CS 162 Final Project - Exploring the Factuality and Fairness of LLMs

Abbas Bakhshandeh
UCLA
ID #105208768
abbasb@g.ucla.edu

Riley Xin
UCLA
ID #005570778
xinr@g.ucla.edu

Reetinav Das
UCLA
ID #005504665
reetinav@g.ucla.edu

Kevin Liu
UCLA
ID #305621696
kevinyliu@g.ucla.edu

Abstract

This project explores the Phi-2 model capabilities in determining whether claims are fair as well as factual. Various different prompting methods were utilized in order to evaluate which method achieved best performance in classifying the claims. While a baseline accuracy and F1 score was achieved using a base model, the accuracy was then optimized for through different testing and experimentation methods. The best performance was a testing accuracy of 72% and a training accuracy of 72.94%.

1 Introduction

This project explored the various methods by which the Phi-2 language model could be utilized to identify claims as fair or factual. The Phi-2 model is pre-trained by Microsoft, and is instantiated in our code through Huggingface. The goal for our model was to be able to identify whether an input claim provided as text was fair or factual. Before any experimentation, due to the completion of Milestone 2, an accuracy of 70% and an F1 score of 68% was found. While we explored different prompting methods and the project, we were also primarily focused on trying to improve the accuracy metric as much as possible. The baseline prompting methods that were explored include Zero Shot Evaluation, Few Shot Evaluation, and Zero Shot With Evidence Evaluation. Additionally, many variations of these techniques and other experiments were taken in order to explore how the model's accuracy could be improved. Before we began our exploration, we completed Milestones 1, 2 and 3 as specified in the project description. In Milestone 1, we implemented the evaluation metrics of accuracy and F1-score to be able to evaluate our model. In Milestone 2, we implemented the formation of prompts for inputs to the Phi-2 model, language generation from the Phi-2 model, zero shot prompting. In Milestone 3, we implemented few shot prompting, data generation, and zero shot with evidence evaluation. Over all explorations, the highest test accuracy was 72% and the highest training accuracy was 72.94%.

2 Methods

In this section, we list each method and exploration we completed in addition to the final accuracy and F1 score. The accuracies for all explorations are included in the Results section (3). A detailed discussion of these results are completed in the Discussion section (4). We also give sample commands that were used in the process of creating the predictions from the model. Sample commands for accomplishing these tasks and explorations can be found in Appendix A.

2.1 Zero Shot Evaluation Prompt

As the baseline method, zero shot prompting was implemented. Zero shot prompting involves creating a prompt which only includes the claim without any additional evidence or data generation. In addition to the claim, the prompt also asks the model whether it supports or refutes the claim. Without any modifications, we found the training accuracy to be 66.21%, and the F1 score to be 65.02%.

2.2 Few Shot Evaluation Prompt

In few-shot evaluation prompting, a claim is combined with examples to form a prompt. The model is then asked to support or refute the claim. The number of examples that are included in the prompt can be inputted by the user as an argument in the input command. The performance of the few shot evaluation prompting was an accuracy of 70.61% and an F1 score of 64.9%.

2.3 Zero Shot With Evidence Evaluation

In zero-shot with evidence evaluation prompting, a claim along with evidence about the claim (which has been generated by the model) is combined to

form the prompt. The evidence is created by first calling the phi zero shot evidence prompt. In our case, we used this prompt to create evidence for the entire set of train_claims.jsonl. This evidence was then paired with the original train_claims.jsonl in order to get final predictions using the zero shot with evidence evuation.The model is then asked to support or refute the claim. The performance was an accuracy of 63.24% and an F1 score of 65.25%.

2.4 Exploration 1 - Adding Task Name to Prompt

In the train_claims.jsonl file that is provided, there is a "domain" property for each claim. For the first exploration, we attempted to include this task name in the input prompt to the Phi-2 Model, to see how it would affect performance. This is essentially a variation of zero shot with evidence evaluation, where the evidence in this case is only the given task name for the claim. In regards to the training data, the performance was an accuracy of 71.5% and an F1 score of 66.7%. For the test data (as tested on Gradescope), the performance was an accuracy of 70% and an F1 score of 67%.

2.5 Exploration 2 - Reducing Prompt Length for Zero Shot

For exploration 2, we dramatically reduced the prompt size for zero shot. Primarily, the first sentence was reduced in the default prompt for zero shot. The resultant performance was a training accuracy of 58.29% and an F1 score of 62.86%. Due to the low training accuracy the testing accuracy was not measured on Gradescope.

2.6 Exploration 3 - Adding Domain and Task Name to the Prompt

In this exploration, we added the task and domain name to the original zero shot prompt. For exploration 3, the performance was an accuracy of 70.88% and an F1 score of 61.87%

2.7 Exploration 4 - Utilizing Language Generated, Task Name, and Domain Name information Combined with Zero Shot

In this exploration task, we used the language generated, task name, and domain name information - which were all provided in the original training claims file - with the zero shot prompting method. For exploration 4, the training performance was an accuracy of 72.82% and an F1 score of 63.63%.

2.8 Exploration 5 - Language Generated Information and Task Name

In exploration 5, a combination of information from the input claims was used that wasn't tried before (the language generated and task name information). Essentially, the domain name information was left out, and this led to slightly worse performance than in exploration 4 where all were combined. For exploration 5, the performance was an accuracy of 71.4% and an F1 score of 67.76%.

2.9 Exploration 6 - All Available Information and Asking to Think Before Answering

For exploration 6, the best performing previous exploration (exploration 4) was taken and the line "Think first before answering." was added at the end of the prompt in constants.py. This means in totality, the prompt contained the language generated info, task name info, and domain name info. We thought perhaps if the model was told to think first before rushing to a conclusion it would lead to better performance. This was inspired by the "chain of thought" command that has been used to achieve better results in prompting (Wei et al., 2022). The performance was an accuracy of 72.97% and an F1 score of 63.94%.

2.10 Exploration 7 - All Available Information and Asking to Think Before Answering as well as to Focus on Being Accurate

This was the same prompt as Exploration 6, except an additional line was added to "Focus on answering accurately". Since the highest performance was on exploration 6 up to this point, the goal of this exploration was to attempt to slightly add new relevant remarks to the phi 2 model, such as to focus on accuracy, in an attempt to increase overall accuracy. The performance was an accuracy of 71.24% and an F1 score of 61.28%.

2.11 Exploration 8 - Domain Name, Task Name, as well as to "Make your best educated guess."

This exploration involved removing the information that we believed to be not as relevant, such as the language generation, and also add a line stating "Make your best educated guess". We were hoping that reducing the length of the prompt by removing the language generation info as well as

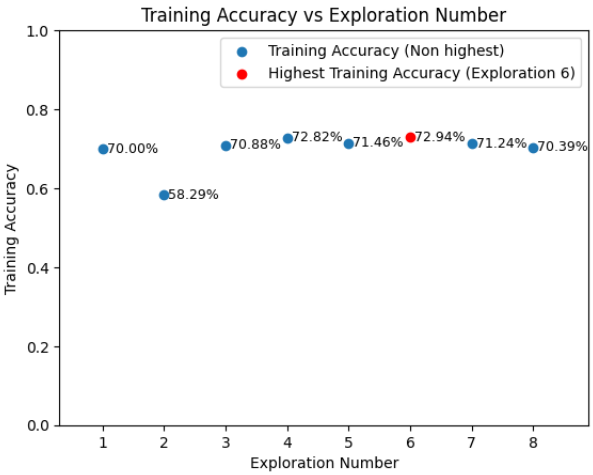

Figure 1: Training Accuracy vs. Exploration Number

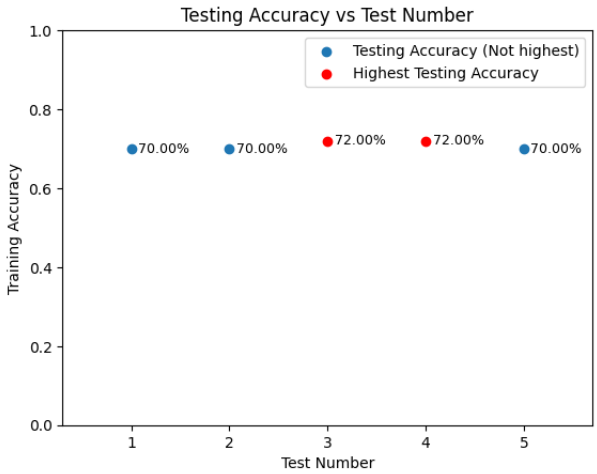

Figure 2: Test Accuracy vs. Test Number

only providing one extra line for guidance would lead to better performance. For exploration 3, The performance was an accuracy of 70.39% and an F1 score of 63.42%.

3 Results

A table of the results for exploration number and training accuracies are shown in Table 1.

Exploration Number	Accuracy (Training)
Zero-Shot	70%
Few-Shot	70.61%
Zero-Shot with Evidence	63.24%
1	71.5%
2	58.29%
3	70.88%
4	72.82%
5	71.46%
6	72.94%
7	71.24%
8	70.39%

A table of the results for selected exploration numbers (due to limited Gradescope submit entries) and testing accuracies are shown in Table 2.

Test Number	Accuracy (Testing)
Zero-Shot	70%
Test 1 - Exploration 1	70%
Test 2 - Exploration 4	72%
Test 3 - Exploration 6	72%
Test 4 - Exploration 7	72%
Test 5 - Exploration 8	70%

4 Discussions

A discussion of the results of the explorations as well as the baseline models, our reasoning behind these results, as well as additional ideas regarding the project are below.

4.1 Best Performance

The best testing accuracy that was a tie achieved was 72% by exploration 4, 6, and 7. The best training accuracy was 72.94% by exploration 6, which was including task name, domain name, language generation information, and additional leading statement "Think first before answering." paired with the original zero shot prompt.

4.2 Exploration 1 - Adding Task Name to Prompt

For the first exploration task, we added the Task Name to the original zero shot prompt. Our results show that the training data had accuracy of 71.5% and an F1 score of 66.7%. In both accuracy and F1 score, the evaluation metrics improved compared to the baseline zero shot case. This makes intuitive sense to use because if the Phi-2 model is given the task information, it can more easily make a determination as to whether the claim is factual or fair. By knowing more precisely the task at hand, and hence what it should be evaluating for, the Phi-2 model is able to make better predictions (which leads to a higher accuracy and F1 score as compared to the baseline case).

4.3 Exploration 2 - Reducing Prompt Length for Zero Shot

The inspiration for this idea was due to observing the trend that zero shot and shorter prompts were performing better than longer, more detailed prompts. By shortening the prompt length heavily (particularly the first line) of the original zero shot prompt, we hoped the model would be more influenced by the claim itself rather than the instructions. By adding the domain information to the prompt for the zero-shot case, we found that the accuracy stayed relatively constant at 58.29% and an F1 score of 62.86%, both very less than the original zero-shot scores (70% accuracy and 67% F1 score). However, the F1 score greatly decreased from 67% to 61.87%. Our results show that this method of exploration did not in fact improve the accuracy results, and worsened the F1 score. One reasoning for this is that we cut down the prompt too much; without the bulk of the instructions / lead-in to the prompt, it is likely the Phi-2 Model got confused leading to slightly worse performance.

4.4 Exploration 3 - Adding Domain Name and Task Name to Prompt

Since the training dataset provided included the domain information for the claim, we hypothesized that including it in the prompt would possibly improve performance. We modified the original zero shot prompt to include a line designating the domain name. This line was added to constants.py: "This is the domain of the claim domain". By adding the domain information to the prompt for the zero-shot case, we found that the accuracy stayed relatively constant at 70.88% (near the 70% for the baseline zero shot case). However, the F1 score greatly decreased from 67% to 61.87%. Our results show that adding the domain name did not improve, and in fact actually worsened the Phi-2 Model performance. One reason this could possibly occur is that the domain name is too broad to actually help the Phi-2 Model make evaluations of the claim.

4.5 Exploration 4 - Utilizing Language Generated, Task Name, and Domain Name information Combined with Zero Shot

In this exploration, we combined the language generated, task name, and domain name options from the training claims file with the original zero shot prompt. Doing this led to a training performance was an accuracy of 72.82% and an F1 score of 63.63%. For the test performance (measured on Gradescope), the accuracy was 72% and the F1 score was 64%. We theorize that this led to a marked improvement in accuracy due to the combination of providing the Phi-2 model a plethora of context information. Being able to know that the sentence is human generated perhaps allows the Phi-2 model to catch errors more easily than if it was unaware. This could explain the increase in accuracy.

4.6 Exploration 5 - Language Generated Information and Task Name

In this exploration, we only included the task name and language generation information in the prompt (with zero shot), and left out the domain name. The accuracy was slightly worse (71.5% vs 72.8%) than when including all three pieces of information in the prompt.

4.7 Exploration 6 - All Available Information and Asking to Think Before Answering

Since exploration 4, where all information was included (language generation, task name, and domain name) was the previous best training accuracy, we though it would be best to experiment further with the prompt. By adding the phrase "Think first before answering" we were hoping that that phi 2 model would pause before rushing to conclusions. Exploration 6 ended with being the best possible training accuracy at 72.94%. For testing accuracy it achieved 72% on Gradescope.

4.8 Exploration 7 - All Available Information and Asking to Think Before Answering as well as to Focus on Being Accurate

Further tinkering along the lines of exploration 4 and exploration 6, we added a line to the zero shot prompt to "Focus on being accurate". This actually led to a slight dip in training accuracy performance, from 72.94% in exploration 6 to 71.24%. We hypothesize the reason for this is that as the prompt gets too large / there are too many leading directions, the phi 2 model begins to get distracted from the original claim. It seems that a careful balance must be struck in order to lead the phi 2 model in the correct direction, while not overwhelming it with information that dilutes the claim and relevant claim information. On Gradescope, the testing

accuracy was 72%.

4.9 Exploration 8 - Domain Name, Task Name, as well as to "Make your best educated guess."

To try to cut down on the prompt (while only including the most relevant info), the leading statements and language generation information was removed from exploration 7. Instead one statement of "Make your best educated guess." was included in the zero shot prompt (along with domain info and task info). The hope with this exploration is that the phi 2 model would interpret this additional statement as a way to make correct logical jumps rather than giving up if it didn't know and give a random answer. This exploration was not as successful since the training accuracy was only 70.39% as opposed to the best (exploration 6 - 72.94%).

4.10 Future Direction

Some ideas that we were thinking of doing but couldn't implement in time was using data from a certain domain to improve the performance on the others. For example, since GPT Toxicity and the hate speech dataset have many similarities, using some representation examples from GPT Toxicity could help with improving the performance on the hate speech dataset.

Our explorations have primarily focused on improving prompting by modifying information feeding into the prompt. We could take advantage of the "large" learning capability of language models by feeding in more training data into the model. One strategy would be to curate data from public sources, such as new forums or media outlets, that would increase the model's exposure to more diverse scenarios. The training and testing accuracy is expected to go up when training on larger data. Another direction would be to enhance the model's fairness by refining fairness evaluation metrics, like incorporating intrinsic and extrinsic bias evaluation metrics, and adding debiasing modules into the model. These steps are designed to provide a more detailed assessment of fairness and actively mitigate biases, thereby enhancing the model's factualness and fairness.

References

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *CoRR*, abs/2201.11903.

A Appendix A

A.1 Sample prompts that were used

To run batch_prompts to create prompts and get predictions from Phi-2 model:
python3 -m src.phi.batch_prompt –model_id_or_path "microsoft/phi-2" –annotations_filepath "/home/abbas/NLP Project/data/train_claims.jsonl" –output_filepath "/home/abbas/NLP-Project/data/predtest1.txt" -prompt_type "zero_eval"

To create predictions file:
python3 -m src.utils.create_predictions_file –pred_filepath "/home/abbas/NLP-Project/data/predtest2.txt" –pred_txt_savepath "/home/abbas/NLP-Project/data/final_predtest2_predictions.txt"

Evidence Creation Command:
python3 -m src.phi.batch_prompt –model_id_or_path "microsoft/phi-2" –annotations_filepath "/home/abbas/NLP-Project/data/train_claims.jsonl" –output_filepath "/home/abbas/NLP-Project/data/zero_evidence4.jsonl" –prompt_type "zero_evidence"

Compare final predictions with train predictions to get accuracy and f1 score:
python3 -m src.utils.eval_utils –gt_filepath "/home/abbas/NLP-Project/data/train_claims.jsonl" –pred_filepath "/home/abbas/NLP-Project/data/train_eval_evidence_out1.txt"

For few shot evaluation:
python3 -m src.phi.batch_prompt –model_id_or_path "microsoft/phi-2" –annotations_filepath "/home/abbas/NLP-Project/data/train_claims.jsonl" –output_filepath "/home/abbas/NLP-Project/data/few_eval_pred1.jsonl" –prompt_type "few_eval"