all the $\langle t, c, q \rangle$ from $\mathcal{K}$:

$$\mathcal{K} = \bigcup_{i=1}^{d} \mathcal{F}([\mathcal{P}_e; x_i], \mathcal{L}) = \bigcup_{i=1}^{d} \{\langle t_{ij}, c_{ij}, q_{ij} \rangle \mid j \in [1, n]\}, \quad (1)$$

$$V = \{t_{ij}, c_{ij}, q_{ij} \mid i \in [1, d]; j \in [1, n]\}. \quad (2)$$

The edges E consist of hard edges E_r and soft edges E_s. As shown in Equations 3 and 4, E_r represents the relationship between the entity text, type, and quantity of each $k \in \mathcal{K}$, and E_s indicates the relationship between entity text in k_{ij} and other entity text or type in the same instance x_i.

$$E_r = \{(t_{ij}, c_{ij}) \cup (t_{ij}, q_{ij}) \mid i \in [1, d]; j \in [1, n]\}, \quad (3)$$

$$E_s = \textstyle\bigcup_{i=1}^{d} \{(t_{ij}, t_{ik}), (t_{ij}, c_{il}) \mid j, k, l \in [1, n]; k, l \neq j\}. \quad (4)$$

By defining hard and soft edges, we can more efficiently identify and replace entity nodes near the current node, improving the correlation between the synthesized instance and the source instance.

Data Synthesis via LLM. Empirical evidence and model performance on standard datasets show that sentence embedding models struggle more with accurately identifying negative samples than positives (Chuang et al., 2022a; Miao et al., 2023). In the contrastive learning methods, the model acquires sentence embedding representation by calculating the distance between sentence-pairs. It aims to minimize the spatial distance between positive pairs and increase the spatial distance between negative pairs. Thus, it is essential to obtain negative samples that closely resemble the source instance in surface-level features, while positive samples should have diverse representations but still convey the same meaning as the source instance.

In this study, we use LLM to generate positive samples through a rewrite prompt. We also focus on the impact of variations in entities and quantities within the samples. Negative samples are generated by the LLM at both the syntactic and fine-grained knowledge levels. The data synthesis prompts are divided into three main types: (1) Rewriting prompt, (2) Syntactic antisense prompt, and (3) Entity revision prompt. The first type is used to create positive samples, while the second and third types are used to create negative samples at the syntactic and knowledge levels, respectively.

The "rewriting prompt" can be classified into three forms: directly requesting LLM to generate a new sentence instance using the "rewrite" instruction, creating the preceding part of the sentence instance, and generating based on the knowledge set of the instance. As the diversity of synthetic samples increases, the likelihood of generating false positives also rises. To address this, the next section involves scoring the generated samples using an evaluation model.

The "syntactic antisense prompt" aims to modify the semantics to create a contradiction at the syntactic level. Such as transforming it into a positive or negative statement using explicit positive or negative words, or by expressing a contrary sentiment. This is an initial approach to synthesizing negative samples that preserves a strong coherence with the source instance in terms of sequence structure. However, it is deficient in generation diversity. To alleviate the issue, the "entity revision prompt" aims to enhance text diversity by replacing the entity text and quantity compared to the source instance. Simultaneously, to ensure the semantic relevance between the synthetic samples and the source instance, replacement entities are selected by searching for neighboring nodes on entity KG. We define $\mathcal{T}(\cdot)$ as the search function, and the replacement entity of t_{ij} are computed as:

$$\mathcal{T}_r(t_{ij}) = \{t_p \mid (t_{ij}, c_{ij}) \in E_r \wedge (t_p, c_{ij}) \in E_r\}, \quad (5)$$

$$\mathcal{T}_s(t_{ij}) = \{t_p \mid (t_{ij}, t_p) \in E_s\}, \quad (6)$$

$$\mathcal{T}_p(t_{ij}) = \{t_p \mid \exists t_k \in \mathcal{T}_s(t_{ij}) \cap \mathcal{T}_s(t_p) \wedge t_p \in \mathcal{T}_r(t_{ij})\}, \quad (7)$$

$$\mathcal{T}(t_{ij}) = \mathcal{T}_r(t_{ij}) \cup \mathcal{T}_p(t_{ij}), \quad (8)$$

where the function $\mathcal{T}_r(\cdot)$ is used to search for entities that share a hard edge with the current entity, while $\mathcal{T}_s(\cdot)$ retrieves entities connected via a soft edge. The function $\mathcal{T}_p(\cdot)$ is designed to find a replacement entity t_p that is of the same type as t_{ij} and shares soft-edge connections with another in-context entity t_k. Finally, the replacement entity is randomly selected from the results of the search function $\mathcal{T}(t_{ij})$. Compared to random entity substitution, our strategy significantly improves the semantic relevance between the synthesized sample and the source instance.

3.2 Model Training

The training process of our model consists of two stages. First, we combine general and domain-specific data to train an evaluation model using standard unsupervised contrastive learning. This improves the uniformity of sentence embeddings in general scenarios and reduces the impact of semantic distribution limitations in the synthesized data, enhancing model robustness. Then, we freeze the evaluation model to filter synthetic data and help the GCSE model eliminate false hard negative sample noise.

General Contrastive Learning. In the first stage, we follow the formulation of SimCSE (Gao et al., 2021) to train the evaluation model. Formally, we define the encoder of the evaluation model as E', each unlabeled sentence instance as x_i, and its positive sample as $x_i^+ = x_i$. The representation of each instance is denoted as $\mathbf{h}' = \mathcal{F}_{E'}(x)$, the representations of x_i and x_i^+ are computed as $\mathbf{h}_i'$ and $\mathbf{h}_i'^+$, respectively. Since the dropout mask in E' is random, $\mathbf{h}_i'$ and $\mathbf{h}_i'^+$ are computed with the same input but with slightly different results. Then, the loss of evaluation model is defined as:

$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i', \mathbf{h}_i'^+)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{h}_i', \mathbf{h}_j'^+)/\tau}}, \tag{9}$$

where N represents the size of each mini-batch, τ is a temperature hyperparameter, and $\mathrm{sim}(\cdot)$ is the cosine similarity function.

Denoising Training. In the second stage, we adopt a copy of the evaluation model as the backbone of GCSE and continue training on synthesized data. In this stage, each input is set as a triplet (x_i, x_i^+, x_i^-), where x_i^+ and x_i^- stand for the positive and negative samples of x_i, respectively. Nevertheless, the synthesized data contains many potential false positive and false negative samples, necessitating the implementation of a filtering process. We use the frozen evaluation model to initially correct these inaccurate samples and build the ultimate triplet dataset. Let $\mathcal{S}(x_i) = \{\hat{x}_{i1}, \ldots \hat{x}_{im}\}$ denotes the synthetic data set of x_i, where m is the size of the set, and x_i^+, x_i^- are calculated as:

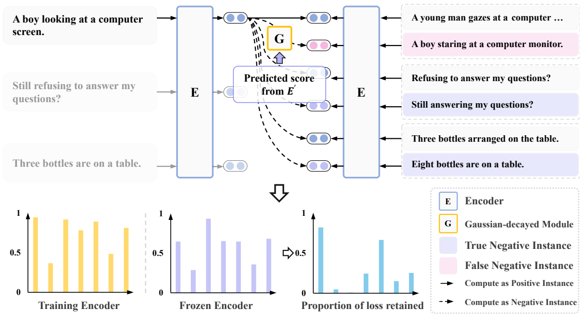


Figure 4: In-batch training with Gaussian-decayed on GCSE.

$$x_i^+ = \begin{cases} \hat{x}_{ij}, & \mathrm{sim}(\mathbf{h}_i', \hat{\mathbf{h}}_{ij}') \geq \alpha, j \in [1, m] \\ x_i, & \text{else} \end{cases}, \tag{10}$$

$$x_i^- = \begin{cases} \hat{x}_{ij}, & \mathrm{sim}(\mathbf{h}_i', \hat{\mathbf{h}}_{ij}') \leq \beta, j \in [1, m] \\ x_k, & k \in [1, N], k \neq i \end{cases}, \tag{11}$$

where α, β are the threshold for positives and negatives, respectively. x_k denotes a randomly selected instance from in-batch data. We can set a high value for α to reduce false positive samples. However, filtering out false negatives in synthetic data is more challenging. In theory, smaller β can reduce more false negatives, but samples with low similarity to the source instance are easy to distinguish due to significant surface-level differences. As a result, training on these samples does not effectively improve the model's ability to distinguish fine-grained false positives. Therefore, we opt for a higher value of β. During training, we use a Gaussian-decayed function to align the distances of hard negative samples between the GCSE encoder E and the frozen encoder E'. As shown in Figure 4, for each mini-batch of triplet inputs, both E and E' compute similarity scores for the negative samples and their corresponding source instances. The loss for each instance in GCSE is defined as:

$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau} + \sum_{\substack{j=1 \\ j \neq i}}^{N} e^{\mathrm{sim}(\mathbf{h}_i, \mathbf{h}_j^-)/\tau} + e^{G(s_i, s_i', \tau, \sigma)}}, \tag{12}$$

$$G(s_i, s_i', \tau, \sigma) = \begin{cases} s_i \left(1 - e^{-\frac{(s_i - s_i')^2 \tau^2}{2\sigma^2}}\right), & s_i \leq s_i' \\ s_i, & s_i > s_i' \end{cases}, \tag{13}$$

where $s_i = \mathrm{sim}(\mathbf{h}_i, \mathbf{h}_i^-)$, $s_i' = \mathrm{sim}(\mathbf{h}_i', \mathbf{h}_i'^-)$. $G(\cdot)$ is the Gaussian-decayed function, where the loss attenuation of the hard negative sample grows as the distance between s_i and s_i' decreases, and σ is a hyperparameter that controls the width of $G(\cdot)$. This implies that when E initially calculates the hard negative sample, it follows the spatial distribution of E' as the "established guidelines" and uses other in-batch negative samples to further increase the spatial distance between negatives, effectively reducing the influence of false negatives. As training progresses, the spatial distribution of true hard negatives between E and E' will progressively increase, and its gradient will be restored.

4 Experiment

4.1 Experiment Setup

Training: In our main experiments, we evaluate model performance under two settings: (1) the default setting, where samples are synthesized using Wikipedia texts following Gao et al. (2021); and (2) a simulated low-resource, high-quality setup using a smaller yet diverse set of domain-specific and general data. Specifically, we use a subset of the NLI dataset from Gao et al. (2021) as the general data, and select the training sets from STS-12 (Cer et al., 2017) (2.2k samples), PAWS (Zhang et al., 2019) (3.5k samples), and SICK (Marelli et al., 2014) (4.5k samples) as domain-specific data. To simulate an unsupervised learning scenario, we only include the unlabeled portions of these datasets. In this experiment, the sample ratio between domain and general data is set to 1:3.

We adopt ChatGLM3-6B (GLM et al., 2024), GLM4-9B-Chat (GLM et al., 2024), Qwen2.5-32B-Instruct (Yang et al., 2024b,a), GPT-3.5 Turbo (OpenAI, 2022) and Deepseek-V3-0324 (DeepSeek-AI, 2024) as LLMs for data synthesis, respectively. We choose BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as the backbone models of GCSE. In the stage of Gaussian-decayed training on synthesized data, the filtering thresholds of α and β are set as 0.9 and 0.75, respectively. The temperature of τ is set as 0.05, and the σ of $G(\cdot)$ is set as 0.01. In the first stage training, the evaluation model is firstly trained on the unlabeled dataset of all general data and domain data. One copy instance of the evaluation model is then utilized as the pre-trained model for GCSE, while the original instance is set to be frozen to filter synthesized data and provide guidance for GCSE. In the second stage, GCSE is trained on the filtered synthesized data, and the sentence embedding is obtained from the last output hidden states of the first token. During the data augmentation phase, we used an NVIDIA A800 80G for LLM-based data synthesis. In the training phase, we conducted training and validation on eight NVIDIA TITAN RTX GPUs.

Evaluation: To validate our method for sentence embeddings, we evaluated the model's performance on semantic textual similarity (STS) tasks, we use the standard evaluation method, measuring model performance with Spearman's correlation, and we adopt SentEval[2] (Conneau and Kiela, 2018) as the evaluation tool, which contains seven STS subsets: STS 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), the STS-Benchmark (Cer et al., 2017) and the SICK Relatedness (Marelli et al., 2014). Additionally, we compared the reranking task performance on Appendix B, and the performance of our model with other methods on transfer tasks in SentEval to evaluate its applicability in Appendix D.

Baselines: We compare our method with mainstream unsupervised sentence embedding baselines: BERT-whitening (Su et al., 2021), SimCSE (Gao et al., 2021), DiffCSE (Chuang et al., 2022b), PromptBERT (Jiang et al., 2022), PCL (Wu et al., 2022a), CARDS (Wang et al., 2022b), DebCSE (Miao et al., 2023) and RankCSE (Liu et al., 2023). In addition, we further compare two baselines: SynCSE (Zhang et al., 2023) and MultiCSR (Wang et al., 2024a), which use LLM for data synthesizing in whole NLI datasets. To verify the effectiveness of our data synthesis method, we choose their results of using GPT-3.5 Turbo for comparison.

4.2 Main Results

STS Tasks: The overall results of the STS tasks are shown in Table 2. Our approach, utilizing synthetic samples from Deepseek-V3-0324 and GPT-3.5 Turbo achieve the best performance across all backbones when compared to other unsupervised baselines. Even with synthetic samples from ChatGLM3-6B, our method still outperforms previous approaches on all backbones. This highlights the applicability of our method, as it can be effectively applied to multiple models. Compared to the standard unsupervised SimCSE, Spearman's correlation of GCSE (ChatGLM3-6B) is improved by an average of 5.40% on the base models and 3.95% on the large models. On the strong base-

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

Model	Method	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	Avg.
BERT-base	whitening†	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
	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
	RankCSE♠	75.66	**86.27**	77.81	84.74	81.10	81.80	75.13	80.36
	SynCSE (GPT-3.5 Turbo)*	75.86	82.19	78.71	**85.63**	81.11	82.35	78.79	80.66
	MultiCSR (GPT-3.5 Turbo)♣	74.86	84.19	79.46	84.70	80.34	83.59	**79.37**	80.93
	GCSE (ChatGLM3-6B)	78.14	85.89	80.71	84.92	81.20	82.89	77.49	81.61
	GCSE (GLM4-9B-Chat)	77.30	86.21	80.60	84.98	81.48	83.22	77.82	81.66
	GCSE (Qwen2.5-32B-Instruct)	77.83	86.07	80.77	85.32	81.51	83.26	78.17	81.85
	GCSE (GPT-3.5 Turbo)	77.88	86.21	**80.91**	84.98	**81.60**	83.38	78.59	81.94
	GCSE (Deepseek-V3-0324)	**78.33**	86.12	80.31	85.32	81.38	**83.62**	78.79	**81.98**
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
	RankCSE♠	75.48	86.50	78.60	85.45	81.09	81.58	75.53	80.60
	SynCSE (GPT-3.5 Turbo)*	74.24	85.31	79.41	85.71	81.76	82.61	79.25	81.18
	GCSE (ChatGLM3-6B)	77.69	86.98	81.68	86.01	81.89	84.28	79.43	82.57
	GCSE (GLM4-9B-Chat)	78.17	87.02	82.08	**86.62**	82.04	84.71	79.53	82.89
	GCSE (Qwen2.5-32B-Instruct)	78.34	87.02	81.88	86.39	82.29	84.80	79.97	82.96
	GCSE (GPT-3.5 Turbo)	**78.60**	**87.27**	82.18	85.90	**82.30**	84.77	80.09	83.02
	GCSE (Deepseek-V3-0324)	78.11	87.22	**82.23**	86.31	82.13	**84.93**	**80.55**	**83.07**
RoBERTa-base	whitening†	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
	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 (GPT-3.5 Turbo)††	74.61	83.76	77.89	85.09	82.28	82.71	78.88	80.75
	MultiCSR (GPT-3.5 Turbo)♣	75.61	84.33	80.10	84.98	82.13	84.54	**79.67**	81.62
	GCSE (ChatGLM3-6B)	76.95	85.59	80.43	85.90	**83.20**	**84.62**	77.28	82.00
	GCSE (GLM4-9B-Chat)	77.83	84.62	80.17	86.21	82.99	84.05	78.33	82.03
	GCSE (Qwen2.5-32B-Instruct)	77.81	84.56	80.23	86.13	83.19	84.38	78.06	82.05
	GCSE (GPT-3.5 Turbo)	**78.03**	83.79	**80.61**	**86.28**	82.76	84.31	79.01	82.11
	GCSE (Deepseek-V3-0324)	77.77	84.33	80.60	86.01	82.75	84.60	78.77	**82.12**
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 (GPT-3.5 Turbo)††	75.45	85.01	80.28	86.55	83.95	84.49	**80.61**	82.33
	GCSE (ChatGLM3-6B)	76.10	86.64	81.21	85.90	83.99	85.51	79.11	82.64
	GCSE (GLM4-9B-Chat)	77.94	87.00	82.34	86.52	84.27	86.19	78.38	83.23
	GCSE (Qwen2.5-32B-Instruct)	77.79	87.45	82.22	87.86	**84.62**	**86.75**	78.30	83.57
	GCSE (GPT-3.5 Turbo)	**78.21**	**87.47**	**82.76**	87.79	84.40	86.15	80.02	**83.83**
	GCSE (Deepseek-V3-0324)	78.11	87.40	82.61	**88.00**	84.15	86.32	80.12	83.82

Table 2: Comparison of Spearman's correlation results on STS tasks. The values in parentheses indicate using data synthesized by different LLMs. The values in bold and underlined indicate the best and second-best values, respectively. "†": results from Miao et al. (2023), "♣": results from Wang et al. (2024a), "♠": results from Liu et al. (2023), "††": results from Zhang et al. (2023). "*": we reproduce the results with the officially released corpus from Zhang et al. (2023). GCSE has significant differences with all comparable baselines on the t-test ($p < 0.5\%$).

line RankCSE, GCSE (ChatGLM3-6B) achieved a 1.90% improvement over its average performance, demonstrating the effectiveness of the LLM data synthesis process.

Furthermore, compared to the two state-of-the-art baseline models SynCSE and MultiCSR, both of which rely on LLMs for data synthesis, our approach consistently achieves better performance across all backbone models. Table 3 further reports our results under a simulated low-resource, high-quality domain-specific setting. The results show that our method, which utilizes local LLMs, achieves higher average Spearman correlations than the GPT-3.5 Turbo-based versions of both baseline models. It is also worth noting that our method uses only 14% of the sample size compared to SynCSE and MultiCSR, which rely on the full NLI datasets. These results demonstrate the effectiveness of our data synthesis method and our domain-oriented sample selection strategy.

4.3 Analysis

Ablation Studies: We analyze the impact of each module or strategy in GCSE (ChatGLM3-6B) under domain-specific setting and report the results in Table 4. First, "w/o stage-2" refers to the results obtained without training in the second stage. This leads to a significant decrease in performance compared to the default model, which is the performance of the evaluation model and is similar to the

Model	Method	Avg.
BERT-base	GCSE (ChatGLM3-6B)	81.35
	GCSE (GLM4-9B-Chat)	81.58
	GCSE (GPT-3.5 Turbo)	**81.92**
BERT-large	GCSE (ChatGLM3-6B)	81.49
	GCSE (GLM4-9B-Chat)	82.05
	GCSE (GPT-3.5 Turbo)	**82.76**
Roberta-base	GCSE (ChatGLM3-6B)	81.23
	GCSE (GLM4-9B-Chat)	81.70
	GCSE (GPT-3.5 Turbo)	**82.18**
Roberta-large	GCSE (ChatGLM3-6B)	82.62
	GCSE (GLM4-9B-Chat)	82.91
	GCSE (GPT-3.5 Turbo)	**83.83**

Table 3: Spearman's correlation results on STS tasks under the low-resource domain data setting. The best results are highlighted in bold.

conventional unsupervised SimCSE. Then, "w randomly" refers to the direct use of the instance itself as a positive sample in the combination dataset of domain and general data, while randomly selecting a negative instance from the dataset. We can observe that its performance in this case is even worse than the evaluation model. This demonstrates that the diversity of positive samples and the quality of negative samples significantly impact the performance of the model. "w/o filtering" indicates the results of training by skipping evaluation model filtering and directly using the data synthesized by LLM. The results show that the performance of the model is significantly affected when false positive and negative samples are introduced without filtering. We investigate the impact of the Gaussian-decayed function by removing it, and the results are shown in "w/o decay". We can observe that the default model performs better overall than when the Gaussian-decayed function is removed, indicating that it can filter out potential false negative sample noise. Finally, we analyze the necessity of including general data and domain data in "w/o general" and "w/o domain" respectively. It can be observed that removing either of them results in a decline in performance, which indicates the significance of domain data and the essentiality of general data in our method.

Analysis of entities and quantities awareness: We analyze GCSE awareness of entities and quantities by constructing a dataset using the data synthesis method in Section 3.1 on the STS-Benchmark development set. Then, the similarity scores of each triplet in the dataset are annotated by two supervised pre-trained models: "sup-simcse-bert-large" and "sup-simcse-roberta-large". The final label is the average score of the similarity calculated by both models. We evaluate Spearman's correlation scores of GCSE and the other three strong baselines on the backbone of the BERT-base model, and the results are shown in Table 5. Our GCSE achieves the best result and outperforms RankCSE by 14.03%. In this case, both SynCSE and GCSE achieve significant improvements over methods without LLM. This might be due to the similarity of the semantic representation space between the training set and the development set, both of which are synthesized via LLM. Nevertheless, GCSE shows a notable enhancement in performance of 2.19% compared to SynCSE, demonstrating that its understanding of the entities and quantities in sentences has enhanced to a certain degree.

4.4 Impact on the ratio between domain and general data

Figure 5 presents the trend of the GCSE Spearman's correlation result as the proportion of general data introduced increases, where "d" represents that only using the domain data. The results show that adding a certain amount of general data improves performance on STS tasks. However, when the size of general data exceeds three times that of domain data, performance starts to decline. This suggests that incorporating a moderate amount of external data enhances the uniformity of sentence embeddings. But as the out-of-domain data grows, the influence of domain-specific data on training weakens. Overall, the results indicate that domain data improves the model's ability to represent target domain sentences, while general data helps with sentence embedding uniformity.

4.5 Impact of the Gaussian-decayed

To further investigate the effectiveness of the Gaussian-decayed function, we analyze the GCSE (ChatGLM3-6B) performance in domain-specific setting against the weight of σ on the synthesized data, both with and without filtering. As shown in Figure 6, we use the synthesized data without filtering to evaluate the efficacy of the Gaussian-decayed function in eliminating false negative samples, and results are presented in Figure 6 (b). It is clear that the model's performance improves as the weight of σ grows. This suggests that a greater σ weight enhances the model's effectiveness in mitigating the impact of false negative samples. It is important to acknowledge that a higher σ does

Method	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	Avg.
GCSE (ChatGLM3-6B)	**76.91**	85.48	**79.49**	**84.28**	**82.65**	**83.90**	**76.72**	**81.35**
w/o stage-2	71.85	83.65	76.84	83.37	78.74	79.10	71.69	77.89
w randomly	71.94	84.03	76.99	83.65	79.11	78.66	69.28	77.67
w/o filtering	74.65	83.54	77.39	83.27	79.97	79.66	74.27	78.96
w/o decay	76.26	**85.98**	79.35	84.09	82.12	83.85	76.00	81.09
w/o general	75.44	85.55	79.19	84.91	80.23	81.57	74.14	80.15
w/o domain	75.59	85.66	78.93	84.09	80.87	82.29	76.00	80.49

Table 4: Ablation studies of STS tasks on BERT-base. Other PLMs yield similar patterns to BERT-base.

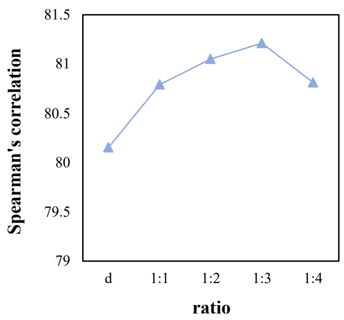


Figure 5: Spearman's correlation against the ratio of domain data to general data on the STS tasks.

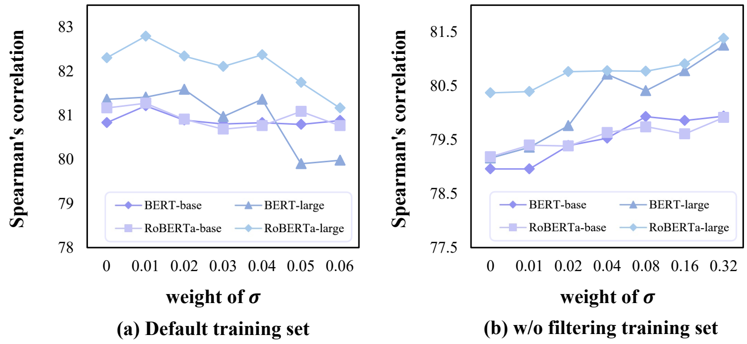


Figure 6: Spearman's correlation against the weight of the Gaussian-decayed on the STS tasks.

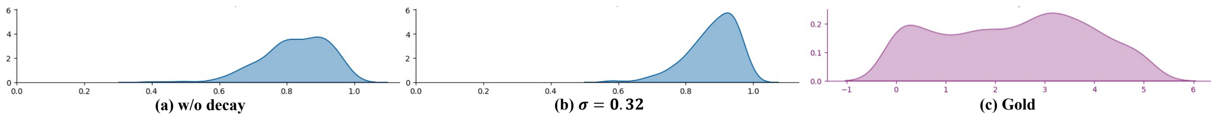


Figure 7: Density plots of the STS-Benchmark development set with labels ≥ 4, which is evaluated by GCSE (ChatGLM3-6B) in domain-specific setting with different σ weights. (c) is the density plot of gold labels.

Method	Spearman's
unsup-SimCSE	75.59
RankCSE	79.74
SynCSE (GPT-3.5 Turbo)	91.58
GCSE (ChatGLM3-6B)	**93.77**

Table 5: Comparison of Spearman's correlation results on the synthetic data of the STS-Benchmark development set.

not necessarily indicate better performance. As shown in Figure 6 (a), an increase in σ at the initial stage contributes to enhancing the model's performance. Nevertheless, as the weight of σ increases, the performance of backbones generally declines, resulting in the model adhering too strictly to the "established guidelines". Consequently, it impacts the efficacy of learning from the hard negative samples. We further use the density plots to visualize the prediction on the STS-Benchmark development set in Figure 7. These models are trained on the synthesized data without filtering. We can observe that in Figure 7 (a), the distribution of prediction results for labels ≥ 4 is significantly shifted to the left. Compared with the results in Figure 7 (b), this issue is effectively alleviated, demonstrating the effectiveness of the Gaussian-decayed function in reducing the influence of false negative samples. To further verify the applicability of the Gaussian-decayed function, we applied it to SynCSE and verified the performance in Appendix F.

5 Conclusion

In this paper, we propose a pipeline-based data augmentation method using LLM to enhance data diversity in sentence representation learning. By leveraging knowledge of entities and quantities, our approach improves the model's ability to capture fine-grained semantic distinctions. The Gaussian-decayed function in our GCSE model further reduces noise in the generated data. Extensive experiments on STS and reranking tasks show that our method achieves state-of-the-art results with fewer synthesized samples and a more lightweight LLM, demonstrating its effectiveness and efficiency.

Limitations

While our data augmentation method achieves promising results across LLMs with varying parameter scales, we observe performance discrepancies depending on the size of the LLM used. These variations may arise from differences in how effectively each model adheres to and aligns with the provided prompts. In future work, we plan to address these limitations by enhancing prompt adherence and alignment across different LLM architectures.

Ethics Statement

Our data augmentation method leverages LLMs to generate data independently of the existing training dataset. However, it is important to note that the generated data may inherit social biases present in the pre-training corpus. Therefore, in practical applications, we recommend conducting manual reviews of the generated data to mitigate the risk of propagating biased information into the sequence labeling models.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, 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.

Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. 1997. Locally weighted learning. *Artif. Intell. Rev.*, 11(1-5):11–73.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *CoRR*, abs/2309.16609:1–59.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, and Xipeng Qiu. 2023. Improving contrastive learning of sentence embeddings from AI feedback. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11122–11138, Toronto, Canada. 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. 2022a. 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.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James R. Glass. 2022b. 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, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4207–4218. Association for Computational Linguistics.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. 2020. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online. Association for Computational Linguistics.

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, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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*, pages 4171–4186. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

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. Association for Computational Linguistics.

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

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

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, San Diego, California. Association for Computational Linguistics.

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, Seattle, Washington, USA, August 22-25, 2004*, pages 168–177. ACM.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. Whiteningbert: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 238–244. Association for Computational Linguistics.

Yongxin Huang, Kexin Wang, Sourav Dutta, Raj Patel, Goran Glavaš, and Iryna Gurevych. 2023. AdaSent:

Efficient domain-adapted sentence embeddings for few-shot classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3420–3434, Singapore. 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. PromptBERT: 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.

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

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1279–1289, San Diego, California. 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.

Xianming Li and Jing Li. 2024a. AoE: Angle-optimized embeddings for semantic textual similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839, Bangkok, Thailand. Association for Computational Linguistics.

Xianming Li and Jing Li. 2024b. BeLLM: Backward dependency enhanced large language model for sentence embeddings. 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 792–804, Mexico City, Mexico. Association for Computational Linguistics.

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), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13785–13802. Association for Computational Linguistics.

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, NL4SE@ESEC/SIGSOFT FSE 2018, Lake Buena Vista, FL, USA, November 4, 2018*, pages 2–5. ACM.

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. *CoRR*, abs/1907.11692:1–13.

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

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, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 1847–1856. ACM.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

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, Dubrovnik, Croatia. Association for Computational Linguistics.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. 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,* *Dublin, Ireland, May 22-27, 2022*, pages 1864–1874. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. Accessed: 2024-11-19.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774:1–100.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. 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 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

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 on Association for Computational Linguistics*, ACL '04, page 271–es, USA. Association for Computational Linguistics.

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 on Association for Computational Linguistics*, ACL '05, page 115–124, USA. Association for Computational Linguistics.

Nina Poerner and Hinrich Schütze. 2019. Multi-view domain adapted sentence embeddings for low-resource unsupervised duplicate question detection. 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 1630–1641, Hong Kong, China. Association for Computational Linguistics.

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*, pages 3973–3983.

Jaydeep Sen, Chuan Lei, Abdul Quamar, Fatma Özcan, Vasilis Efthymiou, Ayushi Dalmia, Greg Stager, Ashish R. Mittal, Diptikalyan Saha, and Karthik Sankaranarayanan. 2020. ATHENA++: natural language querying for complex nested SQL queries. *Proc. VLDB Endow.*, 13(11):2747–2759.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition improves language model embeddings. *CoRR*, abs/2402.15449.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *CoRR*, abs/2103.15316.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *CoRR*, abs/2302.13971:1–27.

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*, SIGIR '00, page 200–207, New York, NY, USA. Association for Computing Machinery.

Bin Wang, C.-C. Jay Kuo, and Haizhou Li. 2022a. Just rank: Rethinking evaluation with word and sentence similarities. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6060–6077, Dublin, Ireland. Association for Computational Linguistics.

Hao Wang and Yong Dou. 2023. Sncse: Contrastive learning for unsupervised sentence embedding with soft negative samples. In *International Conference on Intelligent Computing*, pages 419–431. Springer.

Huiming Wang, Zhaodonghui Li, Liying Cheng, De Wen Soh, and Lidong Bing. 2024a. 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), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 7874–7891. Association for Computational Linguistics.

Kexin Wang, Nils Reimers, and Iryna Gurevych. 2021. TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 671–688, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Improving text embeddings with large language models. In

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11897–11916, Bangkok, Thailand. Association for Computational Linguistics.

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*, SIGIR '22, page 2159–2165, New York, NY, USA. Association for Computing Machinery.

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

Bohong Wu and Hai Zhao. 2022. Sentence representation learning with generative objective rather than contrastive objective. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3356–3368, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 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, Online. Association for Computational Linguistics.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xing Wu, Chaochen Gao, Yipeng Su, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022b. Smoothed contrastive learning for unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4902–4906, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022c. ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Bo Xu, Shouang Wei, Luyi Cheng, Shizhou Huang, Hui Song, Ming Du, and Hongya Wang. 2023. Hsimcse: Improving contrastive learning of unsupervised sentence representation with adversarial hard positives and dual hard negatives. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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.

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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3916–3932. Association for Computational Linguistics.

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

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1298–1308. Association for Computational Linguistics.

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

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

Appendix

A Data Synthesis Prompts

In this section, we provide the specifics of our prompts for knowledge extraction and integration, and data synthesis. The particular prompts are presented in Figure 8 and 9.

B Reranking Tasks

To compare the ranking performance of our method on retrieval tasks, we evaluated the model using the MTEB benchmark (Muennighoff et al., 2023) with four reranking datasets: AskUbuntuDupQuestions (Lei et al., 2016), MindSmallReranking (Wu et al., 2020), SciDocsRR (Cohan et al., 2020) and StackOverflowDupQuestions (Liu et al., 2018), and follow the same settings of Zhang et al. (2023) by using Mean Average Precision (MAP) as the metric.

Table 6 presents the MAP results of our approach and related baselines on the reranking benchmark, and all models are evaluated on the test sets of the reranking benchmark without using the training sets. The results indicate that various approaches exhibit varying performance on different datasets, which can be attributed to the distinct semantic distribution and evaluation scale of each dataset. Our GCSE outperforms SynCSE by 0.39% in average MAP score and achieves the best results in all backbone models, demonstrating the efficacy of our approach in enhancing the precision of unsupervised ranking tasks.

C Visualization of synthetic sample distribution

In this section, we use the supervised SimCSE model to generate sentence embeddings for the synthesized samples and utilize t-SNE to project the vectors into two-dimensional space for a visual analysis of the diversity. To facilitate observation, we group the synthesized samples into three categories: “Rewrite” refers to positive samples synthesized using “Rewriting Prompt 1” and “Rewriting Prompt 2” from Figure 8, while “Antisense” denotes the negative samples generated using “Syntactic Antisense Prompt”. “Revision” denotes the negative samples generated using “Entity Revision Prompt”, “Quantity Revision Prompt” and “Rewriting Prompt 3”, which are related to knowledge modification. And “Source” indicates the original samples from the dataset. We randomly selected 5k “Source” samples and corresponding synthetic samples from our dataset for visualization, and the results are illustrated in Figure 10. We observe that “Rewrite” samples basically cover the spatial distribution of “Source” samples while expanding into the neighborhood space to some extent. “Antisense” and “Revision” samples further enhance the information density within the target semantic space. Comparing Figure 10 (a) and (b), it can be observed that the “Revision” samples cover areas with sparse information, while their overall spatial distribution remains consistent with the semantic distribution of ‘Source” samples. This indicates that the sample synthesis with knowledge effectively increases sample diversity within the semantic space.

D Performance on Transfer Tasks

We also evaluate our GCSE following the same settings as SimCSE on seven transfer tasks: MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), SST2 (Socher et al., 2013), TREC (Voorhees and Tice, 2000), and MRPC (Voorhees and Tice, 2000). The results are shown in Table 7, it can be observed that our GCSE (GPT-3.5 Turbo) achieves the best performance on all backbone models, outperforming second-best methods in average scores of 0.89% with BERT-base, 0.79% with BERT-large, 0.44% with RoBERTa-base, and 0.40% with RoBERTa-large, demonstrating the potential capability in downstream tasks.

E Case Studies

To further verify the improvement in our method’s awareness of entity and quantity, we selected five sample sets from the STS-Benchmark development set that explicitly contained alterations in entity or quantity within the sentence-pair, and presented the prediction cosine-similarity scores of GCSE

Knowledge Extraction Prompt

Instruction: Please help me extract the theme category, subject part, action part, state part, subject count, and entities from the following text at different granularity levels (e.g., "a man on skis" can be extracted as both "a man on skis" and "a man") along with their corresponding types. Output the results in JSON format.

Example:
Input: *"A man playing with a black dog on a white blanket."*
Output:

```
{{
   cls: 'leisure activity',
   suject: [{{text: "A man", type: "person", quantity: 1}}],
   action: [{{text: "playing with a black dog"}}]
   state: [{{text: "on a white blanket"}}]
   entities: [
        {{"entity": "a man on skis", "type": "person"}},
        {{"entity": "a man", "type": "person"}}
   ]
}}
```

Input Text: {input_text}

Output:

Rewriting Prompt 1

Instruction: Act as a {role} and rewrite the following sentence while matching the original text length precisely.

Input: {input_text}

Output Format (JSON): {{\"text\": \"\"}}

Output:

Rewriting Prompt 2

Instruction: Summarize and condense the following sentence while preserving its original meaning.

Input: {input_text}

Output Format (JSON): {{\"text\": \"\"}}

Output:

Rewriting Prompt 3

Instruction: As a skilled storyteller, please invent a plausible and engaging context or sentence that could naturally lead into the following statement.

Input: {input_text}

Output Format (JSON): {{\"text\": \"\"}}

Output:

Figure 8: Examples of prompts used for data synthesis (Part 1).

(ChatGLM3-6B) and related methodologies with the backbone of BERT-base in Table 8. We can observe from the results that the prediction score of our model achieves the minimum root-mean-square (RMS) error compared to the label in most cases, which indicates that our model has a stronger capacity to distinguish information.

F Ablation Studies of Gaussian-decayed and Few-shot Samples

We employ the Gaussian-decayed function on SynCSE and sample SynCSE training data with a sample size the same as our synthetic data to evaluate the efficacy of the proposed Gaussian-decayed function and our domain-oriented selection strategy in the ablation experiment. The data sample size is 64k, and the weight of σ in $G(\cdot)$ is assigned the same value as specified in Section 4.1. The results of various policies implemented in SynCSE are presented in Table 9. "w sampled" denotes the utilization of purely the sampled data in SynCSE, and a performance decrease can be observed when training on a reduced number of samples without extra configurations. "w sampled & G.D." denotes

Rewriting Prompt 4

Instruction: Fill in the bracketed [] sections of this news-themed sentence (one word per []), then evaluate and polish if the result sounds unnatural. If already fluent, keep the original filled version.

Evaluation Criteria:
1. Each [] = exactly one English word
2. News-style vocabulary preferred
3. Polish if:
 - Grammar/syntax errors exist
 - Logical inconsistencies appear
 - News tone is violated

Example:
Input: *"I [] you."*
Output:
{{
 "ori": "I love you.",
 "polish": "I love you."
}}

Input: {input_text}
Output:

Syntactic Antisense Prompt 1

Instruction: Dispute the following statement in {tone_styles} while matching the original text length precisely.

Input: {input_text}

Output Format (JSON): {{\"text\": \"\"}}

Output:

Syntactic Antisense Prompt 2

Instruction: Provide a negative reformulation of this statement that concisely contradicts the original while matching the original text length precisely.

Input: {input_text}

Output Format (JSON): {{\"text\": \"\"}}

Output:

Entity Revision Prompt

Instruction: Replace the phrase {ori_entities} with {replace_entities} and rewrite the entire sentence with proper grammatical adjustments.

Input: {input_text}
Output Format (JSON): {{"text": ""}}
Output:

Quantity Revision Prompt

Instruction: Modify the subject phrase "{subject_text}" by changing its article/quantifier to specify: {new_number}.

Input: {input_text}
Output Format (JSON): {{"text": ""}}
Output:

Figure 9: Examples of prompts used for data synthesis (Part 2).

the additional incorporation of $G(\cdot)$ based on "w sampled". "w G.D." indicates the results by training on the full dataset utilizing $G(\cdot)$. In both configurations, the average performance outperforms the vanilla model, illustrating the module's efficacy. "w sampled & domain & G.D." denotes the concurrent utilization of sample data, domain data, and $G(\cdot)$, with a sample size of 48k for the SynCSE dataset and 16k for the synthesized domain dataset. The results reveal that "w sampled & domain & G.D." attains the second-best performance, suggesting that incorporating domain data can decrease the required training samples while enhancing model efficacy.

Model	Method	AskU.	Mindsmall	SciDocsRR	StackO.	Avg.
BERT-base	SimCSE	51.89	28.68	67.88	<u>39.60</u>	47.01
	PCL	52.46	28.72	68.03	**41.30**	<u>47.63</u>
	SynCSE (GPT-3.5 Turbo)*	<u>52.61</u>	**29.17**	<u>68.46</u>	38.60	47.21
	GCSE (ChatGLM3-6B)	**52.62**	<u>28.79</u>	**70.67**	39.53	**47.90**
BERT-large	SimCSE	53.10	<u>29.59</u>	<u>71.94</u>	<u>40.68</u>	<u>48.83</u>
	PCL	52.03	29.11	70.30	**42.33**	48.44
	SynCSE (GPT-3.5 Turbo)*	<u>53.24</u>	**30.09**	71.45	39.24	48.50
	GCSE (ChatGLM3-6B)	**53.40**	29.43	**73.04**	39.68	**48.89**
RoBERTa-base	SimCSE††	52.78	<u>29.91</u>	65.96	39.25	46.95
	CARDS††	52.94	27.92	64.62	**41.51**	46.75
	PCL††	51.85	27.92	64.70	<u>41.18</u>	46.41
	SynCSE (GPT-3.5 Turbo)††	<u>53.27</u>	**30.29**	<u>67.55</u>	39.39	<u>47.63</u>
	GCSE (ChatGLM3-6B)	**53.44**	29.35	**67.89**	41.13	**47.95**
RoBERTa-large	SimCSE††	<u>55.10</u>	29.23	68.54	<u>42.56</u>	48.86
	CARDS††	53.83	29.07	68.26	**43.24**	48.60
	PCL††	53.43	28.56	66.06	41.54	47.40
	SynCSE (GPT-3.5 Turbo)††	**55.48**	<u>30.27</u>	<u>70.85</u>	40.00	<u>49.15</u>
	GCSE (ChatGLM3-6B)	54.05	**30.30**	**71.23**	41.65	**49.31**

Table 6: Comparison of Mean Average Precision (MAP) results on reranking tasks, where the value highlighted in bold is the best value, and the value underlined is the second-best value. "††": results from Zhang et al. (2023). "*": we reproduce the results with the officially released corpus from Zhang et al. (2023).

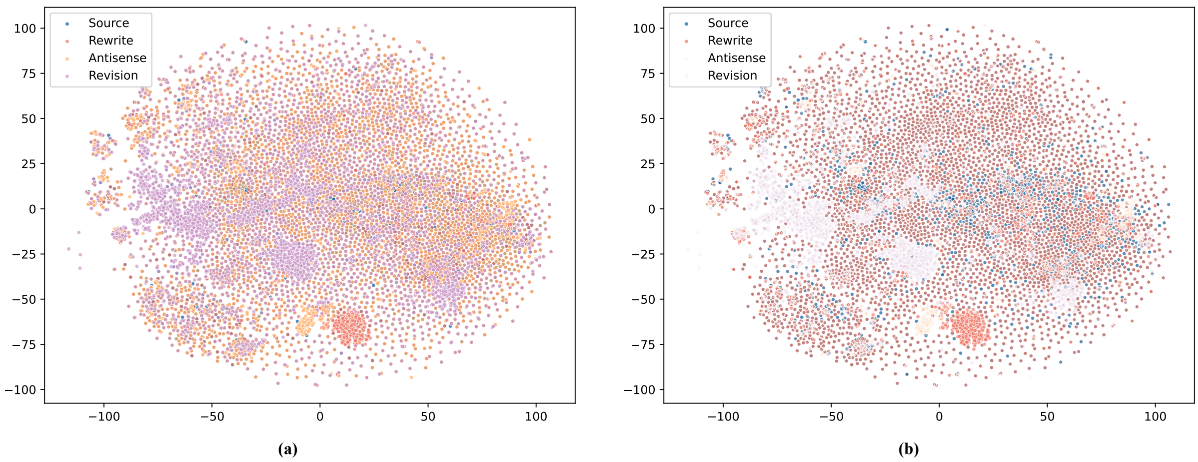


Figure 10: t-SNE visualization of the synthetic sample generated by ChatGLM3-6B, where the transparency of "Antisense" and "Revision" samples in subgraph (b) is reduced to 10% for better observation.

G Unsupervised Sentence Embedding on LLM

In this section, we utilize contrastive learning on multiple LLMs to evaluate the alignment of LLM-generated similarities with the gold labels and the effectiveness of our data augmentation strategy. We use Llama3.2-3B-Instruct (Dubey et al., 2024), Llama3-8B-Instruct (Dubey et al., 2024), ChatGLM3-6B (GLM et al., 2024), GLM4-9B-Chat (GLM et al., 2024) and Qwen2.5-14B-Instruct (Yang et al., 2024b,a) with a low-rank adapter (LoRA) layer for training. The sentence embedding vectors are obtained from the output hidden states of the last position, which is followed by the method of pretended chain of thought (Pretended CoT) (Zhang et al., 2024). We may derive two major conclusions from the results in Table 10: (1) In conventional unsupervised settings, decoder-based LLMs have no significant performance advantage over encoder-based PLMs for sentence representation learning tasks. The model performance does not increase significantly with the increase of the number of model parameters. To reduce expenses, we assert that fully leveraging the capabilities of LLMs for distilling smaller models is the better op-

Model	Method	MR	CR	SUBJ	MPQA	SST2	TREC	MRPC	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
	PCL♠	72.84	83.81	76.52	83.06	79.32	80.01	73.38	78.42
	RankCSE♠	75.66	86.27	77.81	84.74	81.10	81.80	75.13	80.36
	MultiCSR (GPT-3.5 Turbo)♣	82.70	88.15	94.97	90.08	86.87	**87.70**	75.46	86.56
	SynCSE (GPT-3.5 Turbo)*	83.34	88.80	93.88	90.39	88.96	83.60	75.94	86.42
	GCSE (ChatGLM3-6B)	**84.79**	90.03	94.35	89.92	88.37	85.60	75.71	86.97
	GCSE (GLM4-9B-Chat)	84.53	89.96	95.01	89.97	88.67	86.21	76.01	87.19
	GCSE (Qwen2.5-32B-Instruct)	83.94	89.65	94.71	90.31	88.25	86.00	**76.46**	87.05
	GCSE (GPT-3.5 Turbo)	84.71	**90.18**	94.32	90.61	**89.53**	86.09	76.22	87.38
	GCSE (Deepseek-V3-0324)	84.66	90.07	**95.02**	**90.62**	89.16	86.37	76.28	**87.45**
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
	RankCSE♠	75.48	86.50	78.60	85.45	81.09	81.58	75.53	80.60
	SynCSE (GPT-3.5 Turbo)*	85.78	90.47	94.77	90.41	90.50	89.00	75.77	88.10
	GCSE (ChatGLM3-6B)	86.08	90.54	95.00	90.63	91.21	89.60	75.71	88.40
	GCSE (GLM4-9B-Chat)	86.33	90.26	95.08	**90.65**	92.13	**92.08**	75.63	**88.88**
	GCSE (Qwen2.5-32B-Instruct)	86.35	90.73	**95.18**	90.60	91.93	87.80	76.12	88.39
	GCSE (GPT-3.5 Turbo)	85.77	**90.88**	94.35	90.09	**92.91**	88.91	75.12	88.29
	GCSE (Deepseek-V3-0324)	**86.46**	90.46	95.06	90.49	91.93	87.80	**76.87**	88.44
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
	PCL♠	71.13	82.38	75.40	83.07	81.98	81.63	69.72	77.90
	RankCSE♠	73.20	85.95	77.17	84.82	82.58	83.08	71.88	79.81
	MultiCSR (GPT-3.5 Turbo)♣	84.70	90.69	94.40	89.38	89.42	**89.62**	**77.01**	87.89
	SynCSE (GPT-3.5 Turbo)††	85.47	91.44	92.53	89.67	90.94	81.60	76.06	86.82
	GCSE (ChatGLM3-6B)	86.79	92.03	94.35	89.92	92.37	85.60	75.71	88.11
	GCSE (GLM4-9B-Chat)	**86.91**	**92.14**	**94.62**	89.76	**92.60**	86.21	76.17	**88.34**
	GCSE (Qwen2.5-32B-Instruct)	86.32	91.58	94.37	90.04	92.42	84.00	76.12	87.84
	GCSE (GPT-3.5 Turbo)	86.66	91.57	94.44	**90.82**	92.45	84.93	76.18	88.15
	GCSE (Deepseek-V3-0324)	86.42	91.56	94.41	89.23	92.18	87.52	76.13	88.21
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
	RankCSE♠	73.20	85.83	78.00	85.63	82.67	84.19	73.64	80.45
	SynCSE (GPT-3.5 Turbo)††	87.24	92.16	93.75	90.81	91.87	84.00	**76.29**	88.02
	GCSE (ChatGLM3-6B)	87.60	**92.43**	94.66	90.36	**92.37**	88.80	75.30	88.79
	GCSE (GLM4-9B-Chat)	85.55	90.39	94.70	90.37	90.32	**92.65**	73.19	88.17
	GCSE (Qwen2.5-32B-Instruct)	**87.73**	92.18	**94.72**	90.68	92.26	90.00	74.20	**88.82**
	GCSE (GPT-3.5 Turbo)	87.12	91.98	94.01	90.71	92.25	88.75	74.55	88.48
	GCSE (Deepseek-V3-0324)	**87.73**	92.18	94.29	**90.95**	92.15	88.80	73.28	88.48

Table 7: Comparison of different sentence embedding models accuracy on transfer tasks. "♠": results from Liu et al. (2023), "♣": results from Wang et al. (2024a), "††": results from Zhang et al. (2023). "*": we reproduce the results with the officially released corpus from Zhang et al. (2023).

Premise	Hypothesis	Gold	SimCSE	RankCSE	SynCSE	GCSE
A woman is cooking eggs.	A woman is cooking something.	3.00	4.37 (1.372)	4.23 (1.320)	3.66 (0.662)	**3.24 (0.236)**
Two little girls are talking on the phone.	A little girl is walking down the street.	0.50	3.38 (2.881)	3.64 (3.139)	1.97 (1.468)	**1.85 (1.351)**
A chef is preparing some food.	A chef prepared a meal.	4.00	**4.27 (0.270)**	4.59 (0.588)	4.56 (0.561)	4.41 (0.408)
Five kittens are eating out of five dishes.	Kittens are eating food on trays.	2.75	3.81 (1.056)	3.71 (0.957)	3.28 (0.535)	**3.12 (0.373)**
A woman is cutting some herbs.	A woman is chopping cilantro.	2.80	3.58 (0.777)	3.58 (0.967)	3.11 (0.313)	**2.61 (0.185)**

Table 8: Case studies on model prediction similarity with gold labels in the STS-Benchmark development set, where Gold represents the label score of the sentence pair (ranging from zero to five). The similarity scores of all models are multiplied by a coefficient of five for better comparison, and the value in parentheses denotes the RMS error between the predicted score and the label. Words highlighted in blue denote the entity alteration in the sentence-pair, whereas words in yellow indicate the quantities that change inside the sentence-pair.

Method	STS-12	STS-13	STS-14	STS-15	STS-16	STS-B	SICK-R	Avg.
SynCSE (GPT-3.5 Turbo)*	75.86	82.19	78.71	85.63	**81.11**	82.35	**78.79**	80.66
w sampled	75.48	85.60	78.76	84.78	80.38	82.12	76.46	80.51
w sampled & G.D.	75.71	85.24	79.09	85.15	80.82	82.68	77.54	80.89
w G.D.	**75.89**	85.26	79.24	85.67	80.79	82.63	78.19	**81.10**
w sampled & domain & G.D.	75.88	**86.02**	**79.46**	**86.10**	80.27	**82.87**	76.91	81.07

Table 9: Ablation studies of sample size and the Gaussian-decayed function by utilizing SynCSE. "*": we reproduce the results with the officially released corpus from Zhang et al. (2023).

Model	Avg.	Model	Avg.
Unsupervised		*Data Augmentation*	
Llama3.2-3B-Instruct LoRA	71.34	Llama3.2-3B-Instruct LoRA	78.26
Llama-3-8B-Instruct LoRA	**72.73**	Llama-3-8B-Instruct LoRA	78.24
ChatGLM3-6B LoRA	69.38	ChatGLM3-6B LoRA	79.04
GLM4-9B-Chat LoRA	71.77	GLM4-9B-Chat LoRA	**79.52**
Qwen2.5-14B-Instruct LoRA	68.49	Qwen2.5-14B-Instruct LoRA	78.02

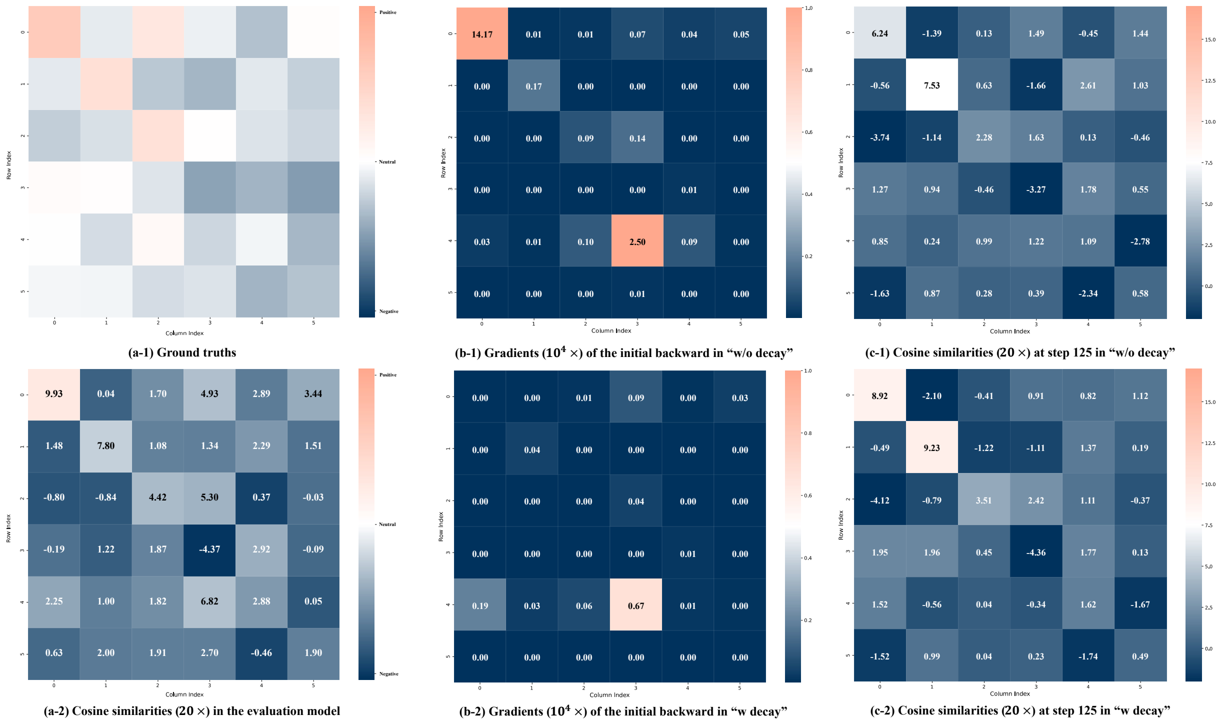
Table 10: Performance comparison of different LLMs on STS tasks, where results of "Unsupervised" refers to models trained on the same unsupervised settings as Gao et al. (2021), and "Data Augmentation" refers to models trained with the synthetic data generated by ChatGLM3-6B.

tion. (2) The application of our data augmentation technique to sentence representation learning tasks in LLMs significantly enhances performance relative to the "Unsupervised" settings, which further proves the applicability and efficacy of our strategy.

H Visualization of Prediction Scores and Gradient Comparisons

To further analyze the effectiveness of the Gaussian-decayed function in mitigating the impact of false negative noise, we visualized the changes in predicted scores and gradients during the training process using heatmaps. In the training procedure of GCSE, each input consists of a source sample, its corresponding positive sample, and a hard negative sample. We visualize the cosine similarity scores and gradient heatmaps for negative samples within a batch in Figure 11. Each cell of a heatmap represents the relationship between the source sample and the negative sample, and the diagonal cells highlight the relationships between source samples and their hard negatives. Since synthetic samples lack manual annotations, we use supervised SimCSE models (Gao et al., 2021) based on different backbones to compute their similarity scores as the ground truth. We normalized the output scores of each model with min-max scaling and averaged them as the final scores to address distributional differences across models, and the results are shown in Figure 11 (a-1). It can be observed that several hard negatives on the diagonal display scores biased towards positive similarity, indicating the presence of false negative noise. In the framework of contrastive learning, when optimized using standard contrastive loss, these hard negatives are positioned further from the source samples in the semantic space, negatively impacting the model's representational capacity. Figure 11 (a-2) displays the normalized cosine similarity scores of hard negatives in the initial step as calculated by the evaluation model in GCSE. The initial score distribution of hard negatives shows a strong correlation with the ground truth, suggesting that these scores could efficiently guide GCSE in gradient correction.

Figures 11 (b-1) and (b-2) present the backward gradient values of the model trained without and with the Gaussian-decayed function, respectively. For better visualization, all gradient values are amplified by 10^4, and all similarities are amplified by 20 by the temperature. By comparing the gradients of hard negative samples in these two figures, it can be observed that the gradient values on false hard negatives are significantly smaller when the Gaussian-decayed function is applied. Additionally, Figures 11 (c-1) and (c-2) present a comparison of cosine similarity scores after 125 training steps with and without the Gaussian-decayed function. The scores for false hard negatives are sig-



(a-1) Ground truths

(b-1) Gradients (10^4 ×) of the initial backward in "w/o decay"

(c-1) Cosine similarities (20 ×) at step 125 in "w/o decay"

(a-2) Cosine similarities (20 ×) in the evaluation model

(b-2) Gradients (10^4 ×) of the initial backward in "w decay"

(c-2) Cosine similarities (20 ×) at step 125 in "w decay"

Figure 11: Heatmap visualization of the prediction scores and gradients.

nificantly higher when the Gaussian-decayed function is employed, while the true hard negatives had lower scores. The overall score distribution aligns more accurately with the ground truth, and these results demonstrate that the Gaussian-decayed function effectively prevents false negatives from being pushed farther away from source samples in the semantic space, thereby validating its effectiveness in mitigating noise and improving model performance.

I Ablation analysis of filtering thresholds

To study the impact of different filtering thresholds, we evaluate the performance on the backbone of the BERT-base, and the results are shown in Figure 12. When $\alpha > 0.9$, the model's performance declines significantly, primarily because the high threshold filters out too many samples, heavily reducing the number of positive samples. In the range $\alpha \in [0.8, 0.9]$, performance degradation is observed due to noise introduced by false positive samples. Similarly, when $\alpha < 0.8$, the model suffers from a performance drop caused by an excessive number of false positives being included in the training process. The threshold for β demonstrates a noticeable impact on model performance when it deviates from 0.75. Specifically, when $\beta > 0.75$, the model's performance declines significantly due to the inclusion of excessive false negative noise, which severely affects the model performance. Conversely, when $\beta < 0.75$, the selected negative samples become easier for the model to distinguish, providing limited benefit for enhancing its representation learning capacity. The results highlight the influence of filtering thresholds on sample quality and distribution.

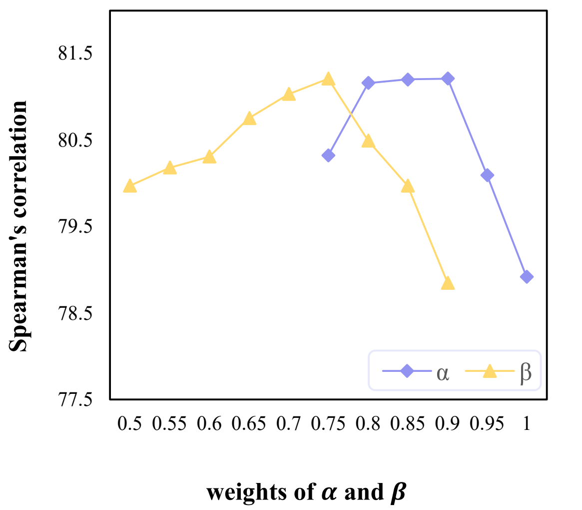


Figure 12: Spearman's correlation against the weight of α and β on the STS tasks. When adjusting the weight of one parameter, the other parameter is fixed at its default value as specified in the experimental settings.

J Score Normalization Methodology

In this work, the labels in datasets are normalized with standard min-max normalization. To address the discrepancy in score distributions among different models, we applied a variant min-max normalization method to align their predicted scores. For each label $l \in [0, \mathrm{MAX}]$, we collect all predicted scores with $l = 0$ as list C_0, and all predicted scores with $l = \mathrm{MAX}$ as list C_1. Specifically, we computed the median prediction scores for C_0 and C_1 as $min_p = \mathrm{median}(C_0)$ and $max_p = \mathrm{median}(C_1)$, respectively. The use of medians, rather than the minimum predicted score for C_0 or the maximum predicted score for C_1, avoids reliance on outlier values that may disproportionately skew the normalization, ensuring a more balanced score distribution. For a given score s, the normalized score s' is calculated as:

$$s' = \text{clip}\left(\frac{s - min_p}{max_p - min_p}, 0, 1\right), \tag{14}$$

where the function clip$(x, 0, 1)$ ensures the normalized score is bounded within $[0, 1]$. This method adjusts the score range to maintain consistency across models while preserving relative score differences.