Investigating Views for Contrastive Learning of Language Representations

Stanford CS224N Custom Project

John Nguyen
Department of Computer Science
Stanford University
nguyenjd@stanford.edu

Abstract

Contrastive learning is a self-supervised learning technique that trains models to learn representations of sentences where similar sentences are close together in vector space and dissimilar sentences are far apart. This project builds upon existing contrastive learning methods applied to transformers in the DeCLUTR paper [1] to analyze whether the sentence embeddings produced by contrastive learning can be comparable or better than standard Masked Language Models. In this project, I apply the SimCLR objective function with slice views, neighbor views, and neighboring slice views to a roberta-longformer model trained on a Wikipedia dataset and compare its performance to a roberta-longformer MLM pretrained model on transfer tasks such as sentiment analysis and topic prediction. I demonstrate that models trained with SimCLR objective function performs better on topic prediction tasks than MLM.

1 Key Information to include

- Mentor: Kendrick Shen

2 Introduction

The goal of contrastive learning is to learn an embedding space where similar sample pairs stay close to each other while dissimilar ones are far apart. Contrastive Learning has more frequently been used in computer vision (CV) to improve visual representations of objects through self-supervision. Pairs of positive samples (views) in CV are generated by augmenting the same sample in multiple ways, such as a crop, rotation, blur, reflection, and masking. For this project, I apply contrastive learning, namely SimCLR, which is a sentence-level objective function to transformer models to learn sentence embeddings, and experiment with a variety of views and different sentence data augmentations. These methods of view selection include neighboring span selection, slice views of the same span, and a combination of both neighboring and slice views for spans. This will be detailed in the Approach section. I then compare the performance these contrastive learning models to baseline roberta-longformer MLM on a variety of transfer tasks, namely the 20 Newsgroup topic prediction task and the SentEval SST2 transfer task. I find that the multiple models trained with SimCLR objective function performed better on topic prediction tasks than the MLM models.

3 Related Work

3.1 Contrastive Learning Methods in Computer Vision

Contrastive learning is commonly used in computer vision. This self-supervised learning method has many advantages over supervised learning, namely the ability to train on much more data without

the need for labels. A paper in 2018, "Unsupervised Feature Learning via Non-Parametric Instance Discrimination" lays a foundation for contrastive learning in CV, as it surpasses state-of-the-art on ImageNet classification by a large margin. [3] The concepts and data pipeline used in this paper is very similar to the data pipeline of my project, and all of the concepts regarding computer vision is applied to in a similar manner to natural language processing.

3.2 Masked Language Models

MLM is a token-level self-supervised objective where the model randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based only on its context. [2] The goal is to predict the word that has been masked out, and the loss is calculated by the difference between the output probability distributions for each output token and the true one-hot encoded labels. MLM has demonstrated the ability to produce good token-level embeddings, but there are limitations on its performance on Next Sentence Prediction (NSP) and topic prediction tasks.

3.3 DeCLUTR

The DeCLUTR paper [1] provides a sentence-level objective function SimCLR, which takes an input sentence and a neighboring span to serve as a pair of positive spans. This provides a sentence-level objective function for training models. DeCLUTR shows that contrastive learning model performs on a some of the SentEval transfer tasks as opposed to MLM. In this project, I use the same SimCLR loss function applied to RoBERTa Longformer models using a variety of different views rather than just one view in the DeCLUTR paper, and compare its performance on a set of transfer tasks.

4 Approach

In this paper, I define the term "span" to be a specific length of text which is selected from a document.

4.1 SimCLR Objective Function

The SimCLR [4] objective function is a contrastive self-supervised learning which does not require the use of a memory bank. It does so by generating pairs between spans within a batch of size N, creating positive pairs and negative pairs. Each span in the positive pair is passed through an encoder to get sentence embeddings to yield z_i and z_j. Then a batch of negatives examples are sampled, and the objective is to maximize the similarity between these two representations z_i and z_j as opposed to the negative samples in the batch. The loss is calculated as follows:

$$sim(u, v) = \frac{u^T v}{\|u\|\|v\|}$$

$$l_{i,j} = -\log \frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{k \neq 1} exp(sim(z_i, z_k)/\tau)}$$

where z_i and z_j are views. This is unique from MLM because SimCLR uses a large batch of negative views to compute loss and contrast the positive views against, whereas MLM uses masked tokens to compute loss for a single example.

4.2 Span Selection for Positive and Negative Views

SimCLR relies on a batch of positive and negative views to calculate loss. In this project, I create positive and negative views from 3 different methods of span selection.

Slice Views: I define slice views as masked views of the same span.

```
Example: I went to the bakery to buy a loaf of bread.
View 1: I <MASK>  the bakery  <MASK>  loaf of bread.
View 2: I went to <MASK> to buy a <MASK>.
```

This example is not set to the correct span length, but it demonstrates the method of generating a positive slice view. Random tokens within the selected span are masked out, and a positive pair cannot have the same tokens masked out. I chose to masked out about 15% of the tokens randomly.

Neighboring Views: I define neighboring views to be the same method used in the DeCLUTR paper, where a positive example are two neighboring spans within a document.

```
Example: I went to the bakery to buy a loaf of bread.
It was very crispy and tasty. The loaf costed $5.00.

View 1: I went to the bakery to buy a loaf of bread.
View 2: It was very crispy and tasty.
```

Here we see that the positive views are adjacent spans within the text.

Neighboring Slice Views: I define neighboring slice views to be a combination of both the slice views and neighboring slice views.

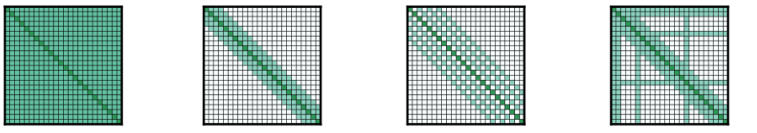
```
Example: I went to the bakery to buy a loaf of bread.
It was very crispy and tasty. The loaf costed $5.00.

View 1: I <MASK>  the bakery  <MASK>  loaf of bread.
View 2: It was <MASK> crispy and <MASK>.
```

In neighboring slice views, the spans must be adjacent within the document, and each of the spans are masked, following the masking protocol of slice views.

Negative Views: I define negative views as any span which does not match the 3 views mentioned above. These are generated in the SimCLR algorithm to contrast against the positive views. The views could be non-adjacent spans from the same document (hard negative), spans from different documents (easy negative). Hard negative views emerge because within a Wikipedia document, non-adjacent spans can still have the same topic, whereas spans a different wikipedia document is likely to have a different topic. I allow for both hard negatives and easy negatives when training the models.

4.3 Model Architecture

I chose to use the Longformer model for this project instead of a standard transformer. [5]. One major advantage of the Longformer is that it scales linearly with token size. This is because the attention is modified from the standard n^2 attention to a sliding window attention. Since the Longformer requires less memory for attention, I was able to dedicate more GPU resources to increasing batch size and sequence length.

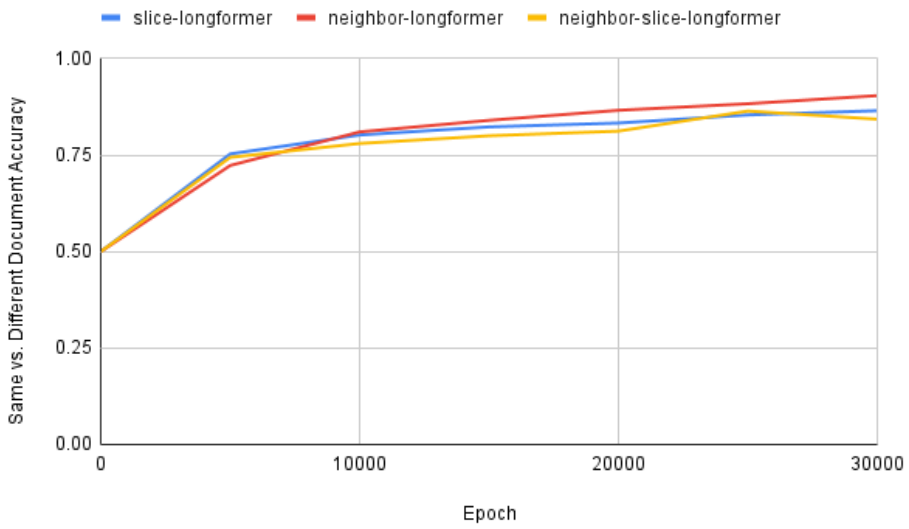


(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

The Longformer that I use is the roberta-longformer-base-4096, which is provided through Huggingface (https://huggingface.co/markussagen/xlm-roberta-longformer-base-4096). I chose RoBERTa over BERT based off of the performance in the DeCLUTR paper. Using the 3 methods of selecting views aforementioned, I train 3 different models which I will refer to as slice-longformer, neighbor-longformer, and neighbor-slice-longformer, each which correspond respectively to the 3 selected methods. I then evaluate and compare these three models' performance on transfer tasks to a roberta-longformer MLM of similar size.

4.4 Baselines

As a baseline, I evaluate each of my models (slice-longformer, neighbor-longformer, and neighbor-slice-longformer) on same vs. different document accuracy. This prediction task takes two random spans within a batch and shows it to the model to predict whether they came from the same document. The results of the baseline looks like



The neighbor-longformer performed the best at same vs different document prediction, which makes sense intuitively because the objective function is learning to to separate neighboring spans from non-neighboring spans. Given that neighboring spans tend to have similar topics and lead to closer sentence embeddings, these models should learn whether spans are in the document versus a different document. Slice views are generated from the same span, and so it has less inter-sentence context compared to neighbor-longformer. The neighbor-slice-longformer performed the worst, which was interesting. It could be because the masking of neighboring spans reduces the amount of information the model can learn from.

5 Experiments

5.1 Dataset and Evaluation Method

For training, I use the Wikipedia dataset via HuggingFace. The dataset are built from https://dumps.wikimedia.org/ where each example is a full Wikipedia article that has been cleaned. The cleaning process strips markdown and unwanted sections from the article, so the result is headings with text. It has over 300,000 articles which is randomly split into 80/20 train/test set. A sample unedited span from this dataset is: "A language is a structured system of communication. Languages are the primary means of communication of humans, and can be conveyed through speech (spoken language), sign, or writing. The structure of language". Note that before any preprocessing, the span length is 32 words in length. This sample will then be processed through different span selection methods.

Sentiment Analysis: For Sentiment Analysis, I use the SentEval SST2 dataset as an evaluation metric to test all of my contrastive learning models as well as the BERT MLM baseline model. [6]. SST2 is a binary classification task with either positive or negative labels. It consists of 67,000 training examples and 1,800 testing samples. One positive example is "equals the original and in some ways even betters it" and a negative example is "thoroughly awful experience". I train a logistic regression model on top of all 3 longformer models as well as roberta-longformer MLM for this transfer task.

Topic Prediction: Additionally, I evaluate the accuracy of each model on the 20 Newsgroups topic prediction task as referenced in the Datasets section.For the Topic Prediction

transfer task, I use the HuggingFace Newsgroup dataset [7]. The 20 Newsgroups dataset contains approximately 20,000 newsgroup documents, partitioned evenly across 20 different newsgroups. These newsgroups topics range from computer graphics to hockey to religion (https://github.com/huggingface/datasets/blob/master/datasets/newsgroup/newsgroup.py). One sample training example is

```
Subject: Re: Wings News and Playoff Thoughts From: kwolfer@eagle.wesleyan.edu In article
<1r8u6v$prv@msuinfo.cl.msu.edu>, twork@egr.msu.edu (Michael Twork) writes: >>roots in Detroit. He
would be a valuable asset to the Wings and Perhaps the >>Rangers could get a Zombo in return? > > >
> Wake up and smell the Norris!! Rick Zombo was traded to the Blues for Vince > Riendo (sp?) last
season. > > - Mike > Sorry Mike! What defensemen would the Wings be willing to give up for Beezer?
>
```

which is labeled for the topic: hockey.

5.2 Experimental details

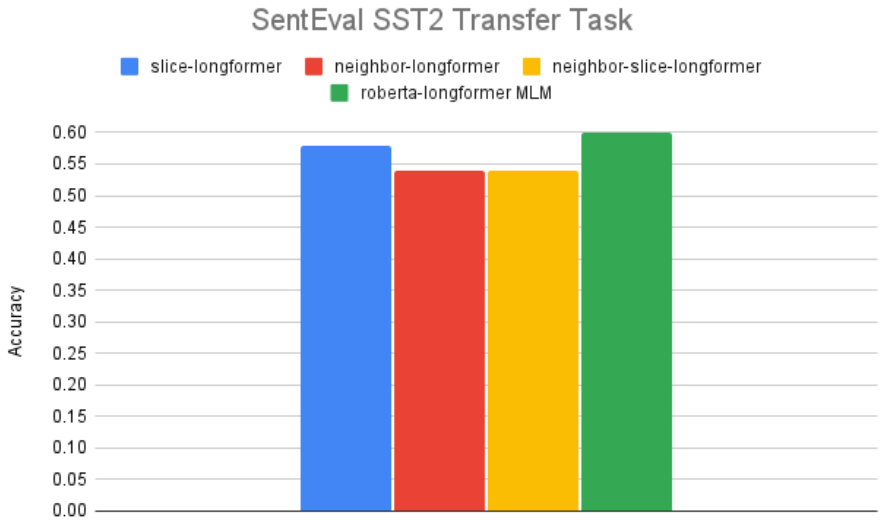
Fine-Tuning Pretrained Models: In order to compare the quality of sentence embeddings, I first created a baseline, which was a RoBERTa Longformer model trained with the MLM objective. I originally started with hyper-parameters: 4 attention heads, 3072 intermediate size, 2 hidden layers, an attention window of 16 and hidden size layers of 128, which was in the ballpark of the suggested hyper-parameters from the DeCLUTR paper for a small model. From these starting hyper-parameters, I slowly increased the size until I got to the following configurations: 4 attention heads, 3072 intermediate size, 4 hidden layers, an attention window of 16 and hidden size layers of 768. This was the maximum model size that I was able to train on a single GPU. Then, I trained the 3 Longformer model using the same hyperparameter configurations: 4 attention heads, 3072 intermediate size, 4 hidden layers, an attention window of 16 and hidden size layers of 768, in order to maintain consistency among the baseline model and my test model, each with different views for the SimCLR objective. The SimCLR models trained on a spans of length 32 and a batch size of 128, which was optimized also for GPU space.

Training Classifiers for Transfer Tasks: Once these models were fine-tuned on Wikipedia, I trained a logistic regression model on top of these model for both SST2 and 20 Newsgroups topic prediction. For SST2, I trained a binary classifier on top of all 4 models with a learning rate of 0.00002, Adam optimizer with betas=(0.9,0.999) and epsilon=1e-08. This was chosen off a hyperparameter grid search based on the recommendations of other experiments run on SST2 [8].

For 20 Newsgroup, the logistic regression model was trained with learning rate = 0.01, momentum = 0.9, weight decay = 0.0001. These hyperparameters were chosen based off recommendations from the DeCLUTR paper as well as other experiments ran on the 20 Newsgroup dataset. [9]. The output of this logistic regression model was a 20 dimensional vector corresponding to the 20 different topics.

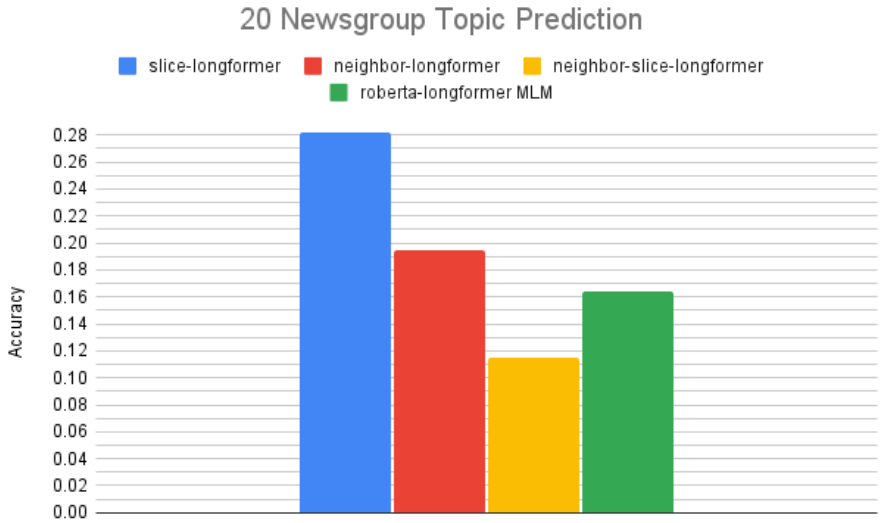
5.3 Results

Sentiment Analysis: For SST2, I had the following results



Since SST2 is a binary classification task, performance of random chance was 50%. This means that all of the models performed only slightly better than chance accuracy.

Topic Prediction: On the 20 Newsgroups topic prediction task, I had the following results



Performance by random chance on the Newsgroup dataset is 5% accuracy. All of the models performed much better than chance, with slice-longformer outperforming all of the other models.

6 Analysis

Sentiment Analysis: All of the models performed a little above chance, which was 50% accuracy. Quantitatively, the roberta-longformer MLM performed the best, but not by a statistically significant marging. Qualitatively, this could stem from the dataset that I trained on. Wikipedia articles are written in an unbiased tone. The lack of exposure to emotionally-charged training data could make it hard for all of these models to pick up on the nuance of sentiment analysis. Compared to the variety of data used in the DeCLUTR paper [1], the wikipedia dataset could very well be the causal factor in all of the models' poor performance.

Topic Prediction: All of the models performed above chance, which is was 5% accuracy. The slice-longformer performed the best, then neighbor-longformer, robert-longformer MLM, and then neighbor-slice-longformer. These results are promising because it demonstrates that models using the SimCLR objective can perform better on topic prediction tasks than the longformer model using the MLM objective. The slice-longformer significantly outperformed all of the other models, which was unexpected. I was expecting the neighbor-longformer to perform the best of the 3 SimCLR models on the topic prediction task, because neighboring spans within a Wikipedia article would have similar topics and thus learn sentence-embeddings using inter-span contexts. However, the slice-longformer performed the best, which means that qualitatively, the random masking of a single span helped the model learn sentence-embeddings which corresponded better to topics because the spans were always omitting information, therefore forcing the model to use surrounding context for topic prediction. In regards to the neighbor-longformer, the sentence-level embeddings generated from comparing inter-sentence positive and negative samples allow it to pick up on longer range topics and latent semantics that the MLM model could not learn as well. Thinking about the SimCLR objective with neighboring spans as views, this makes sense because we are trying to bring sentences that neighbor each other close together in representation. Sentences that are next to each other in a wikipedia article generally have similar subject / topic, which can be picked up by the neighbor-longformer. The neighbor-slice-longformer performed the worst among all of the models. This could be due to the fact that masking neighboring spans omits too much information for the model to learn about overall topic.

7 Conclusion

In this project, I explore the use of different views for models trained on SimCLR contrastive loss. The SimCLR models did not perform well on sentiment analysis, which could be a result of the Wikipedia dataset that the models were trained on. However, the slice-longformer and neighbor-longfomer achieves higher accuracy on topic prediction than the MLM longformer due to the ability to learn from inter-sentence rather than intra-sentence language structures, which is promising. In the future, I would like to explore a wider variety of transfer tasks to see how well the text representations of contrastive learning models generalize. Additionally, I would like to scale up the size of the models and analyze the more complex models on the same transfer tasks. Lastly, I would add a greater variety of training data to see if the performance of these models on sentiment analysis improves.

References

[1] John Giorgi, Osvald Nitski, Bo Wang, Gary Bader (2021)
DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations
https://arxiv.org/pdf/2006.03659.pdf

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2019)
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
https://arxiv.org/pdf/1810.04805.pdf

[3] Zhirong Wu, Yuanjun Xiong, Stella Yu, Dahua Lin (2018)
Unsupervised Feature Learning via Non-Parametric Instance Discrimination
https://arxiv.org/pdf/1805.01978.pdf

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton
A Simple Framework for Contrastive Learning of Visual Representations
https://arxiv.org/pdf/2002.05709.pdf

[5] Iz Beltagy, Matthew Peters, Arman Cohan (2020)
Longformer The Long-Document Transformer
https://arxiv.org/pdf/2004.05150.pdf

[6] *SentEval SST2 Dataset*
https://github.com/facebookresearch/SentEval

[7] *Huggingface 20 Newsgroup Dataset*
https://huggingface.co/datasets/newsgroup

[8] *Bert-Base-Cased Fintuned*
https://huggingface.co/gchhablani/bert-base-cased-finetuned-sst2

[9] *Fine-Tuning Bert Text Classificaiton (20 Newsgroup)*
https://www.linkedin.com/pulse/fine-tuning-bert-text-classification-20news-group-sharmila-upadhyaya