Evaluating Pre-trained Models for Unsupervised Contrastive Learning in Semantic Textual Similarity: A Comparative Study

Alice, Guangrui LU
guangrlu2-c@my.cityu.edu.hk

1. Abstract

This report rigorously evaluates trending pre-trained models that utilize contrastive learning techniques for semantic textual similarity tasks. We used a corpus of one million Wikipedia sentences on a Google Virtual Machine with a V100 GPU for model training. The models were tested using a variety of semantic text similarity tasks from the SentEval toolkit, including the STS benchmark tasks from 2012 to 2016 and the SICK-Relatedness (SICK-R) dataset.

Our findings reveal three key insights: (1) Despite the flexibility added by Rotary Position Embedding (RoPE) in the llama and gemma models, it proves less effective for the SimCSE contrastive learning architecture; (2) Larger pre-trained models do not necessarily yield better STS scores; (3) The robustness of the roBERTa model underscores the potential of robust pre-trained models in maintaining performance stability in contrastive learning settings.

2. Introduction

Semantic textual similarity (STS) represents a critical area of inquiry within the realm of natural language processing (NLP), with implications that extend to a multitude of applications such as information retrieval, text summarization, question-answering frameworks, text categorization, and word sense disambiguation. These applications underscore the foundational role of STS in enhancing our understanding and processing of human language (G Majumder et al., 2016)[6].

Both unsupervised and supervised learning are adopted in this field. Previous studies include encoder-decoder models, Deep Averaging Network encoder, transformer-based encoder, contrastive learning with BERT and angle optimized embedding with the large language model LlaMA2.

Skip-thought vectors (Kiros et al., 2015)[4] used text from books to train an encoder-decoder model which reconstructed the surrounding sentences of an encoded passage at the sentence level. Conneau et al. (2018)[2] used supervised data to train universal sentence representations on a natural language inference(NLI) task. A sentence encoder model based on bi-directional LSTM architecture with max-pooling was adopted. Universal Sentence encoder (Cer et al., 2018)[1] included two models. One is a transformer based encoding model. The other encoder model uses a deep averaging network (DAN). Training was made on both unsupervised data and supervised data from Stanford Natural Language Inference (SNLI) corpus. The study mainly focused on transfer-learning tasks. The best performance was achieved by making use of both word and sentence level transfer. Yang et al. (2018)[7] used 10 years' conversational data from Reddit since 2007 on two conversational response prediction models: DAN and transformer to convert sentences into embeddings. The encoders were tested on response prediction tasks. They also conducted multitask learning over conversational and NLI data from SNLI. Gao et al. (2022)[3] proposed a simple contrastive learning of sentence embeddings framework. It was coupled with pre-trained language models. Both unsupervised and supervised approaches using NLI datasets were adopted. Li et al. (2023)[5] considered angle optimization in a complex space. The objective function combines cosine, in-batch negative and angle difference. Combining the novel loss function with the large language model LlaMA2 makes the work outperforms in the STS evaluation tasks.

3. Related Works

This report is highly correlated to the previous work from Gao et al.(2022)[3], with some exploration on different choice of models. In their paper, BERT-like models are used as encoder to produce sentence embeddings. More specifically, they use the Masked-Language-Model (MLM) prediction heads inside these pre-trianed language models to output their hidden states and further convert them to the sentence embedding from such state. The unsupervised simple contrastive learning passes the same sentence to the pre-trained encoder twice to get two different dropout embeddings as "positive pairs". And the other embeddings in the same batch as negative. Because of its straightforward architecture, almost any pre-trained transformer models can be used as an encoder to test their potentials on STS benchmark.

Contrastive Learning is the last optimization that brings

alignments in embeddings. With an unsupervised dataset wiki1M, a softmax temperature-scaled cross-entropy loss are optimized for these pairing process.

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

where:

h_i is the embedding of the anchor sentence input.

h_i^+ is the embedding of a "positive" sentence. Here for unsupervised case, it is from the same sentence with different dropouts.

h_js are embeddings of the negative sentences, i.e. other sentences in the mini-batch.

$sim(,)$ denotes cosine similarity.

τ is a temperature parameter that scales the similarity scores.

N is the number of sentences in the batch.

4. Methodology

Acknowledged to the overwhelming effect of contrastive learning, more attempts on embeddings encoder were implemented on this experiment. Different from Gao's work to use BERT or RoBERTa as pre-trained models to encode input sentences, I propose the usage of llama, gemma and albert. Although llama and gemma are not optimized for Masked Language Modeling (MLM) tasks, there is still alternative API in huggingface. I used their prediction head for Causal Learning to output the hidden state, and then reveal the sentence embeddings. Due to limited computational resources, only llama2-7B and gemma-7B are adopted as comparison.

Albert is a language model in the BERT family, with similar optimization predication head for MLM tasks. I tested its performance on Contrastive Learning on same procedure as what they did in the SimCSE paper for BERT and roBERTa. Albert-based-v2 is adopted from huggingface, with 11M parameters.

5. Experiments

The experiments setup is logged on the table here1. It should be noted that, the experiment was originally performed on colab pro. But colab does not give user full access to Rust setup, which is essential to build tokenizer with the tokenizer api from huggingface. Moreover, due to limited computational resource, all models get implemented to Contrastive Learning once, with one set of hyperparameters as shown above.

The results are shown as below. The overall performance review is shown as in the figure2. RoBERTa-large unsupervised simCSE outperforms other models in most tasks in my experiments3. Although it could be the case that, Masked Language Modeling (MLM) head API is used for BERT

Pre-trained Models	RoBERTa	BERT	ALBERT	Llama 2	Gemma
Training Set	1 million Wikipedia sentences				
Device	Google Compute Engine, Virtual Machine (single V100-16GB GPU)				
Batch size	512	64			
Learning rate	1e-5				
Temperature	0.05				

Figure 1. Logs for Setup

family, while CausalLM head API is used for non-BERT family. This might give BERT family some advantage regarding of their different optimization purpose. Moreover, due to limited computational resources, only one set of hyperparameters is used for all models, this may prevent them to perform at their best.

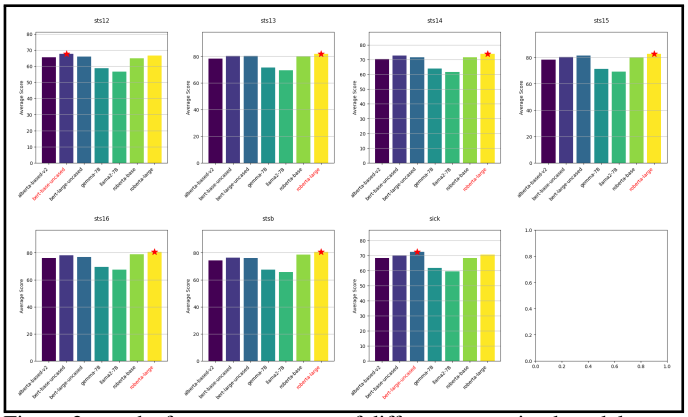


Figure 2. graphs for average scores of different pre-traiend models as the encoder for contrastive learning

	sts12	sts13	sts14	sts15	sts16	stsb	sick
alberta-based-v2	65.595430	78.372666	70.363729	78.195756	76.058694	74.210192	68.364721
bert-base-uncased	67.644819	80.424152	72.683877	80.247040	78.118291	76.281789	70.346418
bert-large-uncased	65.825490	80.441564	71.602235	81.453165	76.846042	75.963023	72.517727
gemma-7B	58.636461	71.600204	64.006215	71.152682	69.600229	67.356714	61.807556
llama2-7B	56.512006	69.609443	61.711329	69.217648	67.368911	65.690872	59.473813
roberta-base	64.920905	80.063749	71.517796	79.920514	78.732900	78.674828	68.392137
roberta-large	66.464714	81.927801	73.929494	82.710892	80.582071	80.571855	70.815224

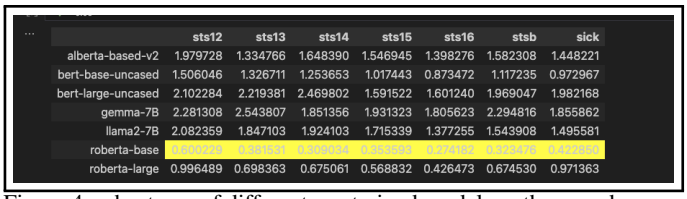
Figure 3. table of average scores of different pre-traiend models as the encoder for contrastive learning

The overall stability review is shown as below4. RoBERTa-base unsupervised SimCSE has the least standard deviation in every single task, and thus is the most robust model. The result aligns with its pre-trained models' optimization goal.

6. Discussions

6.1. Next Sentence Predication (NSP)

According to the evaluation result, BERT, ALBERT and RoBERTa have close performance, while their architecture



	sts12	sts13	sts14	sts15	sts16	stsb	sick
alberta-based-v2	1.979728	1.334766	1.648390	1.546945	1.398276	1.582308	1.448221
bert-base-uncased	1.506046	1.326711	1.253653	1.017443	0.873472	1.117235	0.972967
bert-large-uncased	2.102284	2.219381	2.469802	1.591522	1.601240	1.969047	1.982168
gemma-7B	2.281308	2.543807	1.851356	1.931323	1.805623	2.294816	1.855862
llama2-7B	2.082359	1.847103	1.924103	1.715339	1.377255	1.543908	1.495581
roberta-base	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
roberta-large	0.996489	0.698363	0.675061	0.568832	0.426473	0.674530	0.971363

Figure 4. robustness of different pre-traiend models as the encoder for contrastive learning

differ from one another. BERT has a known optimization goal as Next Sentence Prediction, but RoBERTa removes it, while ALBERT replaces this optimization objective with Sentence Order Predicaiotn (SOP).

Although NSP optimizes the contextual relationships between sentences, it seems beneficial only when long paragraphs are involved. In our STS evaluation, without NSP, both Contrastive Learnings with albert and roberta achieves similar or better results compared to Contrastive Learning with BERT.

6.2. Rotary Position Embeddings (RoPE)

Unlike BERT family, Llama and Gemma adapts Rotary Position Embeddings (RoPE) as part of its architecture. RoPE multiplies each token embedding by a rotation matrix that depends on its position. This integrates positional information directly with token embeddings in a dynamic manner. It should reduce the anisotropic drawback itself, by allowing similarity focused equally well on token embeddings as on relative position.

However, the expected result is not shown on my experiment, with two possible reasons:

With one set of CL training and one set of hyperparameters, llama and Gemma does not reach its capability.

Contrastive Learning aligns the vectors in different dimensions too well and outcast the effect of RoPE.

7. Future Work

Future research could explore model-specific hyperparameter tuning to fully capitalize on the unique capabilities of each pre-trained model. Additionally, investigating the impact of larger models could further enhance our understanding of the scalability and effectiveness of contrastive learning techniques in STS.

8. References

References

[1] D. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder, 2018.

[2] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data, 2018.

[3] T. Gao, X. Yao, and D. Chen. Simcse: Simple contrastive learning of sentence embeddings, 2022.

[4] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors, 2015.

[5] X. Li and J. Li. Angle-optimized text embeddings, 2023.

[6] G. Majumder, P. Pakray, A. Gelbukh, and D. Pinto. Semantic textual similarity methods, tools, and applications: A survey. *Computacion y Sistemas*, 20(4):647–665, 2016.

[7] Y. Yang, S. Yuan, D. Cer, S. yi Kong, N. Constant, P. Pilar, H. Ge, Y.-H. Sung, B. Strope, and R. Kurzweil. Learning semantic textual similarity from conversations, 2018.