CS230

Improving Text Representations using Contrastive Unsupervised Learning

Project Category: Natural Language Processing

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

1 Introduction

The problem I am investigating is applying contrastive learning to improve text representations. Contrastive Learning is a growing field under unsupervised learning, has been used in computer vision to improve visual representations of objects. However, it has not been widely explored in natural language processing. This project applies contrastive learning methods to improve the quality of text representations, as an alternative to the popular masked language modeling (MLM). MLM is a token-level objective which does not perform too well on topic prediction transfer tasks. In this project, I explore a variety of views and objective functions for contrastive learning and evaluate the text representations on a set of transfer tasks, such as sentiment analysis and topic prediction.

2 Related Work

One of the key papers, "Deep Contrastive Learning for Unsupervised Textual Representations" (https://arxiv.org/pdf/2006.03659.pdf) applies contrastive learning to text representations, but it also combines in MLM loss in its objective function.

There is also the paper "Unsupervised Feature Learning via Non-Parametric Instance Discrimination" (https://arxiv.org/pdf/1805.01978.pdf) which applies contrastive learning to images.

Lastly I referenced the paper "Longformer The Long-Document Transformer" (https://arxiv.org/pdf/2004.05150.pdf) for suggestions on hyper-parameter tuning.

3 Dataset and Features

I am using the Wikipedia dataset via HuggingFace (https://huggingface.co/datasets/wikipedia). 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 I randomly split into 80/20 train, test split.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

4 Methods

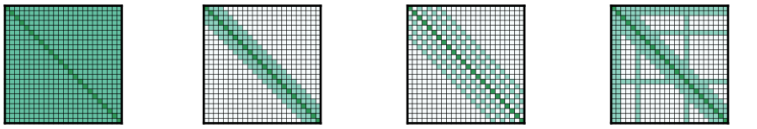
All of the models that were used in this project came from HuggingFace Transformers API. I wanted to compare the quality of word embeddings from a model trained with MLM objective versus a model trained using contrastive learning objective. In order to do so, I had to first create a baseline, which was a BERT 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.

I then trained a Longformer model using the same configurations in order to maintain consistency among the models. This was trained using the SimClr objective function which looks like:

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

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

where z_i and z_j are views. This objective function relies are a large batch size of pairs, where it brings together augmentations/views within the same pairing and creates negative samples by mixing the pairings within the batch. For the views, I used span lengths of 64 tokens and a batch size of 512, since SimCLR performs better with a bigger batch size. The span lengths and batch size was also configured to be the maximum size that could train on a single GPU. This trade-off of batch size and sequence length was the reason why I chose to use the Longformer instead of another BERT model as a comparison. 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

There were 2 main methods in choosing views for SimCLR. The first method was selecting neighboring spans from the same Wikipedia article as positive views and non-neighboring spans as negative views from the same Wikipedia document. The second method was selecting a span within a Wikipedia article and generating two positive spans, each randomly masking out multiple words within that original selected span. The negative views would be randomly masked out spans from different Wikipedia articles. Below is an example of view selection.

View Selection

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

Slice views:

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

Span views:

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

Negative views:
Negative views were different document spans within the same batch, which was used for calculating SimCLR loss.

In total, I had 4 models. The first model was the baseline BERT trained on MLM. The second was a Longformer trained on SimCLR with slice views. The Third was a Longformer trained on SimCLR with span views. The last model was a Longformer trained on SimCLR with a combination of span and slice views.

5 Experiments/Results/Discussion

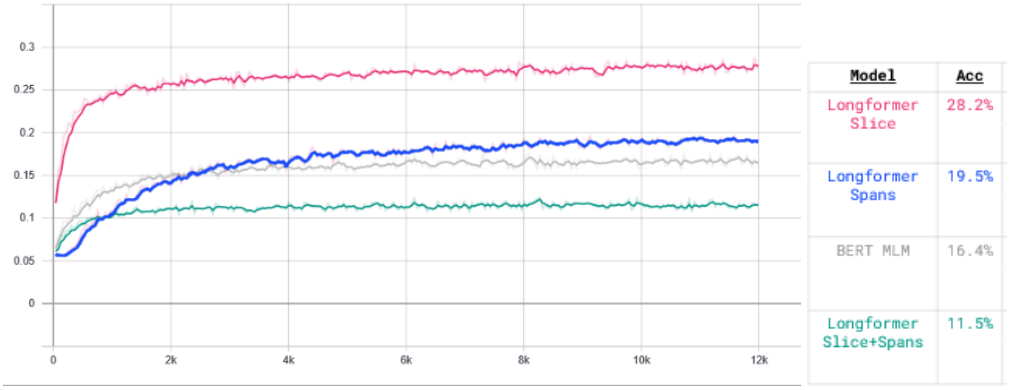
During training, the Longformer models were assessed on the metric of being able to recognize whether 2 spans were from the same document or different document, achieving approximately 82% accuracy on this. This served as a good sanity check for learning using the SimCLR objective function.

After training, I evaluated all models on SentEval SST2, which is a sentiment analysis transfer tasks. The SST2 transfer task consists of classifying whether a particular movie review is positive or negative.

Model	Accuracy (%)
BertMLM	53.34
Longformer Span	53.1
Longformer Slice	50.74
Longformer Span+Slice	53.05

All models only performed slightly above chance with similar accuracy. This was an interesting result which potentially stems from training data. Since I was training on Wikipedia articles, and Wikipedia articles are written in an unbiased tone. The lack of exposure to emotionally-charged training data could make it hard for these models to pick up on the nuance of sentiment analysis.

In addition to sentiment analysis, I evaluated all models on a topic prediction transfer task. I used the Newsgroup dataset, which contains text on 20 different topics ranging from religion to science and technology. The goal was to correctly classify a span of text with the appropriate topic. In order to run this transfer task, I trained a logistic regression model on top of the BERT and Longformer models with an SGD optimizer with learning rate = 0.01, momentum = 0.9, weight decay = 0.0001. The output of this logistic regression model was a 20 dimensional vector corresponding to the 20 different topics.



(Accuracy vs. Iterations)

All of the models performed above chance, which is is 5% with 20 different topics. The slice views performed the best, then span views, then BERT MLM, then Slice+Spans. These results are promising because it demonstrates that models using the SimCLR objective perform better on topic prediction tasks than models using the MLM objective. The wide span views and inter-sentence positive and negative examples allow the contrastive model 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 these models.

To conclude, the application of contrastive learning methods achieves higher accuracy on topic prediction than MLM due to the ability to learn from inter-sentence rather than intra-sentence language structures. The next steps of this project is to explore a wider variety of transfer tasks to see if the text representations of contrastive learning models generalize well. 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.

I would like to thank Alex Tamkin and Mike Wu for their guidance, as well as Jonathan Li for his advice on this project.