Comparing Prompt Engineering Techniques

Anum Ahmad (aqa2001) and **Tramy Dong (td2748)** and **Kyra Ramesh Krishna (kr3026)**

1 Introduction

In-context learning has emerged as a defining capability of pre-trained large language models (LLMs), enabling them to perform tasks solely based on prompt input, without requiring any gradient updates. This makes LLMs adaptable to a wide range of tasks when prompts are engineered well. Exploring this area can help us improve model performance, reduce computational costs, and inform better design principles, making LLMs more accessible and effective for broader applications.

While prior studies have explored techniques like few-shot and chain-of-thought prompting, the factors contributing to prompt effectiveness remains an underexplored area in the natural language processing (NLP) community. This project seeks to systematically evaluate the impact of different prompt engineering strategies on model performance. Specifically, we investigate one-shot, few-shot, meta, and chain-of-thought prompting on a semantic classification task to address the central research question: How do different prompt engineering techniques (one-shot, few-shot, meta, and chain-of-thought) influence the performance of an LLM?

In addition to our primary research question, we would like to explore supplemental questions to deepend our understanding of prompt engineering:

1. Can prompts be programmatically generated (e.g., using techniques like Auto-CoT) to achieve comparable performance to manually crafted prompts?

2. How do small and large models perform relative to each other to different prompting techniques?

3. Are certain prompting techniques more effective for specific task types, such as classification versus question answering?

4. How does a fine-tuned encoder model perform relative to a generative language model in performance under different prompting strategies?

By comparing several techniques and models, we hope to better understand the factors that influence model performance and share findings that can be valuable for both academic research and practical real-world applications.

2 Datasets

For our experiment, we will use the poem _sentiment dataset, a collection of poem verses from Project Gutenberg, annotated with sentiment labels. The dataset, created by Emily Sheng and David Uthus for their paper, "Investigating Societal Biases in a Poetry Composition System.", presents a unique challenge in NLP due to the abstract and subjective nature of sentiment analysis in poetry. We selected this dataset because it will allow us to evaulate both linguistic sensitivity and contextual understanding in response to various engineered prompts.

Given additional time, we may also explore additional sentiment or classification datasets available on HuggingFace, allowing us to generalize our findings across a wider range of tasks and contexts.

3 Models

The collection of models we will import and run our prompts on includes Flan-T5, GPT-3.5-turbo-instruct, and Phi-2. Each of these models offers advantages and functionalities that will be valuable for various natural language processing tasks.

Flan-T5 is the instruction fine-tuned version of the Text-to-Text Transfer Transformer (T5) model developed by Google Research. This model is designed to perform well on NLP tasks by transforming each task into text-to-text format. Flan-T5

is an optimized version that implements instruction fine-tuning, the process of training a model on datasets specifically created to improve understanding of instruction prompts. In other words, Flan-T5 achieves good performance, especially in few-shot prompting, where it can understand and generate accurate outputs with minimal (3-4) examples. Another advantage of this model is its size, despite being smaller in scale compared to models like PaLM 62B, Flan-T5 is still able to deliver accurate results.

GPT-3.5-turbo-instruct is another powerful model we will use. The GPT-3.5 series has been optimized for versatility across chat-based and non-chat tasks. GPT-3.5-turbo-instruct has specifically been enhanced to understand and generate human-like responses through fine-tuning with the Chat Completions API. The model's ability to generate coherent and relevant output will be helpful when used in our various prompting tests. Compared to earlier GPT models, GPT-3.5-turbo-instruct has demonstrated better alignment with user intent and enhanced controllability.

Lastly, Phi-2 is a Microsoft-developed transformer model consisting of 2.7 billion parameters. While have relatively fewer parameters than typical models (13 billion), Phi-2 offers unique advantages in terms of flexibility and ethical AI exploration. However, Phi-2 has not been fine-tuned like most modern models, but this does not prevent it from being a sufficient model for our project. Microsoft specifically designed the model to be smaller, more accessible, and good at tackling challenges such as reducing harmful outputs, understanding biases, and enhancing the model's controllability in diverse scenarios.

By integrating these three models, we aim to leverage their complementary strengths when prompt engineering. Each model contributes unique capabilities and will be a solid foundation for exploring the limits and possibilities of prompt engineering techniques.

4 Preliminary Results/Analysis

We have implemented a sample Google Colab program to process the poetry dataset mentioned earlier for sentiment analysis. The labels are negative (0), positive (1), no_impact (2), and mixed (3). The code can be viewed at.
https://colab.research.google.com/drive/1c5XmrUMpe3K-ViL3pnzUFVzckPIDZB8?usp=sharing

For the environment setup, we installed necessary libraries like datasets and transformers (changed for each model). We also implemented multiple prompt templates, zero_shot_prompt(text), few_shot_prompt(text), chain_of_thought_prompt(text), and meta_prompt(text) to construct prompts for the given text in the respective formats. The function generates _response(prompt) will feed the model the formatted prompt and return the response which we will store in a dictionary and use for future analysis. We also tried out a few different models and because they seemed to not provide the right type of output, we went to office hours to troubleshoot. We tried using the *distilbert-base-uncased* and the regular *distilibert* model but it either provided no output or attempted to continue the sentence. We also tried GPT2 (https://colab.research.google.com/drive/18rmQlRux23DYwWi3wil but it would just repeat the prompt instead of answering. We will be attempting this again with more instruction-tuned models. We explain this more in the detailed plan of work on how we plan to fix this for future results and analysis. While we aren't able to produce much analysis on the successful models, what we can explain is that we learned a lot about how to understand what models can understand instructions and what models cannot. This information will be critical as we move onto using API based models in the next few days!

5 Detail Plan of Work

So far, we have conducted extensive research into various prompting techniques, gaining valuable insights into the strengths and limitations of different models, as well as the resources required to execute our experiment effectively.

Our plan for the next phase of the project is to allocate the next three to four days to experimenting with API-based models. This will allow us to better understand their behavior, as the majority of our work to date has focused on models available through HuggingFace. We have already invested considerable time testing these models and have attended office hours to explore alternative strategies for testing our hypotheses. Following a discussion with Arkadiy during office hours, we determined that a promising direction for our experiment would be to investigate whether we can fine-tune a classification model to achieve perfor-

mance comparable to, or even surpass, that of an un-tuned, instruction-tuned API model. We will spend a few days fine-tuning the classification models on a subset of the dataset and then run the tests again to see if the model has an improved score. We will also test chain of thought prompting on the large language models. Once we have successfully implemented and tested these models, we will assess their performance using the F1 classification metric to evaluate accuracy. We will also use a precision recall curve to see the effectiveness of the classification model.

If there is time, we also intend to explore a different problem type by applying our approach to math-related questions using the MathQA dataset. This will enable us to investigate how different models and fine-tuning strategies perform across diverse problem domains. While we anticipate that fine-tuning may be less effective for mathematical problem-solving tasks, we look forward to empirically testing this hypothesis.

6 Related Work

Zhang, Zhuosheng, et al. *Automatic Chain of Thought Prompting in Large Language Models*

This paper compared zero-shot chain of prompting (where you just write "Let's think step by step") and manual chain of thought prompting (where you provide a few examples showing how to break down a type of problem) on a math word-problem database. It aimed to show that diversity in chain of thought and providing a breakdown of the rationale can improve results. It showed that by initially clustering the data and then retrieving the few-shot example prompts based on similarity (a technique called Automatic chain of thought) they were able to match or exceed performance of having to manually find similar examples.

Zhang, Wenxuan, et al. *Sentiment Analysis in the Era of Large Language Models: A Reality Check.*

This was a paper that Arkadiy recommended to read during office hours. It was trying to solve a very similar problem as we are, comparing sentiment analysis with smaller models and LLMs. In this paper they also compared different types of few shot learning (1,5,10 examples). They found that the LLMs did outperform the smaller models, but it was not by that much. Additionally, they did note that in some cases, where tasks requiring structured sentiment output LLMs actually performed worse than small models like T5 in both automatic and human evaluations.

Brown, Tom, et al. *Language Models Are Few-Shot Learners*.

The paper highlights that GPT-3, when applied to various tasks, performs well without requiring any gradient updates or fine-tuning. Tasks are specified purely through text interaction, with examples provided in a few-shot manner. GPT-3 demonstrates strong performance across tasks like translation, question-answering, and including word unscrambling. They also tested out some math questions using basic arithmetic. The few shot prompting with GPT 3.0 performed the best. The paper also discusses how bias present in these models can influence the results, emphasizing the need to be aware of potential biases in model behavior and performance.

Sun, Xiaofei, et al. *Text Classification via Large Language Models.*

This paper explores a method to try and make LLM's better at text classification problems using a method called Clue And Reasoning Prompting (CARP). CARP is a way of prompting that first tries to use s keywords, tones, semantic relations, and references, and then uses this for final decisions. This method worked well when tested on Rotten Tomatoes and Yelp data. The single shot using CARP was able to perform better than a RoBBERTa model that was fine-tuned. Additionally, the few shot tests outperformed zero shot methods. Additionally, this experiment tested if out of domain training effects performance. It found that the supervised model did worse when being tested on a different domain, while the LLM was able to adapt and not have as much of a decrease in accuracy when the domain was changed.

Dong, Qingxiu, et al. *A Survey on In-Context Learning.* This was one of the papers Professor Muresan sent to us. This was a comprehensive review of current methods of in-context learning (ICL). This paper described ICL as similar to learning by analogy, and explains that instead of training happening in the training and fine-tuning stages of the model, the parameters remain the same. The authors noted that this can reduce cost to run models and make them faster since the training step will not be as computationally heavy and parameters do not need to updated for each new problem set. However, it also noted that this type of learning may be difficult due to many LLM having a token limit for the input. Additionally, for languages that might not have as many examples or sparse datasets

to provide examples it can prove difficult to apply this method.

Zhang, Zhuosheng, et al. *Automatic Chain of Thought Prompting in Large Language Models*

This paper compared zero-shot chain of prompting (where you just write "Let's think step by step") and manual chain of thought prompting (where you provide a few examples showing how to break down a type of problem) on a math word-problem database. It aimed to show that diversity in chain of thought and providing a breakdown of the rationale can improve results. It showed that by initially clustering the data and then retrieving the few-shot example prompts based on similarity (a technique called Automatic chain of thought) they were able to match or exceed performance of having to manually find similar examples.

7 Team Member Contributions

All team members actively participated in brainstorming, researching, implementing, and writing reports for the project. Anum researched several models to import, Tramy implemented prompt template functions, and Kyra researched the different prompting techniques.

8 References

Georgian Impact Blog, *The Practical Guide to LLMs: FLAN-T5*. Medium, 2023. Retrieved from https://medium.com/georgian-impact-blog/the-practical-guide-to-llms-flan-t5-6d26 cc5f14c0

Kiran Varghesev, *Phi-2 Microsoft's 2.7 billion-parameter Small Language Model*. Medium, 2023. Retrieved from https://medium.com/@kiranvarghesev/phi-2-microsofts-2-7-billion-parameter-small-language-model-c3dd0e6bf745

Krtarun Singh, *GPT-3.5 Turbo Instruct: A Powerful New Tool for Professionals*. Medium, 2023. Retrieved from https://medium.com/@krtarunsingh/gpt-3-5-turbo-instruct-a-powerful-new-tool-for-professionals-2e931f5e5874

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, and other. 2023. Few-shot Fine-tuning vs. In-context Learning: A Fair Comparison and Evaluation. arXiv preprint arXiv:2305.16938. Retrieved from https://arxiv.org/pdf/2305.16938.

Zhuosheng Zhang, Aston Zhang, Mu Li, Alex Smola. 2023. Automatic Chain of Thought Prompting in Large Language Models. arXiv preprint arXiv:2210.03493. Retrieved from https://arxiv.org/pdf/2210.03493.