Prompt Engineering

Abdul Hafiz Issah
Colorado School of Mines
aissah@mines.edu

1 Introduction

In the rapidly evolving landscape of artificial intelligence, Large Language Models (LLMs) have emerged as pivotal tools for a multitude of natural language processing tasks. These models, such as the GPT (Generative Pre-trained Transformer) series, have demonstrated remarkable capabilities in understanding, generating, and manipulating human language [1]. However, despite their impressive performance, LLMs often lack nuanced understanding and tend to generate biased or undesirable outputs in certain contexts. Prompt engineering offers a promising avenue to mitigate these shortcomings. By crafting tailored prompts or instructions, prompt engineering seeks to guide LLMs toward desired behaviors and improve their performance on specific tasks or domains [3]. The objective of this project is to investigate the variations in embeddings produced by a language model when presented with different prompts conveying identical contextual meanings. The hope is that this would give an idea of some prompting techniques that might ensure better outputs from LLMs.

2 Literature Review

Prompt engineering, a burgeoning field within natural language processing (NLP), has gained substantial attention in recent years due to its potential in fine-tuning language models (LMs) for specific tasks and domains. This literature review provides an overview of key studies exploring prompt engineering techniques and their implications for enhancing LM performance.

At the core of prompt engineering lies the concept of shaping LM behavior through carefully crafted prompts or instructions. This approach builds upon the theoretical frameworks of transfer learning and fine-tuning, leveraging pre-trained LMs as powerful language understanding tools [3]. By providing tailored prompts, researchers aim to guide LMs toward desired behaviors while minimizing catastrophic forgetting and domain shift. Prompt engineering encompasses a variety of methodological approaches, ranging from prompt design and selection to optimization techniques. [2] propose the use of prompt templates to encode task-specific information, enabling LMs to generate more accurate and contextually relevant responses. Additionally, [4] explores the impact of prompt size and complexity on LM performance, highlighting the importance of optimizing prompt structures for different tasks and datasets.

Empirical studies have demonstrated the effectiveness of prompt engineering in improving LM performance across various NLP tasks. [1] show that carefully constructed prompts can enable LMs to achieve state-of-the-art performance on few-shot learning tasks, surpassing traditional fine-tuning approaches. Furthermore, [3] investigates the role of prompt programming in expanding the capabilities of large LMs, showcasing its potential for enabling more flexible and adaptive language understanding.

While prompt engineering holds promise for enhancing LM performance, ethical considerations must be carefully addressed. [5] highlight the potential for adversarial attacks through prompt manipulation, underscoring the importance of robustness and fairness in prompt design. Moreover, issues of bias and fairness in LM outputs require vigilant attention to ensure equitable representation and decision-making [2].

Looking ahead, the field of prompt engineering presents numerous avenues for future research and innovation. Scholars are exploring novel approaches such as prompt programming and adaptive prompt generation to further enhance LM capabilities [3]. Additionally, interdisciplinary collaborations between NLP researchers, ethicists, and policymakers are essential for addressing the societal implications of prompt engineering and ensuring responsible AI development.

In summary, prompt engineering represents a promising paradigm for enhancing LM performance and advancing the state of the art in NLP. By leveraging tailored prompts and methodological innovations, researchers can unlock new possibilities for language understanding and generation while addressing ethical considerations and societal impact.

3 Proposed work

The aim in this project is to explore the distances between embeddings of prompts to understand how language models (LMs) behave when presented with different input stimuli. The goal is to uncover patterns in LM response generation and representation learning by observing how distinct embeddings are generated for prompts conveying identical contextual meanings. By analyzing these embedding distances, there is a chance to uncover the underlying mechanisms shaping LM behavior and derive insights to enhance model performance through prompt engineering across different tasks and domains.

Here, we consider the angles between the eigenvectors of different prompt pairs modified according to various contexts. Here are the types of token pairs that were considered:

1. **Synonymous Tokens:** Pairing tokens with similar meanings but different lexical forms, such as "big" and "large," to assess the LM's sensitivity to synonyms.
2. **Antonymous Tokens:** Pairing tokens with opposite meanings, such as "hot" and "cold," to evaluate the LM's ability to distinguish between contrasting concepts.
3. **Hyponym-Hypernym Tokens:** Pairing tokens representing a hierarchical relationship, such as "dog" and "animal," to examine the LM's understanding of broader and narrower semantic categories.
4. **Temporal Tokens:** Pairing tokens representing different time frames, such as "past" and "future," to investigate how the LM encodes temporal relationships and context.
5. **Geographical Tokens:** Pairing tokens associated with different locations or regions, such as "Paris" and "New York," to evaluate the LM's understanding of spatial relationships and geographic context.
6. **Sentiment Tokens:** Pairing tokens with contrasting sentiment polarities, such as "happy" and "sad," to assess the LM's ability to capture emotional nuances and sentiment analysis.

An example for synonyms is the prompt pair "The large wolf" and "The big wolf". We find the embeddings of these two prompts at various layers in the language model, find the eigenvectors of the respective embeddings, and find the angle between them as an indication of the similarity of embeddings. For this project, we use the "**tiny-stories-3M**" model, and for each prompt pair category, we consider 100 prompt pairs.

4 Results

In this section, we look at the results and what they might mean. Focusing first on the variation of angles across different eigenvectors, figure 1 illustrates three plots elucidating the angles between eigenvectors at distinct layers. Specifically, the top-left plot depicts angles pertaining to the first eigenvector, while the top-right and bottom plots represent angles associated with the second and third eigenvectors, respectively. The noticeable thing here is that all three plots look identical. Consequently, we proceed to narrow our comparative analysis solely to the first eigenvectors in subsequent discussions of our findings.

The similarity between synonym and antonym pairs seems to be a natural comparison to make. Figure 2 shows the angles for the synonym prompt pairs on the left and antonym prompt pairs on the right. They show a similar pattern where the range of angles increases from layer 0 to layer 5, reduces at

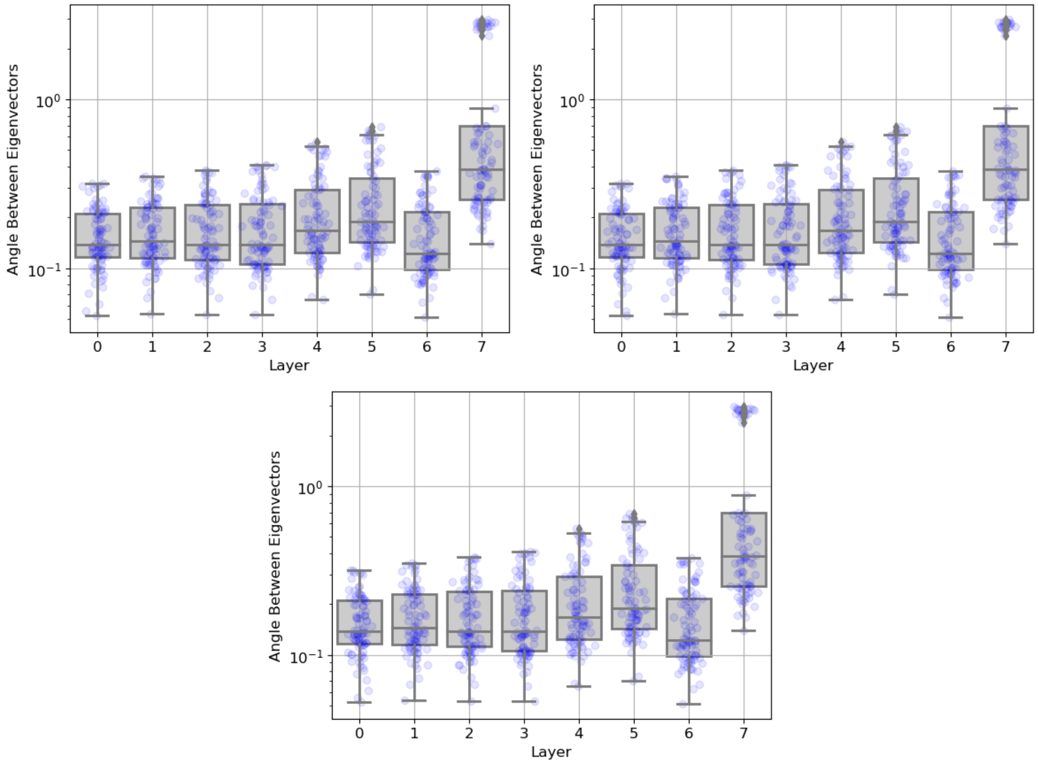

Figure 1: The variation in angles between corresponding prompts with modified synonyms.

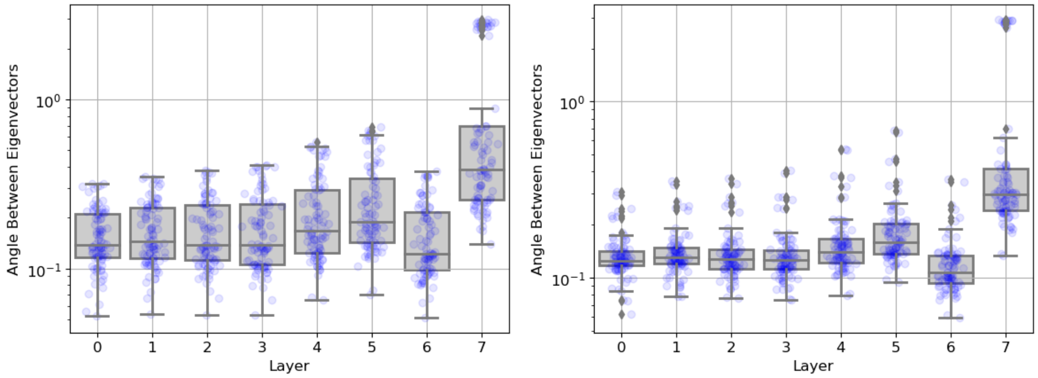

Figure 2: The variation in angles between corresponding prompts with modified synonyms (left) and antonyms (right).

layer 6, and attains the highest values in the last layer. However, an unexpected difference is the higher degree of variability in angles at the various layers for the synonym pairs. One would think that embeddings for synonym pairs would generally be closer to each other than antonym pair prompts.

Another natural comparison to make is the variation of spatial and temporal token pairs. Figure 3 shows the angles for the temporal prompt pairs on the left and prompt pairs with different locations on the right. These show a pattern similar to the synonym and antonym pairs with respect to the layers. The range of values for the angles for location prompt pairs is however significantly smaller than the range for temporal prompt pairs. They are also clustered closer together. This indicates that location keywords have less of an impact in text generation as opposed to keywords indicating time.

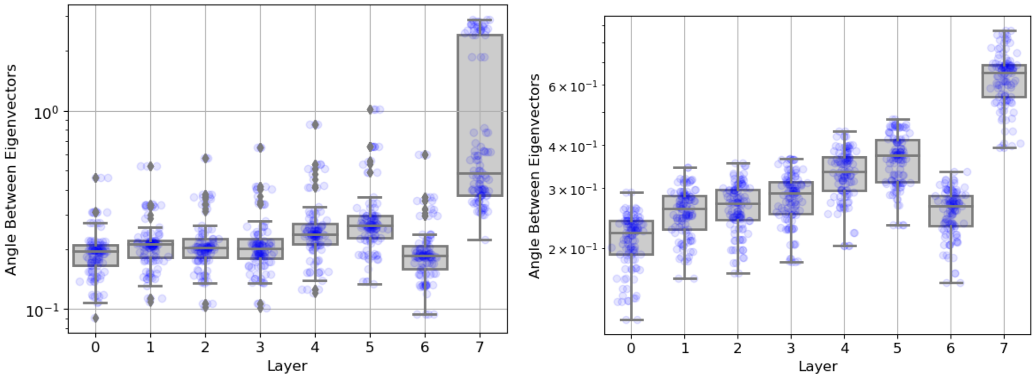

Figure 3: The variation in angles between corresponding prompts with modified temporal keywords (left) and location words (right).

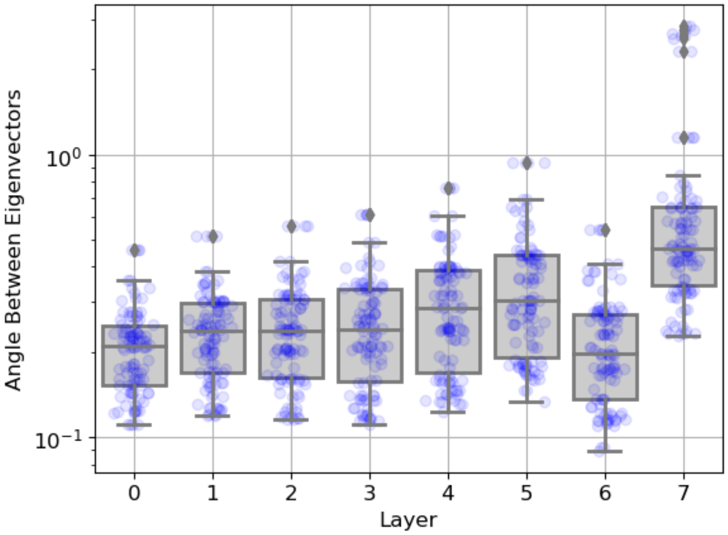

Figure 4: The variation in angles between corresponding prompts indicating different sentiments.

The plot showing the variation for prompts of varying sentiments in figure 4 again shows a similar pattern as the other plots. However, the spread in the distribution of angles for each layer signifies some sensitivity of the model to the different prompts considered.

The hyponym-hypernym pairs test is an indication of the relation of some general knowledge to a more detailed piece of information. The plot for this in figure 5 shows a wide range of angles but also shows a cluster of points around the average angle. The cluster may be an indication of the relationship between the hierarchy of information built into the model.

5 Conclusions

From the results of the experiments, we made a few observations. Synonym prompt pairs surprisingly show a wider range of variability than antonym prompt pairs; our model seems to be more sensitive to time than location words; the model shows some sensitivity to sentiments and likely has directions dedicated to the hierarchy of information. The associated code is at **https://github.com/aissah/LLMs_final.git**

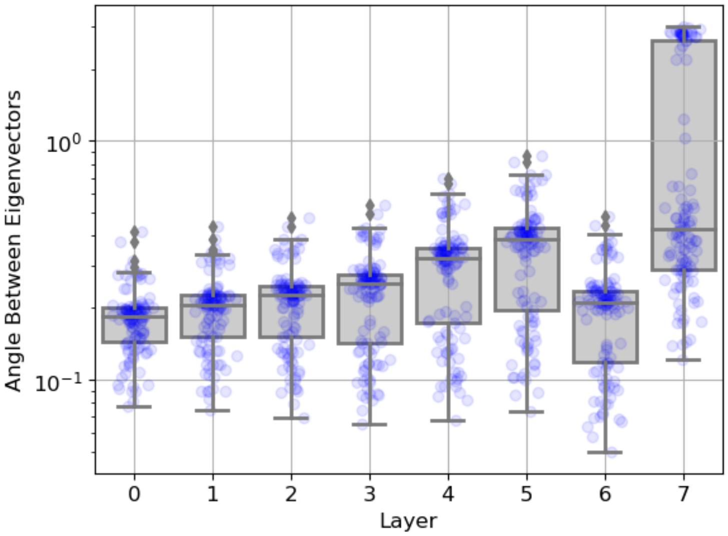

Figure 5: The variation in angles between corresponding prompts indicating a word and corresponding general version.

6 Contributions

Completed by **Abdul Hafiz Issah**.

References

[1] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: `2005.14165 [cs.CL]`.

[2] Fabio Petroni et al. *How Context Affects Language Models' Factual Predictions*. 2020. arXiv: `2005.04611 [cs.CL]`.

[3] Laria Reynolds and Kyle McDonell. *Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm*. 2021. arXiv: `2102.07350 [cs.CL]`.

[4] Timo Schick and Hinrich Schütze. *It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners*. 2021. arXiv: `2009.07118 [cs.CL]`.

[5] Eric Wallace et al. *Universal Adversarial Triggers for Attacking and Analyzing NLP*. 2021. arXiv: `1908.07125 [cs.CL]`.