Using Synthetic Data and Sparse Autoencoders To Interpret Large Language Models

Michael Fatemi [1] Aditya Kak [1] Brandon Yang [1]

[1]University of Virginia gsk6me, mbf3zk, jqm9ba @virginia.edu

Motivation

1. Large Language Models (LLMs) remain largely elusive to current understanding.
2. Polysemantic neurons respond to diverse inputs, complicating the analysis of networks based on individual neuron activity.
3. Sparse Autoencoders (SAEs) reveal that several features exist in superposition.

Modern Large Language Models (LLMs) display impressive abilities across different domains. In this work, we design a novel approach to enhance the interpretability of both small and large language models. We draw inspiration from Sparse Autoencoders (SAEs) [2]. We use both a pretrained GPT-2 sparse autoencoder[3] by OpenAI, as well as a custom-fine-tuned SAE built on top of gemma-2b. For both of these models, we generated synthetic text corpora in text classification, syntactic, and domain-specific texts. We then compare the activations in the hidden states of SAEs for these different text corpora and compare the results. Overall, we find that sparse features tend to get activated more when compared to neutral sentences, and the inner workings of GPT-2 and gemma-2b suggest that these models "think" in English. This preliminary work highlights innovation towards mechanistic interpretability in language models, and we hope to continue our work and expand to domains such as model pruning and SAEs for attention heads.

Method

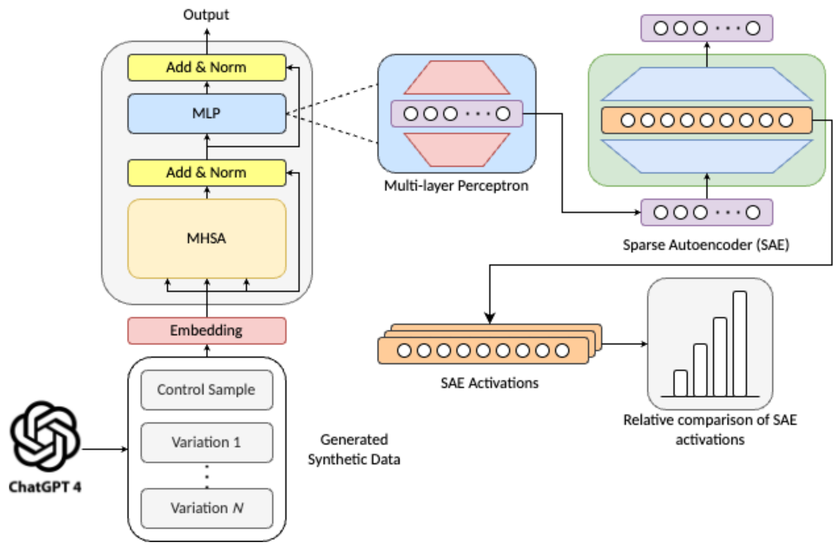

Figure 1. Data Generation and Analyzation Pipeline

Our data analyzation pipeline can be broken down into the following steps:

1. We generate synthetic task-specific data using GPT-4.
2. We batch process input tokens and feed them to the transformer model with SAE.
3. We aggregate the hidden state activations from the SAE and conduct comparative analyses, both within the same data variations and against activations elicited by other synthetic datasets.

Rivanna Compute Usage

To further research into the space of LLM interpretability we aim to create a SAE for the open-source model gemma-2b, and compare its general results with those of the publically available GPT-2 SAE. Currently within the NLP space, SAEs for models of this size and recency are nonexistent, with GPT-2 being less than half the size of gemma-2b and over 5 years older. Due to the large compute needs of the project we focused on the first six, most important layers of gemma, training the SAE over **150M** tokens. To accomplish this we utilized **3 NVIDIA A100** GPU's with an average GPU memory usage of **38.5GB** per GPU throughout training for a total of **36 hours**. Additionally, we utilized over **3TB** of storage, encompassing raw and tokenized text as well as model checkpoints.

Preliminary Results

SAE Activation Comparison & Analysis

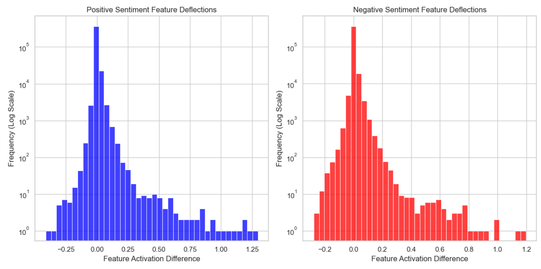

Figure 2. Differences in Activations for Positive and Negative-Sentiment Text

Figure 3. Activation between Subjects (Math, Physics, and Chemistry)

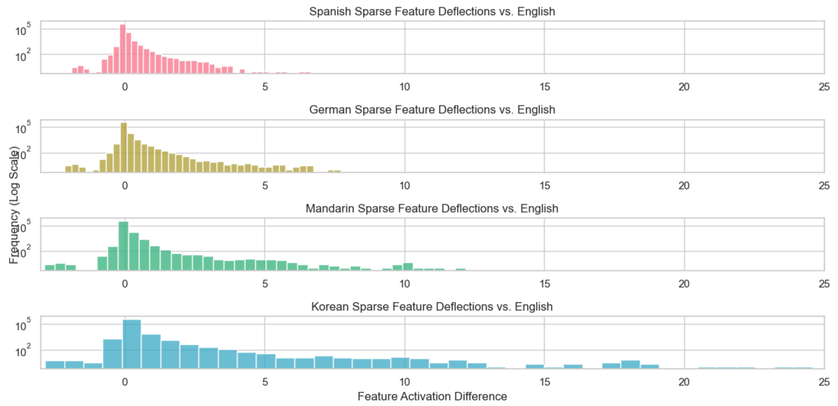

Figure 4. Language Deflections vs. English

Future Work

Though some reference literature indicates that approximately 100 million tokens have been effective in forming Sparse Autoencoders (SAEs), other studies have achieved greater success with a much larger amount of tokens, approximately 8 billion, for models smaller than Gemma. We aim to improve existing frameworks through data-parallel training (e.g., scaling to multiple A100 GPUs).

We also want to detect causality between sparse features in earlier vs. later layers. Prior work has been able to discover "circuits", or computational graphs representing changes in information through a transformer, through meticulous data curation and effort. There have been some early attempts to locate these automatically [1] and/or use sparse feature representations [4] (compared to compressed MLP activations). While a framework is still developing, we are optimistic that we will be able to discover more interpretable and complex circuits by scaling up sparse autoencoders to larger models, and creating more efficient patching methods.

References

[1] Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability, 2023.

[2] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[3] Tom Dupré la Tour. Sparse autoencoder for gpt2 small. `https://github.com/openai/sparse_autoencoder`, 2024.

[4] Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models.