Applications of NLX and Large Language Models Final Report

Muhammad Asghar (masghar@andrew.cmu.edu)
Davis Craig (dmcraig@andrew.cmu.edu)
Hiba Hassan (hibah@andrew.cmu.edu)
Grace Eunji Kim (eunjik@andrew.cmu.edu)
Abdul Rehman (marehman@cmu.edu)
Talia Qaiser (tqaiser@andrew.cmu.edu)
Alex Yang (yihsuehy@andrew.cmu.edu)

Table of Content

Introduction

- Project motivation and objectives.
- Background on AI policy and LLMs.

Methodology

- Data collection and preparation procedures.
- Prompting techniques.
- Fine-tuning and model development approaches.
- Evaluation metrics.

Results

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

Discussion

- Strengths and weaknesses of different approaches.
- Challenges encountered and solutions.
- Implications for AI policy development and research.

Conclusion

- Summary of key takeaways

Acknowledgements

Packages Used

```Unset
argparse
bitsandbytes
datasets
functools
peft
torch
transformers
scipy
```

Introduction

The objectives of this project were to learn and deploy a large language model that was capable of understanding and answering questions related to AI policy. We developed a chat environment that was fine-tuned and refined using prompt engineering techniques.

Documents addressing AI policy were aggregated and question and answer (Q&A) pairs were constructed and formatted in JSON. This file was then used to train and fine-tune a Llama2-7b large language model.

The topics regarding AI policy were collected from US and international governmental websites, policy think tanks and research organizations, international organizations, and public policy forums and repositories.

Methodology

Data Collection and Preparation

Take for instance this policy paper addressing opportunities and blind spots in the White House's blueprint for an AI Bill of Rights. Our formatted JSON looks like this.

```python
[
  {
    "input": "How has the White House Office of Science and Technology Policy (OSTP) attempted to address responsible AI policy, and what challenges remain in enforcing the Blueprint for an AI Bill of Rights?",
    "output": "The OSTP published a Blueprint for an AI Bill of Rights outlining core principles for responsible AI use. However, challenges persist in determining how grievances will be reprimanded and if the non-binding document will prompt congressional action."
  },
  {
    "input": "Which agencies have begun to adopt the blueprint/guidelines?",
    "output": "The Department of Defense (DOD): Implemented Ethical Principles for Artificial Intelligence. The U.S. Agency for International Development (USAID): Developed an Artificial Intelligence Action Plan. The Equal Employment Opportunity Commission (EEOC): Launched an AI and algorithmic fairness initiative in partnership with the Department of Labor. The Department of Energy (DOE): Established its own office to implement guidelines. The Department of Veterans Affairs (VA): Established its own office to implement guidelines. The Department of Health and Human Services (HHS): Established its own office to implement guidelines."
  },
  {
    "input": "Why is it important for law enforcement to be fully part of the national blueprint guidance?",
    "output": "It's crucial to include law enforcement in the national blueprint guidance to prevent discrimination, ensure accountability, and protect civil liberties. Excluding law enforcement could perpetuate biases and hinder oversight, while incorporating them ensures that AI technologies are deployed ethically and transparently, fostering public trust and confidence."
  },
  {
    "input": "What would congressional action on the subject look like?",
```

```
    "output": "Congressional action on the subject would involve passing
legislation to codify the principles outlined in the Blueprint for an AI Bill
of Rights and expanding coverage to include law enforcement and national
security. This legislation would establish enforceable guidelines for data
privacy, auditing automated decisions, and ensuring accountability, addressing
the challenges in enforcing criteria driven by proprietary interests and
providing a credible enforcement regime."
  },
  {
    "input": "What specific sectors or domains regarding civil rights are
emphasized in the blueprint?",
    "output": "The civil rights of interest highlighted by the blueprint
primarily revolve around lending, housing, and hiring."
  }
]
```

Prompt Engineering

We used an inference-based model that was trained on 5 prompts.

1. Zero-Shot Prompting
2. One-Shot Prompting
3. Chain-of-Thought Prompting
4. General Knowledge Prompting
5. Prompt Chaining

Model 1: Zero-Shot Prompting

This type of prompting gives a task description to the model without any example and asks it a question.

Task Description: Give a holistic AI policy for a problem when prompted. It should have some or all of the following components if relevant to the problem:

1. Ethical Principles
2. Transparency and Explainability
3. Fairness and Bias Mitigation
4. Accountability and Responsibility
5. Privacy Protection

6. Security Measures
7. User Consent and Control
8. Data Governance
9. Collaboration and Stakeholder Engagement
10. Compliance with Regulations
11. Robust Testing and Validation
12. Human-in-the-Loop
13. Continuous Improvement
14. Emergency Protocols

Prompt: What are the key steps in the development of an AI policy framework?

Model 2: One-Shot Prompting

The prompt has a few examples that the model uses to generate the answer.
Here we made prompts for different components of a holistic AI policy.

Prompts

1. *Ethics, Transparency, Fairness and Accountability*

Task Description: Give me a comprehensive AI policy that prioritizes ethical considerations and meets stringent transparency, fairness, and accountability standards.

Example: "The AI development policy for governmental organizations prioritizes fairness, transparency, and accountability. All AI initiatives must undergo rigorous ethical assessments, ensuring alignment with human rights and privacy standards. Transparency measures include disclosing data sources and algorithmic processes. Accountability is ensured through regular audits and oversight mechanisms to address any ethical or fairness concerns."

Prompt: Give me an AI development policy for a governmental organization.

2. *Privacy, Security, and Consent*

Task Description: Craft a comprehensive AI policy that prioritizes safeguarding privacy, implementing robust security measures, and ensuring informed user consent in all AI-driven applications.

Example: "AI systems used in financial services must prioritize user privacy by securely handling sensitive financial data, implementing encryption protocols, obtaining explicit user consent for data processing, and regularly auditing for compliance with privacy regulations such as GDPR or CCPA."

Prompt: What kind of consumer data protection policies should an online retail organization consider?

3. *Data Governance, Compliance, and Model Testing*

Task Description: Develop a comprehensive AI policy that prioritizes robust data governance, regulatory compliance, and rigorous testing of AI models to establish a coherent and ethical AI framework.

Example: "Our AI policy mandates strict adherence to data governance principles, including transparent data collection practices, secure storage protocols, and responsible data usage. We comply with all relevant regulations such as GDPR and HIPAA, ensuring that our AI models are ethically developed and deployed. Additionally, our policy requires comprehensive testing of AI algorithms to assess their accuracy, fairness, and safety before implementation in real-world applications."

Prompt: What kind of consumer data protection policies should an online retail organization consider?

<u>Model 3: Chain-of-Thought Prompting</u>

The prompt has a chain of thoughts that captures the reasoning that the model uses to generate the answer. Chain-of-thought prompting can be combined with zero-shot, one-shot, or few-shot prompting

Question: How does a healthcare organization address privacy concerns and ensure compliance with healthcare regulations in the development and deployment of AI systems, especially in handling sensitive patient data?

Answer: A healthcare organization can prioritize patient privacy and adhere to stringent data protection regulations throughout the development and deployment of AI systems. It can implement robust security measures to safeguard sensitive patient data. The AI policies explicitly should outline the procedures for data collection, storage, and processing, ensuring compliance with healthcare regulations. Regular audits and assessments should be conducted to monitor and enhance the security and privacy posture of AI systems, demonstrating the commitment to maintaining the highest standards in healthcare data management.

Prompt: How does a financial institution ensure data security and regulatory compliance when implementing AI technologies, especially in handling sensitive financial information and transactions?

Model 4: General Knowledge Prompting

In this type we first generate ‘knowledge’ and then ask it a question based on that ' knowledge’.

Input: "Imagine a user has asked about the importance of user consent in collecting personal data. Explain why ethical considerations require explicit consent and how this impacts user trust."

Knowledge: "Ethical considerations demand that all personal data collection processes involve explicit consent from the user. This means before collecting any personal data, users should be fully informed about what data is being collected and for what purpose. Obtaining explicit consent is crucial because it respects the user’s autonomy and privacy rights. Moreover, it builds trust between the user and the platform. When users know that their data is only collected with their informed consent, they are more likely to trust and engage with the platform. This trust is foundational to ethical digital practices and enhances user experience by ensuring their rights are protected."

Input: "Discuss the significance of making the algorithms and data usage policies clear to users. How does transparency in these areas support accountability and user confidence?"

Knowledge: "Transparency regarding algorithms and data usage policies is critical to gaining and maintaining user trust. When platforms clearly explain how they use data and how their algorithms work, users can make informed decisions about their engagement with these platforms. Transparency supports accountability because it allows users to understand and critique the processes involved in data handling and decision-making. This openness builds user confidence and encourages platforms to adhere to the highest standards of responsibility and fairness, knowing they are subject to the public's scrutiny."

Input: "Explain the role of governance in AI development and how it ensures that AI technologies are used responsibly. Consider laws, regulations, and ethical guidelines in your response."

Knowledge: "Governance in AI development plays a pivotal role in ensuring that AI technologies are used responsibly. Good governance involves setting up laws, regulations, and ethical guidelines that dictate how AI should be developed, deployed, and used. This framework ensures that AI technologies benefit society while minimizing risks and preventing misuse. For example, regulations might require AI systems to undergo ethical review and compliance checks before they are brought to market, ensuring they do not perpetuate bias or harm. By following these structured guidelines, AI developers and users can ensure that technological advancements are aligned with societal values and ethical principles, fostering innovation that is responsible and beneficial for all."

Question: "Explain the importance of compliance and regulation in AI development and how it ensures that AI technologies are used responsibly. Consider laws, regulations, and ethical guidelines in your response."

Model 5: Prompt Chaining

Prompt chaining allows AI systems to handle tasks that are too intricate for a single prompt. By dividing these tasks into smaller, more manageable steps, LLMs can navigate through each segment, culminating in a comprehensive solution.

Prompt 1: You are an AI policy-making chatbot and you will give a holistic AI policy for a problem when prompted. You should give a clear policy for the specific industry when prompted. Your language should be assertive and direct. Please output the policy using <quotes></quotes>. If the user does not add the industry, ask the user to add an industry and context.

Prompt 2: Given the policy you have crafted (delimited by <quotes></quotes>) for the relevant industry, see if most of these points have been included in it or not. If not, then add to the policy.

1. Ethical Principles
2. Transparency and Explainability
3. Fairness and Bias Mitigation
4. Accountability and Responsibility
5. Privacy Protection
6. Security Measures
7. User Consent and Control
8. Data Governance
9. Collaboration and Stakeholder Engagement
10. Compliance with Regulations
11. Robust Testing and Validation
12. Human-in-the-Loop
13. Continuous Improvement
14. Emergency Protocols

Fine-Tuning the LLM

Of the many fine-tuning methods, we opted to use a Parameter-Efficient Fine-Tuning (PEFT) model which is designed to take in billions of parameters and fine-tune on low-resource hardware. The model used was Facebook's Llama2-7b from Hugging Face. Initially, we used a single NVIDIA V100 16GB GPU at the Pittsburgh Supercomputing

Center to train our model. However as we ran into technical issues connecting to and setting up a cluster, we had to work from Google Colab.

We built on a notebook posted on [OVHcloud's blog] to define several functions: loading the model, creating prompt formats, preprocessing the batch and datasets, creating the bits and bytes and PEFT configuration, and lastly printing the trainable parameters. The model was created with the following parameters:

```python
limit memory to 4GB
quantization_config = bnb_config
device_map = "auto"
```

Our prompt format: To process our 131 question answer pairs we created a function that formatted the inputs & outputs into a prompt. This prompt provided a brief description, a question prefix followed by the input, an answer prefix followed by the answer, and an end token. This prompt format helped our model learn the task of question answering, and could be used down the line to teach the model additional instructions.

```python
INTRO_BLURB = "Below is a question followed by an answer. Write a response that appropriately completes the request."

QUESTION_KEY = "Question:"
ANSWER_KEY = "Answer:"
END_KEY = "### End"

blurb = f"{INTRO_BLURB}"
question = f"{QUESTION_KEY}\n{sample['input']}"
answer = f"{ANSWER_KEY}\n{sample['output']}"
end = f"{END_KEY}"

parts = [part for part in [blurb, question, answer, end] if part]

formatted_prompt = "\n\n".join(parts)

sample["text"] = formatted_prompt

return sample
```

Our bits and bytes configuration: We went with 4 bit quantization to maximize the benefits of memory efficient tuning.

```python
load_in_4bit = True,
 bnb_4bit_use_double_quant = True,
 bnb_4bit_quant_type = "nf4",
 bnb_4bit_compute_dtype = torch.bfloat16
```

Our PEFT configuration using Low-ranked Adaptation (LoRA):

```python
config = LoraConfig(
    r=32, # dimension of the updated matrices
    lora_alpha=64, # parameter for scaling
    target_modules=modules,
    lora_dropout=0.1, # dropout probability for layers
    bias="none",
    task_type="CAUSAL_LM",
)
```

When we began our training we started with 20 steps and an r hyperparameter value of 8. For our final training run we increased the training to 5 epochs and the rank to 32. As an output, our AI policy adapter had 79 million fine tuned weights, which amounts to 2.23% of the original model. The config and weights were saved as *adapter_config.json* and *adapter_model.safetensors* which were then uploaded to Hugging Face. For inference we simply downloaded this adapter from hugging face and merged it with the base model.

Results

The performance of the two models were evaluated in three ways (base, base+PE, fine-tuned+PE). We utilized the references acquired by plugging the questions (prompts) into ChatGPT 4.0.

Automatic Metrics

BLEU Scores:

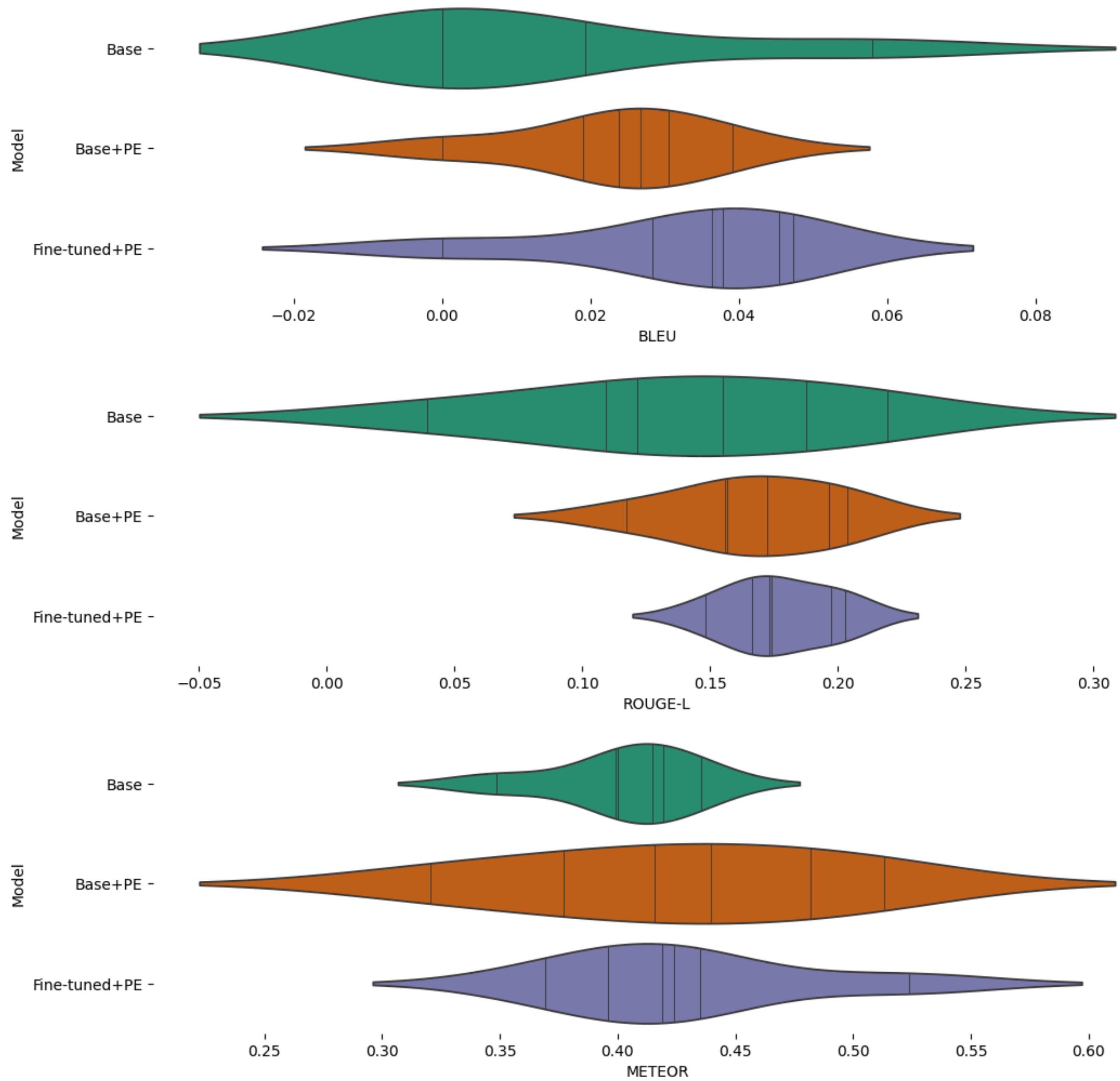
The BLEU scores across models follow a similar trend as seen with the other metrics, with the Fine-tuned+PE model outperforming the others. However, the scores are generally low, which could indicate that exact n-gram matches with the references are rare. On the other hand, the BLEU scores by prompt are consistently low for all models, which to our understanding may be that either the model is still underfitting or the questions are challenging for the models to generate close matches to them, or that the BLEU metric is not capturing the quality of the generated text well for this particular task.

ROUGE-L Scores:

The distribution of ROUGE-L scores also places the Fine-tuned+PE model at the top, albeit with a less pronounced difference from the Base and Base+PE models. This suggests that the fine-tuned model might be capturing longer sequences of text that match the references better than the other models. As with the METEOR metric, the ROUGE-L scores for each prompt show variability. Prompts q4 and q5 appear to yield higher ROUGE-L scores across models, indicating that these prompts may be more conducive to generating longer matching sequences of text.

METEOR Scores:

The violin plot for METEOR scores across models shows that the Fine-tuned+PE model generally achieves the highest scores, followed by the Base+PE, and then the Base model. This indicates that fine-tuning with prompt engineering tends to produce text with higher semantic and syntactic quality. The METEOR scores by prompt indicate that the performance varies significantly depending on the specific prompt. Some prompts (q3 and q4) have a much wider distribution of scores, suggesting that the models might be particularly sensitive to the content or structure of these prompts.



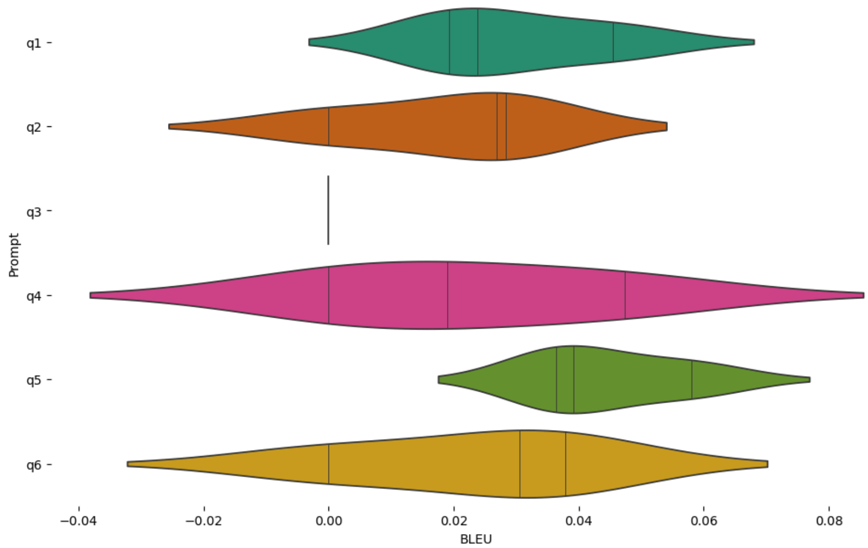
Findings:

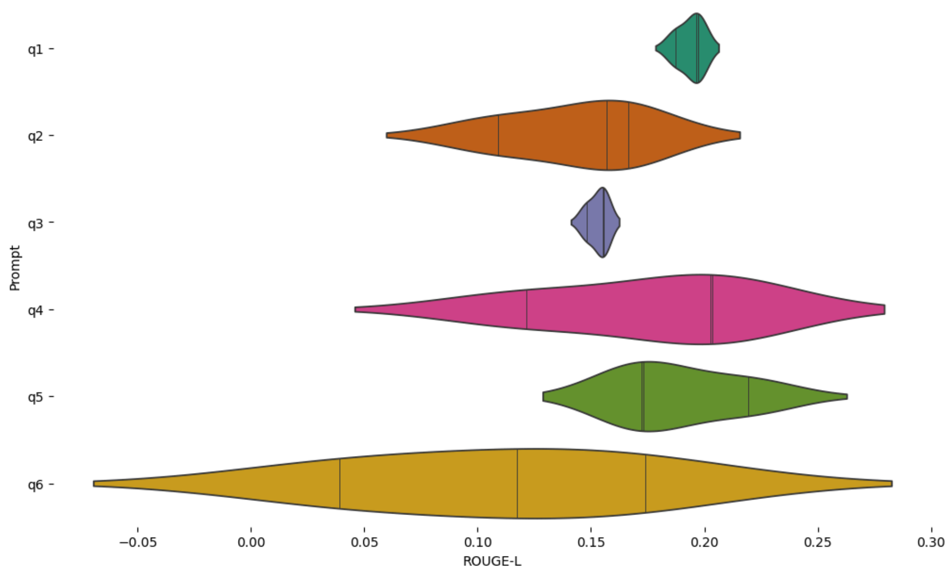
- The Fine-tuned+PE model demonstrates the best overall performance across all three metrics, suggesting that fine-tuning combined with prompt engineering is effective for this text generation task.
- The Base model shows the lowest performance without surprise since it is presumably the least specialized or optimized.
- The Base+PE model shows some improvement over the Base model, indicating that prompt engineering alone does have a positive impact, though not as much as when combined with fine-tuning.

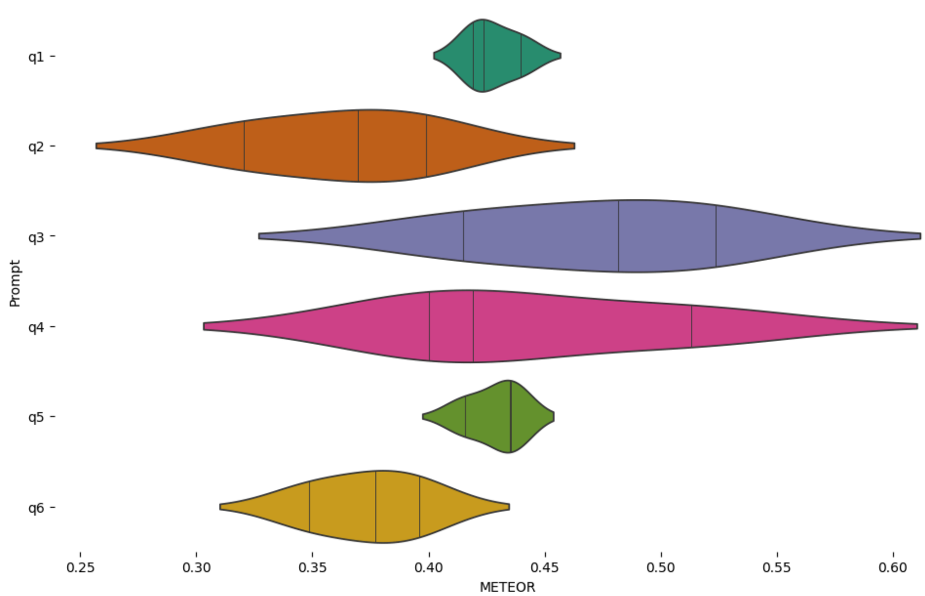
- Performance varies significantly by prompt, which implies that the choice of prompt and how it is structured has a considerable impact on the models' ability to generate high-quality text.
- The way our model responded may tend to replicate itself and initialize a new conversation itself which is irrelevant to our original prompt. This may be the major issue for the model that is hindering the potential to achieve in higher metrics scores.

Overall, while quantitative automatic metrics provide valuable insight into model performance, we also did some qualitative analysis (e.g. human evaluation) to fully understand the capabilities and limitations of the models, especially when the metrics show a wide range of scores.

	Prompt	Model	BLEU	ROUGE-L	METEOR
0	q1	Base	0.0193	0.1874	0.4193
1	q1	Base+PE	0.0238	0.1966	0.4396
2	q1	Fine-tuned+PE	0.0455	0.1974	0.4239
3	q2	Base	0.0000	0.1090	0.3990
4	q2	Base+PE	0.0268	0.1568	0.3204
5	q2	Fine-tuned+PE	0.0284	0.1663	0.3694
6	q3	Base	0.0000	0.1551	0.4149
7	q3	Base+PE	0.0000	0.1556	0.4819
8	q3	Fine-tuned+PE	0.0000	0.1480	0.5237
9	q4	Base	0.0000	0.1215	0.4002
10	q4	Base+PE	0.0190	0.2036	0.5132
11	q4	Fine-tuned+PE	0.0473	0.2028	0.4190
12	q5	Base	0.0580	0.2194	0.4355
13	q5	Base+PE	0.0391	0.1721	0.4156
14	q5	Fine-tuned+PE	0.0364	0.1731	0.4350
15	q6	Base	0.0000	0.0391	0.3485
16	q6	Base+PE	0.0305	0.1172	0.3771
17	q6	Fine-tuned+PE	0.0379	0.1739	0.3961


q1
q2
q3
q4
q5
q6
Prompt
−0.04
−0.02
0.00
0.02
0.04
0.06
0.08
BLEU


q1
q2
q3
q4
q5
q6
Prompt
−0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
ROUGE-L


q1
q2
q3
q4
q5
q6
Prompt
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
METEOR

Human Evaluation

We gathered a total of 38 evaluation feedbacks. The bar graph below compares the performance of three different configurations of the llama-7b model across five evaluation criteria, which is on a scale between 1 to 5, where a higher value indicates better performance.

<u>Relevance</u>

This measures how well the model's responses are related to the queries or tasks at hand. The Base Model scores around 3.0, suggesting that its responses are moderately relevant. The Base + PE approach shows improvement, scoring above 3.5, indicating that prompt engineering helps the model to produce more relevant responses. The Fine-tuned + PE model shows the best performance in relevance, almost reaching a score of 4.0, which suggests that fine-tuning specifically for the task or dataset in question, combined with prompt engineering, provides the most relevant responses.

<u>Coherence</u>

Coherence evaluates how logically connected and consistent the responses are. The Base Model again shows moderate performance, slightly above 3.0. The Base + PE model sees a notable increase, approaching a 4.0 score, suggesting that prompt engineering significantly improves the model's ability to produce coherent text. The Fine-tuned + PE model matches the Base + PE model's performance, indicating that fine-tuning does not significantly impact coherence when prompt engineering is already applied.

<u>Informativeness</u>

This criterion assesses the richness of information in the model's responses. The Base Model's performance is just under 3.5, indicating a moderate level of informativeness. Both the Base + PE and the Fine-tuned + PE models score similarly, just below 4.0. This suggests that while both prompt engineering and fine-tuning add to the model's ability to provide informative content, they do not differ much from each other when combined with prompt engineering.

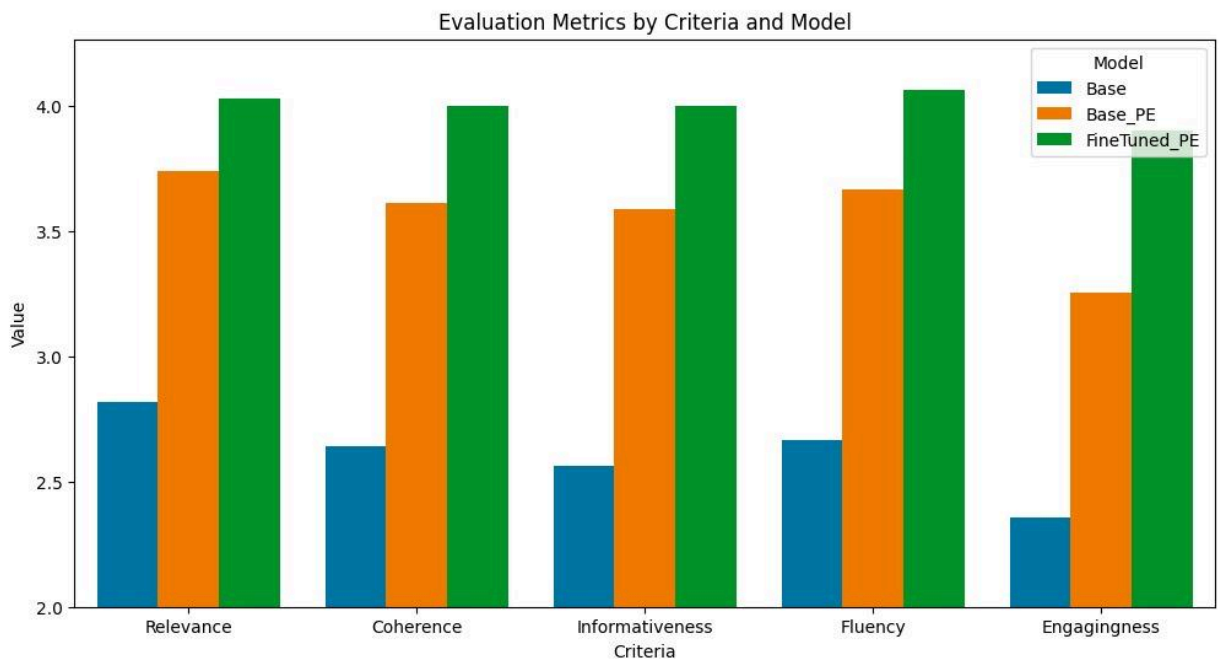
<u>Fluency</u>

Fluency measures the smoothness and readability of the text. The Base Model's score near 3.5 suggests it generates fairly fluent text, but there's room for improvement. Both the Base + PE and the Fine-tuned + PE models score just below 4.0, which indicates that prompt engineering greatly enhances fluency, and that fine-tuning does not make a significant difference in this aspect.

Engagingness

Engagingness gauges the ability of the model to produce responses that are interesting and likely to sustain the user's interest. The Base Model scores the lowest on this metric, just above 2.5, suggesting its responses are the least engaging. The Base + PE model shows some improvement, scoring around 3.0, while the Fine-tuned + PE model demonstrates a significant increase, with scores close to 4.0, indicating that the combined effects of fine-tuning and prompt engineering make the responses much more engaging.

In summary, prompt engineering contributes significantly to the improvement of all metrics over the Base Model. The graph illustrates that both prompt engineering and fine-tuning significantly enhance the llama-7b model's performance across various evaluation metrics, with the Fine-tuned + PE model generally outperforming the other two approaches.



Discussion

There are some observations after experiencing the fine-tuning ourselves:

1. Should let the model learn when to stop generating.
   a. Implementing mechanisms to control the verbosity and length of a model's output can significantly enhance its usability and relevance. The current models we have can often generate more content than necessary, exceeding the limited tokens, making itself cut-off midway, leading to redundancy or deviation from the core topic.
   b. Some techniques such as setting maximum token limits or incorporating stop signals within the training data would be something we might can explore in the future. Additionally, developing algorithms that assess the coherence and relevance of the content as it's being generated can help in determining optimal stopping points.
2. The model may generalize better with more policy context in training, or more input training text in general.
   a. Enriching the model's training data with diverse policy contexts can improve its ability to generalize across different scenarios and make more informed decisions.
   b. Integrating a wider range of scenarios, guidelines, and ethical considerations into the training dataset can make the model learns to navigate complex decision-making processes more effectively, leading to outputs that are not only relevant but also adhere to desired policies and ethical standards.
   c. There may be two possible ways of doing this:
      i. Include all QA pairs provided into the training process.
      ii. Include the articles or documents themselves into the training process.
3. Do some instruction Fine-Tuning to the model.
   a. If computational resources allow, enhancing instruction fine-tuning can significantly improve the model's performance by making it more responsive to user commands, rather than just trying to replicate the structure of the prompt and giving back irrelevant answer.
4. Do some Hyperparameter Tuning to make the model fit better.
   a. Hyperparameter tuning involves experimenting with different configurations to find the optimal settings for the model. Parameters such as learning rate, batch size, and the architecture's depth and width can be adjusted. The goal is to find a balance that allows the model to learn effectively without overfitting or underfitting the data.

Conclusion

In summary, our exploration into fine-tuning a Llama2 7b model highlighted the critical role of computational resources while also demonstrating that armed with fairly little experience, fine tuning can be accomplished to a certain extent. Despite initial challenges with access, we successfully fine-tuned the model using specific data and compared it with the base model. While we observed some improvement, we acknowledge the potential for greater enhancements through more comprehensive fine-tuning methods, including refined prompts and a thorough hyperparameter grid search. This underscores the importance of computational resources and methodological rigor in fine-tuning Large Language Models effectively. Based on the experiences of our group we feel that finetuning an LLM for a policy chat purpose is worth pursuing further with the recommendations we've provided above in the discussion. As the world is becoming increasingly more adept at the development of bespoke LLMs, policy is an area where AI has the potential to create a significant impact for the better.

Our code can be found on GitHub.
Our documents gathered for training can be found on our Drive.
Our accompanying slide deck.

Acknowledgements

The team would like to express our gratitude to the teaching assistants of the course Siva Komaragiri and Harshit Nanda for their tireless efforts to support the group and guarantee a positive learning experience for all. We would also like to thank Professor Anand Rao for creating an engaging and thoughtful project.

References

- https://lightning.ai/pages/community/tutorial/optimizing-llms-from-a-dataset-perspective/
- https://www.lakera.ai/blog/llm-fine-tuning-guide
- https://zohaib.me/a-beginners-guide-to-fine-tuning-llm-using-lora/
- https://solutyics.com/fine-tuning-a-language-model-for-question-answering-a-comprehensive-guide/
- https://huggingface.co/blog/peft
- https://www.promptingguide.ai/techniques/consistency
- https://cobusgreyling.medium.com/12-prompt-engineering-techniques-644481c857aa
- https://blog.ovhcloud.com/fine-tuning-llama-2-models-using-a-single-gpu-qlora-and-ai-notebooks/