

# Contrastive Learning of Sentence Embeddings from Scratch

Junlei Zhang<sup>1,2\*</sup> Zhenzhong Lan<sup>2</sup> Junxian He<sup>†3</sup>

<sup>1</sup>Zhejiang University <sup>2</sup>School of Engineering, Westlake University

<sup>3</sup>The Hong Kong University of Science and Technology

{zhangjunlei, lanzhenzhong}@westlake.edu.cn, junxianh@cse.ust.hk

## Abstract

Contrastive learning has been the dominant approach to train state-of-the-art sentence embeddings. Previous studies have typically learned sentence embeddings either through the use of human-annotated natural language inference (NLI) data or via large-scale unlabeled sentences in an unsupervised manner. However, even in the case of 1; unlabeled data, their acquisition presents challenges in certain domains due to various reasons. To address these issues, we present SynCSE, a contrastive learning framework that trains sentence embeddings with synthesized data. Specifically, we explore utilizing large language models to synthesize the required data samples for contrastive learning, including (1) producing positive and negative annotations given unlabeled sentences (*SynCSE-partial*), and (2) generating sentences along with their corresponding annotations from scratch (*SynCSE-scratch*). Experimental results on sentence similarity and reranking tasks indicate that both SynCSE-partial and SynCSE-scratch greatly outperform unsupervised baselines, and SynCSE-partial even achieves comparable performance to the supervised models in most settings.<sup>1</sup>

## 1 Introduction

The objective of sentence representation learning is to derive sentence embeddings that can benefit a wide range of downstream tasks, including reranking (Lee et al., 2021; Barker et al., 2021), natural language understanding (Cer et al., 2018), and retrieval (Misra et al., 2016; Thakur et al., 2021; Wang et al., 2022a). Methods built on contrastive learning, such as SimCSE (Gao et al., 2021) and PromCSE (Jiang et al., 2022b), have dominated the field due to their competitive performance (Zeng

\*Work done during Junlei’s visit to HKUST.

†Corresponding author.

<sup>1</sup>Code and the synthesized datasets are available at <https://github.com/hkust-nlp/SynCSE>.

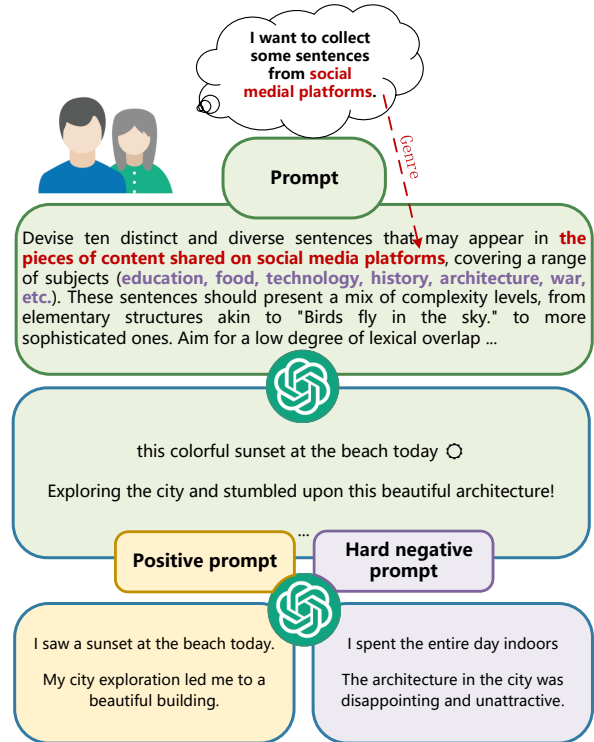


Figure 1: An overview of the data synthesis process of SynCSE-scratch. We specify a desired domain and genre, and our framework will generate diverse unlabeled data for that domain along with their positive and negative annotations.

et al., 2022; Limkonchotiawat et al., 2022; Wu et al., 2022a; Wang et al., 2022c; He et al., 2023).

Contrastive learning trains sentence representations through distinguishing positive samples from negative ones. In this framework, the quality of these positive and negative annotations plays a critical role. Supervised approaches typically gather these annotations from labeled natural language inference (NLI) datasets (Jiang et al., 2022a; Limkonchotiawat et al., 2022) – however, such sources are generally unavailable for most settings, and manually creating them is cost-prohibitive. As a result, unsupervised methods that solely rely on unlabeled sentences attract significantly more attention re-

cently (Gao et al., 2021; Zhou et al., 2022; Wu et al., 2022a) – they mostly develop methods to automatically obtain positive and negative samples to facilitate contrastive learning. A representative example is SimCSE (Gao et al., 2021), which leverages perturbed hidden states as the positive samples and in-batch sentences as negatives to perform contrastive learning. To differentiate between in-batch negatives and the annotated negatives, the latter are often termed “hard negatives”, which have proven to be significantly advantageous in enhancing sentence embeddings (Wang et al., 2022b,c).

Despite considerable advances in recent years, the performance of these unsupervised methods still falls short when compared to their supervised counterparts. Moreover, the unavailability of large-scale unlabeled data for the targeted domain often poses additional limitations to these approaches. To overcome these challenges, we introduce SynCSE, an unsupervised contrastive framework that trains sentence embeddings with synthesized data. Concretely, we propose to prompt large language models (LLMs) such as ChatGPT (OpenAI, 2022) to synthesize the samples needed for contrastive learning. This is inspired by recent successes of prompting large language models (LLMs) to perform various tasks (Chung et al., 2022; Ouyang et al., 2022; OpenAI, 2023), especially the superior performance of LLMs over crowd-workers on text annotation (Gilardi et al., 2023). We investigate two variants of SynCSE in this work that correspond to two practical scenarios: (1) *SynCSE-partial*, where large-scale unlabeled sentences are available and LLMs are prompted to produce positive and hard negative annotations, and (2) *SynCSE-scratch*, where large-scale unlabeled sentences are not available, prompting LLMs to generate sentences and their corresponding annotations from scratch. The latter represents a particularly challenging yet practical scenario where we aim to learn sentence embeddings without any data samples.

We conduct comprehensive experiments on the standard Semantic Textual Similarity (STS) benchmark, along with four reranking tasks and four domain adaptation tasks. Our results demonstrate that both SynCSE-partial and SynCSE-scratch substantially outperform the unsupervised baselines in all cases – for example, SynCSE-partial and SynCSE-scratch exceed the unsupervised SimCSE baseline by 5.37 and 4.18 absolute points respectively on STS. Particularly, SynCSE-partial often equals its

supervised counterpart on STS, marking the first instance of an unsupervised method matching supervised results on this benchmark. We release our synthesized datasets to facilitate further research to learn better sentence embeddings.

## 2 SynCSE

### 2.1 Background

We base our approach on the formulation of SimCSE (Gao et al., 2021), which is one of the most common and effective contrastive learning frameworks to learn sentence embeddings. Formally, we denote the unlabeled sentence as  $x_i$  and its positive sample as  $x_i^+$ . Let  $\mathbf{h}_i$  and  $\mathbf{h}_i^+$  denote the representations of  $x_i$  and  $x_i^+$  respectively, then the unsupervised SimCSE loss is defined as:

$$-\log \frac{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^M e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau}}, \quad (1)$$

where  $M$  denotes the mini-batch’s size,  $\tau$  is a temperature hyperparameter, and  $\text{sim}(\cdot, \cdot)$  stands for a similarity function. Unsupervised SimCSE passes the same  $x_i$  twice to the encoder to form  $(\mathbf{h}_i, \mathbf{h}_i^+)$  pairs due to random dropout, and other sentences within the same mini-batch are considered as negative samples as shown in Eq. 1. Supervised SimCSE further extends  $(x_i, x_i^+)$  with hard negative samples  $x_i^-$  to constitute the triplet datasets  $\{x_i, x_i^+, x_i^-\}_{i=1}^N$  and define the supervised loss:

$$-\log \frac{e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_i^+)/\tau}}{\sum_{j=1}^M (e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^+)/\tau} + e^{\text{sim}(\mathbf{h}_i, \mathbf{h}_j^-)/\tau})}. \quad (2)$$

In supervised SimCSE, the  $(x_i, x_i^+, x_i^-)$  triplets are typically from annotated NLI datasets, where  $x_i$  is the premise,  $x_i^+$  and  $x_i^-$  are the entailment and contradiction hypotheses. Supervised SimCSE significantly outperforms the unsupervised one due to the enhanced quality of positive and hard negative samples. However, such annotated data are typically unavailable in most settings, and manually annotating triplets  $(x_i, x_i^+, x_i^-)$  can be resource-intensive, rendering unsupervised approaches the most promising choices in practice. In this work, we focus on the supervised loss in Eq. 2, but synthesize  $(x_i^+, x_i^-)$  given  $x_i$  or even generate  $(x_i, x_i^+, x_i^-)$  triplets from scratch, aiming to approach the performance of supervised models with an unsupervised method. We describe our data synthesis process next.

---

### Hard negative prompts pools

---

**Prompt1:** 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.

**Prompt2:** 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.

**Prompt3:** 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.

**Prompt4:** Generate a sentence that conveys an altering, contrasting or opposite idea to the given input sentence, while ensuring the new sentence is logical, realistic, and grounded in common sense.

---

Table 1: Hard negative prompts pools. During the generation of hard negative samples, a hard negative prompt is randomly sampled each time.

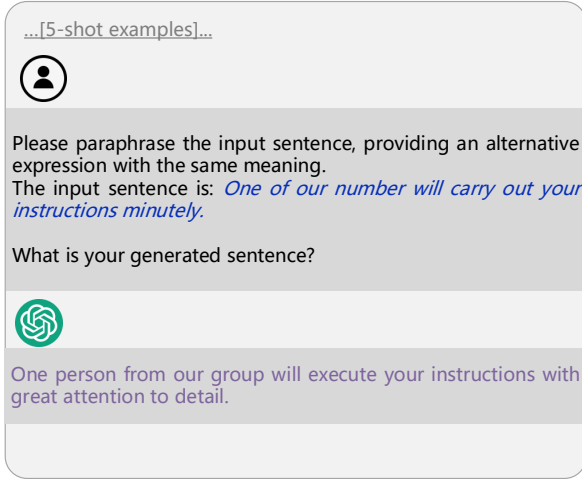


Figure 2: Few-shot examples of generating positive examples of the input sentence. We adopt 5-shot for generation.

## 2.2 Data Synthesis from ChatGPT

We propose to prompt ChatGPT (OpenAI, 2022) to synthesize the required data in contrastive learning, inspired by recent successes of prompting LLMs to fulfill multiple tasks (Chung et al., 2022; OpenAI, 2023). Concretely, we introduce two variants of SynCSE: (1) SynCSE-partial which synthesizes  $(x_i^+, x_i^-)$  given  $x_i$ , and (2) SynCSE-scratch which synthesizes  $(x_i, x_i^+, x_i^-)$  from scratch. SynCSE-scratch is practically useful since large-scale unlabeled data are not always available in the domain of interest due to copyright restrictions, data distribution issues, or messy formats. We describe these two variants below. In general, using SynCSE-scratch as an example, the complete data generation process includes two parts: (1) generating unlabeled sentences in the target domain; (2) generating

positive/hard negative labels with prompt/example pools.

## 2.3 SynCSE-partial

**Synthesizing positive and hard negative examples:** We prompt ChatGPT in a few-shot setting to annotate positive and hard negative samples given a sentence  $x_i$ , an illustrative example is shown in Figure 2. The structure of the prompts for generating positive and hard negative examples remains the same; the only difference lies in the prompts. In our implementation with the ChatGPT model, we have designed a few-shot prompt in a multi-turn chat format.

**Example and prompt pools:** A significant challenge in creating synthetic datasets lies in enhancing the dataset’s diversity. Ye et al. (2022b) suggested that merely increasing the size of the synthetic dataset might not lead to better performance, with one reason being the lack of diversity. Datasets labeled by groups of annotators can naturally help to mitigate this problem due to the variance in understanding and interpretation of prompts among different annotators. This variance results in diverse outputs, even for the same input. For example, Williams et al. (2018) utilized 387 annotators to create the MultiNLI dataset. Even with the same prompt, these annotators provided varied outputs due to their individual understanding of the prompt and their unique world knowledge, leading to a more diverse dataset. In an attempt to mimic this variation among different annotators, we employ example pools and prompt pools. Specifically, we designed four types of positive/hard negative prompts (an example of hard negative prompts are

showed in Table 1) and 18 few-shot exemplars for each of the prompt (generated using GPT-4). During each data generation process, we sample one prompt and five exemplars to construct a distinct input prompt. Details of these pools can be found in Appendix A.

## 2.4 SynCSE-scratch

Creating a synthetic dataset from scratch, where the necessary unlabeled sentences for annotation are absent, presents a substantial challenge. We address this problem in two stages: initially, we generate unlabeled sentences, and subsequently, we apply the procedure discussed in §2.3 to annotate positive and hard negative samples of these sentences.

To ensure data diversity during the generation of unlabeled sentences, we employ a strategy that specifies the genres and topics when generation, combined with the utilization of example and prompt pools. This strategy is intended to minimize repetition and redundancy between the new data and the generated data so far. More specifically, as illustrated in Figure 1, given a text genre, we randomly select six topics from a pre-defined list to be included in the prompt (the list of genres and topics used in this paper can be found in Appendix B). The term "etc." in the prompt ensures that the generated sentences are not strictly limited to these six topics. We adopt one-shot prompting to generate several sentences at once. As long as given different genres or topics when adding data compared to the existing data, the added data will likely have low redundancy with the existing data, thereby enhancing the overall diversity of the dataset. The examples used for generating raw sentences were produced by GPT-4.

## 3 Experiment

### 3.1 Training

We evaluate three different settings in the experiments, including SynCSE-partial, SynCSE-scratch, as well as a combination of SynCSE-scratch with existing annotated datasets in a supervised setting. While both SynCSE-partial and SynCSE-scratch represent unsupervised settings, in the combination setting we augment previous annotated datasets with the synthesized data produced in SynCSE-scratch, to examine whether SynCSE-scratch could provide help for a supervised scenario as well.

We refer to the NLI dataset (MNLI+SNLI) used

by SimCSE as SimCSE\_NLI. In the creation of the SynCSE-partial dataset, for a fair comparison, we utilized the unlabeled sentences  $x$  from SimCSE\_NLI, and generated positive/hard negative examples for them using the algorithm detailed in §2.3. For SynCSE-scratch, we generate the same number of examples as in the SynCSE-partial case, as detailed in §2.4. While our method can easily scale up the dataset, for a fair comparison, we ensure the data volume used for SynCSE-scratch and SynCSE-partial is equivalent to that of SimCSE\_NLI. For the combination of the SynCSE-scratch and SimCSE\_NLI datasets, we simply merge these two datasets to evaluate whether our generated dataset can aid the manually annotated one.

Given that SimCSE serves as a general method in contrastive learning, we consistently use SimCSE as the backbone method for SynCSE. We note that SynCSE is general and could be combined with more advanced algorithms as well, such as with PromCSE (Jiang et al., 2022b) and CARDS (Wang et al., 2022c). We emphasize that, after training the models on the NLI dataset, we freeze the models and directly evaluate our embeddings on all the different tasks and setting below – we do not specifically train sentence embeddings on each setting separately. For the STS and transfer learning tasks, we use the same hyperparameters as SimCSE. Since SimCSE did not conduct reranking experiments, we directly use the default parameters of MTEB (Muennighoff et al., 2023) to evaluate embeddings on the reranking tasks.

### 3.2 Evaluation Settings

**Semantic Textual Similarity Tasks:** Following the procedure outlined in SimCSE, we evaluate our model, trained on the synthetic NLI dataset, across seven semantic textual similarity (STS) tasks: STS 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), the STS Benchmark (Cer et al., 2017), and SICK Relatedness (Marelli et al., 2014). It is important to note that no data from these STS tasks were used during training. Our model was trained solely on our synthetic NLI dataset. The sentence embeddings, which we evaluate on the STS tasks, are obtained from the  $[CLS]$  representation. During the training process, we average the development scores from the STS Benchmark and SICK Relatedness to form the evaluation matrix. This matrix is used to select the best models. The other hyper-

Model	Method	STS12	STS13	STS14	STS15	STS16	STSB	SICK-R	Avg
<i>Unsupervised methods</i>									
RoBERTa-base	unsup-SimCSE <sup>†</sup>	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
	RankCSE <sup>§</sup> <sub>listNet</sub>	72.88	84.50	76.46	84.67	<b>83.00</b>	83.24	71.67	79.49
	RankCSE <sup>§</sup> <sub>listMLE</sub>	72.74	84.24	75.99	84.68	82.88	83.16	71.77	79.35
	L2P-CSR <sup>♣</sup>	74.97	83.63	78.28	84.86	82.03	82.77	71.26	79.69
	PromptRoBERTa <sup>††</sup>	73.94	<b>84.74</b>	77.28	84.99	81.74	81.88	69.50	79.15
	PCL <sup>‡‡</sup>	71.54	82.70	75.38	83.31	81.64	81.61	69.19	77.91
	CARDS <sup>°</sup>	72.49	84.09	76.19	82.98	82.11	82.25	70.65	78.68
	ConPVP <sup>•</sup>	73.20	83.22	76.24	83.37	81.49	82.18	74.59	79.18
	SynCSE-partial (SimCSE based)	<b>76.11</b>	84.49	<b>79.61</b>	<b>85.26</b>	82.60	<b>83.94</b>	<b>81.57</b>	<b>81.94</b>
	SynCSE-scratch (SimCSE based)	74.61	83.76	77.89	85.09	82.28	82.71	78.88	80.75
RoBERTa-large	unsup-SimCSE <sup>†</sup>	72.86	83.99	75.62	84.77	81.80	81.98	71.26	78.90
	RankCSE <sup>§</sup> <sub>listNet</sub>	73.23	85.08	77.50	85.67	82.99	84.20	72.98	80.24
	RankCSE <sup>§</sup> <sub>listMLE</sub>	73.40	<b>85.34</b>	77.25	85.45	82.64	84.14	72.92	80.16
	L2P-CSR <sup>♣</sup>	73.65	84.08	78.29	85.36	82.15	83.70	73.47	80.10
	PCL <sup>‡‡</sup>	73.76	84.59	76.81	85.37	81.66	82.89	80.33	79.34
	CARDS <sup>°</sup>	74.63	86.27	79.25	85.93	83.17	83.86	72.77	80.84
	ConPVP <sup>•</sup>	74.75	84.09	77.88	83.13	83.44	83.64	74.31	80.18
	SynCSE-partial (SimCSE based)	<b>76.03</b>	84.27	80.03	85.37	83.62	84.26	<b>81.14</b>	82.10
	SynCSE-scratch (SimCSE based)	75.45	85.01	<b>80.28</b>	<b>86.55</b>	<b>83.95</b>	<b>84.49</b>	80.61	<b>82.33</b>
	<i>Supervised methods</i>								
RoBERTa-base	sup-SimCSE <sup>†</sup>	76.53	85.21	80.95	86.03	82.57	85.83	80.50	82.52
	sup-SimCSE	75.61	85.19	79.58	85.85	82.39	85.30	80.39	82.04
	PromptRoBERTa <sup>††</sup>	76.75	<b>85.93</b>	<b>82.28</b>	86.69	82.80	<b>86.14</b>	80.04	<b>82.95</b>
	PromptCSE + EH <sup>♡</sup>	75.96	84.99	80.44	<b>86.83</b>	81.30	84.40	80.96	82.13
	SynCSE-scratch + SimCSE_NLI	<b>76.79</b>	84.93	80.14	86.28	<b>83.38</b>	85.75	<b>81.02</b>	82.61
RoBERTa-large	sup-SimCSE <sup>†</sup>	77.46	87.27	82.36	86.66	83.93	86.70	81.95	83.76
	sup-SimCSE	76.62	86.90	82.05	86.10	83.97	86.10	82.04	83.40
	PromptCSE + EH <sup>♡</sup>	<b>79.56</b>	<b>88.97</b>	<b>83.81</b>	<b>88.08</b>	84.96	<b>87.87</b>	82.43	<b>85.10</b>
	SynCSE-scratch + SimCSE_NLI	77.13	87.61	82.82	87.67	<b>85.66</b>	87.22	<b>82.45</b>	84.37

Table 2: Results on the STS benchmark. Spearman’s correlation is reported. The “unsup-” and “sup-” correspond to unsupervised and supervised settings, respectively. “†”: results from (Gao et al., 2021); “§”: results from (Liu et al.); “♣”: results from (Zhou et al., 2023); “††”: results from (Jiang et al., 2022a); “‡‡”: results from (Wu et al., 2022a); “°”: results from (Wang et al., 2022c); “•”: results from (Zeng et al., 2022); “♡”: results from (Jiang et al., 2022b). The term “SynCSE-scratch + SimCSE\_NLI” represents our synthetic data combined with the NLI dataset used in SimCSE. The SynCSE-partial/scratch experiments were implemented on the basis of SimCSE. Some baselines did not conduct some experimental setups. We report the results that exist in their papers.

parameters are kept consistent with those used in SimCSE.

**Reranking tasks:** We further evaluate the synthetic dataset on four reranking tasks: AskUbuntuDupQuestions (Lei et al., 2016), MindSmallR-reranking (Wu et al., 2020), SciDocsRR (Cohan et al., 2020), and StackOverflowDupQuestions (Liu et al., 2018). We directly evaluate the model, which is frozen after training on the NLI dataset, on reranking tasks, without using the training sets of reranking tasks. The resulting ranking is scored for each query and averaged across all queries. In line with the methodology of MTEB (Muennighoff et al., 2023), we utilize Mean Average Precision (MAP) as the primary metric.

**Baselines:** We compare our approach with state-of-the-art sentence embedding learning methods: RankCSE (Liu et al.), L2P-CSR (Zhou et al., 2023), PCL (Wu et al., 2022a), CARDS (Wang et al., 2022c), ConPVP (Zeng et al., 2022), and PromptRoBERTa (Jiang et al., 2022a). While we base our approach on SimCSE, we emphasize that our approach *is orthogonal to the baseline algorithms* and our synthesized datasets may be combined with them to further boost the performance. We directly report the results from their respective papers.

### 3.3 Semantic Textual Similarity

**Main results:** As shown in Table 2, Both SynCSE-partial and SynCSE-scratch significantly

Model	Method	AskU.	MindSmall	SciDocsRR	StackO.	Avg
<i>Unsupervised methods</i>						
RoBERTa-base	unsup-SimCSE	52.78	29.91	65.96	39.25	46.95
	CARDS	52.94	27.92	64.62	<b>41.51</b>	46.75
	PCL	51.85	27.92	64.70	41.18	46.41
	SynCSE-partial (SimCSE based)	<b>53.95</b>	29.97	65.21	37.84	46.74
	SynCSE-scratch (SimCSE based)	53.27	<b>30.29</b>	<b>67.55</b>	39.39	<b>47.63</b>
RoBERTa-large	unsup-SimCSE	55.10	29.23	68.54	42.56	48.86
	CARDS	53.83	29.07	68.26	<b>43.24</b>	48.60
	PCL	53.43	28.56	66.06	41.54	47.40
	SynCSE-partial (SimCSE based)	54.78	30.23	68.90	38.28	48.05
	SynCSE-scratch (SimCSE based)	<b>55.48</b>	<b>30.27</b>	<b>70.85</b>	40.00	<b>49.15</b>
<i>Supervised methods</i>						
RoBERTa-base	sup-SimCSE	52.55	29.87	<b>68.43</b>	37.52	47.09
	SynCSE-scratch + SimCSE_NLI	<b>52.74</b>	<b>30.40</b>	67.65	<b>38.17</b>	<b>47.24</b>
RoBERTa-large	sup-SimCSE	54.72	<b>30.89</b>	<b>71.69</b>	38.24	48.89
	SynCSE-scratch + SimCSE_NLI	<b>55.26</b>	30.40	71.53	<b>39.84</b>	<b>49.26</b>

Table 3: Results on the reranking benchmark. Mean Average Precision (MAP) is reported.

Dataset	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg
GenSE ‡	72.09	<b>85.24</b>	<b>79.84</b>	83.25	<b>82.88</b>	83.24	75.33	80.27
DINO †	70.27	81.26	71.25	80.49	77.18	77.82	68.09	75.20
SynCSE-partial (SimCSE based)	<b>76.11</b>	84.49	79.61	<b>85.26</b>	82.60	<b>83.94</b>	<b>81.57</b>	<b>81.94</b>
SynCSE-scratch (SimCSE based)	74.61	83.76	77.89	85.09	82.28	82.71	78.88	80.75

Table 4: Performance comparison of RoBERTa-base trained on various datasets, using the STS benchmark for evaluation. The reported metric is Spearman’s correlation. The “†” symbol is used to indicate results reported in DINO. For SimCSE, we adopted the MNLI+SNLI dataset used in (Gao et al., 2021). “‡”: GenSE released an NLI synthetic dataset comprising over 60 million samples. For a fair comparison, we randomly sampled from it the same number of samples used in the SimCSE dataset.

outperformed all the unsupervised baselines by more than 2 absolute points. Even when compared with supervised settings, our approach achieved performance near that of manual annotation on RoBERTa-base, falling behind by only about 1 point on RoBERTa-large. It’s worth noting that while the supervised SimCSE training dataset (SNLI) and STS test data share a significant overlap in domains (for instance, both STSb and SNLI extensively used Flicker30k data (Plummer et al., 2015)), the domains were not explicitly known while generating the SynCSE-scratch dataset. Interestingly, SynCSE-partial does not always beat SynCSE-scratch as demonstrated in the RoBERTa-large case, which implies the potential of SynCSE-scratch as a promising approach to learn sentence embeddings without using any real data samples. By augmenting annotated NLI

data with the SynCSE-scratch synthetic dataset, our approach outperformed sup-SimCSE significantly, reaching a performance of 84.37% with RoBERTa-large, suggesting that our synthetic data is complementary to human-labeled NLI datasets. “PromptCSE+EH” (Jiang et al., 2022b) achieves competitive performance in the supervised setups. As an orthogonal contribution, however, SynCSE may be combined with the loss function they proposed to further advance the results.

### 3.4 Reranking

Table 3 shows the results of the reranking tasks. Compared to the STS task, the domain of the reranking task data is more divergent from that of the NLI data used for training, as a result, SynCSE-scratch actually outperforms SynCSE-partial significantly, which implies the advantage of SynCSE-

Model	Method	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg
RoBERTa-base	ZeroGen	51.68	71.45	58.80	67.04	70.04	65.00	66.88	64.41
	SynCSE-scratch (SimCSE based)	71.81	83.43	76.90	83.39	<b>82.33</b>	<b>82.89</b>	77.39	79.73
RoBERTa-large	ZeroGen	50.97	70.90	59.97	69.59	68.79	65.43	65.72	64.48
	SynCSE-scratch (SimCSE based)	<b>74.61</b>	<b>83.76</b>	<b>77.89</b>	<b>85.09</b>	82.28	82.71	<b>78.88</b>	<b>80.75</b>

Table 5: Performance comparison of SynCSE-scratch and ZeroGen, using the STS benchmark for evaluation. The Spearman’s correlation is reported.

Model	Method	BIOSSES (Spearman’s correlation)	StackOverflowDupQuestions (Mean Average Precision)
RoBERTa-base	unsup-SimCSE (Wikipedia domain)	68.86	39.25
	SynCSE-scratch (SimCSE based)	<b>80.12</b>	43.22
RoBERTa-large	unsup-SimCSE (Wikipedia domain)	71.96	42.56
	SynCSE-scratch (SimCSE based)	77.73	<b>45.67</b>

Table 6: Performance comparison of the RoBERTa trained on the Wikipedia domain (using the publicly available unsup-SimCSE checkpoint) and specialized domains data generated by SynCSE-scratch.

scratch when in-domain unlabeled sentences are unavailable. SynCSE-scratch also surpasses other unsupervised baselines while SynCSE-partial underperforms them. Moreover, the combination of SynCSE-scratch with manually annotated datasets still facilitates further performance enhancement, substantiating that our method can aid in augmenting existing datasets.

### 3.5 Comparison with Other Synthetic Datasets

In addition to comparing with the MNLI+SNLI datasets used in SimCSE, we also compare our method with three other baselines that leverage synthetic NLI data: (1) GENSE (Chen et al., 2022) aims to automatically annotate the positive and hard negative examples with a LLM trained on existing NLI labeled dataset. We sample the same number of examples from their published dataset as those used in SynCSE; (2) The objective of DINO (Schick and Schütze, 2021) is to generate synthetic data for sentence embeddings as well. In DINO’s most effective configuration, they generate the positive or hard negative samples and assign a similarity score to them based on the prompts used. As they have not made an NLI-style dataset available, we directly report results from their paper, and (3) ZeroGen (Ye et al., 2022a) proposes an efficient unsupervised dataset generation method. We selected those examples that have been provided in both "entailment" and "not\_entailment" sentences to construct sentence pairs, totaling 46,311 pairs, as training data. To ensure a fair comparison, we randomly sampled an equal number of examples gen-

erated by SynCSE-scratch. We compare the generated sentences of our methods with them in Table 11. From the table, we can find that our generated sentence can generate more diverse annotations. As depicted in Table 4, both SynCSE-scratch and SynCSE-partial have achieved performance on the STS task that surpasses that of DINO, GenSE. In a practical setting when generating a dataset from scratch (SynCSE-scratch), we compare our method with ZeroGen (Table 5), and the results show our method significantly outperforms the baseline.

### 3.6 Applying to Specialized Domains

SynCSE is advantageous when dealing with specialized domains where unlabeled data is unavailable. In such cases, traditional methods are not directly applicable. To evaluate SynCSE in this scenario, we conduct experiments on two another datasets focused on specialized domains – the BIOSSES (Soğancıoğlu et al., 2017) dataset of a semantic textual similarity task for the biomedical domain, and the StackOverflowDupQuestions (Liu et al., 2018) dataset of a reranking task for the programming questions domain. Specifically, our experimental design is based on the assumption that we only have access to the names of the target domains (i.e., “biomedicine” and “Stack Overflow website”) without any data available. We run SynCSE-scratch in these settings. Concretely, we first generate 37k unlabeled sentences in the respective domain following the procedure described in Section §2.4, then generate positive and hard negatives for these sentences, and train the models. We use the publicly available unsupervised

Model	Method	MR	CR	SUBJ	MRQA	SST	TREC	MRPC	Avg
<i>Unsupervised methods</i>									
RoBERTa-base	unsup-SimCSE <sup>†</sup>	83.37	87.76	<b>95.05</b>	87.16	89.02	<b>90.80</b>	75.13	86.90
	L2P-CSR <sup>♠</sup>	79.67	88.30	94.27	87.70	87.50	81.14	76.47	85.01
	PCL <sup>‡‡</sup>	81.83	87.55	92.92	87.21	87.26	85.20	76.46	85.49
	PromptRoBERTa <sup>††</sup>	83.82	88.72	93.19	<b>90.36</b>	88.08	90.60	76.75	87.36
	ConPVP <sup>•</sup>	82.44	88.30	93.20	88.74	87.70	87.33	76.15	86.27
	SynCSE-partial (SimCSE based)	85.41	<b>91.44</b>	93.39	89.91	<b>91.21</b>	84.40	<b>76.87</b>	<b>87.52</b>
SynCSE-scratch (SimCSE based)	<b>85.47</b>	<b>91.44</b>	92.53	89.67	90.94	81.60	76.06	86.82	
RoBERTa-large	unsup-SimCSE <sup>†</sup>	84.66	88.56	<b>95.43</b>	87.50	89.46	<b>95.00</b>	72.41	87.57
	L2P-CSR <sup>♠</sup>	80.12	88.53	94.07	88.92	87.04	83.05	76.84	85.51
	PCL <sup>‡‡</sup>	84.47	89.06	94.60	89.26	89.02	94.20	74.96	87.94
	SynCSE-partial (SimCSE based)	87.18	92.02	94.16	90.76	91.65	86.80	<b>76.87</b>	<b>88.49</b>
	SynCSE-scratch (SimCSE based)	<b>87.24</b>	<b>92.16</b>	93.75	<b>90.81</b>	<b>91.87</b>	84.00	76.29	88.02
<i>Supervised methods</i>									
RoBERTa-base	sup-SimCSE <sup>†</sup>	85.08	<b>91.76</b>	94.02	89.72	92.31	<b>91.20</b>	76.52	88.66
	sup-SimCSE	85.05	90.97	94.20	89.37	91.49	88.60	76.87	88.08
	PromptRoBERTa <sup>††</sup>	<b>85.74</b>	91.47	<b>94.81</b>	<b>90.93</b>	<b>92.53</b>	90.40	<b>77.10</b>	<b>89.00</b>
	SynCSE-scratch + SimCSE_NLI	85.51	91.52	93.33	89.87	92.48	83.40	76.06	87.40
RoBERTa-large	sup-SimCSE <sup>†</sup>	88.12	92.37	95.11	90.49	92.75	<b>91.80</b>	76.64	89.61
	sup-SimCSE	87.89	<b>92.61</b>	<b>95.20</b>	90.77	92.86	90.80	<b>77.22</b>	<b>89.62</b>
	SynCSE-scratch + SimCSE_NLI	<b>88.22</b>	92.56	94.76	<b>90.98</b>	<b>93.08</b>	88.00	76.81	89.20

Table 7: Transfer task results of different sentence embedding models (measured as accuracy). The labels “unsup-” and “sup-” correspond to unsupervised and supervised settings, respectively. “<sup>†</sup>”: results from (Gao et al., 2021); “<sup>♠</sup>”: results from (Zhou et al., 2023); “<sup>‡‡</sup>”: results from (Wu et al., 2022a); “<sup>††</sup>”: results from (Jiang et al., 2022a); “<sup>•</sup>”: results from (Zeng et al., 2022). The term “SynCSE-scratch + SimCSE\_NLI” represents our synthetic data combined human labeled NLI dataset used in SimCSE.

Method	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg
Naive Generation	64.65	75.86	62.94	72.79	71.61	72.76	71.57	70.31
SynCSE-scratch	<b>70.89</b>	<b>83.79</b>	<b>76.48</b>	<b>83.28</b>	<b>81.97</b>	<b>82.36</b>	<b>76.14</b>	<b>79.27</b>

Table 8: Performance comparison of our synthetic dataset generation and the “Naive Generation” method.

SimCSE model checkpoint that was trained on the Wikipedia domain for comparison. This is because we assumed no access to unlabeled sentences in these domains, which is a practical setting. Our observations (Table 6) show that SynCSE-scratch outperforms the unsupervised SimCSE baseline significantly in both domains. This experiment further demonstrates the superiority of our method on new domains where no data is available – traditional unsupervised approaches like SimCSE tend to experience a domain transfer drop in performance in such scenarios.

### 3.7 Analysis

In this subsection, we provide an in-depth analysis of SynCSE. All results presented here are based on the RoBERTa-base model.

**Transfer tasks:** Following SimCSE, we execute seven transfer learning 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 (Voorhees and Tice, 2000). These experiments are carried out with the same settings as used in SimCSE. As shown in Table 7, SynCSE-partial outperforms all unsupervised baselines.

**Comparison with the naive generation process:**

To validate the effectiveness of our data synthesis process, we conduct an ablation experiment, where (1) we do not specify topics or genres when generating unlabeled sentences, and (2) we do not vary the prompt and exemplars but fix them the same (that are randomly selected from the pools) when generating the positive and hard negative labels.



	Counselor 1	H2	H3	H4	H5	Avg
Fraction of ethically unsafe data	1%	0%	0%	1%	0%	0.4%

Table 9: The result of the fraction of ethically unsafe data annotated by one psychological counselor and four postgraduate students. H\* means the index of postgraduate annotators.

Other settings are kept the same as in SynCSE-scratch. We perform the ablation experiment on 22k examples. We denote the baseline without diversity control as “Naive Generation” and show them in the Table 8, our method outperforms the Naive Generation baseline by an average of 8.96%, demonstrating the critical role of diversity control in our data synthesis process.

#### Ethical considerations of the synthetic dataset:

To evaluate the safety of our synthetic dataset, we ask five annotators (one of which is a psychological counselor and the other four are postgraduate students) to annotate whether the generated sentences have ethical problems. Specifically, we randomly select 100 sentences from those generated by SynCSE-scratch, and each sentence is independently evaluated by the five people for potential ethical problems. As the Table 9 suggests, only a minor portion of the data is classified as ethically unsafe, indicating that our synthetic dataset upholds a certain level of safety concerning ethical issues. This is not surprising since ChatGPT, the backend in our experiments, is already heavily aligned to avoid producing text with ethical or safety issues.

## 4 Related Work

Prior approaches for sentence embedding fall into two main categories: (1) supervised learning with labeled sentences, and (2) unsupervised sentence embedding with unlabeled sentences. Among these, works based on contrastive learning have proven to be the most effective. For unsupervised methods, SimCSE uses dropout masks to construct positive pairs for learning, while negative examples use in-batch negative examples. Some works employ data augmentation techniques on input sentences, such as word repetition (Wu et al., 2022b), case flipping (Wang et al., 2022c), or a combination of multiple data augmentation strategies to offset the bias caused by mono-augmentation (Wu et al., 2022a). PromptBERT (Jiang et al., 2022a)

uses prompts instead of the  $[CLS]$  token to extract embeddings.

However, these unsupervised methods significantly lag behind their supervised counterparts. Supervised approaches usually derive positive and hard negative samples from labeled NLI datasets (Wang and Lu, 2022; Gao et al., 2021; Jiang et al., 2022a), but these datasets are limited in quantity and domain. Additionally, annotating a new NLI dataset is costly, especially in fields that require trained annotators. Chen et al. (2022) trained a T5 (Chung et al., 2022) model capable of producing positive and hard negative samples, while Ye et al. (2022b) implemented a continuously updated model to modify prompts for generation. However, the performance of these algorithms is still constrained by the performance of generators, which need labeled NLI data for training. Differing from these methods, which necessitate training an additional model, Wang et al. (2022b) proposed a rule-based algorithm capable of generating hard negative annotations. However, its diversity is limited to the prescribed rules. Gilardi et al. (2023) used ChatGPT for dataset annotation. However, their exploration was limited to tasks with explicit answer labels such as "RELEVANT" or "IRRELEVANT". They did not attempt to annotate datasets that required diverse responses. Schick and Schütze (2021) also propose to generate both annotations and unlabeled sentences, while they do not focus on the contrastive learning framework.

## 5 Discussion

In this work, we propose SynCSE, a novel contrastive learning framework for learning sentence embeddings with synthetic data. We prompt LLMs to synthesize unlabeled sentences and their positive and hard negative examples. Furthermore, by utilizing example and prompt pools, we can specify the genre and topic of generated sentences, thereby enhancing the quality of the synthetic dataset. Experiments on both sentence similarity and reranking tasks demonstrate the effectiveness of SynCSE. The performance of SynCSE in this study strongly suggests the potential of synthetic datasets generated by the increasingly advanced LLMs of today. We envision that, through the effective use of prompting strategies with LLMs, synthetic datasets produced by these models could potentially serve as promising alternatives to real-world data across a wide range of 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, 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.
- Ken Barker, Parul Awasthy, Jian Ni, and Radu Florian. 2021. [IBM MNLP IE at CASE 2021 task 2: NLI reranking for zero-shot text classification](#). In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 193–202, Online. Association for Computational Linguistics.
- Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. [SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.
- 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.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- 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.
- 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.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.
- Hongliang He, Junlei Zhang, Zhenzhong Lan, and Yue Zhang. 2023. Instance smoothed contrastive learning for unsupervised sentence embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12863–12871.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.
- Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022a. [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.
- Yuxin Jiang, Linhan Zhang, and Wei Wang. 2022b. [Improved universal sentence embeddings with prompt-based contrastive learning and energy-based learning](#). In *Findings of the Association for Computational*

- Linguistics: EMNLP 2022*, pages 3021–3035, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Ann Lee, Michael Auli, and Marc’Aurelio Ranzato. 2021. **Discriminative reranking for neural machine translation**. 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 7250–7264, Online. Association for Computational Linguistics.
- 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.
- Peerat Limkonchotiwat, Wuttikorn Ponwitayarat, Lalita Lowphansirikul, Can Udomcharoenchaikit, Ekapol Chuangsuwanich, and Sarana Nutanong. 2022. **ConGen: Unsupervised control and generalization distillation for sentence representation**. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6467–6480, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jiduan Liu, Jiahao Liu, Qifan Wang, Jingang Wang, Wei Wu, Dongyan Zhao, and Rui Yan. Rankscse: Unsupervised sentence representations learning via learning to rank.
- Xueqing Liu, Chi Wang, Yue Leng, and ChengXiang Zhai. 2018. Linkso: a dataset for learning to retrieve similar question answer pairs on software development forums. In *Proceedings of the 4th ACM SIGSOFT International Workshop on NLP for Software Engineering*, pages 2–5.
- Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. **A SICK cure for the evaluation of compositional distributional semantic models**. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Amrita 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.
- 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.
- Graham Neubig and Zhiwei He. 2023. Zeno GPT Machine Translation Report.
- OpenAI. 2022. **Chatgpt: Optimizing language models for dialogue**. *OpenAI Blog*.
- OpenAI. 2023. Gpt-4 technical report. *arXiv*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- Bo Pang and Lillian Lee. 2004. **A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts**. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.
- Bo Pang and Lillian Lee. 2005. **Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales**. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.
- Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.
- 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.
- 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.
- Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. 2017. Biosses: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58.
- 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)*.

- Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.
- 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, Yangguang Li, Zhen Huang, Yong Dou, Lingpeng Kong, and Jing Shao. 2022b. Sncse: Contrastive learning for unsupervised sentence embedding with soft negative samples. *arXiv preprint arXiv:2201.05979*.
- Tianduo Wang and Wei Lu. 2022. **Differentiable data augmentation for contrastive sentence representation learning**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7640–7653, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Wei Wang, Liangzhu Ge, Jingqiao Zhang, and Cheng Yang. 2022c. Improving contrastive learning of sentence embeddings with case-augmented positives and retrieved negatives. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2159–2165.
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39:165–210.
- Adina Williams, Nikita Nangia, and Samuel 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.
- 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, Xubo 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, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022b. **ESim-CSE: 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.
- Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022a. **ZeroGen: Efficient zero-shot learning via dataset generation**. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jiacheng Ye, Jiahui Gao, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2022b. **ProGen: Progressive zero-shot dataset generation via in-context feedback**. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3671–3683, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jiali Zeng, Yongjing Yin, Yufan Jiang, Shuangzhi Wu, and Yunbo Cao. 2022. **Contrastive learning with prompt-derived virtual semantic prototypes for unsupervised sentence embedding**. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7042–7053, Abu Dhabi, United Arab Emirates. 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.
- Kun Zhou, Yuanhang Zhou, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Learning to perturb for contrastive learning of unsupervised sentence representations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

## A Prompt pools

In order to increase the diversity of input prompts, we designed a variety of prompts for generating positive samples, hard negative samples, and unlabeled data, which are adopted during generation based on certain probabilities. The specific prompts are displayed in Tables 12, 1, and 15. Given that generating image captions differs somewhat from generating other types of text, we have designed unique prompts for image captions to further enhance diversity, as illustrated in Table 16.

## B Genres and Topics

**Genres:** When generating unlabeled sentences, to make the newly generated sentences as different as possible from existing data, we specify the genre and topic of the new sentences. As long as the genre and topic of the new sentences are different from existing ones, the probability of these new sentences providing more new information to the dataset becomes higher. In this paper, we use 20 different genres (Table 10) and 31 different topics (Table 2). Before generating sentences, we use GPT-4 to generate 30 examples for each genre as one-shot example sentences. When using them, we first specify a genre and fill it into the "[*the description of the genre*]" in the prompt of Table 15, then randomly choose 6 from the topic list to fill into "[*topic<sub>i</sub>*]" . These descriptions are adapted from the genre specifications provided by GPT-4, thus, creating new descriptions does not require a significant effort.

**Topics:** We leveraged GPT-4 to generate an array of diverse topics, and 37 of these were randomly selected as the thematic grounding for our generation of unlabeled sentences. Concretely, these themes are: nature, technology, food, sports, culture, history, animals, environment, politics, finance, education, social issues, global issues, entertainment, healthcare, war, mathematical and electrical engineering, crime, relationships and emotional bonds, magic and mythical creatures, personal life stories, business strategies, fitness and mental health, global warming and conservation, various forms of art and cultural practices, teaching methodologies and learning styles, recipes and culinary techniques, ethical dilemmas and existential questions, space exploration and celestial phenomena, legal issues and courtroom drama, examination of past events and civilizations, ancient myths and legends,

scientific theories, life stories of notable individuals, COVID-19, immigration policies, and mental health.

Genre descriptions	
1	in-person conversations
2	letters
3	reports, speeches, letters, and press releases from public domain government websites
4	fiction
5	image descriptions
6	video descriptions
7	news from websites or newspapers
8	reviews and critiques for shopping
9	headlines of news
10	dialogue of technical or Instructional tutorial
11	informative and expository texts which provide guidance or explanations related to a wide range of topics
12	STEM examination questions
13	travelogue
14	historical description
15	plots involving political intrigue and maneuvering
16	formal writings that present research findings or scholarly discussion
17	speeches given by politicians, often with the intent of persuading or informing an audience about political topics
18	written works such as poetry, drama, and novels
19	pieces of content shared on social media platforms
20	short image captions
21	messages paid for by a business or individual to promote a product, service, or event

Table 10: The list of genre descriptions.

## C Hyperparameters

We employed gpt-3.5-turbo-0301 for sentence generation. For the generation of unlabeled sentences, we set the temperature to 1.3, top\_p to 1.0, and both presence\_penalty and frequency\_penalty to 0.3. The input prompts were one-shot prompts; in the example, 10 sentences were generated at once, and during the generation process, 20 sentences were generated at once. During the generation of positive sample annotations, we set the temperature to 1.0 and top\_p to 0.9. In the generation of neg-

Method	Entailment	Contradiction
<i>Input: A young man is getting ready to release a red kite.</i>		
DINO	A young man releasing a red kite. A man getting ready to release a red kite.	A red kite releasing a red kite. It was a big deal to him and he didn't know how he would explain it to his parents
GenSE	The man is prepared to fly the kite. A man is planning to fly a kite.	A man is playing basketball. The woman is flying a kite
SynCSE-scratch(ours)	A young man is preparing to let go of a red kite. A man prepares to fly a crimson kite.	A young man getting ready to release a blue kite. A young man gets ready to catch a red kite that has been released.
<i>Input: One of the hotel's rooms</i>		
DINO	The hotel room One of the hotel rooms	The other one is on fire I have no idea what that is.
GenSE	A room inside a hotel. A hotel room.	It's not the hotel's room. There is no room at the hotel.
SynCSE-scratch(ours)	A room in the hotel. A hotel room.	None of the hotel's rooms. All of the hotel's rooms were fully booked for the weekend.

Table 11: Comparison of different data synthesis methods. For samples of DINO and GenSE, we cite the generation sentences reported in (Ye et al., 2022b).

### Positive prompts pools

**Prompt1:** Please paraphrase the input sentence or phrase, providing an alternative expression with the same meaning.

**Prompt2:** Rewrite the following sentence or phrase using different words and sentence structure while preserving its original meaning.

**Prompt3:** Create a sentence or phrase that is also true, assuming the provided input sentence or phrase is true.

**Prompt4:** 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.

Table 12: Positive prompts pools. During the generation of positive samples, a prompt is sampled with a certain probability and inserted into the few-shot input prompts in Table 2, which are input in the form of multi-turn dialogues.

	Unlabeled Sentence	Positive Label	Hard Negative Label	All
cost (% per sentence)	0.00007	0.00067	0.00076	0.0015

Table 13: The cost analysis of our method generating sentences with gpt-3.5-turbo.

ative sample annotations, we set the temperature to 1.0 and top\_p to 0.95. Both positive and negative sample generations were 5-shot. Our training framework is based on SimCSE, which forcibly truncates parts of the sentence exceeding 32 words during training. To maintain a fair comparison, we filter out sentences with more than 32 words before training with the SimCSE framework after generating sentences with SynCSE-scratch.

## D Cost of the synthesize data

We used gpt-3.5-turbo to synthesize data that is not very expensive, currently costing 0.0015 dollars per 1K tokens for input and 0.002 dollars per 1K tokens for output. Concretely, there are three parts in the data generation process: unlabeled sentences, positive labels, and hard negative labels. Since the length of each input varies, to quantify the cost, we randomly sampled 40 inputs and calculated the average cost per sentence. As detailed in Table 13, our method cost a total of around 1.5 \$ for generating 1000 sentences, and the total cost of producing the 276k sentences used in our experiments of SynCSE-scratch in Table 2 is around 414 \$. In the domain specialized task (Table 6), we just generate 37k sentence pairs and significantly surpass SimCSE in the target domain, and the cost is around 55 \$. We would like to highlight that

<b>Data</b>	0%	20%	40%	60%	80%	100%
<b>Avg. STS</b>	82.04	82.10	82.73	82.58	<b>82.75</b>	82.61

Table 14: Performance of SimCSE\_NLI when combined with varying amounts of our synthetic SynCSE-scratch dataset. We report the performance on the avg STS results on the test set.

the rate per sentence above is much cheaper than manually labeling data; for instance, in machine translation tasks, human translation (around \$0.1 per word) can be thousands of times costlier than using gpt-3.5-turbo (Neubig and He, 2023).

## E Synthetic data amount

We also analyzed the impact on performance when augmenting the volume of generated data on the manually curated dataset, as shown in Table 14. Since the domain of SynCSE-scratch is established upon its completion, the performance ceases to increase after a certain amount of SynCSE-scratch data is added to SimCSE. This may be due to the fact that the added data is randomly sampled, which likely already covers the domain of SynCSE-scratch.

---

**Prompt pool for generating unlabeled sentences with specified genres and topics.**

---

**Prompt1:** Devise *[number]* distinct and diverse sentences that may appear in *[the description of the genre]*, covering a range of subjects (*[topic<sub>1</sub>]*, *[topic<sub>2</sub>]*, *[topic<sub>3</sub>]*, *[topic<sub>4</sub>]*, *[topic<sub>5</sub>]*, *[topic<sub>6</sub>]*, etc.). These sentences should present a mix of complexity levels, from elementary structures akin to "Birds fly in the sky." to more sophisticated ones. Aim for a low degree of lexical overlap and an extensive vocabulary. Incorporate a variety of sentence modes - declarative, interrogative, exclamatory, imperative, and descriptive. The sentences should oscillate in tone between informative, persuasive, descriptive, and narrative, and should present varied perspectives. Vary the length of the sentences, ranging from concise phrases of 3-5 words to longer sentences containing 20-40 words.

---

**Prompt2:** Construct *[number]* varied and diverse sentences that may appear in *[the description of the genre]* encompassing a multitude of topics such as *[topic<sub>1</sub>]*, *[topic<sub>2</sub>]*, *[topic<sub>3</sub>]*, *[topic<sub>4</sub>]*, *[topic<sub>5</sub>]*, *[topic<sub>6</sub>]* and so on. Aim for low lexical repetition and a rich vocabulary variety. Ensure to blend different sentence structures - declarative, interrogative, exclamatory, imperative, and descriptive. The sentences should convey different tones, including informative, persuasive, descriptive, and narrative, and should be expressed from various perspectives. Vary the length of the sentences, ranging from concise phrases of 3-5 words to longer sentences containing 20-30 words.

---

**Prompt3:** Create *[number]* varied and diverse sentences that may appear in *[the description of the genre]*, spanning an array of topics such as *[topic<sub>1</sub>]*, *[topic<sub>2</sub>]*, *[topic<sub>3</sub>]*, *[topic<sub>4</sub>]*, *[topic<sub>5</sub>]*, *[topic<sub>6</sub>]* and so on. Include a mix of sentence styles - declarative, interrogative, exclamatory, imperative, and descriptive. Vary the length of the sentences, ranging from concise phrases of 3-5 words to 25-35 words.

---

**Prompt4:** Compose *[number]* assorted and diverse sentences that may appear in *[the description of the genre]*, touching upon various themes like *[topic<sub>1</sub>]*, *[topic<sub>2</sub>]*, *[topic<sub>3</sub>]*, *[topic<sub>4</sub>]*, *[topic<sub>5</sub>]*, *[topic<sub>6</sub>]* and so on. The sentences should reflect a spectrum of complexity levels, from basic structures such as "The sun sets in the west." to more elaborate forms. Aspire for minimal lexical redundancy and a wide array of vocabulary. Incorporate a blend of sentence types - declarative, interrogative, exclamatory, imperative, and descriptive. The sentences should shift in tone, cycling between informative, persuasive, descriptive, and narrative, and should vary in perspective. Vary the length of the sentences, ranging from concise phrases of 3-5 words to longer sentences containing 25-45 words.

---

Table 15: Prompt pool for generating unlabeled sentences with specified genres and topics. When generating unlabeled sentences, we randomly sample a prompt. Here, "*[number]*" indicates the number of sentences generated each time, "*[the description of the genre]*" provides a specific description of the given genre, and "*[topic<sub>i</sub>]*" is a topic sampled from the topic list.



---

**Prompt pool for generating image captions.**

---

**Prompt1:** Please randomly generate  $[number]$  diverse sentences in the style of  $[the\ description\ of\ the\ genre]$ , similar to image captions in the Flickr30k dataset, covering a wide range of subjects, actions, contexts, and settings. The sentences do not need to be semantically related. Please make sure to generate sentences with a uniform distribution in length, ranging from short phrases to longer ones.

---

**Prompt2:** Kindly generate  $[number]$  unique sentences reflecting the style of  $[the\ description\ of\ the\ genre]$ , drawing inspiration from the diverse scenarios found in the Flickr30k dataset. The sentences should cover an extensive array of themes, actions, surroundings, and situations. Please create a mix of simple sentences like "A man strums a guitar", along with more elaborate ones.

---

**Prompt3:** Kindly generate  $[number]$  distinct sentences with simple structures, each may appear in  $[the\ description\ of\ the\ genre]$ . These sentences should touch on a wide array of topics, actions, and environments without necessarily having a semantic link. Strive to provide sentences of various lengths, from short to moderately long, all while maintaining simplicity and clarity.

---

**Prompt4:** Please randomly generate  $[number]$  diverse and descriptive sentences which may appear in  $[the\ description\ of\ the\ genre]$ , ensuring they mirror the structure of the following image captions:  $[example_1]$ ,  $[example_2]$ ,  $[example_3]$ ,  $[example_4]$ . Each sentence should vividly portray the primary activity in a hypothetical image or segment of a video, articulating the main subject(s), their actions, and any important objects or secondary subjects involved.

---

Table 16: The Prompt pool for image caption generation. We designed a separate prompt for generating image captions to enhance diversity. Here, " $[number]$ " denotes the number of sentences to generate, " $[the\ description\ of\ the\ genre]$ " provides a description for generating captions, and " $[example_4]$ " is a randomly sampled example sentence.