Contrastive Learning with Sub-optimal Positive Samples in NLP tasks

Qiwen Zhang
Center for Data Science
New York University
qz2274@nyu.edu

Siyuan Sheng
Center for Data Science
New York University
ss16133@nyu.edu

Karen Fan
Center for Data Science
New York University
kf1066@nyu.edu

Luyang Shang
Center for Data Science
New York University
ls6096@nyu.edu

Shiqi Yang
Center for Data Science
New York University
sy3506@nyu.edu

Abstract

In our paper, we introduce a novel approach to enhance NLP models through 'Hard Positive Augmentation.' This method, inspired by advancements in computer vision, diverges from traditional contrastive learning by generating challenging positive samples using varied dropout masks in an encoder model. Trained on a diverse Wikipedia dataset and evaluated against Semantic Textual Similarity benchmarks, our approach significantly improves sentence embedding quality. A significant finding is the superiority of the dot product method over cosine similarity in identifying hard positive samples, marking a substantial advancement in NLP model performance and setting a new direction for research in unsupervised learning techniques. The github link is https://github.com/QiwenZz/simcse_w_hard_positive.

1 Introduction

This project is driven by the growing field of neural network research, taking inspiration from a significant computer vision study that used Generative Adversarial Networks (GANs) to generate challenging positive samples. This study revealed the potential of these samples in improving model robustness. Building upon this finding, our project aims to explore the impact of incorporating "hard positive samples" in the field of Natural Language Processing (NLP). The underlying logic is based on the common foundations of deep learning models across domains, suggesting that the success observed in computer vision could extend to other areas as well. The objective of our project is to enhance the existing contrastive learning technique by introducing a novel method called Hard Positive Augmentation to NLP tasks.

In traditional contrastive learning techniques, models rely on augmentation methods to create sequences similar to the original data point. However, our project hypothesizes that among the generated sequences from different augmentation methods random selection process in traditional contrastive learning may not effectively capture highly similar sequences, potentially leading to suboptimal performance. Alternatively, we propose to create multiple positive examples by passing batches of examples to the encoder model utilizing various dropout masks. The model will then calculate the similarity scores between the original embedding (anchor) of positive examples and the corresponding generalized positive examples. The identification of a "hard positive sample" follows a parallel concept to that of hard negative samples; specifically, it corresponds to the instance with the least similarity score among all generated positive samples. We then introduce this hard positive sample into our model, aiming to enhance the similarity score between it and the anchor data point.

This method is aligned with the philosophy of reinforcing the model's capacity to capture subtle similarities among positive instances.

The experimental outcomes are consistent with our hypothesis, revealing an enhancement in model performance by maximizing the similarity score between the hard positive sample and the anchor data point, as compared to the control group using a single positive sample. An interesting observation is the superiority of the dot product method over cosine similarity in terms of both computational efficiency—especially evident with a higher number of candidate positive samples—and experimental outcomes, achieving higher scores across all trials.

2 Related Work

Contrastive learning, pivotal in computer vision, optimizes embeddings by aligning similar samples (positives) and distancing dissimilar ones (negatives). A key advancement is the use of hard negatives and hard positives. Hard negatives are subtly different and improve model performance, while hard positives, often identified through adversarial methods like GANs, enrich the feature space [1, 2, 3, 4].

In NLP, contrastive learning is also applied, utilizing hard negatives to refine model efficacy. However, the concept of hard positives remains largely unexplored in this domain. While NLP has seen significant advances with models like Word2Vec and BERT, the integration of hard positive techniques, as seen in computer vision, represents a novel and untested approach in NLP [5, 6].

3 Approach

In the dynamic realm of unsupervised learning within Natural Language Processing (NLP), contrastive learning, exemplified by models like SimCSE, has emerged as a critical technique. However, this method may miss sequences with higher similarity to the original, which could limit the model's effectiveness. Addressing this gap, we introduce "Hard Positive Augmentation" and explore an innovative approach. Here we explain the details of the 2 methods mentioned above.

3.1 SimCSE

SimCSE, or Simple Contrastive Sentence Embedding, forms the cornerstone of our approach and acts as our base model. It is an unsupervised model aimed at generating sentence embeddings, primarily utilizing the capabilities of BERT-based architectures. SimCSE's core mechanism involves using dropout as an augmentation tool, generating two distinct embeddings from the same sentence. These embeddings are then aligned closer in the embedding space, thereby refining the model's proficiency in discerning sentence similarities. However, this approach tends to focus on aligning embeddings from identical input sentences, potentially missing out on the richer learning experiences offered by varied yet semantically akin sentences.

3.2 Hard Positive Augmentation

To enhance the SimCSE framework, we introduce "Hard Positive Augmentation." This method involves the strategic use of dropout masks to selectively inactivate nodes within the network, thus generating a range of augmented sequences. We then identify the sequence with the lowest similarity score to the original input as the 'hard positive' sample. This process not only diversifies the training data but also challenges the model to grasp more intricate semantic connections. Incorporating these demanding positive pairs is designed to markedly advance the model's discriminative capabilities and its robustness in comprehending complex sentence structures.

By synergizing the robust embedding generation of SimCSE with our pioneering Hard Positive Augmentation technique, we aim to redefine the standards in unsupervised NLP model training, fostering a deeper and more nuanced understanding of language.

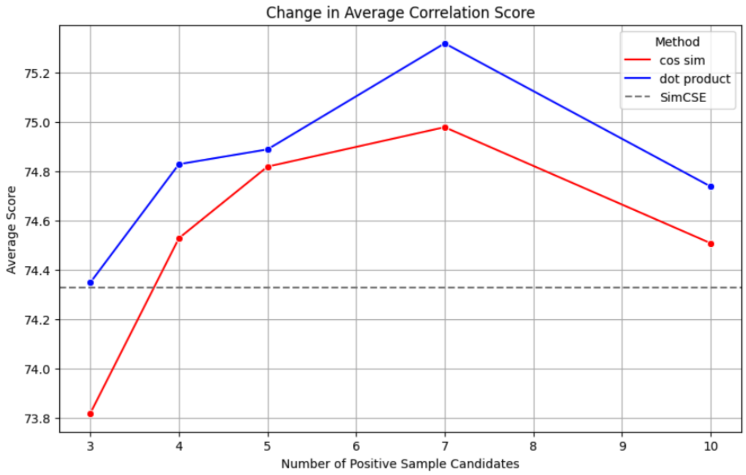

Figure 1: Change in Average Correlation Score

4 Experiments

4.1 Data

Our experimental approach involves utilizing a pretrained BERT model, which we fine-tune on a dataset comprising one million sentences randomly selected from Wikipedia. The dataset is chosen for its diversity and comprehensiveness, making it ideal for training sentence-level representations.

4.2 Evaluation method

The evaluation metrics employed are based on the Semantic Textual Similarity (STS) benchmarks, ranging from STS12 to STS16, along with the stsbenchmark and SICKRelatedness tasks. These benchmarks provide a score reflecting the semantic similarity between pairs of sentences, as judged by human annotators. The performance of the model is assessed using Spearman's rank correlation coefficient between the cosine similarity of the sentence embeddings and the human-annotated similarity scores.

4.3 Experimental details

The experiments were conducted using the BERT-base-uncased model. Key hyperparameters include:

- Learning Rate: 3e-5, Batch Size: 64, Sequence Length: 32
- Temperature for Contrastive Learning: 0.05
- Model seed: [42, 48, 3407]

We introduce a 'hard positive' selection mechanism, prior to the contrastive loss calculation, using either cosine similarity or dot product. The number of hard positive candidates was varied (3, 4, 5, 7, 10) to analyze its impact on the model's performance. For each combination of candidate number and selection mechanism, we conducted three separate experiments, each initialized with a different model seed from the set [42, 48, 3407]. The final results for each configuration were obtained by computing the average performance across these three runs.

4.4 Results

Figure 1 shows the our main experimental outcomes, detailed result can be accessed at table 1.

These results indicate that incorporating hard positives into the SimCSE framework can potentially enhance the quality of sentence embeddings, as evidenced by the improvements in the STS tasks. Furthermore, the dot product method outperforms cosine similarity in this context. It is also observed that there is an optimal range of hard positive candidates for enhancing the model's performance (around 5 for cosine similarity and 7 for dot product). Beyond these points, increasing the number of candidates does not yield additional benefits and could potentially compromise the model's efficiency.

5 Analysis

From the result, it can be concluded that the incorporation of hard positive selection can increase the performance to a certain extent as it stands true for almost all experiments that adding the hard positive component beats the SimCSE baseline. This aligns with our hypothesis that if we use those samples that are hard for the model to identify them to be similar as the positive sample pairs to train the contrastive loss, the model's representation learning ability should be enhanced.

It should be noted that the increment of the positive sample candidates number should always result in a same or rising performance because a larger selection space contains more pairs with largely different embeddings. However, it's observed in both cosine similarity and dot product that the performance decreases after increasing the number of candidates from seven to ten. This might indicate that positive sample pair with too different embeddings could damage the representation learning process.

When looking into the comparison between using dot product versus cosine similarity as the distance metric to determine the most dissimilar embedding pair, it's observed that the use of dot product performed better than the use of cosine similarity. This is reasonable because cosine similarity removes the consideration of vector magnitude which can be very influential when determining how different two embeddings are. This also indicates that the output embeddings from BERT can be largely different in terms of magnitudes.

6 Conclusion

The main findings of our project is that the selection of hard positive sample result in an enhancement in model performance. Dot product as the distance metric to determine the difference between embeddings leads to a better result than cosine similarity. The performance generally increases as we increase the number of positive sample candidates until reaching a point.

One limitation of this method is the time it takes to train. We can see that regarding the limited improvement in performance, the time it takes to train increases linearly with the increase of candidates number for the dot product method and exponentially for the cosine similarity method. The other limitation is that we only explored a restricted set of values for the number of candidate hard positive samples. In the future, we could expand upon this parameter through more extensive experiments and runs. Additionally, theoretical analyses could be performed to ensure a more comprehensive understanding.

References

[1] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005.

[2] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples, 2021.

[3] Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning, 2020.

[4] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings, 2022.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

7 Appendix

Num Candidate	Train Time	STS12	STS13	STS14	STS15	STS16	STS	SICK-R	Average Score
SimCSE	4,371.97	65.35	81.26	71.61	79.67	77.59	75.03	69.81	74.33
Cos Sim									
3	6,917.71	65.85	80.10	70.63	79.84	76.70	74.72	68.96	73.82
4	8,810.57	66.44	80.38	71.62	80.09	77.90	75.56	69.73	74.53
5	10,763.49	67.50	80.50	72.13	80.43	78.24	75.63	69.30	74.82
7	16,684.17	67.54	80.49	71.69	80.67	77.77	76.47	70.25	74.98
10	27,242.13	67.16	79.73	71.16	79.58	77.86	76.09	70.02	74.51
Dot Product									
3	5,139.16	65.81	80.37	71.32	79.43	77.84	75.29	70.37	74.35
4	5,648.41	66.07	80.94	71.57	79.96	78.77	76.16	70.33	74.83
5	6,085.94	67.21	80.85	71.91	80.22	78.66	75.91	69.50	74.89
7	7,141.78	67.90	80.97	72.33	80.35	78.41	77.66	70.53	75.32
10	8,412.60	67.10	80.51	71.26	80.11	77.98	75.76	70.49	74.74

Table 1: Results of SimCSE and its variations with hard positive

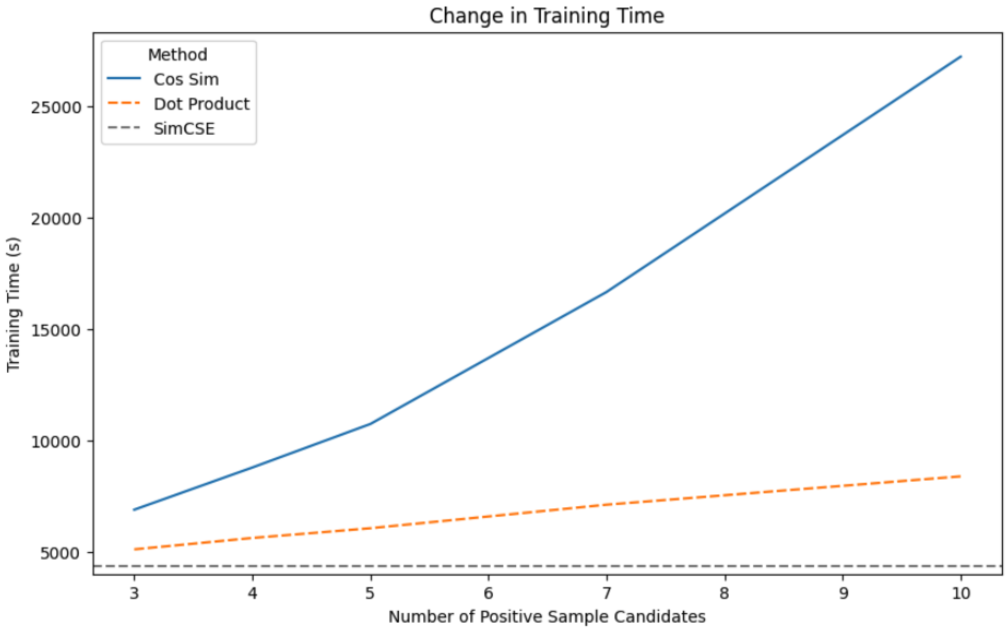

Figure 2: Change in Training Time