Narrowing the Gap between Supervised and Unsupervised Sentence Representation Learning with Large Language Model

Mingxin Li[1], Richong Zhang[1,2*], Zhijie Nie[1,3], Yongyi Mao[4]
[1]SKLSDE, School of Computer Science and Engineering, Beihang University, Beijing, China
[2]Zhongguancun Laboratory, Beijing, China
[3]Shen Yuan Honors College, Beihang University, Beijing, China
[4]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
{limx,zhangrc,niezj}@act.buaa.edu.cn, ymao@uottawa.ca

Abstract

Sentence Representation Learning (SRL) is a fundamental task in Natural Language Processing (NLP), with the Contrastive Learning of Sentence Embeddings (CSE) being the mainstream technique due to its superior performance. An intriguing phenomenon in CSE is the significant performance gap between supervised and unsupervised methods, with their only difference lying in the training data. Previous works attribute this performance gap to differences in two representation properties (alignment and uniformity). However, since alignment and uniformity only measure the results, they fail to answer "*What* aspects of the training data contribute to the performance gap?" and "*How* can the performance gap be narrowed?". In this paper, we conduct empirical experiments to answer these "*What*" and "*How*" questions. We first answer the "*What*" question by thoroughly comparing the behavior of supervised and unsupervised CSE during their respective training processes. From the comparison, we identify the similarity pattern as a key factor to the performance gap, and introduce a metric, called *Relative Fitting Difficulty* (RFD), to measure the complexity of the similarity pattern. Then, based on the insights gained from the "*What*" question, we tackle the "*How*" question by increasing the pattern complexity of the training data. We achieve this by leveraging the In-Context Learning (ICL) capability of the Large Language Model (LLM) to generate data that simulates complex patterns. By utilizing the hierarchical patterns in the LLM-generated data, we effectively narrow the gap between supervised and unsupervised CSE. We release our codes and appendix at https://github.com/BDBC-KG-NLP/NGCSE.

Introduction

Sentence Representation Learning (SRL) is a crucial task in Natural Language Processing (NLP), which learns representations (or embeddings) for sentences in the feature space. It is a fundamental task that underpins many NLP applications, including Semantic Textual Similarity (STS) (Wang and Isola 2020), Information Retrieval (IR) (Cer et al. 2018), and text classification (Pang and Lee 2004). Contrastive learning of Sentence Embeddings (CSE) has been recently introduced into SRL (Yan et al. 2021; Gao, Yao, and Chen 2021) and has drawn much attention as it significantly improves the performance of sentence embeddings. CSE can be trained in both supervised and unsupervised settings, where the primary difference is the training data. However, with only this difference, supervised CSE can outperform unsupervised CSE on STS tasks by a large margin. Gao, Yao, and Chen (2021) explain this performance gap by referring to two properties (alignment and uniformity) from (Wang and Isola 2020). Specifically, compared to the representations trained by the unsupervised method, they find that the representations trained by supervised data exhibit better alignment, uniformity or both (as shown in Figure 2), thereby resulting in better performance on STS tasks. This explanation still rests on the final results, and cannot explain the mechanism that led to these results. In this paper, we focus on the training data and its relationship with the performance gap. Specifically, we pose two questions: "*What* aspects of the training data contribute to the performance

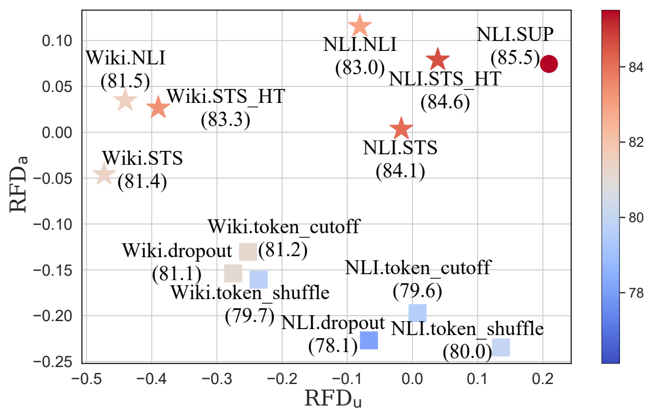


Figure 1: $\mathrm{RFD_u}$-$\mathrm{RFD_a}$ plot of models based on $\mathrm{BERT_{base}}$. The colors of the points and the numbers in brackets represent Spearman's correlation in the evaluation data, i.e., the validation split of STS Benchmark dataset. RFD is a metric we propose to measure the complexity of the similarity pattern in the training data. This metric helps us answer the "*What*" question and further leads us to address the "*How*" question. "■" is trained with unsupervised data, "●" is trained with supervised data, and "★" is trained with data generated from the LLM.

*Corresponding author

gap?" and "*How* can the performance gap be narrowed?", and answer them with empirical experiments in this study.

To answer the "*What*" question, we record the variety of alignment and uniformity in the training processes of both supervised and unsupervised CSE, where we identify the similarity pattern, i.e., how a dataset defines similar and dissimilar sentence pairs, as a key factor to the performance gap. The more complex the similarity pattern of a training dataset, the higher the performance that training with such a dataset can yield. We also find that the complexity of the similarity pattern (pattern complexity for short) can be measured by the relative magnitude of alignment and uniformity between the training data and the evaluation data. More specifically, the similarity pattern of the supervised training data is more difficult to fit than that of the evaluation data, resulting in higher alignment and uniformity values in the training data than those in the evaluation data. In contrast, the similarity pattern of the unsupervised training data is simpler to fit, resulting in lower alignment and uniformity values. Therefore, we define a metric called *Relative Fitting Difficulty* (RFD) to measure the pattern complexity and provide the answer to the "*What*" question: the increase of the pattern complexity leads to the performance gap between supervised and unsupervised CSE.

Based on the insight gained from answering the "*What*" question, we answer the "*How*" question by introducing complex similarity patterns into the unsupervised training data. This is achieved by leveraging the In-Context Learning (ICL) capability of the Large Language Model (LLM) (Brown et al. 2020) to simulate the similarity patterns in STS (Agirre et al. 2012) and NLI (Gao, Yao, and Chen 2021) datasets. Furthermore, we notice the hierarchical nature of the STS dataset, where the semantic similarity between two sentences is measured with a score ranging from 0 to 5, rather than simply classified as similar or dissimilar. This finding motivates us to simulate the hierarchical pattern of the STS dataset. And to utilize the hierarchical pattern, we propose a loss called Hierarchical Triplet (HT) loss to ensure that such a pattern can be learned during training, which helps us further narrow the performance gap.

Briefly, our main contributions are as follows:

- We propose a new metric, i.e., *Relative Fitting Difficulty* (RFD), to measure the complexity of the similarity pattern and demonstrate that the higher RFDs on both alignment and uniformity correlate with better performance on STS tasks;
- We narrow the performance gap on STS tasks between supervised and unsupervised CSE by introducing the training data with complex similarity patterns, which is obtained by the ICL capability of LLMs, and introduce a novel loss function called Hierarchical Triplet (HT) loss to utilize the hierarchical patterns effectively;
- We conduct extensive further experiments to validate our findings on RFDs and to verify the effectiveness of our proposed methods in narrowing the performance gap.

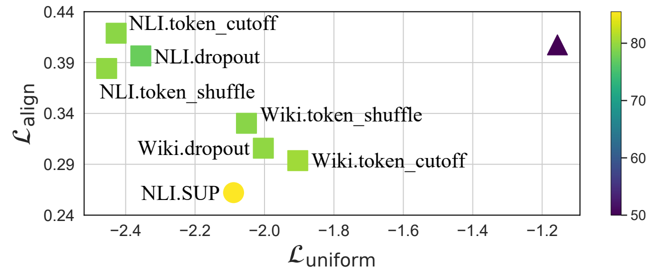


Figure 2: Uniformity-alignment plot of models based on $\text{BERT}_{\text{base}}$. The colors of the points and the numbers in brackets represent Spearman's correlation in the validation split of STS Benchmark dataset. "▲" is the pre-trained model, "■" is trained with unsupervised data, and "●" is trained with supervised data.

Background

In this section, we first detail the differences in training and performance between supervised and unsupervised CSE to help readers understand the background better, and then give some notations for the convenience of the later narrative.

Performance Gap on the STS Tasks

Contrastive learning of Sentence Embeddings (CSE) (Yan et al. 2021; Gao, Yao, and Chen 2021) is a prevalent technique in SRL for its superior performance. CSE refines the sentence embeddings in two ways: 1) pulling the anchor sentence s_i and its semantically similar sentence (or **positive sentence** s_i^{p}) closer; 2) pushing the anchor sentence s_i and its semantically dissimilar sentence (or **negative sentence** s_i^{n}) apart. The commonly used contrastive learning loss, InfoNCE (Oord, Li, and Vinyals 2018), can be expressed as

$$\mathcal{L}_{\mathrm{C}} = -\log \frac{e^{f(s_i)^\top f(s_i^{\mathrm{p}})/\tau}}{e^{f(s_i)^\top f(s_i^{\mathrm{p}})/\tau} + \sum_{j=1}^{N} e^{f(s_i)^\top f(s_{i,j}^{\mathrm{n}})/\tau}}, \quad (1)$$

where τ is a hyper-parameter and $\{s_{i,j}^{\mathrm{n}}\}_{j=1}^{N}$ is a set of negative sentences corresponding to s_i.

There are two settings of CSE, namely Supervised CSE (S-CSE) and Unsupervised CSE (U-CSE). Many improvements have been proposed based on both S-CSE and U-CSE (Jiang et al. 2022; Wu et al. 2022), but we focus on the typical paradigm in this study. For U-CSE, the augmented view of s_i is treated as its positive sentence s_i^{p}, and a randomly sampled sentence is treated as its negative sentence s_i^{n}. The common data augmentation methods used to generate this augmented view include dropout, token shuffle, and token cutoff (Yan et al. 2021). U-CSE is typically trained with a Wikipedia dataset (Gao, Yao, and Chen 2021), consisting of one million sentences extracted from Wikipedia. For S-CSE, it relies on human-annotated s_i^{p} and s_i^{n}, with the supervision signal from the Natural Language Inference (NLI) task (Conneau et al. 2017). Specifically, given a premise, the entailment hypothesis is treated as s_i^{p} and the contradiction hypothesis is treated as s_i^{n}. Gao, Yao, and Chen (2021) collect a widely used NLI dataset (Gao, Yao, and Chen 2021) for CSE.

The performance of CSE is usually evaluated with the SentEval toolkit (Conneau and Kiela 2018), which includes the STS tasks and the transfer tasks. The STS task quantifies the semantic similarity between two sentences with a score ranging from 0 to 5 and takes Spearman's correlation as the metric for performance. There are seven STS datasets are included for evaluation: STS 2012-2016 (Agirre et al. 2012, 2013, 2014, 2015, 2016), STS Benchmark (Cer et al. 2017), SICK Relatedness (Marelli et al. 2014). The transfer tasks evaluate the transfer capability of sentence embeddings by performing logistic regression. There are also seven datasets included for the evaluation of transfer task: MR (Pang and Lee 2005), CR (Hu and Liu 2004), SUBJ (Pang and Lee 2004), MPQA (Wiebe, Wilson, and Cardie 2005), SST-2 (Socher et al. 2013), TREC (Voorhees and Tice 2000), and MRPC (Dolan and Brockett 2005).

Both S-CSE and U-CSE exhibit strong performance in the transfer tasks, but there exists a significant performance gap in the STS task, even when the sole difference between them lies in the training data. Gao, Yao, and Chen (2021) borrow two properties, alignment and uniformity, from the empirical work of Wang and Isola (2020) to better understand it, and the two properties can be expressed as

$$\mathcal{L}_{\text{align}} \triangleq \mathop{\mathbb{E}}_{(s,s^{\text{p}})\sim p_{\text{pos}}} \|f(s) - f(s^{\text{p}})\|_2^{\alpha} \tag{2}$$

$$\mathcal{L}_{\text{uniform}} \triangleq \log \mathop{\mathbb{E}}_{(s_i,s_j)\overset{\text{i.i.d.}}{\sim} p_{\text{data}}} e^{-t\|f(s_i)-f(s_j)\|_2^2} \tag{3}$$

where α and t are two hyper-parameters, p_{pos} is the distribution of positive sentence pairs, and p_{data} is the distribution of sentences. These properties measure the quality of the sentence embeddings. Gao, Yao, and Chen (2021) has shown that the performance gap results from the better alignment and uniformity of S-CSE in comparison to U-CSE (as shown in Figure 2). This explains the result of the performance gap, but does not shed light on the question of "*What* aspects of the training data contribute to the performance gap?". In this study, we seek to answer this "*What*" question. Furthermore, based on the insights from the "*What*" question, we will explore "*How* can the performance gap be narrowed?".

Notation

We conduct experiments under various training data settings to study how the training data affects the performance of CSE. To maintain consistency, we organize all settings using the same naming format: "**[Data-Domain].[Similarity-Pattern]**", where [Data-Domain] represents how the anchor sentences are collected, and [Similarity-Pattern] represents how the positive and negative sentences are defined. Two data domains are included: (1) Wiki, consisting of sentences from Wikipedia (Gao, Yao, and Chen 2021); (2) NLI, consisting of the premises from the NLI dataset (Gao, Yao, and Chen 2021). And similarity patterns are divided into three types, including supervision signals (denoted as SUP), data augmentations, and our proposed pattern simulation techniques, which will be explained in later sections.

What Aspects of the Training Data Contribute to the Performance Gap?

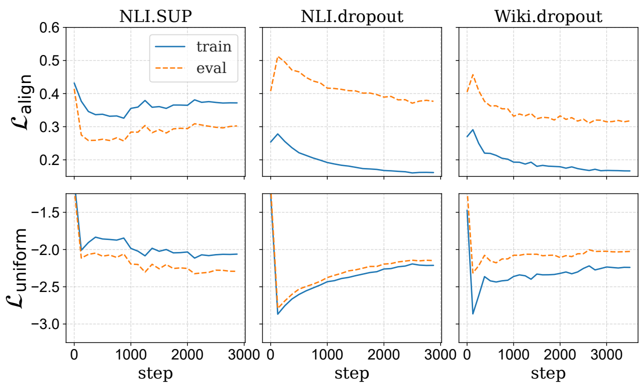


Figure 3: Alignment and uniformity in both the held-out training data and the evaluation data during the training process. We only plot the results of "NLI.SUP", "NLI.dropout", and "Wiki.dropout" here for comparison. The results of "token_shuffle" and "token_cutoff" are similar to those of "dropout", so we plot them in the appendix.

Observation in the Training Process

In this section, we study how the training data affects the performance of CSE. To ensure the results are only correlated with training data, we select pre-trained BERT$_{\text{base}}$ model (Devlin et al. 2019) as the backbone and ensure identical settings across all training, with details illustrated in the **appendix**. Then, we record changes in alignment and uniformity in both the training and evaluation data during the training process and observe how these changes vary between S-CSE and U-CSE. Note that the widely-used training data for S-CSE ("NLI.SUP") and U-CSE ("Wiki.dropout", "Wiki.token_cutoff", "Wiki.token_shuffle") differ in both their data domains and similarity patterns. To study the impact of these two factors separately, we introduce three additional training data settings to U-CSE: "NLI.dropout", "NLI.token_cutoff", and "NLI.token_shuffle". Part of the results are shown in Figure 3.

We first fix the similarity pattern to investigate the impact of the data domain, e.g., comparing "NLI.dropout" with "Wiki.dropout". The results in Figures 2 and 3 show that the performance of the STS task and the trends in alignment and uniformity are similar across different data domains, indicating that the data domain does not influence the performance significantly. Next, we fix the data domain to investigate the impact of the similarity pattern, e.g., comparing "NLI.SUP" with "NIP.dropout". From the comparison, we can observe the performance gap in Figure 2, indicating that the similarity pattern is a key factor in such phenomenon. Furthermore, Figure 3 shows that the changes in alignment and uniformity during the training process in S-CSE are quite different from those in U-CSE. Specifically, S-CSE exhibits higher alignment and uniformity values in the held-out training data than those in the evaluation data. In contrast, U-CSE exhibits lower alignment and uniformity values in the held-out training data than those in the evaluation data. We argue that this difference in the training process, in fact, reflects the difference in the complexity of the similarity pattern (pattern complexity for short).

Relative Fitting Difficulty

The similarity pattern in S-CSE, which is defined by supervision signals, is far more complex than the similarity pattern in U-CSE, which is defined by data augmentations. Moreover, the similarity pattern in S-CSE training data is more difficult to fit than that of the evaluation data, while the similarity pattern in U-CSE training data is simpler to fit than that of the evaluation data. This difference in the pattern complexity results in the difference in the training process between S-CSE and U-CSE. Therefore, we introduce a metric called *Relative Fitting Difficulty* (RFD) to act as an indicator of the **pattern complexity**. RFD is defined as the difference in fitting difficulty between the held-out training data and the evaluation data, i.e., the relative magnitude of alignment and uniformity between the held-out training data and the evaluation data during the training process.

Let the alignment and uniformity of a sentence encoder f at time step t in the held-out training data be denoted by $a_{\mathrm{h}}(f,t)$ and $u_{\mathrm{h}}(f,t)$, while those in the evaluation data be denoted by $a_{\mathrm{e}}(f,t)$ and $u_{\mathrm{e}}(f,t)$. We can then define the RFD for alignment over a set of time steps $T = \{t_i\}_{i=1}^{M}$ as

$$\mathrm{RFD}_{\mathrm{a}}(f,T) = \frac{1}{M}\sum_{i=1}^{M} a_{\mathrm{h}}(f,t_i) - a_{\mathrm{e}}(f,t_i), \qquad (4)$$

and the RFD for uniformity as

$$\mathrm{RFD}_{\mathrm{u}}(f,T) = \frac{1}{M}\sum_{i=1}^{M} u_{\mathrm{h}}(f,t_i) - u_{\mathrm{e}}(f,t_i). \qquad (5)$$

We calculate the RFD for the six U-CSE settings and one S-CSE setting mentioned in the last subsection, and present the results in Figure 1 using "■" and "●". By comparing the results within each data domain, we can observe two facts: (1) when one setting has both lower $\mathrm{RFD}_{\mathrm{a}}$ and $\mathrm{RFD}_{\mathrm{u}}$ values than another setting, it will have lower performance in the STS task accordingly; (2) When either $\mathrm{RFD}_{\mathrm{a}}$ or $\mathrm{RFD}_{\mathrm{u}}$ value increases, the performance in the STS task tends to improve. These observations show a tendency that higher RFD values correspond to better performance. In other words, compared to U-CSE, S-CSE has higher fitting difficulty in alignment and uniformity, i.e., higher pattern complexity, which leads to better performance in the STS task. In fact, we conjecture it may be the answer to the what question.

However, this answer is drawn from only seven points in two data domains. If we need to get this answer more conclusive, we need to experiment under more settings and to get more RFD coordinates and their corresponding STS task performance. Therefore, in the next section, we will introduce some artificial settings and explore the correlation between their RFDs and STS performance to further corroborate the conclusions of this section.

How Can the Performance Gap Be Narrowed?

In this section, we answer the "*How*" based on the insights gained from the "*What*". At the same time, we will provide additional validation for our answer to the "*What*" question.

Pattern Simulation With LLM

Our answer to the "*What*" question reveals the correlation between the pattern complexity in training data and the STS task performance. Therefore, to narrow the performance gap, we propose to increase the RFD of U-CSE by introducing complex similarity patterns (patterns for short) into U-CSE. To realize this, we leverage the In-Context Learning (ICL) (Brown et al. 2020) capability of LLM to simulate the patterns in NLI and STS datasets. We adopt the `gpt-3.5-turbo-0613` as the LLM through the official API [1] from OpenAI with default parameters.

Figure 4 illustrates the overall procedure of pattern simulation and dataset generation. The datasets are generated from two types of sources: a corpus source and a pattern source. For the corpus source, we consider the two data domains used in the previous experiments: (1) Wiki, which consists of sentences from Wikipedia, and (2) NLI, which consists of the premises from the NLI dataset. For the pattern source, we also consider two classes of patterns: (1) STS patterns, which adopt the training split of the STS12 dataset (Agirre et al. 2012) as the source of patterns, and (2) NLI patterns, which adopt the same NLI dataset as previously as the source patterns.

When generating datasets, we randomly sample 20,000 sentences from the corpus source and use the LLM to simulate STS and NLI patterns separately. We refer to these sentences as source sentences. For every source sentence s_i, we subsequently generate a positive sentence s_i^{p} and a negative sentence s_i^{n} using the following process:

1. **Sampling Pattern Example**: To simulate STS patterns, We randomly sample three sentence pairs with STS scores above 4 as examples for the pattern of positive sentence pairs (referred to as positive examples), and three sentence pairs with STS scores below 1 as examples for the pattern of negative sentence pairs (referred to as negative examples). Similarly, to simulate NLI patterns, we randomly sample three premises and their entailment hypotheses as positive examples, and three premises and their contradiction hypotheses as negative examples;
2. **Generating Positive Sentence** s_i^{p}: To generate s_i^{p} that simulates STS patterns, we prompt the LLM with positive examples to generate a sentence that is semantically similar to s_i. And to generate s_i^{p} that simulate NLI patterns, we prompt the LLM with positive examples to generate a sentence that is an entailment hypothesis to s_i;
3. **Generating Negative Sentence** s_i^{n}: To generate s_i^{n} that simulates STS patterns, we prompt the LLM with negative examples to generate a sentence with a distinct meaning compared to s_i^{p}. And to generate s_i^{n} that simulate NLI patterns, we prompt the LLM with negative examples to generate a sentence that contradicts s_i^{p}.

[1] https://openai.com/api

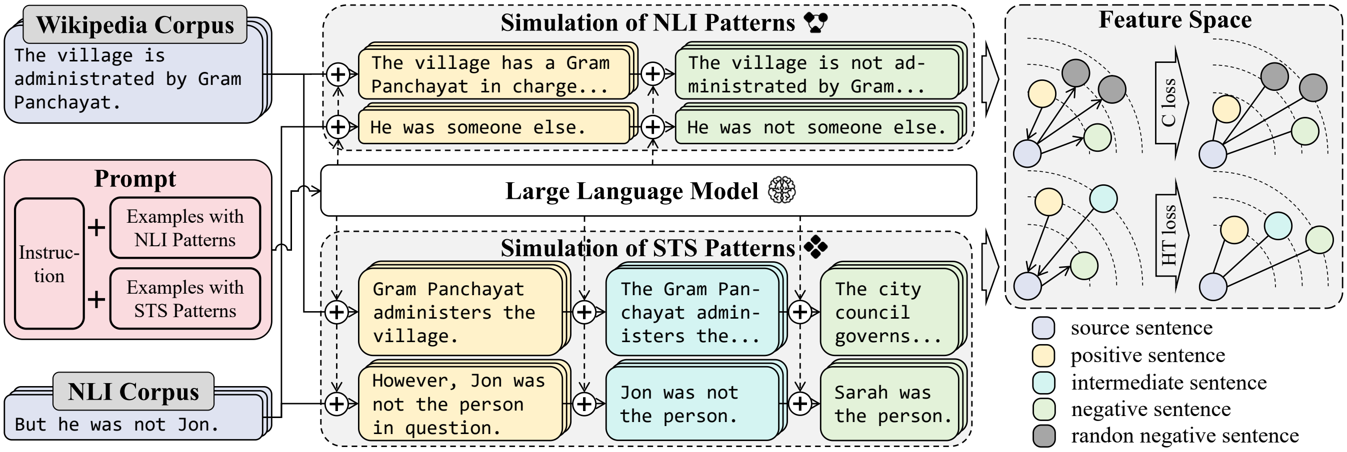


Figure 4: The procedure of pattern simulation and pattern utilization. We simulate the STS and NLI patterns separately with the ICL capability of the LLM. Then, we adopt a combination of contrastive loss and HT loss proposed by us to utilize the pattern. The prompt consists of three examples randomly sampled from the pattern source and its detail is shown in the appendix.

The data generated in these processes is called "LLM-generated data" for simplicity. We combine the LLM-generated data with the remaining sentences in the source corpus to form hybrid datasets, resulting in a total of four hybrid datasets. We name the training data settings of these four datasets by: "Wiki.STS", "Wiki.NLI", "NLI.STS", and "NLI.NLI". Note that (1) For "Wiki.NLI" and "NLI.NLI", the supervision signals in the NLI dataset are used; (2) For "NLI.STS", we only utilize the premises, which do not contain any supervision signals in the NLI dataset.

Then we employ the four training data settings to perform CSE under the same setting as in the **Observation** section. This allows us to examine whether the introduced complex patterns can help narrow the performance gap between S-CSE and U-CSE. Additionally, we calculate both RFD_a and RFD_u for all four settings to further validate our answer to the "*What*" question. The results are plotted in Figure 1 using "★". From the results, it is evident that all the results trained by the hybrid datasets outperform the U-CSE, indicating success in narrowing the performance gap. Moreover, all of these new settings exhibit larger RFD values (RFD_a or RFD_u or both) compared to U-CSE, which indicates that we indeed introduce complex patterns to the training data that lead to an increase in performance. Also, these observations can be viewed as evidence to support our answer to the"*What*" question.

Pattern Utilization With Hierarchical Triplet Loss

In the previous subsection, we have managed to narrow the performance gap to some extent. However, there is still something not being fully utilized, which is the hierarchical nature of the STS patterns. Instead of defining the positive sentence pair and negative sentence pair, the STS task adopts a score ranging from 0 to 5 to reflect the semantic similarity between two sentences, which makes the similarity pattern in STS dataset hierarchical. To maintain such hierarchical nature of STS patterns, we revise our process of pattern simulation (as shown in Figure 4). Specifically, we prompt the LLM to generate an intermediate sentence s_i^m which contains the less details compared to the positive sentence s_i^p, and we randomly sample three sentences from the source of STS patterns with STS scores between 1 and 4 as examples in the prompt. Then, we propose a method to utilize all three sentences s_i^p, s_i^m and s_i^n by adopting a sequence of triplet losses. This approach ensures that the hierarchical pattern can be learned by the sentence encoder. We refer to this loss as the Hierarchical Triplet (HT) loss, and we provide its formal definition below.

For a source sentence s_i and the three sentences generated based on it s_i^p, s_i^m and s_i^n, the HT loss is defined as

$$\begin{aligned}\mathcal{L}_\mathrm{HT} = \frac{1}{2}\big(&\max(f(s_i)^\top f(s_i^\mathrm{m}) - f(s_i)^\top f(s_i^\mathrm{p}) + m_1, 0) + \\ &\max(f(s_i)^\top f(s_i^\mathrm{n}) - f(s_i)^\top f(s_i^\mathrm{m}) + m_2, 0)\big),\end{aligned} \tag{6}$$

where m_1, m_2 are two hyper-parameters that control the margin of the triplet loss, and f is the sentence encoder that maps sentences into a hypersphere. The HT loss is combined with the contrastive loss 1 to form the final loss:

$$\mathcal{L} = \mathcal{L}_\mathrm{C} + \beta\mathcal{L}_\mathrm{HT}, \tag{7}$$

where β is a hyper-parameter controls the weight of $\mathcal{L}_\mathrm{HT}$. Note that $\mathcal{L}_\mathrm{HT}$ is calculated only on the LLM-generated data, which covers 20,000 instances in the hybrid dataset.

We perform CSE with this final loss on "Wiki.STS" and "NLI.STS" under the same setting as in the **Observation** section, These training settings are denoted as "Wiki.STS_HT" and "NLI.STS_HT" respectively. For all settings in this section, we set $m_1 = 5e-3$, $m_2 = 1e-2$ and $\beta = 1$. Similarly, we calculate the RFD of these settings and plot the results in Figure 1 using ★ points. It can be observed that training with $\mathcal{L}_\mathrm{HL}$ increases both RFD_a and RFD_u, and then improves the performance. The rise of RFD values can be explained as follows: the common pattern only determines which sentence pair is similar and

Method	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Supervised methods								
InferSent†	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
SBERT†	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
ConSERT*	74.07	83.93	77.05	83.66	78.76	81.36	76.77	79.37
SimCSE*	75.30	84.67	80.19	85.40	80.82	84.25	80.39	81.57
PromptBERT†	**75.48**	**85.59**	**80.57**	**85.99**	**81.08**	**84.56**	**80.52**	**81.97**
Unsupervised methods								
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
PromptBERT†	71.56	84.58	76.98	84.47	80.60	81.60	69.87	78.54
InfoCSE‡	70.53	84.59	76.40	<u>85.10</u>	<u>81.95</u>	82.00	71.37	78.85
LM-based methods								
Dino§	72.61	81.92	75.09	80.42	76.26	77.10	70.43	76.26
CLAIF§	70.62	81.51	76.29	85.05	81.36	<u>84.34</u>	<u>78.22</u>	79.63
Our methods								
Wiki.STS_HT	$72.46_{\pm 0.15}$	<u>84.88</u>$_{\pm 0.40}$	<u>77.80</u>$_{\pm 0.63}$	$83.85_{\pm 0.66}$	$81.11_{\pm 0.44}$	$81.90_{\pm 0.18}$	$76.56_{\pm 0.26}$	$79.79_{\pm 0.23}$
NLI.STS_HT	<u>72.94</u>$_{\pm 0.19}$	$84.32_{\pm 0.27}$	$77.71_{\pm 0.29}$	$84.20_{\pm 0.40}$	$80.85_{\pm 0.27}$	$82.21_{\pm 0.19}$	$78.04_{\pm 0.35}$	<u>80.02</u>$_{\pm 0.22}$

Table 1: Spearman's correlation on STS Tasks. All models adopt $\mathrm{BERT}_{\mathrm{base}}$ as the backbone. †: results from (Jiang et al. 2022), ‡: results from (Wu et al. 2022), §: results from (Cheng et al. 2023), *: results from their original paper. We bold the highest results among all models and underline the highest results among the models that are not supervised.

dissimilar, while the hierarchical pattern determines which sentence pair is more similar and dissimilar than another. In other words, the hierarchical pattern extends the ideas of the common pattern, thereby increasing the pattern complexity and raising RFD values.

Through the above subsections, we significantly narrow the performance gap between S-CSE and U-CSE. We now provide our answer to the "*How*" question: By utilizing the ICL capability of LLM, we can simulate the patterns in the NLI and STS datasets, thereby introducing complex patterns to the unsupervised training dataset. This process narrows the performance gap to some extent. Subsequently, we thoroughly exploit the hierarchical patterns in the STS dataset with the HT loss, further narrowing the performance gap.

Final Performance

In this section, we compare our methods with various well-known and state-of-the-art baselines:

Unsupervised baselines include a post-processing method, BERT-whitening (Su et al. 2021), as well as contrastive learning methods like ConSERT (Yan et al. 2021), SimCSE (Gao, Yao, and Chen 2021),PromptBERT (Jiang et al. 2022), and InfoCSE (Wu et al. 2022).

Supervised baselines include some traditional supervised methods such as InferSent (Conneau et al. 2017), SBERT (Reimers and Gurevych 2019), and some of the contrastive learning methods mentioned above, which can also be utilized in a supervised setting.

LM-based baselines includes Dino (Schick and Schütze 2021), which generates training data with the Pre-trained Language Model (PLM), and CLAIF (Cheng et al. 2023), which generates training data with the LLM.

In the previous sections, all experiments were conducted under the same settings for a fair comparison. While in this section, we run the "Wiki.STS_HT" and "NLI.STS_HT" training settings under a group of hyper-parameters and decide the best combination of hyper-parameters with the evaluation data, i.e., the validation split STS Benchmark dataset. The details are provided in the **appendix**. We get sentence embeddings following Jiang et al. (2022) to achieve better and more stable performance. We evaluate our method following the standard evaluation protocol mentioned in the **Background** section and compare our method with the baselines on the STS tasks in Table 1 (with $\mathrm{BERT}_{\mathrm{base}}$ as backbone) and 2 (with $\mathrm{RoBERTa}_{\mathrm{base}}$ (Liu et al. 2019) as backbone). By comparing our method with both supervised and unsupervised baselines, we observe that although our method is still inferior to state-of-the-art supervised methods, it outperforms all unsupervised baselines by a large margin. This indicates that we successfully narrow the performance gap between supervised and unsupervised CSE. When compared to data-generation-based methods, our method outperforms them by generating only 20,000 instances, which is significantly fewer than them.

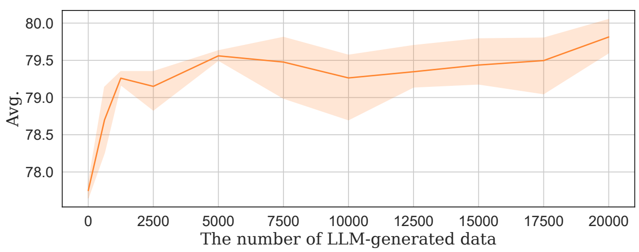
Further Study

In this section, we investigate how each component of our method impacts the performance, and how the embeddings transfer to the downstream tasks. We conduct the experi-

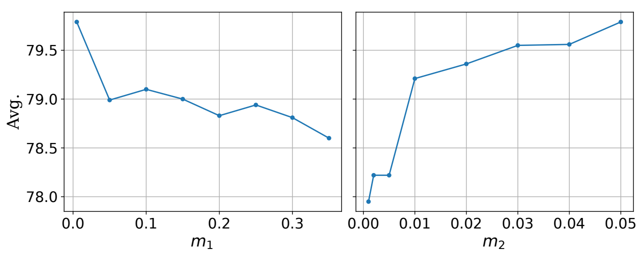
Method	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
SRoBERTa†	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
SimCSE*	76.53	85.21	80.95	86.03	82.57	85.83	80.50	82.52
PromptRoBERTa†	**76.75**	**85.93**	**82.28**	**86.69**	82.80	**86.14**	**80.04**	**82.95**
RoBERTa-whitening†	57.83	63.24	57.23	71.36	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
PromptRoBERTa†	73.94	84.74	77.28	84.99	81.74	81.88	69.50	79.15
Dino§	71.24	81.55	75.67	81.42	78.77	80.10	71.31	77.15
CLAIF§	68.33	82.26	77.00	85.18	**83.43**	85.05	78.02	79.90
Wiki.STS_HT	75.68$\pm$0.41	84.97$\pm$0.63	78.08$\pm$0.69	84.82$\pm$0.23	83.41$\pm$0.37	83.79$\pm$0.25	77.66$\pm$0.20	81.20$\pm$0.21
NLI.STS_HT	74.54$\pm$0.51	85.10$\pm$0.42	79.10$\pm$0.15	85.48$\pm$0.19	82.93$\pm$0.28	83.87$\pm$0.17	78.31$\pm$0.27	81.33$\pm$0.09

Table 2: Spearman's correlation on STS Tasks. All models adopt $\text{RoBERTa}_{\text{base}}$ as the backbone. †: results from (Jiang et al. 2022), §: results from (Cheng et al. 2023), *: results from their original paper. We bold the highest results among all models and underline the highest results among the models that are not supervised.

ments on "Wiki.STS_HT" with $\text{BERT}_{\text{base}}$ as the backbone.



(a) The average Spearman's correlation on STS tasks w.r.t the number of LLM-generated data in the hybrid training dataset.



(b) The average Spearman's correlation on STS tasks w.r.t the margins (m_1 and m_2) in HT loss.

Figure 5: The study of hyper-parameters.

The Number of LLM-Generated Data

In this section, we want to investigate the impact of the number, which is 20,000 previously, of LLM-generated data on performance. To this end, We run our method under different numbers of LLM-generated data and plot the results in Figure 5a. From this figure, we can observe a scaling effect that as the number of LLM-generated data increases, the performance of our method tends to improve.

The Margins of HT Loss

There are two margins in the HT loss, m_1 and m_2. In pattern simulation, both s_i^{p} and s_i^{m} can be regarded as positive samples to the source sentence. As such, we set a small m_1 to ensure that the distance between $f(s_i^{\mathrm{p}})$ and $f(s_i^{\mathrm{m}})$ is not large when learning their hierarchical pattern. Conversely, s_i^{n} can be treated as a negative sample to the source sentence, so we set a large m_2 to ensure that the distance between $f(s_i^{\mathrm{m}})$ and $f(s_i^{\mathrm{n}})$ is sufficiently large when learning their hierarchical pattern. In this section, we invert these settings to investigate the impact of a large m_1 and small m_2 on performance. The results are plotted in Figure 5b, and they conform to our expectation that large m_1 and small m_2 would adversely affect the performance.

	Avg.
Wiki.STS	**79.79**$\pm$0.23
CL.*single_psitive*	78.72$\pm$0.24
CL.*multiple_positive*	79.14$\pm$0.27

Table 3: The average Spearman's correlation on STS tasks when HT loss is replaced with Contrastive Loss (CL).

HT Loss vs. Contrastive Loss

The HT loss is proposed to ensure that the sentence encoder can learn the hierarchical STS pattern. However, training with the HT loss includes more positive samples (i.e., s_i^{m}) than training without the HT loss. To investigate whether the improvement in performance brought about by the HT loss is solely due to the presence of more positive samples, we run our method with only the contrastive loss in two settings: *single positive*, where only s_i^{p} is treated as positive samples, and *multiple positive*, while both s_i^{p} and s_i^{m} are treated as positive samples. The results are shown in Table 3. "CL.*multiple positive*" outperforms "CL.*single positive*", indicating that more positive samples indeed improve performance. However, it still underperforms when compared to training with the HT loss, suggesting that the HT loss indeed brings more effective signals to the training process.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
InferSent†	81.57	86.54	92.50	**90.38**	84.18	88.20	75.77	85.59
SBERT†	**83.64**	**89.43**	94.39	89.86	**88.96**	89.60	76.00	**87.41**
SimCSE*	82.69	89.25	94.81	89.59	87.31	88.40	73.51	86.51
PromptBERT†	83.14	89.38	94.49	89.93	87.37	87.40	76.58	86.90
Avg. BERT embed†	78.66	86.25	94.37	88.66	84.40	<u>**92.80**</u>	69.54	84.94
BERT-[CLS] embed†	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
SimCSE*	81.18	86.46	94.45	88.88	85.50	89.80	74.43	85.81
PromptBERT†	80.74	85.49	93.65	89.32	84.95	88.20	76.06	85.49
InfoCSE*	81.76	86.57	<u>**94.90**</u>	88.86	87.15	90.60	76.58	86.63
Dino§	79.96	85.27	93.67	88.87	84.29	88.60	69.62	84.33
CLAIF§	81.64	87.98	94.24	89.34	86.16	89.80	<u>**77.16**</u>	86.62
Wiki.STS_HT	$82.12_{\pm 0.63}$	$87.96_{\pm 0.27}$	$94.82_{\pm 0.18}$	$90.10_{\pm 0.13}$	$86.84_{\pm 1.38}$	$89.20_{\pm 1.74}$	$75.88_{\pm 1.81}$	$86.70_{\pm 0.76}$
NLI.STS_HT	$\underline{82.36}_{\pm 0.61}$	$\underline{88.19}_{\pm 0.11}$	$94.62_{\pm 0.15}$	$\underline{90.15}_{\pm 0.19}$	$\underline{87.75}_{\pm 0.45}$	$90.60_{\pm 0.60}$	$76.93_{\pm 0.53}$	$\underline{87.23}_{\pm 0.24}$

Table 4: Results on the transfer tasks. All models adopt $\mathrm{BERT_{base}}$ as the backbone. We use the names of hybrid datasets to denote our models. †: results from (Jiang et al. 2022), §: results from (Cheng et al. 2023), *: results from their original paper. We bold the highest results among all models and underline the highest results among the models that are not supervised.

Transfer Tasks

Our study focuses on improving sentence embeddings in the STS task performance, so it remains a mystery how well the learned sentence embeddings can be applied to the downstream tasks. In this subsection, we evaluate the performance of transfer tasks for our methods following the standard evaluation protocol mentioned in the **Background** section, and compare our methods with the baselines in Table 4. The statistics in the table show that, by improving the STS task performance, our methods can learn sentence embeddings that are suitable for the downstream tasks.

Related Work

Sentence Representation Learning (SRL) (Conneau et al. 2017; Reimers and Gurevych 2019; Li et al. 2020) is a fundamental task in NLP, aiming to learn representations for sentences that maintain semantic information. The supervised (Conneau et al. 2017; Reimers and Gurevych 2019) and unsupervised (Li et al. 2020; Su et al. 2021) settings of SRL used to diverge a lot, where supervised SRL (Conneau et al. 2017; Reimers and Gurevych 2019) focused on how to utilize NLI datasets and unsupervised SRL (Li et al. 2020; Su et al. 2021) focused on how to mitigate the anisotropy problem (Li et al. 2020). With the introduction of contrastive learning into SRL (Yan et al. 2021; Gao, Yao, and Chen 2021), many recent works (Gao, Yao, and Chen 2021; Jiang et al. 2022) can be applied to both supervised and unsupervised SRL, building a bridge between these two settings. Although these works boost the performance of SRL under the contrastive learning paradigm, they do not explore the underlying processes leading to the performance gap between supervised CSE (S-CSE) and unsupervised CSE (U-CSE), which motivates our study. Our study also provides a method to improve U-CSE, which can be related to the studies (Wu et al. 2022) that specifically focus on U-CSE.

Our study utilizes the LLM in SRL, and a recent study by (Cheng et al. 2023) employs this approach as well. However, our study differs from theirs in both the method of data generation and the intention of using the LLM. They generate data by predicting masks, while we do so by pattern simulation. They use the LLM to enhance the performance of CSE, while we use the LLM to narrow the performance gap between supervised and unsupervised CSE, and concurrently validate our findings about the fitting difficulty. There is also an early work (Schick and Schütze 2021) that generates datasets with Pre-trained Language Models (PLM) in a way similar to ours. Though our methods seem similar, their intention is to mitigate the need for human-generated data, which is different from our intention.

To the best of our knowledge, we are the first to study what aspects of the training data contribute to the performance gap between supervised and unsupervised CSE.

Conclusion

In this study, we investigate the training process of S-CSE and U-CSE, where we find that the similarity pattern of training data is a key factor to the STS task performance. Then, we define a new metric called Relative Fitting Difficulty (RFD) to quantify the complexity of the similarity pattern in the training data, and prove that higher RFD values correlate with improved performance. Building on this insight, we successfully narrow the performance gap between S-CSE and U-CSE by introducing STS and NLI patterns to the unsupervised data. Moreover, we introduce a Hierarchical Triplet (HT) loss to utilize the hierarchical STS patterns, further narrowing the gap. The fact that we train better sentence embeddings with hierarchical STS patterns than with NLI patterns indicates that a more advanced model may be trained by replacing the long-used NLI dataset with a carefully-crafted hierarchical STS dataset. Such a dataset, previously difficult to create due to the lack of sentences with hierarchical semantic similarities, is now attainable thanks to the powerful LLM.

Acknowledgements

This work was supported by the STI 2030-Major Projects under Grant 2022ZD0120200, in part by the National Natural Science Foundation of China (No. U23B2056), in part by the Fundamental Research Funds for the Central Universities, and in part by the State Key Laboratory of Software Development Environment.

References

Agirre, E.; Banea, C.; Cardie, C.; Cer, D. M.; Diab, M. T.; Gonzalez-Agirre, A.; Guo, W.; Lopez-Gazpio, I.; Maritxalar, M.; Mihalcea, R.; Rigau, G.; Uria, L.; and Wiebe, J. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In Cer, D. M.; Jurgens, D.; Nakov, P.; and Zesch, T., eds., *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, 252–263. The Association for Computer Linguistics.

Agirre, E.; Banea, C.; Cardie, C.; Cer, D. M.; Diab, M. T.; Gonzalez-Agirre, A.; Guo, W.; Mihalcea, R.; Rigau, G.; and Wiebe, J. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In Nakov, P.; and Zesch, T., eds., *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, 81–91. The Association for Computer Linguistics.

Agirre, E.; Banea, C.; Cer, D. M.; Diab, M. T.; Gonzalez-Agirre, A.; Mihalcea, R.; Rigau, G.; and Wiebe, J. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In Bethard, S.; Cer, D. M.; Carpuat, M.; Jurgens, D.; Nakov, P.; and Zesch, T., eds., *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, 497–511. The Association for Computer Linguistics.

Agirre, E.; Cer, D. M.; Diab, M. T.; and Gonzalez-Agirre, A. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In Agirre, E.; Bos, J.; and Diab, M. T., eds., *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, 385–393. The Association for Computer Linguistics.

Agirre, E.; Cer, D. M.; Diab, M. T.; Gonzalez-Agirre, A.; and Guo, W. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In Diab, M. T.; Baldwin, T.; and Baroni, M., eds., *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA*, 32–43. Association for Computational Linguistics.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.-F.; and Lin, H.-T., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Cer, D.; Yang, Y.; Kong, S.-y.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Sung, Y.-H.; Strope, B.; and Kurzweil, R. 2018. Universal Sentence Encoder. *CoRR*, abs/1803.11175. ArXiv: 1803.11175.

Cer, D. M.; Diab, M. T.; Agirre, E.; Lopez-Gazpio, I.; and Specia, L. 2017. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In Bethard, S.; Carpuat, M.; Apidianaki, M.; Mohammad, S. M.; Cer, D. M.; and Jurgens, D., eds., *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, 1–14. Association for Computational Linguistics.

Cheng, Q.; Yang, X.; Sun, T.; Li, L.; and Qiu, X. 2023. Improving Contrastive Learning of Sentence Embeddings from AI Feedback. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, 11122–11138. Association for Computational Linguistics.

Conneau, A.; and Kiela, D. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In Calzolari, N.; Choukri, K.; Cieri, C.; Declerck, T.; Goggi, S.; Hasida, K.; Isahara, H.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; Piperidis, S.; and Tokunaga, T., eds., *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 670–680. Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Dolan, W. B.; and Brockett, C. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005*. Asian Federation of Natural Language Processing.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 6894–6910. Association for Computational Linguistics.

Hu, M.; and Liu, B. 2004. Mining and summarizing customer reviews. In Kim, W.; Kohavi, R.; Gehrke, J.; and DuMouchel, W., eds., *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, 168–177. ACM.

Jiang, T.; Jiao, J.; Huang, S.; Zhang, Z.; Wang, D.; Zhuang, F.; Wei, F.; Huang, H.; Deng, D.; and Zhang, Q. 2022. PromptBERT: Improving BERT Sentence Embeddings with Prompts. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 8826–8837. Association for Computational Linguistics.

Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; and Li, L. 2020. On the Sentence Embeddings from Pre-trained Language Models. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 9119–9130. Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692. ArXiv: 1907.11692.

Marelli, M.; Menini, S.; Baroni, M.; Bentivogli, L.; Bernardi, R.; and Zamparelli, R. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In Calzolari, N.; Choukri, K.; Declerck, T.; Loftsson, H.; Maegaard, B.; Mariani, J.; Moreno, A.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, 216–223. European Language Resources Association (ELRA).

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR*, abs/1807.03748. ArXiv: 1807.03748.

Pang, B.; and Lee, L. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In Scott, D.; Daelemans, W.; and Walker, M. A., eds., *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain*, 271–278. ACL.

Pang, B.; and Lee, L. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In Knight, K.; Ng, H. T.; and Oflazer, K., eds., *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, 115–124. The Association for Computer Linguistics.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 3980–3990. Association for Computational Linguistics.

Schick, T.; and Schütze, H. 2021. Generating Datasets with Pretrained Language Models. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 6943–6951. Association for Computational Linguistics.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1631–1642. ACL.

Su, J.; Cao, J.; Liu, W.; and Ou, Y. 2021. Whitening Sentence Representations for Better Semantics and Faster Retrieval. *CoRR*, abs/2103.15316. ArXiv: 2103.15316.

Voorhees, E. M.; and Tice, D. M. 2000. Building a question answering test collection. In Yannakoudakis, E. J.; Belkin, N. J.; Ingwersen, P.; and Leong, M.-K., eds., *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 24-28, 2000, Athens, Greece*, 200–207. ACM.

Wang, T.; and Isola, P. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, 9929–9939. PMLR. ISSN: 2640-3498.

Wiebe, J.; Wilson, T.; and Cardie, C. 2005. Annotating Expressions of Opinions and Emotions in Language. *Lang. Resour. Evaluation*, 39(2-3): 165–210.

Wu, X.; Gao, C.; Lin, Z.; Han, J.; Wang, Z.; and Hu, S. 2022. InfoCSE: Information-aggregated Contrastive Learning of Sentence Embeddings. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 3060–3070. Association for Computational Linguistics.

Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; and Xu, W. 2021. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 5065–5075. Association for Computational Linguistics.