Large Language Models can Contrastively Refine their Generation for Better Sentence Representation Learning

Huiming Wang[*1] **Zhaodonghui Li**[†2,3] **Liying Cheng**[2] **De Wen Soh**[1] **Lidong Bing**[2]

[1]Singapore University of Technology and Design

[2]DAMO Academy, Alibaba Group

[3]Nanyang Technological University, Singapore

huiming_wang@mymail.sutd.edu.sg dewen_soh@sutd.edu.sg

{zhaodonghui.li, liying.cheng, l.bing}@alibaba-inc.com

Abstract

Recently, large language models (LLMs) have emerged as a groundbreaking technology and their unparalleled text generation capabilities have sparked interest in their application to the fundamental sentence representation learning task. Existing methods have explored utilizing LLMs as data annotators to generate synthesized data for training contrastive learning based sentence embedding models such as SimCSE. However, since contrastive learning models are sensitive to the quality of sentence pairs, the effectiveness of these methods is largely influenced by the content generated from LLMs, highlighting the need for more refined generation in the context of sentence representation learning. Building upon this premise, we propose MultiCSR, a multi-level contrastive sentence representation learning framework that decomposes the process of prompting LLMs to generate a corpus for training base sentence embedding models into three stages (i.e., sentence generation, sentence pair construction, in-batch training) and refines the generated content at these three distinct stages, ensuring only high-quality sentence pairs are utilized to train a base contrastive learning model. Our extensive experiments reveal that MultiCSR enables a less advanced LLM to surpass the performance of ChatGPT, while applying it to ChatGPT achieves better state-of-the-art results. Comprehensive analyses further underscore the potential of our framework in various application scenarios and achieving better sentence representation learning with LLMs.

1 Introduction

As a fundamental task, sentence representation learning (Conneau et al., 2017; Reimers and Gurevych, 2019; Gao et al., 2021) aims to learn universal sentence embeddings that can benefit various downstream tasks, such as semantic similarity comparison (Agirre et al., 2012, 2013, 2014, 2015, 2016; Cer et al., 2017; Marelli et al., 2014), and information retrieval (Le and Mikolov, 2014; Misra et al., 2016; Thakur et al., 2021; Wang et al., 2022). Recent advancements, particularly in contrastive learning-based methods such as SimCSE (Gao et al., 2021) and its variants (Chuang et al., 2022; Zhou et al., 2022; Tan et al., 2022; Wu et al., 2022; Jiang et al., 2022; Liu et al., 2023), have demonstrated to be the most efficient and effective ones. In contrastive learning, the quality of sentence pairs has a large impact on the overall performance (Chen et al., 2022). In particular, supervised contrastive learning methods trained on natural language inference (NLI) datasets (Bowman et al., 2015; Williams et al., 2018) can outperform their unsupervised versions by a large margin (Gao et al., 2021). However, obtaining large amounts of high-quality sentence pairs can be costly in both time and resources, particularly considering that various application domains can only be better handled with domain-specific training data.

The recent emergence of large language models (LLMs), such as the Flan series (Chung et al., 2022), LLaMA (Touvron et al., 2023) and Chat-

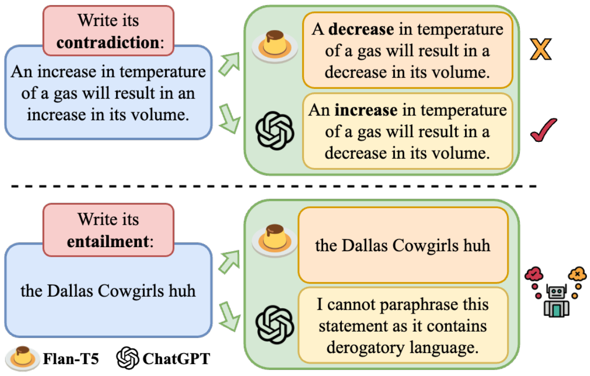

Figure 1: Two example sentences with the generations from Flan-T5 and ChatGPT given different instructions.

[*]Work done while Huiming Wang was an intern at DAMO Academy, Alibaba Group.

[†]Zhaodonghui Li is under the Joint PhD Program between Alibaba and Nanyang Technological University.

GPT (OpenAI, 2022), has brought a paradigm shift in natural language processing due to their impressive performance. Consequently, there has been a growing interest in harnessing the power of LLM for sentence representation learning. Cheng et al. (2023) proposed to directly measure the semantic similarities of sentence pairs with LLMs for a base model to imitate. However, this imitation also places great demands on the LLM's understanding of semantics and constrains its application in a wider range of LLMs. Instead of utilizing LLMs for semantic similarity scoring, a recent line of work has explored leveraging LLMs to generate sentence pairs in the NLI style with provided premises (Schick and Schütze, 2021; Zhang et al., 2023) and demonstrates SOTA performance.

Despite their exciting results, these methods heavily depend on the quality of generated content from LLMs, leading a huge performance gap between different LLMs. Moreover, concerns about the accuracy and quality of the generated content from LLMs remain unsolved and have drawn significant attention from the community (Zheng et al., 2023; Shi et al., 2023), which are more pronounced with relatively less advanced LLMs. For example, in Figure 1, contradicting the first sentence requires a clear understanding towards its meaning and only ChatGPT successfully produce its contradiction. When faced with the second provided sentence, both ChatGPT and Flan-T5 encounter difficulties due to the limited information provided. Consequently, Flan-T5 only repeats the premise while ChatGPT entirely gives up the generation. In a task that is sensitive to the quality of sentences like sentence representation learning, there highlights a need for a more robust framework that can automatically refine the outputs of LLMs for better contrastive sentence representation learning.

Motivated by observations above, in this work, we propose **Multi-Level Contrastive Sentence Representation Learning (MultiCSR)**, a novel three-stage framework that contrastively refines the generations of LLMs at distinct stages: sentence generation, sentence pair construction and in-batch training, while fine-tuning a contrastive learning model like SimCSE. Concretely, while generating a sentence given a specific instruction like "Write its entailment:", we also deploy the noisy variants of this instruction to identify obvious error that LLMs will make during generating the outputs following this instruction. In a contrastive sentence generation procedure, next-token prediction logits will be systematically deviated by comparing the logits derived from the original instructions with those from the noisy instructions. This process detects and avoids the obvious error tendencies of LLMs, and refines their generations to align more closely with the intended instruction, rather than providing an even opposite generation, such as the first generation of Flan-T5 in Figure 1.

Despite the effectiveness of contrastive generation strategy, directly training a base sentence embedding model on the generated corpus gives poor results in our experiments, both because of the inevitably noisy generations, and the overlooked relations between sentences (i.e., sentence pairs), since a contrastive learning loss works essentially by modeling the relations between sentences while pulling the positive pairs closer and pushing the negative pairs apart. To take the relations between sentences into consideration, in sentence pair construction stage, we show that LLMs can be utilized to self-curate the set of their newly generated sentences by measuring the semantic similarities of sentence pairs, ensuring that only sentence pairs in highest quality are included into the final training stage. To further prevent the *false-negative* issue (Zhou et al., 2022) raised during the in-batch training, where randomly selected negative samples are indeed semantically similar to the original sentence, we utilize a pre-trained sentence representation model to provide the similarity mask and contrastively filter *false negatives* during training.

In summary, our contributions include: (1) We propose a new and promising direction to improve sentence representation learning by refining the generated content of LLMs. (2) We for the first time decompose the process of prompting LLMs to generate a corpus for training base sentence embedding models into three stages (i.e., sentence generation, sentence pair construction, in-batch training), and integrate the idea of contrast into each stage for refinement. (3) We conduct extensive experiments on standard semantic textual similarity (STS) tasks (Agirre et al., 2012, 2013, 2014, 2015, 2016; Cer et al., 2017; Marelli et al., 2014) and several transfer tasks (Conneau and Kiela, 2018) with two representative LLMs (i.e., Flan-T5 and ChatGPT). We further perform a comprehensive analysis of the behavior of MultiCSR, demonstrating its effectiveness from various perspectives. We hope that, our proposed method provides additional insights into achieving high-quality sentence representation learning corpus by refining LLMs' generations.

2 Related Work

2.1 Sentence Representation Learning with LLMs

There has been a recent exploration in utilizing LLMs for sentence representation learning. Cheng et al. (2023) prompted LLMs to measure the semantic similarities of sentence pairs, and fine-tuned base models to "immitate" these judgements of LLMs with mean squared error. However, this method also places great demands on the model's understanding of semantics. Thus, we conduct a pilot experiment to see whether LLM's generated similarities are well-aligned with the ground-truth labels and include the results in Table 1. We can see from the results that even ChatGPT equipped with ICL can not outperform those contrastive learning methods with only base models.

Instead of treating LLMs as evaluators, a recent line of work leverages LLMs as data generators, with the generated entailment and contradiction hypotheses being used to train a contrastive learning method (Schick and Schütze, 2021; Zhang et al., 2023). The success of these works is based on the fact that models trained on NLI datasets demonstrate superior performance (Conneau et al., 2017; Reimers and Gurevych, 2019; Gao et al., 2021; Chen et al., 2022). Nevertheless, the performance of these methods will heavily rely on the quality of generated content, which also poses a huge challenge to the instruction-following and generation capabilities of LLMs. Thus, refining the generated data before utilizing them into training is essential, which usually requires substantial effort and computational resources, highlighting the need for more efficient approaches. In this work, we introduce MultiCSR, which can automatically refine the generation of LLMs and ensures that only high-quality sentence pairs are utilized for training a contrastive learning method.

2.2 Contrast in Text Generation

Recently, with the rapid development of LLMs, the idea of contrast for improving text generation has been studied in various settings (Welleck et al., 2019; Liu et al., 2021; Li et al., 2023; Yona et al., 2023; Kim et al., 2024). Among them, contrastive decoding (Li et al., 2023) studies maximizing the next-token probability by contrasting the predictions from a high-performing "expert" model against those from a less accurate "amateur" model. Despite its effectiveness, the need for at least two models of different scales limits their applications in broader scenarios. In this work, inspired by instructive decoding (Kim et al., 2024) which places emphasis on the potential of instructions in the input text, we show that, within the contrastive generation procedure, the instructions used for generating the opposite hypotheses can be extremely effective in identifying the obvious error tendencies of LLMs for further refining their generations in the context of sentence representation learning.

Model	Avg.
$\text{SimCSE}_{\text{RoBERTa}}$	76.57
PromptRoBERTa	**79.15**
Flan-T5-XL	63.24
Flan-T5-XL w/ ICL	68.76
ChatGPT	71.58
ChatGPT w/ ICL	76.19

Table 1: Performance comparison of different models and directly utilizing LLMs w/o and w/ in-context learning (ICL) to measure the similarities on STS tasks.

3 Methodology

In this section, we present MultiCSR, a framework designed to enhance the quality of generated content of LLMs. By contrastively refining their generations at distinct stages of training a contrastive learning method, MultiCSR ensures that only high-quality sentence pairs are utilized in the final training stage, achieving a better sentence representation learning with LLMs. The whole workflow can be visualized as Figure 2.

3.1 Stage 1: Contrastive Generation

During the generation procedure, when presented with a concatenation of an instruction I and an input sequence $\boldsymbol{x}$, the objective of an LLM C_θ is to generate the corresponding output sequence $\boldsymbol{y} = [y_1, ..., y_n]$. For each token y_t, C_θ will compute its logits as $l_t = C_\theta(I, \boldsymbol{x}, \boldsymbol{y}_{<t})$. The probability of output sequence $\boldsymbol{y}$ can be expressed as:

$$p_\theta(\boldsymbol{y}|I, \boldsymbol{x}) = \prod_{t=1}^{n} p_\theta(y_t|I, \boldsymbol{x}, \boldsymbol{y}_{<t}), \qquad (1)$$

where $p_\theta(y_t|I, \boldsymbol{x}, \boldsymbol{y}_{<t})$ represents the normalized probability of the sampled (e.g., greedy sampled) next token y_t derived from the softmax over l_t. Within this process, more refined generations can be achieved by ensuring that the model's generation essentially aligns with the given instructions.

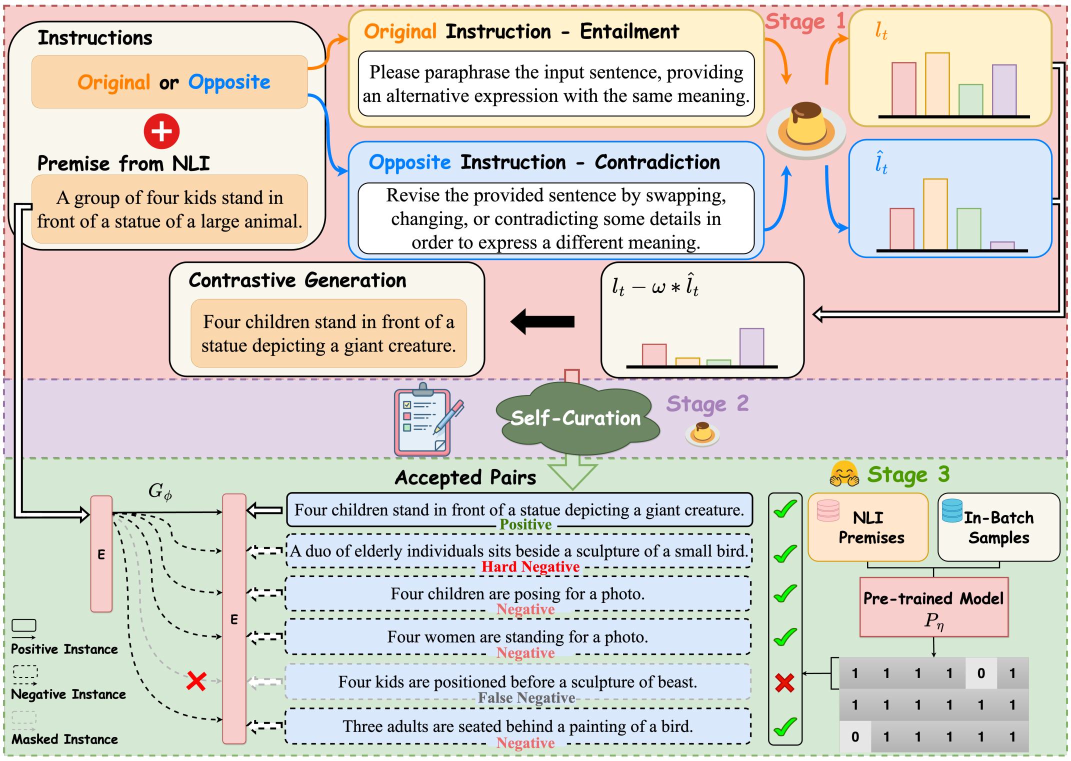

Figure 2: Overview of our three-stage framework MultiCSR. **Stage 1: Contrastive Generation.** We refine each token's logits with the opposite instruction to align more closely with the intended instruction. **Stage 2: Contrastive Sentence Pair Construction.** By prompting LLMs to evaluate the semantic similarities of generated sentence pairs, we ensure that only sentence pairs satisfying the pre-defined rules are left to form a curated set. **Stage 3: Contrastive In-Batch Training.** We leverage the similarity mask provided by a pre-trained sentence representation model to contrastively filter *false negatives* during the in-batch training.

Thus, it is better to understand what kind of noisy generations C_θ will produce when following the instructions, and we can realize better alignment by eliminating these trending noises from the next-token distribution. As inspired by Kim et al. (2024) which indicates that specific noisy variants of the original instruction can help induce the corresponding undesired behaviors of LLMs, we designed and conducted analysis towards several noisy instructions $\hat{I}$ in the context of hypothesis generation. We can acquire the noisy next-token logits as:

$$\hat{l}_t \leftarrow C_\theta(\hat{I}, \boldsymbol{x}, \boldsymbol{y}_{<t}). \tag{2}$$

By comparing the logits from the original instructions and those from the noisy variants, we can detect and correct noises for C_θ, achieving a more refined generation. During our contrastive generation, the logits l_t will be contrasted with $\hat{l}_t$, and the next-token y_t will be sampled from the probability distribution of the final logits $l_t - \omega * \hat{l}_t$.

In the context of sentence representation learning, for each premise $\boldsymbol{x}$ of NLI, we will generate its corresponding *entailment* $\boldsymbol{x}_+$ and *contradiction* $\boldsymbol{x}_-$ hypotheses. Through the experiments of all noisy variants, we find that the instructions used to generate the opposite hypotheses demonstrate to be the most effective. Thus, when mentioning noisy instructions, we specifically refer to these opposite instructions throughout our paper. For example, as shown in Figure 2, while generating the *entailment* hypothesis of $\boldsymbol{x}$, we leverage the instruction to generate *contradiction* as the noisy instruction $\hat{I}$. We include the detailed instructions in Appendix A.

3.2 Stage 2: Contrastive Sentence Pair Construction with Self-Curation

Although the generated sentences of C_θ are refined, the relations within sentence pairs remain uncertain. Since contrastive learning methods are modeling the distances of sentence pairs in the embedding space, it is particularly important to ensure that the sentence pairs are in high quality and the distances within the sentence pairs are suitable for effectively training a base model G_ϕ with contrastive learning. Thus, during the sentence pair construction stage, we perform self-curation and select high quality sentence pairs using C_θ itself.

Given a generated triplet $(\boldsymbol{x}, \boldsymbol{x}_+, \boldsymbol{x}_-)$, we follow

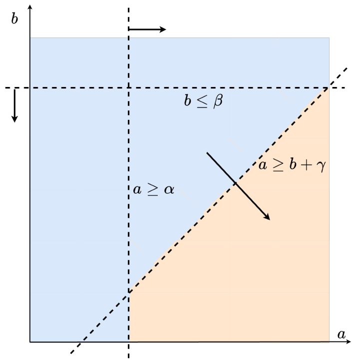

Figure 3: One example of our self-curation strategy. During the final training stage, we only use the sentence pairs in orange region.

the same generation procedure of Equation 1 and utilize C_θ to assign the semantic similarity scores for $(\boldsymbol{x}, \boldsymbol{x}_+)$ and $(\boldsymbol{x}, \boldsymbol{x}_-)$ separately with the same instruction prompt $\boldsymbol{d}$ (more details in Appendix A):

$$a \leftarrow C_\theta([\boldsymbol{x}; \boldsymbol{x}_+; \boldsymbol{d}]), \; b \leftarrow C_\theta([\boldsymbol{x}; \boldsymbol{x}_-; \boldsymbol{d}]). \quad (3)$$

The bounds of a and b are specified in $\boldsymbol{d}$ as $[0, 5]$, where a score of 5 means the semantics of a sentence pair are completely the same and 0 means these two sentences are totally different.

Based on the semantic similarity scores of two sentence pairs, we can then select a subset of triplets $(\boldsymbol{x}, \boldsymbol{x}_+, \boldsymbol{x}_-)$ with a pre-defined strategy to form the curated training corpus. One example is shown in Figure 3 and formalized as follows:

$$(\boldsymbol{x}, \boldsymbol{x}_+, \boldsymbol{x}_-) \in \mathcal{T}, \text{ if } \begin{cases} a \geq \alpha \\ b \leq \beta \\ a \geq b + \gamma \end{cases}, \quad (4)$$

where $\mathcal{T}$ is the curated training corpus, α and β are the thresholds to control the absolute similarities of positive pair $(\boldsymbol{x}, \boldsymbol{x}_+)$ and negative pair $(\boldsymbol{x}, \boldsymbol{x}_-)$ respectively, and γ will serve as the margin which represents the relative similarity distance of $(\boldsymbol{x}_+, \boldsymbol{x}_-)$. By performing self-curation as a contrastive sentence pair construction procedure, we get a lightweight but high-quality training corpus.

3.3 Stage 3: Contrastive In-Batch Training

With the curated corpus, we can fine-tune a base model G_ϕ to learn better sentence representations. To introduce enough challenge to the model training, we follow Gao et al. (2021) and take $(\boldsymbol{x}, \boldsymbol{x}_+)$ as a positive pair and $(\boldsymbol{x}, \boldsymbol{x}_-)$ as a *hard negative* pair, and the entailment and contradiction hypotheses of other premises inside a batch of size N will be treated as other in-batch negatives of $\boldsymbol{x}$ as $(\boldsymbol{x}, \boldsymbol{x}_+^k)$ and $(\boldsymbol{x}, \boldsymbol{x}_-^k)$, where $\boldsymbol{x}^k \neq \boldsymbol{x}$. For simplicity, we denote the representations $G_\phi(\boldsymbol{x})$, $G_\phi(\boldsymbol{x}_+)$ and $G_\phi(\boldsymbol{x}_-)$ as h, h_+ and h_-, respectively. Then the training objective ℓ is defined as:

$$-\log \frac{e^{\mathrm{sim}(h,h_+)/\tau}}{\sum_{k=1}^{N}\left(e^{\mathrm{sim}(h,h_+^k)/\tau} + e^{\mathrm{sim}(h,h_-^k)/\tau}\right)}, \quad (5)$$

where τ is temperature parameter and $\mathrm{sim}(,)$ is the similarity of two sentence embeddings from G_ϕ.

As introduced, the above in-batch negatives from $(\boldsymbol{x}, \boldsymbol{x}_+^k)$ and $(\boldsymbol{x}, \boldsymbol{x}_-^k)$ may involve *false negatives*, where $\boldsymbol{x}_+^k$ or $\boldsymbol{x}_-^k$ can be indeed semantically similar with $\boldsymbol{x}$ due to the random in-batch sampling. As an example, in Figure 2, the entailment hypothesis "Four kids are positioned before a sculpture of beast" has a high semantic similarity with the first premise, so we treat this pair $(\boldsymbol{x}, \boldsymbol{x}_+^k)$ as *false negative* and need to mitigate its effects during training.

To alleviate this problem, we incorporate the pre-trained sentence representation model P_η to provide a weighted mask for (h, h_+^k) and (h, h_-^k), as shown in Figure 2. We denote the similarity given by pre-trained model P_η as $\mathrm{sim}_\eta(,)$. A mask indicator $M_{\boldsymbol{x},\boldsymbol{x}^k,\cdot}$ is defined as:

$$M_{\boldsymbol{x},\boldsymbol{x}^k,\cdot} = \begin{cases} 0, \; \mathrm{sim}_\eta(h, h_\cdot^k) \geq \sigma, \; \boldsymbol{x}^k \neq \boldsymbol{x} \\ 1, \; \text{else} \end{cases}, \quad (6)$$

where σ is the threshold. In this way, $(\boldsymbol{x}, \boldsymbol{x}_\cdot^k)$ with higher semantic similarity than σ will be masked out during the in-batch training stage. We then use the following training objective to fine-tune our base model G_ϕ:

$$-\log \frac{e^{\mathrm{sim}(h,h_+)/\tau}}{\sum_{k=1}^{N}\sum_{i\in\{+,-\}}\left(M_{\boldsymbol{x},\boldsymbol{x}^k,i}\, e^{\mathrm{sim}(h,h_i^k)/\tau}\right)}. \quad (7)$$

4 Experiment

4.1 Experiment Setup

We evaluate our approach on standard semantic textual similarity (STS) tasks and seven transfer tasks in SentEval[1]. We further evaluate our method on zero-shot information retrieval tasks in BEIR (Thakur et al., 2021). Following the settings of Gao et al. (2021), we use Spearman's correlation to measure the performance of different approaches. We choose BERT$_{base}$ (Devlin et al., 2019) and RoBERTa$_{base}$ (Liu et al., 2019) as our

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

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
				BERT-base				
BERT-whitening♡	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
ConSERT♡	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE♡	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
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
InfoCSE♢	70.53	84.59	76.40	85.10	81.95	82.00	71.37	78.85
RankEncoder♠	74.88	85.59	78.61	83.50	80.56	81.55	75.78	80.07
RankCSE♣	**75.66**	**86.27**	77.81	84.74	81.10	81.80	75.13	80.36
MultiCSR (Flan-T5)	73.30	83.40	77.63	84.69	80.78	82.81	78.23	80.12
SynCSE (ChatGPT)*	72.71	84.79	77.96	**85.50**	**82.16**	83.16	76.08	80.34
MultiCSR (ChatGPT)	74.96	84.29	**79.56**	84.80	80.44	**83.69**	**79.47**	**81.03**
				RoBERTa-base				
RoBERTa-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
RankCSE♣	73.20	**85.95**	77.17	84.82	**82.58**	83.08	71.88	79.81
MultiCSR (Flan-T5)	73.75	84.61	79.32	**85.84**	81.60	83.64	78.33	81.02
SynCSE (ChatGPT)*	**78.08**	79.27	78.25	85.77	81.72	82.88	79.18	80.74
MultiCSR (ChatGPT)	75.71	84.43	**80.20**	85.08	82.23	**84.64**	**79.77**	**81.72**

Table 2: Performance comparison of MultiCSR on STS tasks. We implement our frameworks based on SimCSE (Gao et al., 2021). ♡: results from Jiang et al. (2022), ♣: results from Liu et al. (2023), ♢: results from Wu et al. (2022), ♠: results from Seonwoo et al. (2023). *: we remove their manual cleaning process for fair comparison, and reproduce the results of SynCSE with their officially released corpus, following our same settings.

backbone models G_ϕ. For the unlabeled sentences, we use the sentences of Wikipedia and the premises of NLI from Gao et al. (2021) as unlabeled sentences, and ensure the data volume used by MultiCSR is equivalent to that of SimCSE. For the main results in Table 2, we include only the results with NLI premises following Zhang et al. (2023). In addition, we also discuss the impact of different data resources in Section **??**. While our framework is general and could be combined with more advanced algorithms as well, we utilize SimCSE as our main backbone. In Appendix B, we include more experimental results of applying MultiCSR to different backbones.

We utilize the corresponding versions of SimCSE (i.e., unsupervised and supervised, BERT$_{base}$ and RoBERTa$_{base}$) as P_η in different settings. For LLM C_θ, we include the results of Flan-T5-XL(3B) to show that, with MultiCSR, a relatively tinier and less advanced LLM can even outperform ChatGPT, while applying to ChatGPT achieves a better state-of-the-art performance. In the experiments with ChatGPT, since their logits can not be acquired, we utilize only the last two stages of MultiCSR. We also discuss the effect of smoothing coefficient ω, self-curation thresholds α, β and γ in Appendix C and D separately, and only include the results with $\omega = 0.3$, $\alpha = 3$, $\beta = 3$ and $\gamma = 1$ in main results. We fine-tune our model with a batch size of 256, $\tau = 0.05$ and $\sigma = 0.9$, and choose the best model parameters ϕ based on the development performance from STS-Benchmark following Gao et al. (2021). We conduct ablation studies on σ in Appendix E. We include the performance on transfer tasks and BEIR tasks in Appendix F. We also evaluate MultiCSR in a supervised setting by directly combining our generated corpus with labeled NLI corpus. Because of our proposed contrastive in-batch training, the *false-negative* issue will not be raised with this direct combination. We include a more detailed analysis and our supervised performance in Appendix G.

Baselines We compare our method with many strong baselines including ConSERT (Yan et al., 2021), SimCSE (Gao et al., 2021), DiffCSE (Chuang et al., 2022), PromptBERT (Jiang et al., 2022), InfoCSE (Wu et al., 2022), RankEncoder (Seonwoo et al., 2023), RankCSE (Liu et al., 2023) and a post-processing method BERT-whitening (Su et al., 2021). To make comparison between different corpus construction methods, we further compare MultiCSR with SynCSE (Zhang et al., 2023), which directly leverages the generated sentences of ChatGPT to train SimCSE.

Method	Spearman's	Δ
MultiCSR$_{\text{RoBERTa}}$	**85.82**	0.00
w/o stage 1	84.06	-1.76
w/o stage 2	83.98	-1.84
w/o stage 3	84.79	-1.03
w/o stage 2 & 3	82.74	-3.08
w/o stage 1 & 2	79.86	-5.96
w/o stage 1 & 3	82.45	-3.37
w/o stage 1 & 2 & 3	75.51	-10.31

Table 3: Ablation studies of different components in MultiCSR (Flan-T5) based on SimCSE$_{\text{RoBERTa}}$. The results are based on the development set of STS-B. MultiCSR w/o stage 1&2&3: training SimCSE with the raw generation of Flan-T5-XL.

4.2 Main Results

From the results shown in Table 2, we have the following observations: (1) From the comparison between our approach and previous strong baselines, MultiCSR can significantly enhance the base model SimCSE and raises the averaged Spearman's correlation from 76.25% and 76.57% to 81.03% and 81.72% respectively, achieving a better state-of-the-art performance. It is also important to note that MultiCSR is general and data-oriented, and can be easily applied to various base models and improves their performance as shown in Appendix B. Although we build our method over SimCSE, MultiCSR can achieve comparable or even better results than the strong and competitive models such as RankCSE, which demonstrates the effectiveness of our method. (2) When comparing with SynCSE, which also leverages LLMs for enhancing sentence representation learning, MultiCSR shows to be more effective. By utilizing only Flan-T5-XL(3B), our approach can even outperform SynCSE with ChatGPT on RoBERTa$_{base}$, which also provides an opportunity for broader open-source but less advanced LLMs, rather than blindly pursuing larger and better LLMs (which are mostly closed-source). In addition, MultiCSR can still be valid when applied to ChatGPT, which demonstrates our claim that a curated training corpus is necessary for effectively training a contrastive learning method.

5 Analysis

5.1 Ablation Studies

We investigate the impact of each component in MultiCSR and report the performance in Table 3.

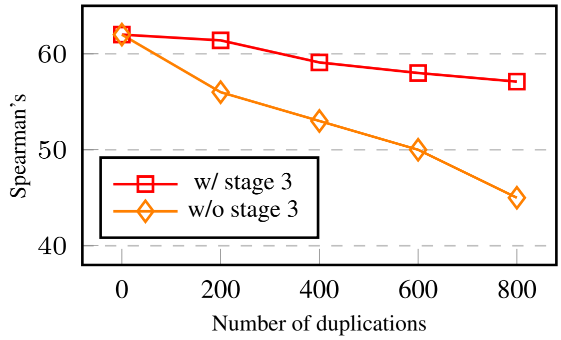

Figure 4: Performance comparison between with and without our constrastive in-batch training stage, with the number of duplications and Spearman's scores on the development set of STS-B reported.

The results of MultiCSR w/o stage 1&2&3 demonstrates our motivation that the raw generated content from LLMs may not satisfy the requirement of training a contrastive learning method. The comparison between this result and those of removing two stages shows that each component plays a crucial role in enhancing the base model.

Among them, since stage 1 and stage 2 are strictly controlling the quality of each generation or each sentence pair, they seem to be more important than stage 3. However, they are in charge of different stages of training a contrastive learning method like SimCSE. In a training batch with N sentence pairs, during the data generation (stage 1) and sentence pair construction (stage 2), they only check whether $2N$ sentence pairs are qualified. But for stage 3, it will consider whether $2N(N-1)$ sentence pairs involve *false negatives* or not. Moreover, the control of stage 3 will not be too strict for a high-quality dataset like NLI where most premises are not similar to each other as shown in the statistical results in Appendix G. For a dataset where only limited data are available, whether using stage 3 or not will result in a huge performance gap. To demonstrate this, we sample a sentence pair from $10K$ ground truth sentence pairs of NLI, and duplicate it for certain times and combine this set with other pairs together to train a SimCSE model. The results are shown in Figure 4. It is important to note that all the sentence pairs are in good condition and will survive after stage 1&2. In this scenario, only our proposed contrastive in-batch training stage can be helpful. In our supervised setting, by utilizing this in-batch masking, we can directly combine the generated corpus with the ground truth NLI dataset as the training corpus, without suffering from the *false-negative* issue, as shown in Appendix G.

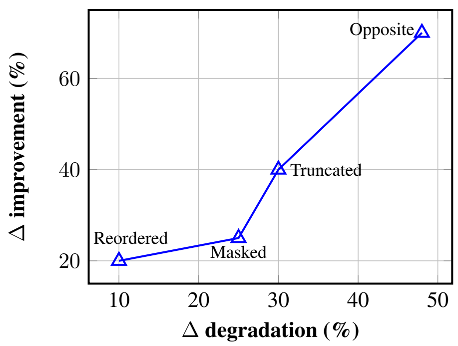

Figure 5: Correlation between performance degradation with noisy instructions and improvement by incorporating them into our contrastive generation stage.

Through the analysis of our ablation studies, each component is necessary for MultiCSR to ensure that a contrastive learning method is only trained on a high-quality and effective corpus for better sentence representation learning.

5.2 Analysis of Noisy Instructions $\hat{I}$

In Section 3.1, we introduce the details of how to refine the generation of LLMs with the help of noisy logits produced with noisy instructions. Through noisy instructions, we want to identify which kinds of mistakes LLMs will make while following the original instructions, and eliminate these noises from the next-token prediction distribution. Thus, we employ the following noisy instruction variants. (1) **Word reordering**. Words from the instruction are randomly selected and shuffled, and this will challenge the model to deal with the shuffled instructions. (2) **Random mask**. We randomly mask 15% tokens in the instruction, and challenge the model to follow an instruction with missing words. (3) **Truncation**. We set a truncation ratio to 0.5 and using only half of the instruction and this helps to identify noises made without a clear instruction. (4) **Opposite**. In sentence representation learning, an LLM is prompted to generate the entailment or contradiction hypotheses given a premise. If a misunderstanding occurs, the model may produce the exactly opposite generation. Thus, we also utilize their opposite prompts as the noisy instructions.

When used in a standard generation process, using noisy instructions results in a significant decline in the quality of generated responses. However, when integrated into our constrastive generation (stage 1), these instructions help prevent some noisy generations and enhance the quality. We conduct human evaluation to compare our constrastive generation against generating with the

Method	Accuracy%
BERT-base	
SimCSE	72.6
SynCSE*	81.9
MultiCSR	84.2
RoBERTa-base	
SimCSE	74.1
SynCSE*	82.3
MultiCSR	85.1

Table 4: Performance comparison of different methods on the low-resource language Tagalog. For SimCSE, we only use the unlabeled sentences for training its unsupervised version. We utilize ChatGPT for data generation of both SynCSE and our MultiCSR. *: utilizing the prompts listed in their paper for generation.

original instructions. For randomly sampled 1000 premises from NLI, we calculate the percentage of sentences with better generations by utilizing our contrastive generation as "improvement" and those with worse generations following the noisy instructions as "degradation". We include their relationship in Figure 5. Interestingly, we find a strong positive correlation between the initial drop in performance and the subsequent improvement when applied to contrastive generation. Notably, the "opposite" instruction, which causes the most significant initial decline, results in the largest performance boost by improving the generation quality of nearly 70% sentences. We assume this is because, during the sentence generation for sentence representation learning, the most common mistake made by LLMs is to produce the opposite generation. By further analyzing the 694 sentences for which our contrastive generation can improve performance, we find that the generations of 431 (62.1%) sentences are opposite following the original instruction. We further include some related case studies in Appendix H.

5.3 Applying to A More Challenging Scenario

In real-world applications, except a few high-resource languages / domains, the majority of languages / domains have limited amount labeled data. In such cases, traditional supervised methods may not be directly applicable. To evaluate MultiCSR in challenging domains, we further apply our MultiCSR to low-resource languages, which can be regarded as a more challenging scenario. We conduct experiments on a rather small language: Tagalog (TL). For training corpus, we leverage

TED2020 from Reimers and Gurevych (2020) with 1167 translated sentence-pairs, and use only the sentences in TL. The evaluation is performed by finding the most similar sentence inside a corpus using cosine similarity, with 1000 test pairs from LASER (Artetxe and Schwenk, 2019). The results are shown in Table 4. For a fair comparison, the only difference of SynCSE and MultiCSR is whether utilizing our proposed self-curation during contrastive sentence pair construction and filtering *false negatives* during the in-batch training. From the results, we can observe significant improvement from utilizing methods with LLMs, and the performance can be further enhanced with MultiCSR. The evaluation of applying MultiCSR to this challenging scenario demonstrates the flexibility of our framework, enabling its broad applications across different domains and even languages.

6 Conclusion

In this paper, we introduce MultiCSR, a novel framework to contrastively refine the generation of LLMs at distinct stages of training a contrastive learning method, ensuring only high-quality and suitable sentence pairs are utilized during the model training. Experiments on standard STS tasks and several downstream tasks demonstrate the effectiveness of MultiCSR. Extensive analysis provided shows the potential of our work and hope to inspire future work in achieving better sentence representation learning with LLMs.

Limitations

Despite the effectiveness of MultiCSR, there are still some potential directions worth exploring and we leave as future work. Firstly, as introduced in 3.2, we employ pre-defined rules to control the absolute and relative similarities of each pair $(\boldsymbol{x}_+, \boldsymbol{x}_-)$ during the contrastive sentence pair construction. However, a composite model can be utilized here. As our generated sentence pairs are in the same formats of NLI datasets (Zhang et al., 2023), we can actually use a NLI classifier or its combination with LLMs for self-curation. Nevertheless, utilizing a pre-trained NLI classifier also poses a requirement for ground truth NLI corpus, contradicting the main purpose of performing unsupervised learning. Thus, we leave this as future work and hope to propose some alternative self-curation strategies. Secondly, in the first stage of our framework, we perform contrastive generation with logits acquired from different instructions. However, for closed-source LLMs, acquiring their logits is impractical. Thus, a prompting methods that can be incorporated into refining the generation of LLMs will be valuable. Recently, a contemporary method, self-correction, has been proposed to address this issue. However, the effectiveness of the techniques in this kind often depends on the fortuitous alignment of prompts or initial conditions, making them labor-intensive, highlighting more efficient approaches. To sum up, these limitations also illustrate the great potential of our method. It is expected to be applied to various domains and better serve more downstream tasks.

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.

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

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, 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. 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.

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.

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2022. In-context examples selection for machine translation. *ArXiv*, abs/2212.02437.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

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. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

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

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

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

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *ACM SIGKDD international conference on Knowledge discovery and data mining*.

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.

Taehyeon Kim, Joonkee Kim, Gihun Lee, and Se-Young Yun. 2024. Instructive decoding: Instruction-tuned large language models are self-refiner from noisy instructions. In *International Conference on Learning Representations*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. 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 6691–6706, Online. 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)*, pages 13785–13802, Toronto, Canada. Association for Computational Linguistics.

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

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *International Conference on Language Resources and Evaluation (LREC)*, pages 216–223.

Amita Misra, Brian Ecker, and Marilyn Walker. 2016. Measuring the similarity of sentential arguments in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–287, Los Angeles. Association for Computational Linguistics.

OpenAI. 2022. Introducing chatgpt.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Association for Computational Linguistics (ACL)*, pages 271–278.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Association for Computational Linguistics (ACL)*, pages 115–124.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Huai hsin Chi, Nathanael Scharli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*.

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 *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

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

Haochen Tan, Wei Shao, Han Wu, Ke Yang, and Linqi Song. 2022. A sentence is worth 128 pseudo tokens: A semantic-aware contrastive learning framework for sentence embeddings. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 246–256, Dublin, Ireland. Association for Computational Linguistics.

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, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

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

Bin Wang, C.-C. Jay Kuo, and Haizhou Li. 2022. 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.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *ArXiv*, abs/1908.04319.

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

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

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

G. Yona, Or Honovich, Itay Laish, and Roee Aharoni. 2023. Surfacing biases in large language models using contrastive input decoding. *ArXiv*, abs/2305.07378.

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

Shen Zheng, Jie Huang, and Kevin Chen-Chuan Chang. 2023. Why does chatgpt fall short in providing truthful answers?

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 Instruction Prompts

In this section, we give the details of our instruction prompts used in C_θ text generation and self-curation. In addition, for fair comparison with SynCSE (Zhang et al., 2023), we utilize the same entailment and contradiction prompts for generation. During the first stage of contrastive generation, we will randomly select one from the entailment prompts and the other one from the contradiction prompts, and acquires the next-token logits with the logits derived from these two instructions.

Entailment Prompt 1

Please paraphrase the input sentence or phrase, providing an alternative expression with the same meaning.

Entailment Prompt 2

Rewrite the following sentence or phrase using different words and sentence structure while preserving its original meaning.

Entailment Prompt 3

Create a sentence or phrase that is also true, assuming the provided input sentence or phrase is true.

Entailment Prompt 4

Please provide a concise paraphrase of the input sentence or phrase, maintaining the core meaning while altering the words and sentence structure. Feel free to omit some of the non-essential details like adjectives or adverbs.

Contradiction Prompt 1

Revise the provided sentence by swapping, changing, or contradicting some details in order to express a different meaning, while maintaining the general context and structure.

Contradiction Prompt 2

Generate a slightly modified version of the provided sentence to express an opposing or alternate meaning by changing one or two specific elements, while maintaining the overall context and sentence structure.

Contradiction Prompt 3

Transform the input sentence by adjusting, altering, or contradicting its original meaning to create a logical and sensible output sentence with a different meaning from the input sentence.

Contradiction Prompt 4

Generate a sentence that conveys a altering, contrasting or opposite idea to the given input sentence, while ensuring the new sentence is logical, realistic, and grounded in common sense.

While we utilize an LLM to measure the semantic similarity, we will use the prompt as the following:

Measuring Similarity Prompt

Scoring the semantic similarity of the following sentences between 0.0 and 5.0, 5.0 means they have the same meaning, 0.0 means they are completely different: (a) "{$\boldsymbol{x}$}", (b) "{$\boldsymbol{x}_y$}":

B MultiCSR with Various Backbones

In our main results from Table 2, we improve the performance of SimCSE by a large margin. It is important to note that MultiCSR is general and can be uniformly applied to different backbones. Thus, in this section, we also conduct experiments with representative contrastive sentence representation learning methods PromptBERT (Jiang et al., 2022), InfoCSE (Wu et al., 2022), RankEncoder (Seonwoo et al., 2023) and RankCSE (Liu et al., 2023). The results shown in Table 5 show that, our approach can consistently improve their performance by a large margin, which also demonstrates the strong generalization ability of our MultiCSR.

C Effect of Smoothing Coefficient ω

Figure 6 shows the influence of the hyperparameter ω on our method's performance. This parameter adjusts the smoothness of logits derived from noisy instructions. For all the evaluation, we utilize opposite prompts as the noisy instructions. Same as Section 5.2, we sample 100 sentences from the NLI premises and conduct human evaluation. From the results, we can see that performance tends to decline with negative ω values, as the model becomes increasingly biased toward the noisy instruction. Conversely, excessively positive values (above 0.4) lead to a quick deterioration in performance. Interestingly, the model's performance stabilizes be-

Model	Avg.
BERT-base	
SimCSE	76.25
+ MultiCSR	80.95
PromptBERT	78.54
+ MultiCSR	81.15
InfoCSE	78.85
+ MultiCSR	80.75
RankEncoder	80.07
+ MultiCSR	80.52
RankCSE	80.36
+ MultiCSR	81.29
RoBERTa-base	
SimCSE	76.57
+ MultiCSR	81.29
PromptRoBERTa	79.15
+ MultiCSR	81.57
RankCSE	79.81
+ MultiCSR	81.95

Table 5: Performance comparison on different backbone contrastive sentence representation learning methods between with and without MultiCSR.

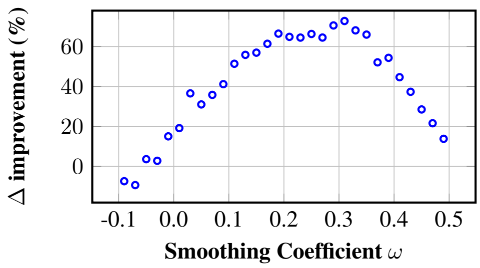

Figure 6: Generation improvement by incorporating noisy instruction with $l_t - \omega * \hat{l}_t$ into our contrastive generation stage.

tween 0.2 and around 0.4, indicating a certain level of robustness to variations in ω within this range. In our main results, we consistently utilize ω as 0.3.

D Self-Curation Strategies

In Section 3.2, we briefly introduce how we select "high-quality" triplets like $(\boldsymbol{x}, \boldsymbol{x}_+, \boldsymbol{x}_-)$ after we get the semantic similarity scores from the LLM C_θ as a and b:

$$(\boldsymbol{x}, \boldsymbol{x}_+, \boldsymbol{x}_-) \in \mathcal{T}, \text{ if } \begin{cases} a \geq \alpha \\ b \leq \beta \\ a \geq b + \gamma \end{cases}. \quad (8)$$

α	β	γ	$\lvert\mathcal{T}\rvert$	Spearman's
5	0	-	30.7k	83.14
5	1	-	37.7k	82.75
5	2	-	37.9k	82.39
5	3	-	47.8k	83.41
5	4	-	108.9k	80.49
4	0	-	48.8k	85.68
4	1	-	60.4k	85.52
4	2	-	60.7k	85.89
4	3	-	72.9k	86.41
3	0	-	51.6k	85.03
3	1	-	63.7k	85.20
3	2	-	64.1k	85.84
2	0	-	51.6k	85.01
2	1	-	63.8k	85.44
1	0	-	54.4k	83.85
4	4	1	134.0k	83.64
4	4	2	70.6k	84.02
4	4	3	60.6k	85.72
4	4	4	55.8k	85.91
3	3	1	76.4k	**86.54**
3	3	2	73.9k	85.42
3	3	3	63.4k	86.21
3	1	3	63.2k	85.07
2	1	2	63.8k	85.62
2	2	1	64.1k	85.85
2	2	2	64.0k	86.08
1	1	1	66.6k	86.14

Table 6: Studies of different self-curation strategies in MultiCSR. The results are the development performance of STS-B.

We firstly conduct extensive experiments on different combinations of hyperparameters α, β and γ. The results are shown in Table 6. Based on the results from Table 1, we know that directly utilizing LLMs to measure the similarities may not be a wise choice, but their provided signals can still help us improve the quality of generated corpus. From the results in Table 6, we can see that choices that are not too extreme (e.g., set α to 5 or β to 0) tend to give better results, which also demonstrate our concern that "*hard negative*" and "*hard positive*" are always important to provide sufficient challenges to learn a better contrastive learning model.

E Ablation Study on Mask Indicator Threshold

In this section, we perform an ablation study of setting different mask indicator thresholds used in Section 3.3 to mask in-batch false negatives. The results are presented in Figure 7. As shown in the figure, introducing the mask indicator mechanism significantly improves the results. When the threshold is set too low, the performance drops as expected, due to the fact that too many sentences are masked and few challenging inputs are given

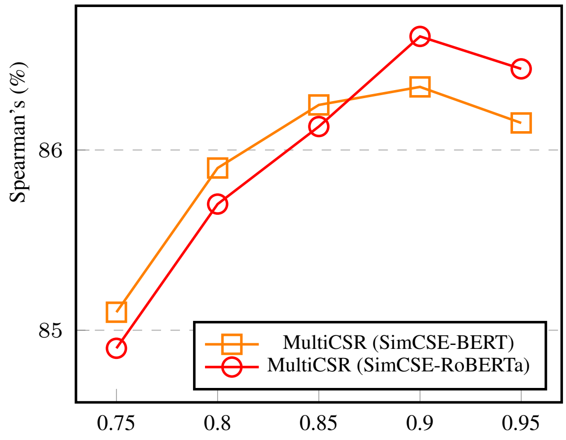

Figure 7: The performance of MultiCSR with different mask indicator thresholds. The results are the development performance of STS-B.

Dataset/Model	SimCSE	MultiCSR
MS MARCO	0.230	0.280
TREC-COVID	0.298	0.362
NFCorpus	0.125	0.159
NQ	0.141	0.192
HotpotQA	0.219	0.252
FiQA-2018	0.091	0.133
ArguAna	0.386	0.427
Touché-2020	0.089	0.139
CQADupStack	0.155	0.201
Quora	0.768	0.798
DBPedia	0.191	0.214
SCIDOCS	0.075	0.107
FEVER	0.122	0.184
Climate-FEVER	0.111	0.150
SciFact	0.329	0.391
Avg.	0.222 (1x)	**0.266 (1.20x)**

Table 7: Performance comparison on zero-shot information retrieval tasks BEIR.

to the model. Meanwhile, the purpose of filtering out false negative sentences is not fulfilled if the threshold chosen is too high. Based on the experimental results on STS-B development set, we set the mask indicator threshold to be 0.9 across our main context.

F Performance on Transfer Tasks and BEIR Tasks

We also evaluate our approach on the following transfer tasks: MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), SST-2 (Socher et al., 2013), TREC (Voorhees and Tice, 2000) and MRPC (Dolan and Brockett, 2005). In these tasks, a classifier is trained on top of sentence representations produced by different methods. We reported the detailed performance of MultiCSR in Table 8. From the experimental results, we can see that MultiCSR achieves the best performance on both BERT$_{base}$ and RoBERTa$_{base}$, which also shows the potential of our method to be applied to different downstream tasks.

To further evaluate our framework on real-world applications, we also test the performance of MultiCSR on zero-shot information retrieval tasks BEIR (Thakur et al., 2021) and include the results in Table 7. We directly utilize our trained checkpoint to test and no in-domain data is used. The results are shown in the following table with nDCG10 scores reported, following BEIR. We use all the tasks public in BEIR. For CQADupStack, we evaluate each StackExchange subforum separately and report the overall average scores. The results show that our method can consistently improve the performance of SimCSE on these tasks, even before we use any unlabeled in-domain data. We believe that better semantic similarity performance is beneficial to the measurement of sentence-pair relationship, and our method can help the base model capture richer semantic information.

G Supervised Settings

In this section, we introduce how we perform supervised settings on our method. Using demonstrations is now a standard way to perform few-shot (Brown et al., 2020; Agrawal et al., 2022) inference on LLMs in various tasks. As a natural semantic retriever, the pre-trained sentence representation model P_η can be utilized to find the most proper demonstrations. Given the source sequence $\boldsymbol{x}$, we first compute its representation as $P_\eta(\boldsymbol{x})$, and search over $\mathcal{W}$ (i.e., NLI premises specifically) to find the most relevant demonstrations, based on $\mathrm{sim}(P_\eta(\boldsymbol{x}), P_\eta(\boldsymbol{x}'))$ where $\mathrm{sim}(,)$ calculates the cosine similarity of two parameterized vectors. We denote the set of L demonstrations as $\mathcal{D} = (d_1, d_2, ..., d_L)$, and each demonstration d will be either the entailment or contradiction hypothesis of the premise $\boldsymbol{x}^k$. To prevent the low similarity demonstrations, we only choose premise $\boldsymbol{x}^k$ with $\mathrm{sim}(P_\eta(\boldsymbol{x}), P_\eta(\boldsymbol{x}^k)) \geq \lambda$, where λ serves as a hyperparameter threshold. Finally, the whole text generation procedure without any parameter updates can be formulated as the following:

$$\boldsymbol{y} \leftarrow C_\theta\left([\mathcal{D}; \boldsymbol{x}; I]\right), \boldsymbol{x} \in \mathcal{W}.$$

We have included the performance of several

Model	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC	Avg.
				BERT-base				
SimCSE	81.18	86.46	94.45	88.88	85.50	**89.80**	74.43	85.81
DiffCSE*	81.76	86.20	94.76	89.21	86.00	87.60	75.54	85.87
PromptBERT	80.74	85.49	93.65	89.32	84.95	88.20	76.06	85.49
RankCSE	83.07	88.27	95.06	89.90	87.70	89.40	76.23	87.09
MultiCSR (SimCSE)	**83.60**	**89.05**	**95.87**	**90.98**	**87.77**	88.60	**76.36**	**87.46**
				RoBERTa-base				
SimCSE	81.04	87.74	93.28	86.94	86.60	84.60	73.68	84.84
DiffCSE*	82.42	88.34	93.51	87.28	87.70	86.60	76.35	86.03
PromptRoBERTa	83.82	88.72	93.19	90.36	88.08	90.60	76.75	87.36
RankCSE	83.32	88.61	**94.03**	88.88	89.07	**90.80**	76.46	87.31
MultiCSR (SimCSE)	**85.75**	**91.25**	93.98	**90.94**	**90.16**	89.90	**76.91**	**88.41**

Table 8: Performance comparison on transfer tasks. *: since they select the model based on the development sets of transfer tasks, we retest their performance with their officially released checkpoints for a fair comparison.

L	λ	Spearman's
	BERT-base	
3	0.6	85.46
5	0.8	85.51
-	-	85.35
	RoBERTa-base	
3	0.6	86.52
5	0.8	86.81
-	-	86.63

Table 9: Studies of different number of demonstrations L and similarity controlled thresholds λ in MultiCSR. The results are the development performance of STS-B.

Model	Avg.
BERT-base	
SBERT	74.89
SimCSE	81.57
MultiCSR w/o stage3	81.62
MultiCSR	82.17
RoBERTa-base	
SRoBERTa	74.21
SimCSE	82.52
MultiCSR w/o stage3	82.37
MultiCSR	82.90

Table 10: Supervised performance on STS tasks.

combinations of different L and λ in Table 9. From the result, we can see that, although using demonstrations can be still helpful in our contrastive generation procedure, the improvement compared with not using demonstrations is not so significant. And inference with more demonstrations will somehow increase the time required for generation. Thus, in all our supervised settings, we choose not to use demonstrations including the results in Table 10.

After generation of $\mathcal{T}$, we would combine $\mathcal{T}$ with ground truth corpus $\mathcal{N}$, which means, for each premise in $\mathcal{T}$, there would be at least one triplet with the same premise in $\mathcal{N}$. As an example, we randomly sample a batch of size N from $\mathcal{T} \cup \mathcal{N}$, the distribution of cosine similarity given by $\mathrm{sim}(P_\eta(\boldsymbol{x}), P_\eta(\boldsymbol{x}'))$ between any premises and (h_+^k, h_-^k) (i.e., $2N(N-1)$ similarity scores here) is shown in Figure 8. From the figure, we can see that even excluding the entailment and contradiction hypotheses of each premise, there are still more than 18.6% sentences inside a batch having a similarity score higher than 0.7. Thanks to mask indicator introduced in Section 3.3, we can directly utilize this mixed corpus for training without suffering from *false-negative* problem. The results of supervised MultiCSR are included in Table 10.

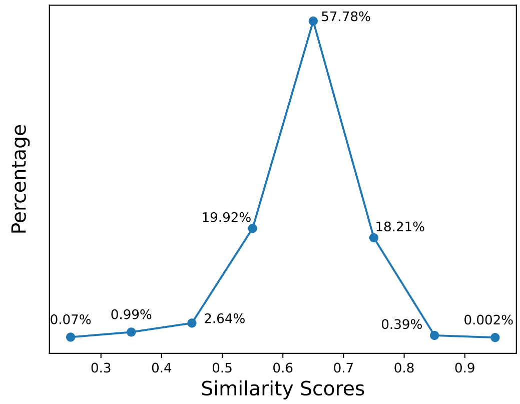

Figure 8: The percentages of consine similarity scores provided by pre-trained supervised SimCSE-RoBERTa on a randomly sampled batch of size 256.

We further provide a cost analysis of the time and space. Since our stage 1&2 works before the in-batch training stage, we provide the analysis of the resource required to support our stage 3, under different batch sizes. We test on a single NVIDIA V100-32GB, and replicate a triplet $10K$ times as the training corpus. The time and GPU memory required for training are shown in Table 12, while the inference time and space of our method remains the same as theirs.

It is worth mentioning that the data generated in the first stage of our framework becomes more

Label	Original & Generated Sentences	Semantic Similarities
Premise	One of our number will carry out your instructions minutely.	-
Entailment	A member of my team will execute your orders with immense precision.	4.5
Contradiction	We have no one free at the moment so you have to take action yourself.	0.0
Premise	He turned and smiled at Vrenna.	-
Entailment	He turned back and smiled at Vrenna.	5.0
Contradiction	He turned and walked away.	0.0
Premise	How do we fix this?	-
Entailment	How can we fix this?	5.0
Contradiction	Let's not worry about fixing this.	1.0
w/o Stage 1	We can't figure out how to fix this.	4.0
Premise	The economy could be still better.	-
Entailment	The economy is not at its best possible state.	4.0
w/o Stage 1	The economy is not good.	0.0
Contradiction	The economy could be worse.	0.0

Table 11: Generated sentence triplets and the semantic similarities between the hypotheses and their premises. The sentence and similarities are generated by on NLI premises. If we set the thresholds of filtering strategy as $\alpha = 3$, $\beta = 3$ and $\gamma = 1$, the third triplet will not appear in the training corpus $\mathcal{T}$ because of unqualified contradiction $4.0 \nleq 3$, and neither for the fourth triplet because of wrong entailment $0.0 \ngeq 3$.

Batch Size	Time(s)		GPU Memory(MB)	
	SimCSE	MultiCSR	SimCSE	MultiCSR
8	86(1x)	99(1.15x)	3863(1x)	5225(1.35x)
16	62(1x)	81(1.31x)	4873(1x)	6080(1.25x)
32	55(1x)	75(1.36x)	5661(1x)	8163(1.44x)
64	52(1x)	68(1.30x)	7353(1x)	11965(1.63x)
128	51(1x)	65(1.27x)	11503(1x)	20137(1.75x)

Table 12: Time and memory cost analysis of perform our stage 3, contrastive in-batch training.

lightweight after the second stage of self-curation. This makes, although the time required to train on the same dataset increases, the time we need to train a model of MultiCSR is actually greatly reduced. We assume that a training set has M sentences, and the batch size is N. In unsupervised SimCSE, for each sentence, it will have 1 positive pair and $N-1$ negative pairs, then there will be a total of $M*N$ pairs of similarity to be calculated during training. In MultiCSR, for each sentence, it will have 1 positive pair and $2N-1$ negative pairs, a total of $M*(2N)$ pairs will be considered. But in our scenario, the value of M varies greatly. For example, if we set the batch size to 64, it takes $0.98h$ to train an unsupervised SimCSE on $1M$ Wikipedia sentences, with $1M*64 = 64M$ pairs. But after our first and second stages with these sentences, only $0.19M$ triplets are left, with $0.19M*(2*64) = 24.32M$ pairs. Based on this, it only takes $0.36h$ to train our model.

H Case Studies

In this section, we present some generated triplets using Flan-T5-XL in Table 11 with the premises from NLI and semantic similarity scores for these triplets. While the first two example triplets are high quality data for training, the last two triplets suffer from either the *false-negative* or the *false-positive* problem if Stage 1 is not applied. For example, in the third triplet, the contradiction sentence generated without Stage 1 "We can't figure out how to fix this." has a 4.0 high similarity score to the premise sentence "How do we fix this?", indicating a *false negative* that can harm the training if included. In our method, we can either avoid these "low-quality" triplets by introducing Stage 1 or self-curation strategy by $\alpha = 3$, $\beta = 3$ and $\gamma = 1$, which we proposed in Stage 2. For another example, we can also exclude the third triplet in training due to unqualified contradiction $4.0 \nleq 3$ and avoid the *false negative* problem. From the case studies we conduct, we can see that all components are helpful in refining the generation of LLMs, ensuring only high-quality sentence pairs are utilized into model training.