Team 31 Project Final Report

Derek Hua, Akhil Manthina, Alan Yu
derekhua@g.ucla.edu, akhiluma@g.ucla.edu, azy7@g.ucla.edu

1. Introduction

The increasing complexity of NLP tasks demands more sophisticated and fine-tuned models. Our project focuses on prompting strategies to determine a Large Language Model's (LLM) capabilities in determining factuality or fairness of claims. This need is particularly pronounced in the context of information fairness and verification, where the implications of the models' decisions can have significant real-world impacts. Models are bound to have implicit biases due to the nature of the data they are trained on, so it is critical for researchers to design prompts to help models detect fair and factual outputs. We focus on the UniLC benchmark, a comprehensive dataset that encompasses a range of NLP tasks, including Hate Speech Detection, Social Bias Inference, GPT Toxicity, Climate-Fever, Health Fact-Checking, and Machine-Generated Fake News. We experimented with various prompting techniques: Zero-Shot, Few-Shot, Zero-Shot-Evidence, and Chain-of-Thought along with various LLMs.

2. Methods

2.1 Model Selection and Evaluation Framework: For zero-shot we started out testing with the Phi-1 model as a baseline, which we realized was not extremely adequate for our fairness assessment tasks due to its inability to complete text generation challenges related to fairness. We then shifted to the more advanced Phi-2 model, which was chosen for its potential to better handle the nuances of zero-shot classification in the context of fairness evaluation. We experimented with different models for both generating evidence and the binary classification itself, testing prompting strategies for each along the way. For evidence generation, we decided to use larger models (versus Phi-2's ~3B). This was to ensure that the model was not merely reinforcing its own biases. Using larger models allowed for more rich outputs to help the smaller models perform reasoning. Specifically, these were OpenAI's GPT 3.5 Turbo and Mistral AI's Mixtral 8x7B, both via API endpoints. As for the actual classification, we used Phi-2 and Mistral-7B Instruct run locally on our machine.

2.2 Prompting Techniques

Our baseline approach was Zero-Shot prompting, where we did not provide any prior specific examples or training on the classification, naively instructing the models to complete the task. In contrast, for Few-Shot we provide the model with prior examples and their corresponding outputs to give basis to its reasoning. Our next approach was Zero-Shot-Evidence prompting. In this approach, we prompted the model with evidence related to the claim. This evidence is generated by an LLM model and provides further context for the inference model to reason with. Finally, in Chain-of-Thought prompting, we provide the model with a series of steps, breaking down the prompt into smaller sections. This prevents LLM hallucinations as LLMs perform better with smaller tasks that do not require complex reasoning.

2.2.1 Zero-Shot

For Zero-Shot, we used the prior prompt templates from the starting github repository, as all we needed to do was embed the claim and task type (fairness or factuality).

Our prompt is as follows:

> Instruct:
> You will be given a claim and using commonsense reasoning, you need to respond with SUPPORTS or REFUTES, depending on whether you support or refute the claim.
>
> Claim:{claim}
> Is the claim {task_type}?
> Respond with SUPPORTS or REFUTES
> Output:

2.2.2 Few-Shot

For Few-Shot Prompting, we created a list of 8 examples for each category. Based on the domain of the input, we embedded the corresponding examples into the prompt. These inputs were designed to be general cases that provided the model with a broad understanding of the subject matter and how to respond. We noticed that if the examples were edge cases, i.e. trials it previously got wrong, performance would degrade and the model would be more likely to

produce no output (more information in results). We tested prompts with 2, 4, and 8 examples at a time.

Our prompt is as follows:

> You will be given a claim and using commonsense reasoning, you need to respond with SUPPORTS or REFUTES, depending on whether you support or refute the claim.
> Following are some examples:
> {examples} *(gathered from a dictionary of examples keyed by domain name)*
>
> Now Your Turn
> Claim:{claim}
> Is the claim {task_type}?
> Respond with SUPPORTS or REFUTES
> Output:

2.2.3 Zero-Shot-Evidence

For evidence generation, we chose GPT 3.5 Turbo and Mixtral 8x7b due to their speed and general efficiency in summarizing text. GPT has more parameters (estimated to be anywhere from 20B to 175B) and benchmarks well. Mixtral's draw is less censorship and an efficient training method, meaning it punches well above its weight class. Furthermore, Mixtral is a Mixture of Experts (MoE) model, increasing its inference speed, useful for generating large amounts of evidence cost-effectively via API endpoint. API models also are more up to date versus hugging-face pretrained models, meaning they would be better at generating factual evidence.

Our prompt is as follows:

> Instruct:
> You will be given a claim and information about the fairness or factuality of the claim.
> You have to generate detailed evidence for the claim given information about it.
>
> Claim: {claim}
> Information: {information}
> Evidence Output:

2.2.4 Chain-of-Thought

Our custom prompt template was Chain-of-Thought, where we broke down the claim into smaller tasks for the model. We were inspired by “Interpretable Unified Language Checking” and created a prompt similar to their few-shot fact generation pipeline [1], which they refer to as ½-Shot Prompting.

However, we discovered that the model would not follow the third task of responding with “SUPPORTS” or “REFUTES” and would generate outputs such as “The claim is fact …” and explain its reasoning. To combat this we changed the prompt to have the model respond with only “Yes” or “No”, which greatly reduced the amount of extraneous output. We theorized that this may be because answering “yes” or “no” to questions is more common in natural language, and thus the models would likely have seen it in training data.

Our prompt is as follows:

```
Instruct:
Claim:{claim}
1. Summarize the claim.
2. Generate a related natural or social {fact_type} fact.
3. Is the claim {task_type}? Respond only with Yes or No.
Output:
```

2.4 Utilizing the Transformers Library: We used the transformers library from Hugging Face for the ease of integration it offers for language models, including Phi-2. This library made it easy to work with the Phi-2 model with downloadable pretrained models along with a tokenizer. It also allowed us to seamlessly download and test other models such as Mistral 7B.

2.5 Evaluation Metrics: To assess the model's performance, we used Overall Accuracy and Overall F1-Score from scikit-learn’s metrics library. Specifically, Accuracy is calculated by the ratio of correctly predicted instances to the total number of instances in the dataset. It provides a straightforward measure of overall model performance. F1-Score combines both precision and recall into a single value, providing a more balanced assessment of a classifier's performance.

3. Results & Discussion

3.1 Performance Metrics

Model & Prompting Technique	Accuracy	F1-score
Phi2-Zero-Shot	0.70	0.67
Phi2-Few-Shot	0.53	0.63
Phi2-Chain-Of-Thought	0.67	0.54
Phi2-Zero-Shot-Evidence (Evidence from Mixtral)	0.72	0.69
Mistral-Zero-Shot-Evidence (Evidence from GPT 3.5)	0.72	0.70
Mistral-Zero-Shot-Evidence (Evidence from Mixtral)	0.78	0.73

3.2 Analysis:

Our baseline of Zero-Shot prompting on Phi2 already achieved a respectable accuracy of 0.70 and f1-score of 0.67. Microsoft's pretraining of the model was already proficient in factuality and fairness detection, especially fairness as there was less of a tendency to hallucinate in those domains.

Our Few-Shot prompting was not as successful, due to issues with the model's outputs. We discovered that the model would be heavily biased toward supporting the claim, performing even worse than Zero-Shot prompting. When looking deeper, we discovered that the reason was that the model was outputting nothing, and thus our classifier would mark everything as "SUPPORTS". We experimented by changing the amount of evidence provided, max output tokens, and even changing the model to a fine-tuned version of Phi2, dolphin 2.6. However, none of our changes achieved any substantial improvements, although increasing the number of examples provided some benefit. We suspect that this issue has to do with Phi not correctly understanding the prompt templates given examples, only understanding an instruct: -> output: structure as outlined on hugging-face templates. We rule out the issue being a matter of context window size because more examples resulted in more valid outputs.

The most promising prompting technique with Phi-2 was Zero-Shot-Evidence, which we focused the majority of our efforts on. We ran into some issues with the obscenity in the claims, resulting in many publicly available LLMs rejecting our prompts. To combat this, we used prompt engineering to get around built-in guardrails, convincing the model that it was generating results for a fictional world. Mixtral, with its reduced guardrails, slightly increased the accuracy and f1-score of the Phi-2 model. Furthermore, when we used a larger model, Mistral, to run inference instead, we achieved even better results achieving our best accuracy of 0.78 and f1-score of 0.73. We did experiment with using GPT 3.5 to generate evidence too, but the performance suffered despite the model being more powerful due to the number of rejected prompts. Thus the inference model had less evidence to base its reasoning on and performed worse.

Our final experiments were done with Chain-of-Thought prompting techniques. As seen in the table the final accuracy and f1-score were lower than the Zero-Shot prompting. We theorize that this may be due to Phi-2's small size and inability to maintain context across long outputs.

References

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, James Glass. 2023. Interpretable Unified Language Checking. [1]