SumCSE: Summary as a transformation for Contrastive Learning

Raghuveer Thirukovalluru[†] **Xiaolan Wang**[‡] **Jun Chen**[‡]
Shuyang Li[‡] **Jie Lei**[‡] **Rong Jin**[‡] **Bhuwan Dhingra**[†]
[†] Duke University [‡]Meta AI
[†]rt195@duke.edu

Abstract

Sentence embedding models are typically trained using contrastive learning (CL), either using human annotations directly or by repurposing other annotated datasets. In this work, we explore the recently introduced paradigm of generating CL data using generative language models (LM). In CL for computer vision (CV), *compositional* transformations (series of operations applied over an image. e.g. cropping + color distortion) which modify the input/image to retain *minimal* information were shown to be very effective. We show that composition of a 'Summary' transformation with diverse paraphrasing/contradicting transformations accomplishes the same and works very well in CL for sentence embeddings. Our final generated dataset (using Vicuna-13B) significantly outperforms the previous best unsupervised method (using ChatGPT) by 1.8 points, and SimCSE, a strong supervised baseline by 0.3 points on the semantic text similarity (STS) benchmark.

1 Introduction

Contrastive learning (CL) is widely used for training sentence embeddings (Gao et al., 2021; Jiang et al., 2022a; Su et al., 2022; Li et al., 2023). CL typically uses data in the form of (`anchor`, `positive`, `negative`) where embedding of `anchor` is made closer to `positive` and away from the `negative`. SimCSE (Gao et al., 2021) and PromptBert (Jian et al., 2022) used manually annotated NLI data (275K) to train sentence similarity models. Reliance on annotated data, however, limits contrastive learning from being performed at scale and from being transferred to other domains. Some works repurposed existing web datasets to be used as contrastive learning data - Su et al. (2022) used super-NI (Wang et al., 2022b); Li et al. (2023) repurposed multiple unsupervised and supervised datasets. Although the scale of these web datasets helped these models achieve state of the art on multiple tasks, these models still depended on class labels/target text to build the CL triples. Research progress was also made into unsupervised training of sentence embedding models - by using rank consistency between different attention masks (Liu et al., 2023), by diverse noise and heuristic augmentations (Zhou et al., 2023; Wu et al., 2022), and by case augmented positives and retrieved negatives Wang et al. (2022a). GenSE (Chen et al., 2022) finetune a T5 model with the NLI data (275K) and then generate a large CL training dataset (61M). Recently, SynCSE (Zhang et al., 2023) used ChatGPT with diverse paraphrasing, contradiction prompts to generate positives and negatives for CL significantly outperforming other unsupervised methods. However, it's still an open question if paraphrases, contradictions are ideal positives, negatives resp..

On the other hand, in CL for computer vision, Chen et al. (2020); Tian et al. (2020) showed the effectiveness of *compositional* transformations that retain *minimum* information necessary for downstream tasks. We draw inspiration from these works to propose composition of a 'Summary' transformation over other diverse 'Paraphrase' and 'Contradiction' transformations, as positives and negatives respectively, in CL for sentence embeddings. Synthetic dataset generated by our unsupervised methodology, `SumCSE` (uses Vicuna-13B), improves over `SynCSE` (uses ChatGPT) (+1.8) and `SimCSE` (+0.3), on STS. Our method shows benefits (+0.9) when directly applied to other existing datasets and achieves an additional (+0.5) over `SumCSE` when scaled (4x synthetic dataset), on STS.

2 Background and Motivation

SimCSE (Gao et al., 2021) used loss as shown in Eq. 1 to perform unsupervised training. Note that

- Summer internship work. Academic collaborators created the dataset. Meta was not involved in using the dataset for training. Meta did not perform the experiments or use or process any of the data described or referenced in the paper.

Findings of the Association for Computational Linguistics: NAACL 2024, pages 3577–3588

Eq. 1 uses only positives and not negatives. Unsupervised SimCSE used different attention masks to create positive sentence representation and train the model. Eq. 1 further evolved into InfoNCE loss in Eq. 2 for supervised SimCSE with both positives and negatives. N is number of in-batch examples.

$$\mathcal{L} = -\log \frac{e^{sim(h_i,h_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(h_i,h_j^+)/\tau}} \quad (1)$$

$$\mathcal{L} = -\log \frac{e^{sim(h_i,h_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(h_i,h_j^+)/\tau} + e^{sim(h_i,h_j^-)/\tau}} \quad (2)$$

Transformations (t) are used to get h_i^+ i.e. $h_i^+ = t(h_i)$. The success of a contrastive algorithm is substantially influenced by these transformations (Chen et al., 2020; Wu et al., 2020). Minimising InfoNCE loss learns representations that maximise the lower bound on mutual information between input and transformation, $\mathcal{I}(h, h^+)$, over the dataset. Tian et al. (2020) used this to theoretically justify the 'InfoMin' principle - an ideal transformation should share with the input *minimal* information necessary to perform well at a downstream task. It should minimise all irrelevant nuisances in the input. Also, SimCLR (Chen et al., 2020) showed that *compositional* transformations work very well for CL training image representations and found that (image_cropping + colour_distortion) worked the best. Cropping follows the InfoMin principle (reduces mutual information with the input). It matches global-local contexts in an image to learn robust representations. Inspired by these concepts, we propose transformations for sentence CL.

3 Methodology

Our goal here is unsupervised generation of positives and negatives given an anchor. The generated synthetic dataset is used to train an embedding model with Eqs. 1/ 2. This trained model is used to evaluate results on STS and other benchmarks.

3.1 Generating Positives

In accordance with the InfoMin principle and to maximise global-local context agreement (Tian et al., 2020; Chen et al., 2020), we propose to use a *Summary* transformation to generate positives from anchor data. Summary of a sentence can filter out irrelevant information while maintaining its core meaning (minimum information required for sentence similarity). To further validate its feasibility, we follow Gao et al. (2021) to evaluate various transformations on STS-B development set (using the `SimCSE` NLI dataset anchors). We use multiple prompts with a generative LM to create these transformations. We specifically pick the transformations which showed promise (as positives) in other works e.g. *Entailment* sentences(annotated) in Gao et al. (2021), *Sentence Structure Change*, *Paraphrase*, etc in Zhang et al. (2023). More details in §A.7.1. Summary outperforms all other transformations in Table 1. Furthermore, composing transformations (a second transformation over the first) does better. In Table 2, average performance of both Summary and Entailment (best performers in Table 1) as the second transformation is better than either of them individually in Table 1. Finally, following Zhang et al. (2023) in using diverse prompts to simulate real life data, we randomly select one of the four summary compositions in last row Table 2. Random summary compositions outperform all other transformations. Our final positives comprise these random summary compositions over the four diverse transformations in Table 2.

Transformation	Performance
Summary	**86.97**
Entailment	86.71
Sentence Structure Change	84.13
Paraphrase	85.13
Concise Paraphrase	86.19
UnSup. SimCSE (attention mask)	82.5

Table 1: Positive Only STS-B validation results using Eq. 1. RoBERTa-large is the base embedding model.

Transformation	Summary	Entailment
Summary	-	87.21
Entailment	87.3	-
Sentence Struc. Change	86.75	86.15
Paraphrase	86.88	87.13
Concise Paraphrase	87.24	86.99
Avg	87.05	86.88
Random	**87.3**	86.89

Table 2: Positive Only STS-B dev. results (Eq. 1). Rows : First transformation. Colums: Second transformation.

3.2 Generating Negatives

Negative examples in CL typically come from a different class than the `anchor` class. In NLP, a contradiction to a sentence is typically used as `negative` (Gao et al., 2021). Inspired by `SynCSE`, we use four diverse contradiction prompts (which ask to generate different, opposing, contrasting meaning sentences) to generate negatives. However, contradiction to the anchor might not be the ideal negative. For example, STS has multiple examples which have a very high similarity despite being

Model	Method	STS12	STS13	STS14	STS15	STS16	STSB	SICKR	Avg.
				Unsupervised					
RoBERTa-large	unsup-SimCSE†	72.86	83.99	75.62	84.77	81.8	81.98	71.26	78.9
	RankCSE$^{\dagger}_{listNet}$	73.23	85.08	77.5	85.67	82.99	84.2	72.98	80.24
	RankCSE$^{\dagger}_{listMLE}$	73.4	85.34	77.25	85.45	82.64	84.14	72.92	80.16
	L2P-CSR†	73.65	84.08	78.29	85.36	82.15	83.7	73.47	80.1
	PCL†	73.76	84.59	76.81	85.37	81.66	82.89	80.33	79.34
	CARDS†	74.63	86.27	79.25	85.93	83.17	83.86	72.77	80.84
	ConPVP†	74.75	84.09	77.88	83.13	83.44	83.64	74.31	80.18
	SynCSE†	76.03	84.27	80.03	85.37	83.62	84.26	81.14	82.1
	SynCSE-scratch†	75.45	85.01	80.28	86.55	83.95	84.49	80.61	82.33
	SynCSE	75.35	84.54	80.05	85.81	83.53	84.75	81.75	82.25
	SumCSE (Ours)	**78.25**	**87.59**	81.62	**87.64**	**85.20**	85.49	**82.72**	**84.07**
				Supervised					
RoBERTa-large	SimCSE	77.58	87.11	**82.47**	86.42	84.32	**86.75**	81.79	83.78
	SynCSE+	76.92	85.63	81.89	86.26	84.34	85.48	82.43	83.28
	GenSE	73.63	85.87	80.48	84.98	84.57	84.77	77.55	81.69

Table 3: SumCSE does better than all other unsupervised, supervised methods on 7 STS tasks. SumCSE uses Vicuna-13B. SynCSE uses ChatGPT. †: Numbers reported in their corresponding papers. **Bold**: best, underline: second best.

Model	Type	Method	MR	CR	SUBJ	MPQA	SST2	TREC	MRPC	Avg.
RoBERTa-large	Unsupervised	SynCSE	86.7	91.79	93.97	90.53	90.83	85.8	**77.68**	88.19
		SumCSE (Ours)	**88.24**	**92.61**	94.61	**91.01**	**93.36**	**93.20**	76.64	**89.95**
	Supervised	SimCSE	87.91	92.56	**95.04**	90.7	92.7	90.8	75.07	89.25
		SynCSE+	86.88	91.95	94.45	90.99	91.43	89.6	76.87	88.88

Table 4: SumCSE outperforms others on transfer tasks (finetuning logistic classifier with frozen text embeddings)

Model	Type	Method	Classif.	Clust.	Rerank.	Pair Classif.	STS	Retr.	Summ.	Avg
# Datasets			11	11	4	3	10	15	1	56
RoBERTa-large	Unsupervised	SynCSE	67.22	34.06	48.43	76.71	78.52	23.17	28.66	49.40
		SumCSE	70.11	35.33	**49.37**	**82.20**	**81.41**	26.66	29.17	52.10
	Supervised	**SimCSE**	**70.17**	**36.15**	48.71	81.01	80.73	**27.50**	**31.88**	**52.31**
		SynCSE+	67.87	34.42	48.74	79.76	80.49	22.83	28.75	50.06

Table 5: SumCSE outperforms SynCSE and SynCSE+ on MTEB (56 diverse text embedding tasks). SumCSE beats SimCSE in three out of seven task categories. Summary transformation more suitable for these categories of tasks.

contradictions. Several works in contrastive learning have shown that there needs to be enough distance between the anchor and the negative for best results with CL (Wu et al., 2020). A summary composition can put enough distance between input and a contradiction while retaining meaning of the contradiction. Similar to positives, we found that summary when composed over the four contradiction prompts (i.e. first generate a contradiction and then generate a summary of it) gives very good validation results in §4.10. We also justify this choice from an anisotropy standpoint in §4.10. For our final negatives, we pick a random summary composition over the four diverse contradiction prompts.

4 Experiments

4.1 Methods Compared

Our methodology, SumCSE is compared with SimCSE and SynCSE (Zhang et al., 2023). We used Vicuna-13B-v1.3 with prompts to generate all positives and negatives in SumCSE. The exact prompts used for different transformations are detailed in §A.7. SumCSE uses the same anchor sentences as SimCSE and has same number of examples around $\sim 275K$. SynCSE only has $\sim 263K$ examples (failure of ChatGPT on 12K examples). Hence we also compare with SynCSE+ an extended version of SynCSE where we pick the remaining examples from SimCSE. Adding SimCSE triples to SynCSE was proposed in Zhang et al. (2023) and is known to have a significant improvement over SynCSE. All above methods use exact same anchor data and same loss function (Eq. 2). They only differ in the positive and negative data. Further we also compare with multiple unsupervised methods: RankCSE (Liu et al., 2023), L2P-CSR (Zhou et al., 2023), PCL (Wu et al., 2022), CARDS (Wang et al., 2022a) and ConPVP (Zeng et al., 2022). We also compare with GenSE (Chen et al., 2022) where we sample 275K examples from its 61M for fair comparison. Most of these unsupervised baselines are complementary to SumCSE and can also be applied on top of the SumCSE dataset.

4.2 Implementation Details

We present results with RoBERTa-large. For consistency, we use the exact same hyperparams and settings in SumCSE as were used in the best performing SimCSE models. For SynCSE and SynCSE+, we use the best settings from Zhang et al. (2023).

SumCSE used Vicuna-v1.3. Note that this is an older version of Vicuna. Older version was used to be consistent with the time period of SynCSE which used ChatGPT from mid 2023. In SimCSE and SumCSE, we follow (Gao et al., 2021) and use STS-B (one of the seven STS benchmark datasets) validation performance to pick the best models. Also, additional MLM loss was not used in any training of SimCSE or SumCSE. For SynCSE and SynCSE+, MLM loss was included and (STS-B, SICKR) average validation was used to pick best models following (Zhang et al., 2023). More details in §A.1.

4.3 Main Results: STS

We follow Gao et al. (2021) to evaluate all models on STS test benchmark comprising seven textual similarity tasks: STS12-16, the STS benchmark and SICK Relatedness. Details in §A.3. We train the models on their annotated/synthetic data and test them on STS. Spearman correlation is used as the performance metric. Table 3 shows the results on STS benchmark. SumCSE does significantly better than both SynCSE and SynCSE+. Note that SynCSE used ChatGPT while SumCSE used Vicuna-13B-v1.3. SumCSE also shows notable improvement over supervised SimCSE.

4.4 Transfer results

Following SimCSE, we further evaluate the transfer capabilities of these embeddings with seven transfer learning tasks: MR, CR, SUBJ, MPQA , SST2, TREC and MRPC. Details in §A.3. A logistic regression classifier is trained and evaluated on frozen embeddings fro different methods. Table 4 shows results. SumCSE does better than other models. Here, MLM loss was not used for SumCSE and SimCSE but was used in SynCSE due to default settings. Gao et al. (2021) says transfer numbers are much higher with the MLM loss.

4.5 MTEB results

Embeddings are required in multiple other tasks as well. Recently, a generalised text embedding benchmark MTEB (Muennighoff et al., 2022) was released. This benchmark has multiple other tasks from Retrieval, Classification and other long text tasks. We evaluate SumCSE on this benchmark in Table 5. SumCSE outperforms SynCSE, SynCSE+ in all tasks. It does better than SimCSE in re-ranking, pair-classification and STS. STS here is a more advanced version including STS17, STS22 in addition to Table 3. Interestingly, SimCSE does better than SumCSE on overall MTEB. We posit this happens because of our choice of transformations (which we optimise for sentence similarity). We observe higher anisotropy for SumCSE which further explains inferior performance in clustering, retrieval. Details in §4.9.

Ablation	Method	Size	STS	Transfer
1	SynCSE	263K	78.82	88.75
	SumCSE	275K	**81.73**	**89.55**
	SimCSE	275K	80.62	88.58
	SynCSE+	275K	79.83	89.15
2	SumCSE	275K	84.07	89.95
	SumCSE$_{large}$	1.1M	**84.63**	89.82
3	SynCSE+	275K	83.28	88.88
	Sum + SynCSE+	275K	**84.18**	**89.86**

Table 6: 1. SumCSE (Positive Only, Eq. 1) outperforms others. Already better than most unsupervised, supervised methods(in Tab. 3) 2. SumCSE has significant gains on large scale (4x) dataset. 3. Summary transformation applied on SynCSE+ dataset results in +0.9 gain on STS.

4.6 Ablation 1: Positive Only

We assessed multiple transformations for positives in Table 1, 2. In this subsection, we analyse how positives in SumCSE compare to positives in other annotated/generated datasets. Table 6 shows the test STS and transfer results with a positive only loss for all methods with RoBERTa-large. SumCSE outperforms other methods. Positive only performance of SumCSE already outperforms most unsupervised methods in Table 3.

4.7 Ablation 2: Large Scale Data

Large scale datasets have played a crucial role in creating SOTA sentence embeddings. Here, we evaluate if our methodology can be used to generate a large scale dataset. Instead of randomly sampling one among four summary positives in §3.1, we consider all of them. For negatives, we randomly sample one among the four. This creates SumCSE$_{large}$, a 4x dataset, and simulates the case of generating a large scale dataset. SumCSE$_{large}$ achieves the best result in this paper, significantly improving over SumCSE on STS while matching transfer numbers in Table 6.

4.8 Ablation 3: Summary on Other Data

Given summary compositions work well in SumCSE, herein, we try to investigate if we can do the same with other datasets. We simply apply the summary transformation to SynCSE+ positives and negatives to generate a new dataset(using Vicuna-13B-v1.3). Table 6 shows that the summary transformation when applied to other datasets works significantly improves the overall performance.

4.9 Analysis 1: Anisotropy

Anisotropy is defined as the average pairwise sentence similarity of sentences in a corpus. While some research has shown that anisotropy is less important for sentence embeddings (Jiang et al., 2022a), it is still an important measure of spread of embeddings. Table 7 shows the anisotropy of different models on a subset of Wikipedia data dump. SumCSE has an higher value of anisotropy compared to other models. Higher anisotropy indicates that the sentence embeddings are all in a close space and explains lower performance in clustering, retrieval tasks of MTEB.

4.10 Analysis 2: Choice of Negatives

Eq. 1 allows for evaluation of the quality of positives independent of negatives. Eq. 2 however depends on both positives and negatives. The choice of negatives thus depends on the positives. In this ablation, we mix and match different positives, negatives to get an understanding of the type of negatives that work best. Table 8 shows the STS-B validation performance and anisotropy of different models. We observe that shorter negatives generally work better with shorter positives - models with higher STSB validation and better anisotropy.

Row 3 in the table shows performance of SumCSE without the summarization step for negatives i.e. without the second transformation. This results in much longer negatives. Looking at SumCSE only numbers (rows 3 and 5), we note that summarizing negatives makes a big difference in validation performance and anisotropy. Comparing rows 4 and 5 shows that summary negatives work better than other shorter negatives. These results further justify the strength of summary.

4.11 Analysis 2: Shared Information

To further show the shared information between anchor, positive and negative, Rouge1 similarity scores are shown in Table. 9. InfoMin principle suggest this number has to be as low as possible while while maintaining enough information to solve downstream task (sentence similarity). Positives in SumCSE have low similarity with the anchor illustrating that they follow InfoMin principle. Interestingly, SimCSE positives also have low similarity with the anchor showing that annotated data in SimCSE might have also benefited from InfoMin.

Method	SimCSE	SynCSE	SynCSE+	SumCSE
Anisotropy	0.094	0.112	0.103	0.123

Table 7: Anisotropy of different models over wikipedia sentences. SumCSE has highest anisotropy

Positive	**Negative**	**PL**	**NL**	**STSB**	**Anisotropy**
SimCSE	SynCSE+	8.06	13.41	85.94	0.175
SumCSE	SynCSE+	7.32	13.41	87.44	0.167
SumCSE	SumCSE**	7.32	15.47	85.6	0.132
SumCSE	SimCSE	7.32	8.23	87.35	**0.105**
SumCSE	SumCSE	7.32	7.78	**88.13**	0.123

Table 8: Performance of negatives from different methods for shorter positives. PL - Positives length, NL-Negatives length. Longer negatives with shorter positives results in very high anisotropy. Shorter negatives worked best with shorter positives. SumCSE** - Only first transformation without second (No Summary).

Method	**RougeAP**	**RougeAN**	**RougePN**
SynCSE	0.67	0.52	0.34
SumCSE	0.45	0.31	0.31
SimCSE	0.44	0.32	0.41
SynCSE+	0.66	0.51	0.42

Table 9: RougeAP, RougeAN, RougePN: Rouge1 similarity between anchor, positive and negative (pairwise). Positives in SumCSE have low similarity with the anchor illustrating that they follow InfoMin principle.

5 Conclusion

In this work, we draw inspirations from computer vision to build transformations that minimise irrelevant information in contrastive learning data. We propose SumCSE which composes a summary transformation over diverse paraphrasing, contradiction transformations to generate CL training triples for sentence embeddings. The proposed unsupervised synthetic dataset, SumCSE, significantly improves over all other unsupervised methods and supervised SimCSE on sentence similarity tasks. SumCSE shows promise when extended to a large scale data and also when applied on other datasets. Future work would involve investigating transformations that work for generalised text embeddings in MTEB and generating a large scale contrastive dataset.

6 Limitations

Lagging Performance on MTEB: Results of SumCSE lagged behind SimCSE on MTEB. Tian et al. (2020) showed both theoretically and empirically that transformations are downstream task dependent. InfoMin (Tian et al., 2020) principle suggests that transformation should have all of the information necessary to perform a specific downstream task. In this research, we optimised our transformations for sentence similarity. Hence, SumCSE shows gains on STS while lagging on some other tasks. We argue that summary composition is not best suited for clustering, retrieval tasks where it underperformed.

Web Scale Experiments: While SumCSE$_{large}$ showed positive gains from large datasets (1.1M), huge web scale datasets (>50M datapoints) were not explored due to computational limitations. Further, we were also restricted by the anchor points that were used for generating positive and negatives. We followed Gao et al. (2021), Zhang et al. (2023) in using the same set of anchor points for this work. Choice of anchors to use for contrastive learning is an interesting research topic by itself. Chen et al. (2022) used open domain data as anchors. Zhang et al. (2023) also proposed SynCSE-scratch to generate anchors in multiple diverse domains. We reserve this exploration with web scale data for future research.

Limited Supervised Baselines: We limit our baselines to SimCSE and GenSE for the supervised case. Most supervised methods like Jiang et al. (2022a), Jiang et al. (2022b) work with SimCSE NLI dataset and propose improvements to the modelling or loss function. These methods are complementary to SumCSE and can be used on top of the SumCSE dataset. A lot of unsupervised methods from §4 are also complementary and can be used with SumCSE. Hence, we limited the baselines based on the the datasets used.

The goal of this research is to share with the community that summary as a transformation works as a strong data augmentation method in contrastive learning for sentence similarity. For future research, we intend to explore generalised transformations that work with a variety of tasks in MTEB and generate a web scale contrastive dataset.

Broader Impact and Discussion of Ethics:
While our model is not tied to any specific applications, it could be used in sensitive contexts such as health-care, etc. Any work using our method is requested to undertake extensive quality-assurance and robustness testing before applying in their setting. To the best of our knowledge, the datasets used in our work do not contain any sensitive information. We followed SynCSE to perform an ethical evaluation of 100 random samples from the SumCSE dataset and manually checked for any ethical problems. We did not find any data with ethical problems in any of the examples. This is expected because Vicuna-13B has been trained not to output any sensitive information.

License: All datasets, methods used fall under Apache License 2.0. This research work abides by terms of the license. Research output of this paper also fall under Apache License 2.0.

Replicability:
Source Code and Datasets available at https://github.com/raghavlite/SumCSE

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

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

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Yiming Chen, Yan Zhang, Bin Wang, Zuozhu Liu, and Haizhou Li. 2022. Generate, discriminate and contrast: A semi-supervised sentence representation learning framework. *arXiv preprint arXiv:2210.16798*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

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.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2022. Contrastive learning for prompt-based few-shot language learners. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5577–5587, Seattle, United States. Association for Computational Linguistics.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022a. Promptbert: Improving bert sentence embeddings with prompts. *arXiv preprint arXiv:2201.04337*.

Yuxin Jiang, Linhan Zhang, and Wei Wang. 2022b. Improved universal sentence embeddings with prompt-based contrastive learning and energy-based learning. *arXiv preprint arXiv:2203.06875*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Jiduan Liu, Jiahao Liu, Qifan Wang, Jingang Wang, Wei Wu, Yunsen Xian, Dongyan Zhao, Kai Chen, and Rui Yan. 2023. Rankcse: Unsupervised sentence representations learning via learning to rank. *arXiv preprint arXiv:2305.16726*.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 1–8.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Hongjin Su, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, Tao Yu, et al. 2022. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839.

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.

Wei Wang, Liangzhu Ge, Jingqiao Zhang, and Cheng Yang. 2022a. 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.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

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

Mike Wu, Milan Mosse, Chengxu Zhuang, Daniel Yamins, and Noah Goodman. 2020. Conditional negative sampling for contrastive learning of visual representations. *arXiv preprint arXiv:2010.02037*.

Qiyu Wu, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, and Daxin Jiang. 2022. Pcl: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings. *arXiv preprint arXiv:2201.12093*.

Jiali Zeng, Yongjing Yin, Yufan Jiang, Shuangzhi Wu, and Yunbo Cao. 2022. Contrastive learning with prompt-derived virtual semantic prototypes for unsupervised sentence embedding. *arXiv preprint arXiv:2211.03348*.

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

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 Appendix

A.1 Training Parameters

All embedding training models used `RoBERTa-large` (350M) or `RoBERTa-base` (110M) as a starting checkpoint (`RoBERTa-base` experiments later in the appendix). Peak learning rate of 5e-5 was used for the former and 1e-5 was used for the later following Gao et al. (2021). Batch size of 512 was used for bothmodels. Default seed value of 42 for all experiments following Zhang et al. (2023).

In `SimCSE` and `SumCSE`, we follow (Gao et al., 2021) and use STS-B (one of the seven STS benchmark datasets) validation performance to pick the best models. Also, additional MLM loss was not used in any training of `SimCSE` or `SumCSE` (though this slightly improved `SumCSE` numbers). For `SynCSE` and `SynCSE+`, MLM loss was included and (STS-B, SICKR) average validation was used to pick best models following (Zhang et al., 2023).

A.2 Compute

All experiments were run on four A6000 GPUs (48gb). `RoBERTa-large` model training took 2 hours for `SumCSE`, `SynCSE`, `SimCSE` on 275K sized data. `RoBERTa-base` model training took 50 mnts. To generate the four positive paraphrases using Vicuna-13B, it took 65 hrs for first transformation and 12 hrs for the second transformation. Runtime numbers for negatives were the same.

A.3 Test Datasets

STS Benchmark: Comprises seven semantic similarity tasks- STS12, STS13, STS14, STS15, STS16 (Agirre et al., 2012, 2013, 2014, 2015, 2016), the STS benchmark (Cer et al., 2017) and SICK Relatedness (Marelli et al., 2014). All models in this paper are trained on an annotated/synthetic dataset and tested on STS. STS 'train' split is not used in any method.

Transfer Tasks: Comprise seven tasks: MR (Pang and Lee, 2005), CR (Hu and Liu, 2004), SUBJ (Pang and Lee, 2004), MPQA (Wiebe et al., 2005), TREC (Voorhees and Tice, 2000) and MRPC (Voorhees and Tice, 2000).

MTEB Benchmark: Comprises 57 sentence embedding methods from Classification, Clustering, Reranking, Pair Classification, STS, Retrieval, Summarization (Muennighoff et al., 2022).

sentence1	sentence2	score(out of 5)
There is no girl with a black bag on a crowded train	A girl with a black bag is on a crowded train	3.7
A man is playing a guitar on stage	There is no man playing a guitar on stage	3.6

Table 10: Few examples from STS where sentence similarity is high despite being a contradiction.

Model	Method	STS12	STS13	STS14	STS15	STS16	STSB	SICKR	Avg.
Unsupervised									
RoBERTa-base	RankCSE$^{\dagger}_{listNet}$	72.88	84.5	76.46	84.67	83	83.24	71.67	79.49
	RankCSE$^{\dagger}_{listMLE}$	72.74	84.24	75.99	84.68	82.88	83.16	71.77	79.35
	L2P-CSR†	74.97	83.63	78.28	84.86	82.03	82.77	71.26	79.69
	PromptRoberta†	73.94	84.74	77.28	84.99	81.74	81.88	69.5	79.15
	PCL†	71.54	82.7	75.38	83.31	81.64	81.61	69.19	77.91
	CARDS†	72.49	84.09	76.19	82.98	82.11	82.25	70.65	78.68
	ConPVP†	73.2	83.22	76.24	83.37	81.49	82.18	74.59	79.18
	SynCSE†	76.11	84.49	79.61	85.26	82.6	83.94	81.57	81.94
	SynCSE-scratch†	74.61	83.76	77.89	85.09	82.28	82.71	78.88	80.75
	SynCSE	76.29	84.33	79.26	84.75	82.83	83.83	81	81.76
	SumCSE (Ours)	77.13	85.39	79.50	86.48	83.88	84.56	81.39	**82.62**
Supervised									
RoBERTa-base	SimCSE	75.75	84.88	80.15	85.38	82.14	84.89	80.39	81.94
	SynCSE+	77.3	83.76	79.57	85.33	82.55	83.87	81.37	81.96

Table 11: SumCSE does better than all other methods on 7 STS tasks. †: Numbers reported in their corresponding papers. **Bold**, underline indicate first, second best.

Model	Method	MR	CR	SUBJ	MPQA	SST2	TREC	MRPC	Avg.
RoBERTa-base	SynCSE	85.01	91.52	92.55	89.84	91.32	83.8	76.23	87.18
	SumCSE	85.82	92.19	93.26	89.67	91.49	86.40	76.93	**87.97**
	SimCSE	85.35	91.82	93.78	89.65	91.21	85.6	75.59	87.57
	SynCSE+	85.49	91.34	93.11	89.77	92.04	84.4	75.94	87.44

Table 12: SumCSE does better than all other methods on 7 Transfer Tasks. **Bold**, underline indicate first, second best.

A.4 Dataset Stats

In this subsection, we discuss about the statistics of different datasets used. Some dataset stats are shown in Table 13.

Method	Size	Anchor Len.	Pos. Len.	Neg. Len.
SynCSE	263K	15.58	14.86	13.68
SumCSE	275K	15.72	7.32	7.78
SimCSE	275K	15.72	8.06	8.23
SynCSE+	275K	15.72	14.55	13.81

Table 13: Dataset statistics of different methods. Anchor Len. - Average length of anchor. Pos. Len. - Average length of positives. Neg. Len. - Average length of Negatives

A.5 Contradiction not ideal Negatives

Table 10 shows a couple of examples of contradictions which have a very high similarity with the input sentence and yet have a high score in STS. This shows that contradictions as were used in the past mihgt not be ideal negatives.

A.6 More models

All results in the main paper came from RoBERTa-large as the base model. To further justify strength of SumCSE we also test it with RoBERTa-base. Table 11 and 12 show the results. The trends are exactly the same as RoBERTa-large.

A.7 Prompts

A.7.1 Positive Prompts

Positive Prompts used to create various transformations mentioned in §3.1 are shown in Table 14. We use zero a shot summarization prompt. For all other prompts, we use a 5 shot chat setup as mentioned in Zhang et al. (2023). We used 10 fixed examples for each prompt and randomly sampled 5 among them to use as demonstrations for each generation following Zhang et al. (2023). The 10 examples were picked from SynCSE, SimCSE data. Using 5 fixed demonstrations directly worked equally well for us. Within the zero shot summary prompts, we used 'seven' as a reference length to build summaries. This was approximately half the average length of anchor data in SimCSE. We posit this satisfies InfoMin principle in retaining minimum information.

A.7.2 Negative Prompts

Negative Prompts were mostly reused from Zhang et al. (2023). Similar to previous case, we randomly picked 5 demonstrations from a set of 10 fixed demonstrations for each prompt. These negative prompts ask the LM to generate different, opposing, contrasting meaning sentences. Negative1, Negative 2 ask to modify some/one or two details and maintain sentence structure. Negative3, Negative4 ask the LM to generate logical outputs with no constraints on sentence structure. The final `SumCSE` contains summaries of the four negatives using the summarization prompt in §A.7.1.

Positive Prompts
Summary USER: Summarize the input sentence in seven words. The input sentence is - <*input*> What is your generated sentence?
Entailment Create a sentence or phrase that is also true, assuming the provided input sentence or phrase is true. USER: <*example_fewshot_input1*> ASSISTANT: <*example_fewshot_output1*> . . USER: <input>
Sentence Structure Change Rewrite the input sentence or phrase using different sentence structure and different words while preserving its original meaning. Please do not provide any alternative or reasoning or explanation. USER: <*example_fewshot_input1*> ASSISTANT: <*example_fewshot_output1*> . . USER: <input>
Paraphrase: Paraphrase the input sentence or phrase, providing an alternative expression with the same meaning. Please do not provide any alternative or reasoning or explanation. USER: <*example_fewshot_input1*> ASSISTANT: <*example_fewshot_output1*> . . USER: <input>
Concise Paraphrase: 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. Please do not provide any alternative or reasoning or explanation. USER: <*example_fewshot_input1*> ASSISTANT: <*example_fewshot_output1*> . . USER: <input>

Table 14: Prompts used to generate Positives

Negative Prompts
Negative 1 Revise the provided sentence by swapping, changing, or contradicting some details in order to express a different meaning, while maintaining the general context and structure. USER: <*example_fewshot_input1*> ASSISTANT: <*example_fewshot_output1*> . . USER: <input>
Negative2 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. USER: <*example_fewshot_input1*> ASSISTANT: <*example_fewshot_output1*> . . USER: <input>
Negative3: 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. USER: <*example_fewshot_input1*> ASSISTANT: <*example_fewshot_output1*> . . USER: <input>
Negative4: Generate a sentence that conveys a altering, contrasting or opposite idea to the given input sentence, while ensuring the new sentence is logical, realistic, and grounded in common sense. USER: <*example_fewshot_input1*> ASSISTANT: <*example_fewshot_output1*> . . USER: <input>

Table 15: Prompts used to generate Negatives