FREEZE!: Pretrained Language Models as Frozen Feature Extractors for Semantic Tasks

Hayden McDonald
hayden_mcdonald@brown.edu

Nicholas Marsano
nicholas_marsano@brown.edu

Oliver McLaughlin
oliver_mclaughlin@brown.edu

December 1, 2024

1 Introduction

Our project is an investigation into a few related areas. In short, we wanted to see if we could develop an architecture that could take the intermediate activations of a pre-trained decoder-only large language model (Henceforth LLM) and use them to do text-embedding tasks. We chose sentence similarity as our primary focus and motivation because of its simplicity. Importantly, in contrast to most other work using LLMs for text embedding, we are *not* fine-tuning the model. We are simply taking a subset of the intermediate activations and treating them as a kind of *"feature extracted"* (frozen) representation.

We chose this idea for several reasons.

1. Our original motivational question was: *"Do 'similar' prompts (to human readers) have 'similar' (under some metric) representations to a given language model?"*. By training a sentence-similarity model on LLM activations, we believed we could make progress on answering this question by *learning* a transformation of the activations that correlates to semantic similarity.

2. *If* we could train a small model to interpret a given LLM's activations and produce a useful text embedding (or set of embeddings) from them, we could skip costly fine-tuning and simply pre-train and fine-tune the small "interpreter" model. Pretrained small LLMs are cheap to do forward passes on and very plentiful, so if we could *just* use the intermediate activations and not have to do backward passes we could get a lot of functionality for very little compute.

3. We also wanted to investigate the question: *Do multiple intermediate activations perform better than just the last?* Typically when doing finetuning or transfer learning you remove the classifier end and use the last intermediate representation as your "feature extracted representation". We wanted to see if there was any performance benefit in using *multiple layer's* worth of representation. This is clearly the case in the finetuning setting [7] but it's unclear if it will work here.

4. From an interpretability standpoint, most distance or similarity measures (E.g cosine similarity) do not correlate strongly with *human ideas* of similarity (Figure 1). By training a similarity learner on LLM activations, grounded in human ideas of similarity, we hoped to produce a more interpretable measure of similarity for LLM activations.

5. There just isn't that much work on this particular topic. For this task you'd typically finetune the LLM using LoRA [5] or simply just use an existing text embedding model. We hoped to discover interesting properties and learn a lot about what it's like to try and *"disentangle"* an LLM's intermediate activations into something useful by pursuing this project. [1] We're not really interested in *what* information is where, we just want to see *if it's useable* for a simple task like STS.

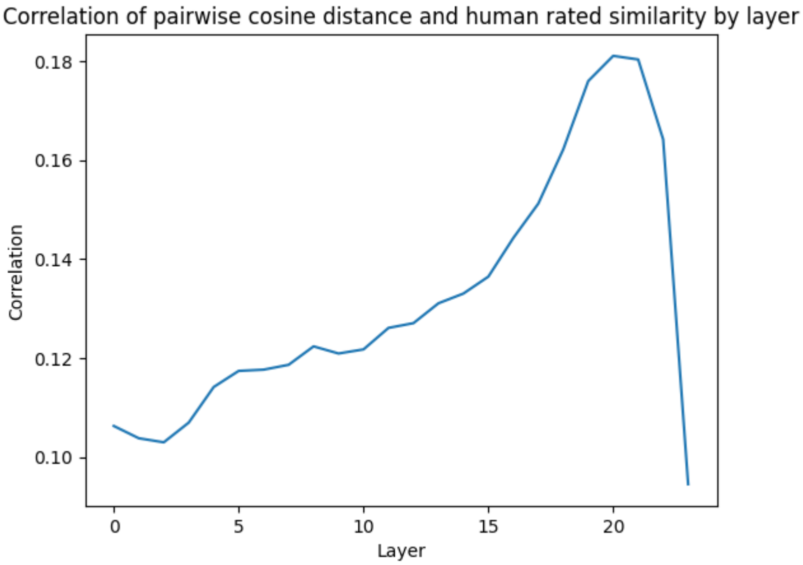

Figure 1: **The problem we are ultimately trying to solve**. Layerwise average correlation between gpt2-medium's embeddings and their cosine similarity vs. human similarity score for the STS training pairs

[1] There was only one paper we could find doing a similar setup – extracing latents from an LLM and applying them to a task – but it is very low quality and does not reveal many of the important details [1]. There are many papers *like* this one ([6] comes to mind) but almost none completely freeze the entire model and "start from scratch" as we did here.

We chose `gpt2-medium` (355M parameters) as our *"base model"* for study because Oliver was familiar with its architecture and it readily fit onto all of our GPUs. We also did minor experiments on `qwen2-0.5B` (391M parameters). Moreover we chose `gpt2-medium` because of its very low quality text generation ability. It was not at all a priori obvious to us that its representations would be fit for this task.

2 Related Work

3 Data

We have three major datasets that we used for both pretraining and supervised STS training.

1. STSB [4]
2. GenericsKB [2]
3. SNLI [3]

4 Methodology

Our model looks like (FIGURE). We do autoencoder pretraining stuff. Sentence-BERT style SNLI pretraining did improve performance over baseline but not better than the standard autoencoders.

Our loss function has evolved significantly since the beginning. We plotted the distribution of our residuals and noticed the spread could be tightened. Since this is a correlative task, we don't care about bias, just variance. So instead of doing MSE we just optimize to minimize the variance of the residuals directly

$$\mathcal{L}(\theta) = -\log(\mathrm{Var}(y - \hat{y}))$$

Where the added log improved stability

5 Results

Our main notion of success was performance on the STSB [4] benchmark. This benchmark uses correlation blah blah

This is a reasonable measurement in general because similarity is a relative notion, so the important part in comparing similarity is your precision (Consistency), not accuracy (How close you are to the literal values of the human judgements)

6 Ethics

7 Reflection

One big takeaway from this project is that as gigantic open source LLMs become more pevalent, it might be possible to chop off most of their layers and train little "interpreter" modules as we did here to leverage the incredibly rich and complex learned represenations at a fraction of the cost and computation time.

8 Division of Labor

- *Hayden.*
- *Nick.*
- *Oliver.*

9 Appendix: *STSB* dataset leakage

References

[1] Mahmoud Basharat and Marwan Omar. Harnessing gpt-2 for feature extraction in malware detection: A novel approach to cybersecurity, 2024.

[2] Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. Genericskb: A knowledge base of generic statements, 2020.

[3] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[4] Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity - multilingual and cross-lingual focused evaluation. *CoRR*, abs/1708.00055, 2017.

[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[6] Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Pretrained transformers as universal computation engines, 2021.

[7] Yixuan Tang and Yi Yang. Pooling and attention: What are effective designs for llm-based embedding models?, 2024.