Report for S24-94812 Final Project “AI Policy Chat”

Joey Xie jiuyuanx@andrew.cmu.edu
Valerie Yuan jiayuyua@andrew.cmu.edu
Jiaying Qian jiayingq@andrew.cmu.edu
Cleon Sun cleons@andrew.cmu.edu
Abhijit Verma abhijitv@andrew.cmu.edu
Yan Luo yanluo@andrew.cmu.edu

Introduction:

- Project motivation and objectives

This "AI Policy Chat" project aims to address the increasing complexity and significance of AI policy-related discussions. As AI technologies continue to advance, policymakers, researchers, and the general public are confronted with intricate questions and challenges related to AI's ethical, legal, and societal implications. The project aims to leverage Large Language Models (LLMs) to create an intelligent conversational agent that can provide nuanced and contextually relevant responses to inquiries about AI policy.

This project has several objectives: (1) Develop a fine-tuned Large Language Model capable of understanding and answering questions related to AI policy. This involves training and fine-tuning the base Llama-2 7b model to process and generate human-like text specific to AI policy topics. (2) Explore prompt engineering techniques to guide the model toward accurate and informative responses. (3) Benchmark the fine-tuned Llama-2 7b model against the vanilla Llama-2 7b model and Llama-2 7b to compare the performance of different models.

- Background on AI policy and LLMs

Artificial Intelligence policy is the set of guidelines, regulations, and principles that govern the development, deployment, and use of AI technologies. As AI continues to advance and permeate various aspects of society, policymakers grapple with the need to strike a balance between fostering innovation and ensuring ethical, responsible, and safe AI practices.Governments around the world are increasingly recognizing the importance of creating a regulatory framework to address the unique challenges posed by AI. These challenges include issues related to privacy, security, bias, accountability, and the impact of AI on the workforce.

One key aspect of AI policy is the consideration of legal and ethical implications associated with Large Language Models (LLMs). LLMs, such as GPT. Ethical concerns surrounding LLMs primarily revolve around issues of bias and fairness. Since LLMs

learn from vast amounts of data, they may inadvertently perpetuate existing biases present in the training data. Policymakers are challenged to develop guidelines that encourage the development of LLMs that are unbiased, transparent, and accountable. This involves addressing concerns related to the unintentional amplification of societal biases and ensuring that LLMs are used in ways that align with ethical standards.

Privacy is another significant aspect of AI policy concerning LLMs. These models, by their nature, process and generate large amounts of text, sometimes incorporating sensitive information. Policymakers need to establish regulations that safeguard individuals' privacy while allowing for the responsible use of LLMs in various applications.

Methodology:

Data collection and preparation procedures.

For our study on AI policy, we gathered and prepared data in a few key steps, aimed at creating a rich and diverse dataset for analysis and model training.

We started by collecting 20 AI policy documents from various trusted sources listed on our course's Canvas page:

- Government Sites: Documents from both international and U.S. government websites give us a wide view of policies.
- Think-Tanks and Research Groups: These provide in-depth analyses and perspectives on AI policy.
- International Organizations: From these, we get a global viewpoint on AI regulations.
- Public Forums: These offer insights into public opinions and discussions on AI policy.

With our documents in hand, we then:

- Made Questions: We came up with around 100 questions that cover a broad range of AI policy topics.
- Found Answers: For each question, we found an answer in our collected documents.
- Varied Types: We made sure to include different kinds of questions (like yes/no, detailed explanations, and comparisons) to make our dataset well-rounded.

Finally, we put all our questions and answers into a JSON file. This makes it easy to use the data later for training our AI models and doing experiments with different ways to ask and answer questions (prompt engineering).

Fine-tuning and model development approaches:

Experiment with different hyper-parameters:

1. Learning rate

   Selecting an optimal learning rate is critical when fine-tuning the llama model as it influences how effectively the model adapts to new data without forgetting its pre-trained knowledge. The right learning rate ensures swift convergence to good performance and prevents overfitting, maintaining the model's ability to generalize to unseen data. This balance is key to leveraging the full potential of LLaMA in specific tasks while retaining its extensive pre-trained capabilities.

   We tried different learning rates of 1e-4 and 2e-4. The result shows fine tuning llama with 1e-4 has relatively better performance as it provides improved or comparable performance in most metrics over the 2e-4 learning rate and maintains more of the model's general capabilities compared to the raw model.

	BLEU_Score	ROUGE-1	ROUGE-2	ROUGE-l
Raw_llama2	0.006988	0.287063	0.056603	0.235454
Fine_tuned_llama2_lr_1e_4_metrics	0.007629	0.205463	0.037827	0.183335
Fine_tuned_llama2_lr_2e_4_metrics	0.005543	0.229239	0.032119	0.204028

2. Lora Configuration
   a. lora_alpha: This parameter controls the scaling factor applied to the low-rank matrices in LoRA. It is used to adjust the magnitude of the updates to the original model parameters. A higher lora_alpha value means that the low-rank updates will have a larger impact on the model.
   b. lora_dropout: This parameter specifies the dropout rate applied to the low-rank matrices in LoRA. Dropout is a regularization technique used to prevent overfitting by randomly setting a fraction of the input units to 0 during training. A higher lora_dropout value means that more elements of the low-rank matrices will be dropped out, leading to stronger regularization.
   c. r: This parameter defines the rank of the low-rank matrices in LoRA. The rank determines the number of columns in the low-rank matrices, which in turn controls the amount of parameter reduction. A lower rank leads to more significant compression of the model parameters, while a higher rank allows for more expressive power but with less compression.
   d. bias: This is a boolean parameter that indicates whether to include bias terms in the low-rank adaptation. If True, bias terms will be added to the low-rank matrices, which can help the model capture additional information that is not accounted for by the weights alone.

      After experimenting, the best combination for Lora Configuration is shown below.

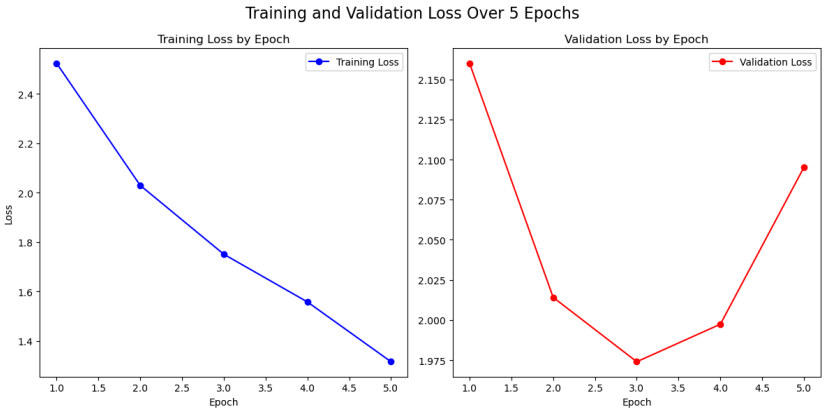
	BLEU_Score	ROUGE-1	ROUGE-2	ROUGE-l
Raw_llama2	0.012922	0.307788	0.077700	0.256869
Fine_tuned_llama2_Lora	0.007254	0.297749	0.066084	0.252001

lora_alpha	lora_dropout	r	bias
32	0.2	8	none

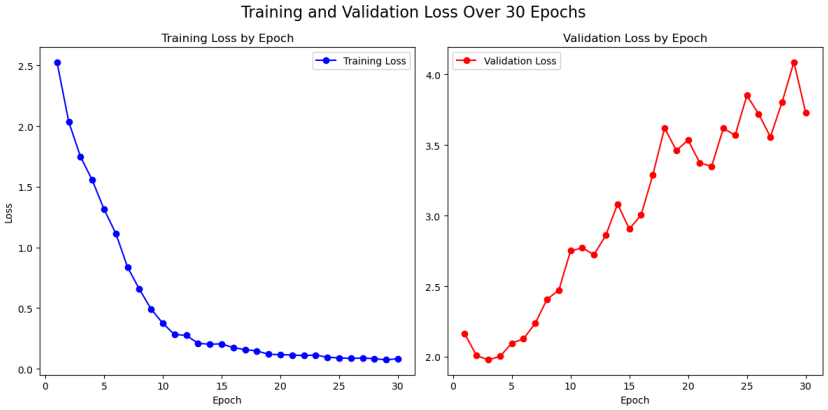
3. Batch Size(through gradient accumulation step)

   Larger batch size usually makes gradient approximation more accurate and stabilize training. Our baseline batch size is 4. Because T4 GPU RAM is limited to 14 GB, large batch like 8 cannot fit into the GPU.
   Gradient accumulation is a technique used to train models with large mini-batches that cannot fit entirely into the GPU memory at once. We can backward the gradient after two gradients of the two small batches are calculated and add them to get a large batch gradient, and update the model.

4. Epochs



In our experiments, we tested the model with varying epochs: 5, 10, 20, and 30. Surprisingly, we found that 3 epochs yielded the best results.



Beyond this point, the validation loss increased, indicating overfitting. This observation aligns with the general practice in language model training, where 1-4 epochs are often sufficient, contrasting with visual models that may require several hundred epochs. The tendency for linguistic data to overfit more quickly than visual data might be due to its dense and rich nature, suggesting a saturation point is reached sooner, possibly exacerbated by extensive pre-training.

	BLEU_Score	ROUGE-1	ROUGE-2	ROUGE-l
Raw_llama2	0.006171	0.298435	0.062240	0.256231
Fine_tuned_llama2_3_epoches	0.007181	0.331065	0.088585	0.261162

Upon fine-tuning the Llama2 language model for three epochs, we observe enhancements across all measured metrics. The BLEU score, indicating translation precision, increased marginally from 0.006171 to 0.007181. The ROUGE-1 score, measuring unigram overlap, rose from 0.298435 to 0.331065, suggesting improved content matching. Similarly, the ROUGE-2 score improved from 0.062240 to 0.088585, reflecting better bigram overlap. Notably, the ROUGE-L score, which assesses the longest common subsequence, also increased from 0.256231 to 0.261162, indicating an improvement in sequence matching. These metrics support the decision to adopt the three-epoch fine-tuned model for enhanced text generation.

Meta-llama:
Previously I used the base model from NousResearch. I also tried the same base model from Meta. Metrics are listed below:

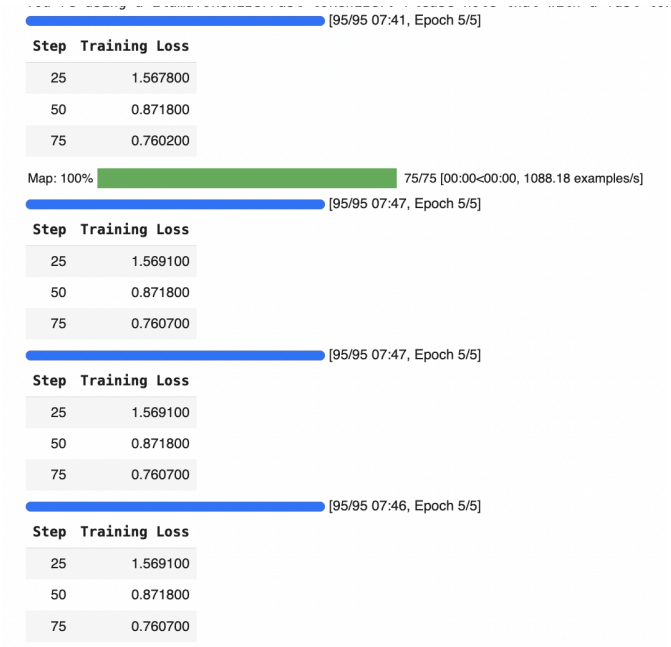
	BLEU_Score	ROUGE-1	ROUGE-2	ROUGE-l
Raw_llama2	0.002833	0.243284	0.046212	0.213330
Fine_tuned_llama2_3_epoches	0.011653	0.238402	0.047726	0.192272

The Meta base model generally outperforms the NousResearch base model across all metrics. It demonstrates a higher BLEU score, indicating better precision or alignment with reference translations. Additionally, its ROUGE-2 and ROUGE-L scores are also higher, suggesting it more effectively captures two-word phrases and the overall structure of the reference text. The ROUGE-1 score is slightly lower for the Meta model compared to the NousResearch model, but the difference is minor, indicating a comparable ability to match single words from the reference. The superior performance of the Meta model might be attributed to various factors such as differences in pre-training data, model architecture, or optimization techniques used during their development.

5. Weight Decay

In our code, we also endeavored to evaluate the effects of weight decay regularization on a LLaMA model, a parameter crucial for the mitigation of overfitting. We experimented with a spectrum of weight decay values, specifically [0.0001, 0.001, 0.01, 0.1], training the model across five epochs and documenting the training loss at intervals within the final epoch. Notably, the training loss exhibited minimal fluctuations across the different weight decay settings, suggesting a marginal influence of this parameter on the model's learning curve within the scope of our dataset and chosen hyperparameters. Such consistency in training loss, despite altering weight decay, may be attributed to several factors: the predominance of the learning rate in the training dynamics, the inherent complexity of the model and dataset which could diminish the propensity for overfitting, and the possibility of statistical noise intrinsic to the training procedure.

```
weight_decay_values = [0.0001, 0.001, 0.01, 0.1]
```



It is also important to consider that weight decay's primary role is to enhance generalization to unseen data, typically reflected in validation loss metrics rather than training loss. Thus, the true measure of weight decay's effectiveness may not be fully captured in our current observations. Moving forward, we aim to expand the range of weight decay values, coupled with an in-depth evaluation of the model's performance on validation sets. This will enable us to better understand the parameter's role in enhancing the model's generalization abilities, ensuring a comprehensive assessment of its performance and underlying robustness.

Prompting techniques:

1. Priming

```
from tqdm.notebook import tqdm
import gc

priming_text = """
The following responses should reflect a deep understanding of AI Policy, including ethical considerations,
regulatory frameworks, and the societal impact of AI technologies.
Answers should be informed, nuanced, and precise, demonstrating a comprehensive grasp of the subject matter.
"""

generations_primed =[]
for i in tqdm(range(len(df_test_all)), "generating..."):
  prompt = f"{priming_text}### Question: {df_test_all['input'][i]}\n Briefly, in 100 words answer the question. ### Answer: </s>"
  # Generate predictions
  inputs = llama_tokenizer(prompt, return_tensors='pt')
  inputs = inputs.to("cuda")
  output = raw_model.generate(**inputs, max_new_tokens=200,temperature=0.2)
  response = llama_tokenizer.decode(output[0].tolist())
  # print(response)
  # break
  generations_primed.append(response)
  del inputs, output
  gc.collect()
  torch.cuda.empty_cache()
```

   a. A priming text was crafted to instruct the model to produce responses with a deep understanding of AI policy, ethical considerations, regulatory frameworks, and societal impacts.
   b. This priming text was prefixed to each question in a loop that processed with the test dataset.
   c. The predicted answers (generations) were collected for evaluation.

	BLEU_Score	ROUGE-1	ROUGE-2	ROUGE-l
generations_without_priming	0.006171	0.298435	0.062240	0.256231
generations_with_global_priming	0.007318	0.364842	0.085737	0.293611

- The implementation of global priming has markedly enhanced model output, as evidenced by increased scores across the board. The rise in BLEU score implies finer alignment with reference texts, while the significant jump in ROUGE-1 indicates better word-level accuracy. Improvements in ROUGE-2 and ROUGE-L scores suggest enhanced coherence in both short and long text sequences. These gains collectively highlight the effectiveness of global priming in refining the model's contextual awareness and overall text generation quality.

2. Control Codes Prompting

   Control codes are special instructions or tokens we prepend or append to our input prompt to guide the model's generation towards a desired format, tone, or content type. For the Llama-2 7b model, this involves specifying the type of response needed (e.g., factual, analytical, policy recommendation) or indicating the focus area (e.g., AI ethics, regulation frameworks, implementation strategies).

   To evaluate the effectiveness of control codes in guiding the model:

   - Developed a set of control codes that represent different aspects of AI policy queries, such as [International Cooperation], [Research Approach], [AI Ethic Guideline], etc.
   - Crafted input prompts that incorporate these control codes and query the model with them.

	input	prompt_engineering
0	Why is international cooperation on AI important?	[International Cooperation] Why is internation...
1	Can you describe the approach taken by the res...	[Research Approach] Can you describe the appro...
2	What concerns do critics have regarding the EU...	[EU AI Act Criticism] What concerns do critics...
3	How does the European Union classify AI system...	[EU AI Classification] How does the European U...
4	What are the Universal Guidelines for Artifici...	[AI Ethics Guidelines] What are the Universal ...
5	What role do whistleblowers and complaints pla...	[Regulatory Process] What role do whistleblowe...

   - Analyzed the responses for relevance, accuracy, and insightfulness regarding the AI policy topics. This involved comparing the model's outputs with and without control codes to see if there's a marked improvement in the quality and relevance of responses.
   - Iterate based on feedback. Refine your control codes and prompt strategies based on the initial rounds of feedback to optimize the model's performance.

	BLEU_Score	ROUGE-1	ROUGE-2	ROUGE-l
Raw_llama2	0.007532	0.227682	0.043049	0.197577
Fine_tuned_llama2_control_codes_prompting	0.014546	0.258092	0.059738	0.219957
prompt_engineering_control_codes_metrics	0.005434	0.287454	0.051924	0.245689

   The evaluation of the Llama-2 language model across three configurations—raw, fine-tuned with examples, and adjusted with control codes—revealed distinct impacts on

performance. The raw Llama-2 served as a benchmark, with modest BLEU and ROUGE scores. Fine-tuning with examples enhanced all metrics, indicating increased precision and better overlap with reference texts. Adjusting with prompt engineering control codes slightly reduced the BLEU score but improved ROUGE scores, particularly ROUGE-L, suggesting an improved generation of contextually relevant content. These outcomes underscore the trade-offs between precision and content relevance in model performance, informing the choice of configuration based on application needs.

3. Task instructions

This section presents the findings from a series of experiments aimed at understanding the effects of priming and task instruction on the performance of language generation models. The evaluation is based on metrics such as BLEU score and ROUGE scores (ROUGE-1, ROUGE-2, and ROUGE-L).
The results of the experiments are summarized in the table below:

Experiment	BLEU Score	ROUGE-1	ROUGE-2	ROUGE-L
Baseline	0.006728	0.213823	0.035865	0.176707
Task Instruction	0.005774	0.292351	0.051474	0.255964

Task Instruction: Providing specific task instructions showed a balanced improvement across all metrics, suggesting that task instruction might be a more effective approach than individual priming in certain contexts.

The experiments demonstrated the nuanced impact of task instruction on language generation. Task instruction emerged as a promising approach, offering a balanced improvement across key performance metrics. These findings highlight the importance of carefully selecting and applying prompt engineering techniques to optimize language generation models.

4. Examples

I adopted a fixed example method to prime LLaMA2 for generating answers. This approach mitigates the risk of inadvertently guiding the model towards specific answers, which could skew its natural language generation capabilities. The fixed example provides a general template of a well-structured question-answer pair that reflects the quality and format we expect the model to emulate. By using this consistent example across all prompts, the model learns to generate responses that are stylistically similar to the example without being influenced by the content of the actual answers.

Example QA pair used:
Question: What are the ethical considerations in AI development?
Answers: Ethical considerations in AI development include fairness, transparency, accountability, privacy, and ensuring AI systems do not perpetuate bias or discrimination.

	BLEU_Score	ROUGE-1	ROUGE-2	ROUGE-l
Raw_llama2	0.006988	0.287063	0.056603	0.235454
llama2_examples_prompting	0.002679	0.295252	0.057654	0.233252

The example prompting method has slightly improved the BLEU score, while the ROUGE scores show a mixed result with marginal improvements in ROUGE-1 and ROUGE-2, but a slight decrease in ROUGE-L. This suggests that example prompting may improve the fluency or relevance of the generated text to some extent, but not consistently across all evaluation metrics.

5. Templates

To enhance the effectiveness of the prompts and elicit more detailed and useful answers with a specific template about AI policy, employing "templating" techniques can be very beneficial. This technique involves structuring the prompt of template to guide the AI in generating a more focused, comprehensive, and structured response.
For example:
The original prompt is: "How does the European Union classify AI systems under its AI Act, and what are the implications for "high risk" AI systems?"

We can engineer the prompt as follows:
Question: What concerns do critics have regarding the EU's AI Act's potential impact on innovation and competitiveness?
Briefly, in 100 words use this template to answer the questions
Discuss the main concerns raised by critics regarding the European Union's AI Act, particularly focusing on its potential impact on innovation and competitiveness within

the EU. Analyze how the regulation might hinder or foster technological advancement and the competitiveness of European AI industries on a global scale.

	BLEU_Score	ROUGE-1	ROUGE-2	ROUGE-l
Raw_llama2	0.006728	0.213823	0.035865	0.176707
templating_prompt_engineering_metrics	0.012847	0.372629	0.091972	0.314735

We can see from the table that we have a higher BLEU, ROUGE scores than raw prompting.

Evaluation metrics:

Automatic Metrics

- BLEU
- ROUGE

Human Evaluation

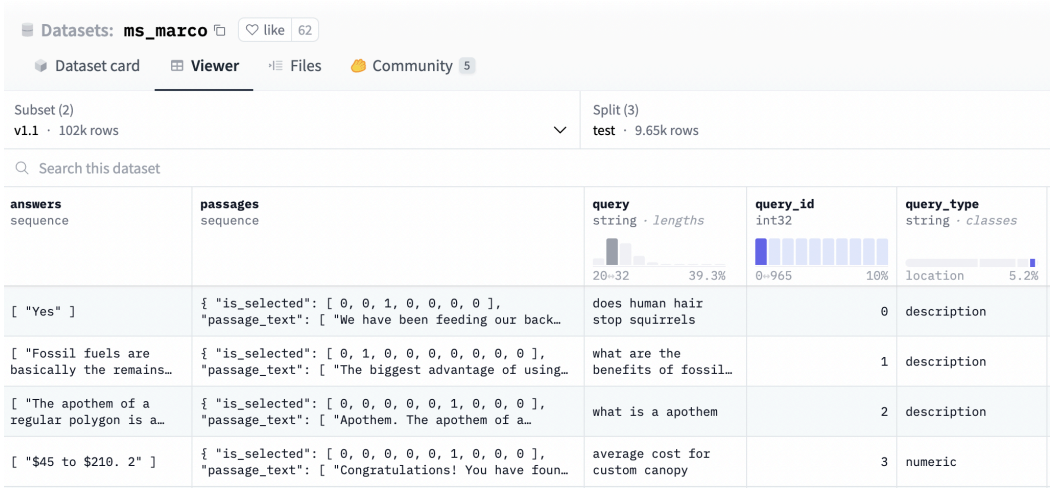
- Relevance: Does the response address the prompt?
- Coherence: Is the response logically consistent and easy to follow?

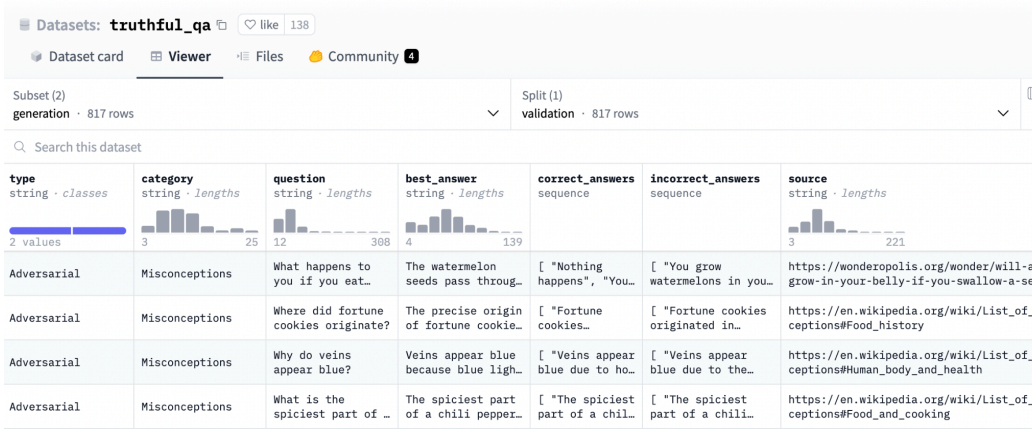
Results:

- Model performance across different methods.
- Comparative analysis of results.
- Key findings and insights.

- Benchmark against vanilla Llama2 7b and Llama2 7b Chat

Two benchmark datasets are used to compare the fine tuned model against Llama2 7b and Llama2 7b Chat, namely, **ms_marco**(https://huggingface.co/datasets/ms_marco/viewer/v1.1/test) and **truthful_qa**(https://huggingface.co/datasets/truthful_qa/viewer), both acquired from hugging face.





As these the volume of these two datasets are significantly larger than our policy dataset. Therefore, in the generation section, only a small portion of the data is used to generate answers and calculate automatic metrics.

For ms_marco:

	BLEU_Score	ROUGE-1	ROUGE-2	ROUGE-l
llama	0.002696	0.197467	0.037944	0.194967
llama_chat	0.000000	0.148073	0.014250	0.133303
finetuned_llama2_chat	0.000000	0.145014	0.006926	0.133466

For truthful_qa:

	BLEU_Score	ROUGE-1	ROUGE-2	ROUGE-l
llama	0.000000	0.054168	0.000000	0.049168
llama_chat	0.019834	0.342512	0.149015	0.306943
finetuned_llama2_chat	0.019288	0.352340	0.174474	0.333537

The "truthful_qa" dataset appears to be focused on questions with factual answers, covering a range of topics such as human hair, fossil fuels, and geometric concepts like the apothem. On the other hand, the "ms_marco" dataset seems to involve a mix of questions, potentially spanning a broader set of topics and requiring more diverse knowledge.

For the MS_macro dataset, vanilla Llama performs better based on the given metrics, while the other two models (llama_chat and finetuned_llama2_chat) have lower scores, especially in terms of BLEU and ROUGE-2, indicating potential room for improvement in generating text that aligns more closely with the reference.

For the truthful_qa dataset, vanilla Llama has low scores across all metrics, suggesting a limited match with the reference in terms of unigrams, bigrams, and longest common subsequences. llama_chat and finetuned_llama2_chat perform better in terms of ROUGE metrics, indicating a higher level of overlap with the reference compared to the llama model.

With regard to human evaluation metrics such as relevance/coherence and informativeness, it is found that models have different performances for the two datasets. For the ms_Marco dataset, Llama2 generates better responses with regards to correctness and relevance. For the truthful_qa dataset, however, fintuned Llama 2 7b

chat provided the most relevant and coherent responses(though not significantly better than the vanilla chat).

Discussion:

- Strengths and weaknesses of different approaches.
    - Significant strengths in leveraging LLMs for AI policy discussions, particularly through fine-tuning of the Llama-2 7b model and innovative prompt engineering techniques.
    - Enabled the creation of a conversational agent capable of providing nuanced and contextually relevant responses.
    - Tailored adjustment of hyperparameters and incorporation of LoRA configurations significantly enhanced model performance.
    - Weaknesses in generalizing the model's performance across various datasets and balancing precision with contextual relevance.
    - Limitations of traditional evaluation metrics like BLEU and ROUGE, which cannot fully capture the qualitative aspects of the generated text.
- Challenges encountered and solutions.
    - Challenges in optimizing the model to prevent overfitting while ensuring generalization across different AI policy contexts.
    - Solutions involved rigorous experimentation with learning rates, LoRA configurations, and various prompting techniques.
    - The strategic approach to fine-tuning and prompt engineering led to improved model performance.
    - Showcased the potential of LLMs in facilitating complex AI policy discussions.
- Implications for AI policy development and research
    - Different models may excel on different datasets, indicating the importance of considering the characteristics of the training data when evaluating model performance.
    - The choice of evaluation metrics plays a crucial role in assessing model performance. Metrics like BLEU and ROUGE may not capture all aspects of model-generated text quality.
    - AI policy development should incorporate ethical considerations, particularly when models are applied in contexts that require accurate and truthful information, as in the case of the truthful_qa dataset.

Conclusion:

- Summary of key takeaways
    - The performance of language models varies significantly across different datasets and domains.
    - Balancing BLEU and ROUGE metrics in model performance is challenging, akin to precision and recall, indicating the complexity of optimizing for both accuracy and relevance in generated content.
    - Best hyperparameter combination
        - Lora_rank=16,
        - Gradient Accumulation=2,
        - Learning_rate=1e-5,
        - Weight_decay=2e-3,
        - Max_grad_norm=0.1,
        - Epoch=10
    - Best prompting techniques?
        - Templates

- Recommendations for future work
    (1) Explore domain-specific fine-tuning strategies to enhance model performance across diverse datasets. Fine-tuning techniques tailored to the characteristics of specific domains or tasks could lead to improved adaptability and better alignment with user expectations.

    (2) Additionally, continued research into the development of more nuanced evaluation metrics could provide a better understanding of a model's qualitative performance, moving beyond the current limitations of metrics like BLEU and ROUGE.

    (3) Further investigation into the ethical implications of language model applications in AI policy is also essential. Ensuring that these models adhere to ethical standards and contribute positively to policy development is paramount.

    (4) Lastly, we advocate for a more iterative and dynamic process of model training and evaluation, incorporating feedback loops that allow for continuous refinement of model performance. This approach would better equip language models to handle the evolving nature of AI policy discussions and the complex questions they raise.

Appendix:

1. https://chat.openai.com/share/cb51d7df-6062-42ba-bdd1-28131d5c4688

2. Our process of human evaluation across different fine-tuning experiments and prompt engineering techniques: https://docs.google.com/spreadsheets/d/17-JDgSJSxjbH-zk6wxtRN3Y2OmibVuHddKCz7MqQaMQ/edit#gid=700621598