Building and Evaluating AI PolicyChat

Final Project Report

Prepared by (ordered alphabetically): Quentin Auster, Meghan Holquist, Nitya Mathur, Colton Lapp, Kateryna Shapovalenko

Table of contents

Introduction

Project motivation and objectives

The goal of this project was to create a Question and Answer (Q/A) chatbot useful in policy analysis. To do this, we first evaluated prompting strategies that would allow us to use pre-trained large language models (LLMs) out of the box. We used pre-trained LLaMa-7B and LLaMA-7B Chat in our analysis. We found that this approach was not sufficient, so we proceeded to fine-tune the pre-trained LLaMA-7B model on a custom dataset of over 400 question-answer pairs we collected along with our colleagues that were specifically related to policy.

We evaluated each of these models—the original LLaMA-7b, LLaMA-7B Chat, and the fine-tuned version of LLaMA-7b—using a combination of automatic metrics (BLEU, METEOR, ROUGE, and ROUGE derivatives) and human evaluation metrics (relevance and coherence) to compare performance. We found that the fine-tuned model provided better performance relative to the pre-trained versions (across all prompting techniques) with minimal additional resources spent on fine-tuning.

Background on AI policy and LLMs

The fast-paced growth of artificial intelligence (AI) and LLM technology creates concern for policymakers and a sense of urgency to attempt to regulate their use. The myriad of concerns ranges all the way from data privacy to the environmental implications of training these models. There are also concerns about the technology's potential use cases. For example, the use of algorithms in the hiring process might exacerbate existing workforce inequities or deepfake technologies leveraged to spread misinformation online.

Given the potential costs and benefits associated with the technology as well as the ambiguity of its implications, there is significant disagreement among policymakers on how to regulate this technology. In October 2022, President Biden's administration released the "Blueprint for an AI Bill of Rights," outlining general guidance on balancing the use of automated systems and protecting individuals' rights. Since then, the discussion on AI policy more generally has expanded, but debate remains on best practices for regulation addressing common concerns.

Nevertheless, there are significant benefits to LLMs and AI, including bolstering technological innovation and improving workflow efficiency. Many of these LLMs have a strong general knowledge base but lack depth on niche topics. As this technology has grown, the availability of these pre-trained models has also increased, offering unique opportunities to build off of and fine-tune the original models toward specific domains or tasks.

Methodology

Data Collection and Preparation

Our group, and other groups, collected data from multiple policy-related documents, ranging from executive orders to policy think tank reports. Although these documents primarily focused on AI policy, we tried to diversify our dataset by gathering articles related to both foreign and domestic AI policy. Our team created twenty total question-answer pairs based on these documents. Each group in the class followed a similar approach and once aggregated, the final dataset consisted of about four hundred question-answer pairs about various policy-related documents. This creates a diverse set of policy-focused question-answer pairs.

Once we had these Q/A pairs, we converted them to JSON and CSV formats to be used for training and evaluation. We read in the custom database and split it into train and test sets using the HuggingFace API. We used an 80/20 train/test split. Q/A pairs were then concatenated into a single string with roughly the format of an (optional) prompting prefix, followed by the question, followed by a unique delimiter, followed by the answer.

Evaluation Metrics

We used the following automatic evaluation metrics to measure the quality of generated responses relative to ground truth responses in the dataset:

- **BLEU**. BLEU is a similarity metric commonly used in translation tasks to measure the similarity between generated and reference text. It does not account for meaning or sentence structure.
- **METEOR**. METEOR is a measure of quality in terms of alignment between generated and reference text. It utilized stemming and accounts for synonyms in addition to using straightforward word matching.
- **ROUGE (and derivatives)**. ROUGE and its derivative metrics (ROUGE bi-gram and ROUGE longest sequence) are commonly used in summarization or translation tasks and measure overlapping unigrams, bigrams, as well as sentence-level structure to compare generated and reference text.

We also used human evaluations of Coherence and Relevance to score a subset of our model's outputs. For these scores, each member of the team independently ranked the model-generated output on a scale of 1 (worst) to 5 (best) on the two metrics. We then averaged scores across team members.

Prompting Techniques

Prompt engineering refers to the method of crafting prompts to feed the LLM in order to aid in generating an output more in line with the user's needs. Our aim for prompting was to generate

answers that were the most similar to our baseline outputs. Characteristics we tried to emulate were: short length of answer, usually stated in paragraph format, logical flow and correct interpretation of what the query is asking for. We improved the quality of our prompts by trying one prompt on one QA pair and iterating on it. Our group tried multiple prompt engineering techniques:

- **No-shot learning**. To see how the model's output would perform in the absence of any additional examples or guidance, we simply used the question itself as the input.
- **Task Instruction**. We clearly defined what the model is supposed to do for our task, more than just answering the query. We tried giving it a persona saying that it is an expert in AI Policy.
    - Give the most concise answers possible to questions about AI policy, considering you are an expert on AI policy.
- **One-shot and few-shot learning**. An example (one-shot) or multiple examples (few-shot) are provided in the prompt as additional context for the model. Specifically, we provided examples of sufficient answers for the model to mirror its output. We experimented with different examples and quantities of them in few-shot learning.
    - Give an answer to my query by modeling the following example:

      My Query: Summarize the UK's legislation on AI.

      Your Answer: The UK's legislation on AI is currently decentralized, with no specific comprehensive law governing AI. Instead, existing laws such as data protection legislation (e.g., the Data Protection Act 2018), equalities and privacy laws (e.g., the Equality Act 2010 and the Human Rights Act 1998), and intellectual property laws (e.g., the Copyright, Designs and Patents Act 1988) play a role in regulating various aspects of AI development and usage. These laws impact data collection, discrimination, human rights implications, intellectual property rights, and the limitations on AI decision-making and surveillance tools in the workplace.
- **Chain-of-Thought Prompting**. We broke down the problem into logical steps, a process that is known to improve reasoning in the LLM. In the initial prompt, the model answered us in steps but since we were only looking for the final answer, we had to add a sentence that instructed the model to not output the intermediate steps. We also gave instructions on what each sentence should talk about.
    - First, analyze the keywords in the query. Secondly, decipher the purpose of the query. Don't explicitly write these. Your final answer should be a maximum of 3 sentences long. The first sentence should summarize what the question is asking. The second sentence should give the main answer to the query. The third sentence can be an additional point if you think some information is very important to the query. Format all the sentences into a single paragraph.

- **Active Prompting**. We slightly modified the traditional definition of active prompting (iteratively providing the model with feedback on responses) to eliminate the involvement of human evaluation and feedback by asking the model to generate multiple responses and pick the best based on defined criteria. According to our domain research, this kind of technique has not been defined yet but based on our calculated metrics, may not qualify as a novel technique (unless tweaked more).
    - Think of 3 possible different answers to the query but do not output them all. Only output the answer that is the shortest, most concrete, relevant to the query, and easy to understand by a college student. Do not include the reason for your pick.

We prepended the corresponding prompting text to example questions that were in the test set, then fed these texts into inference pipelines using both the LLaMA-7B pre-trained and LLaMA-7B Chat models to generate responses

Fine-Tuning

We fine-tuned Meta's LLaMA-7B HuggingFace model on the training set of 333 custom Q/A examples using LoRa/QLoRa (see **Figure 1**).

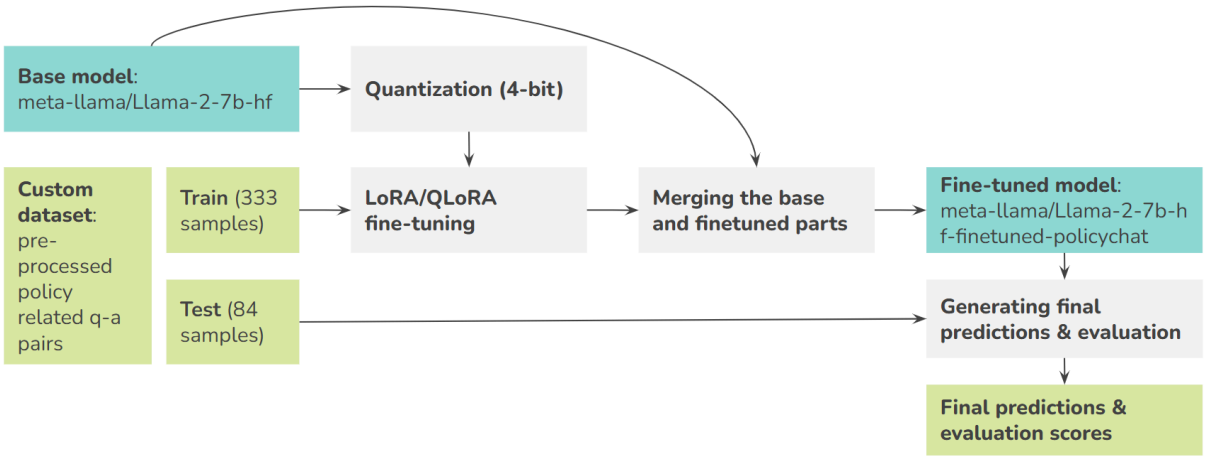

Figure 1. Fine-tuning pipeline using QLoRA.

The process, executed in Google Colab T4 with high RAM, involved several key steps:

- **Initial preparation**: We installed the necessary libraries and set up Hugging Face authentication, ensuring access to the LLaMA-7B base model and tokenizer.
- **Data handling**: The dataset, split into training and test sets, was preprocessed to fit the model's input format. For LLaMA, this meant formatting the data as concatenated question-answer pairs.
- **Fine-tuning setup**: We employed LoRa configurations for fine-tuning, which included setting parameters like LoRA's alpha, dropout, and attention dimension. We also configured training parameters such as the number of epochs, learning rate, batch size,

and gradient accumulation steps. Notably, we opted for a 5-epoch training duration to avoid overfitting (see **Figures 2 and 3**).

- **Model loading and training**: The pre-trained model was loaded with a 4-bit precision base using BitsAndBytesConfig. The training was conducted using the SFTTrainer, with a focus on causal language modeling. The training phase was completed in about 10 minutes, indicating efficient utilization of resources (see **Figure 4**).
- **Model merging and saving**: Post-training, the original and newly trained models were merged. This step was crucial for integrating the fine-tuned aspects with the base model, ensuring a cohesive final model that encapsulates both the original and newly acquired capabilities.
- **Inference and evaluation**: With the fine-tuned model saved to Google Drive, we performed inference tests. Using a custom function, responses were generated for the test dataset questions. The final predictions were then evaluated using BLEU, ROUGE, and other metrics for linguistic quality assessment.

```
lora_alpha   = 16 # Alpha parameter for LoRA scaling
lora_dropout = 0.1 # Dropout probability for LoRA layers
lora_r       = 64 # LoRA attention dimension

peft_config = LoraConfig(lora_alpha   = lora_alpha,
                         lora_dropout = lora_dropout,
                         r            = lora_r,
                         bias         = "none",
                         task_type    = "CAUSAL_LM")
```

Figure 2. Final QLoRA hyperparameters.

```
num_train_epochs              = 5 # Number of training epochs
fp16                          = True # Enable fp16 training
bf16                          = False # Enable bf16 training (set bf16 to True with an A100)
per_device_train_batch_size   = 1 # Batch size per GPU for training
per_device_eval_batch_size    = 1 # Batch size per GPU for evaluation
gradient_accumulation_steps   = 1 # Number of update steps to accumulate the gradients for --- was
gradient_checkpointing        = False # Enable gradient checkpointing
max_grad_norm                 = 0.3 # Maximum gradient normal (gradient clipping)
learning_rate                 = 2e-4 # Initial learning rate (AdamW optimizer)
weight_decay                  = 0.001 # Weight decay to apply to all layers except bias/LayerNorm w
optim                         = "paged_adamw_32bit" # Optimizer to use
lr_scheduler_type             = "cosine" # Learning rate schedule
max_steps                     = -1 # Number of training steps (overrides num_train_epochs)
warmup_ratio                  = 0.03 # Ratio of steps for a linear warmup (from 0 to learning rate)
group_by_length               = False # Group sequences into batches with same length
save_steps                    = 0 # Save checkpoint every X updates steps
logging_steps                 = 25 # Log every X updates steps
seed                          = 42
```

Figure 3. Final TrainingArguments hyperparameters.

```
max_seq_length = 500 # Maximum sequence length to use
packing        = True # Pack multiple short examples in the same input sequence to increase efficiency
data_collator  = DataCollatorForLanguageModeling(tokenizer=tokenizer,
                                                 mlm=False)
def compute_metrics(eval_preds):
    predictions, labels = eval_preds
    decoded_preds = tokenizer.batch_decode(predictions, skip_special_tokens=True)
    decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=True)
    bleu_score = bleu.compute(predictions=decoded_preds, references=decoded_labels)
    rouge_score = rouge.compute(predictions=decoded_preds, references=decoded_labels)
    return {"bleu": bleu_score, "rouge": rouge_score}

trainer = SFTTrainer(model             = model,
                     train_dataset     = finetune_dataset,
                     eval_dataset      = eval_dataset,
                     dataset_text_field = 'text',
                     peft_config       = peft_config,
                     max_seq_length    = max_seq_length,
                     tokenizer         = tokenizer,
                     args              = training_arguments,
                     packing           = True,
                     data_collator     = data_collator,
                     compute_metrics   = compute_metrics)
```

Figure 4. Final SFT hyperparameters.

The fine-tuning process was marked by careful consideration of hyperparameters and model configurations. We manually explored a significant set of hyperparameters and picked the best. The choice of a 5-epoch training duration balanced the need for learning new patterns without overfitting. The use of 4-bit precision and LoRa/QLoRa configurations aimed to optimize training efficiency and model performance. This fine-tuning approach not only enhanced the model's specific capabilities but also offered significant parameter savings, evidenced by the efficient training time and the successful integration of the fine-tuned layers with the base model.

Results and Discussion

Performance Across Prompting Techniques

We found no benefits from the prompting techniques we employed. The pre-trained models appeared to perform better without prompting. In some cases, longer prompt prefixes appeared to confuse the models, and they would repeat much of the same text back. In other strategies, such as active prompting, we saw the models generate text that repeated the strategy's prompt but did not follow it. Additionally, some prompt techniques performed better on some types of questions, which leads us to believe that the format of the query also comes into effect for the quality of prompt response.

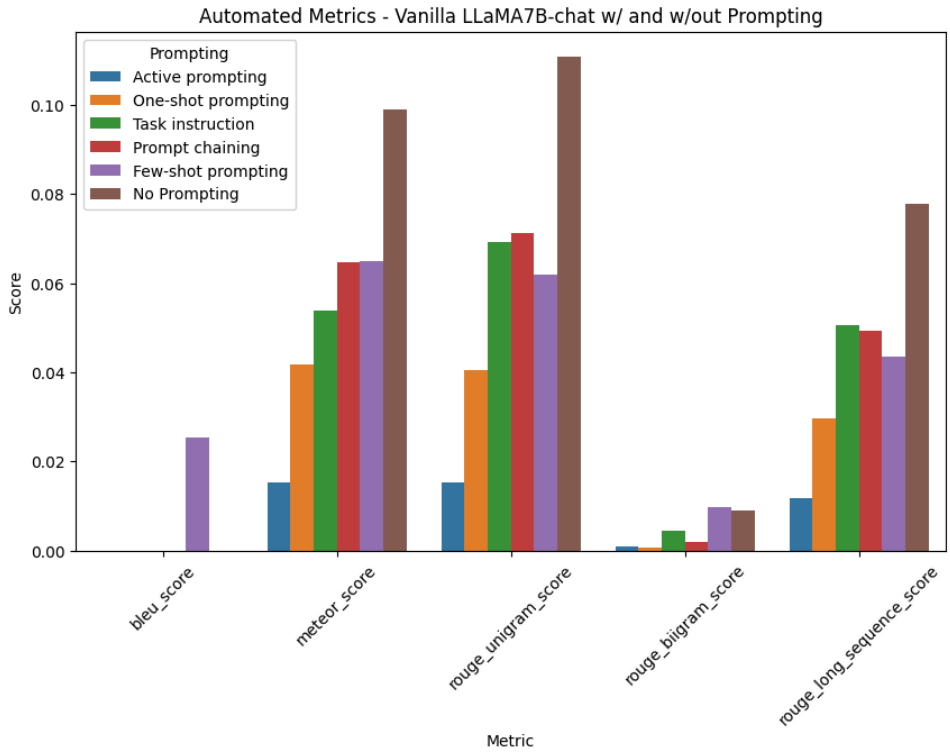

Figure 5. Evaluation of prompting techniques on LLaMA-7B Chat.

Performance Across Models

We found that the fine-tuned model performed much better relative to the best performance of the pre-trained models across all prompting strategies. As discussed above, prompting strategies in general did not improve performance. The original pre-trained LLaMa 7b seemed to regurgitate exam questions or completely irrelevant information, while the fine-tuned version

seemed to have more robust answers. These observations were reflected in both the automatic and human-evaluated metrics.

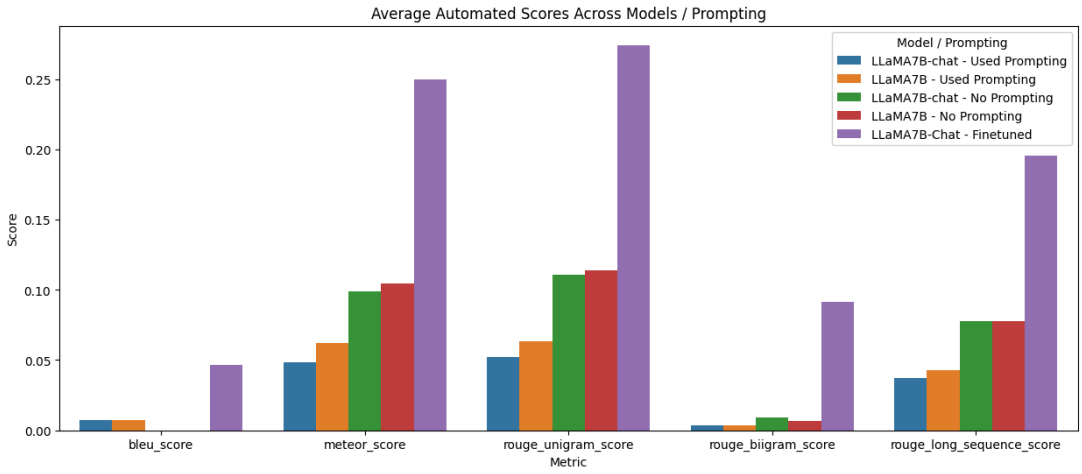

Figure 6. Evaluation of pre-trained models versus the fine-tuned model.

We found similar performance on the human evaluation metrics. Our process was for each of our team members to give a rating for a QA pair run through all models mentioned in Figure 7 based on coherence and relevance. We then calculated the average to determine the final human evaluation scores.

Approach		Coherence	Relevance	Average Score (1 - worst, 5 - best)
Baseline	**LLaMA7B - vanilla**	2.04	1.50	**2.20**
	Chat - vanilla	2.56	2.68	
Prompt-engineering	**LLaMA7B - best prompt**	2.40	2.60	**2.70**
	Chat - best prompt	2.72	3.08	
Fine-tuning	**LLaMA7B - finetuned, no prompt**	3.47	4.06	**3.76**

Key takeaways: Fine-tuned model performed the best on human evaluation.

Figure 7. Human evaluations of models. These evaluations were performed on a small subset of our test data due to resource and time constraints. In a business setting, it would likely be necessary to contract this work out to external partners to scale the testing process.

In a business setting, it could be a significant cost to pay individuals to read through the responses and manually evaluate them. While each answer is easy to score, this would be

extremely time-consuming at scale, and it would be difficult to audit. Additionally, and very importantly, the organization would need to ensure the human evaluators represent a diverse set of individuals to prevent bias in the ratings.

Cost-Benefit Analysis of Methods

We found performance benefits from fine-tuning a pre-trained LLM rather than using it out of the box or with simple prompting strategies. However, there are several costs associated with fine-tuning that are worth mentioning.

First, the creation of a high-quality custom dataset can be both time-consuming and difficult. The final dataset we used consisted of 400+ Q/A pairs, which is a relatively small training dataset, especially if we are interested in creating a highly knowledgeable policy chatbot that will generalize well to new information. Additionally, the quality of the Q/A pairs likely contributes to a fine-tuned model's downstream performance. Our group tried to create a range of diverse question-answer pairs, but our data collection was only a small subset of the class, and other groups may have followed different approaches. It takes significant time and effort to read through each document and ensure the questions created pertain to relevant information found in the document. In a business setting, we would want to carefully choose what third parties and/or internal organizations to work with when creating the dataset given its fundamental importance to the task.

Second, the process of human evaluation of model output was time-consuming and could be costly at scale. Additionally, it would be difficult and potentially expensive to mitigate the effects of low-quality ratings and/or bias on a large set of predictions across candidate models. We attempted to balance evaluations by taking an average across all individuals' scores. However, our scoring was likely biased because we were creating the models in the first place; of course in a business setting, we would likely contract with a third party to evaluate responses. Still, we would want to consider whether this contractor had a diverse group of people evaluating responses to avoid evaluation bias in response scoring. Done across large-scale output over many candidate models, this process would be extremely time-consuming and costly.

This said the curation of a dataset and the evaluation of responses is necessary to evaluate the less-onerous modeling with pre-trained models and prompting techniques. That is, the fact that these two things are costly does not necessarily differentiate fine-tuning from these more primitive procedures.

Given our use of QLoRA to fine-tune the model, this process was surprisingly non-labor, -compute-, or -time-intensive. However, in a business setting, we would need to spend additional time on hyperparameter tuning, which would require additional computing costs.

From the data we collected from this experiment and the consideration of the above factors, we would recommend steps to further improve fine-tuning performance.

Conclusions

In this paper, we created a Q/A chatbot model that would be useful for answering policy-related questions. In doing so, we wanted to consider both technical and business questions. Therefore, we first evaluated whether strong results could be achieved using either pre-trained models out-of-the-box or with simple prompting techniques. We found these approaches did not give satisfactory results. In fact, we found that the prompting techniques we experimented with often did not lead to any improvement over the out-of-the-box model. We then moved on to fine-tuning the model. Using QLoRA techniques to fine-tune, we saw a large improvement in performance with relatively minimal additional costs. The use of quantization and low-rank adaptation for model training meant that only a small number of the parameters had to be re-trained, so fine-tuning was quick (<10 min) and computationally cheap. Finally, we performed a cost-benefit analysis of the methods used. Although we found that constructing a custom dataset and performing human evaluations of generated text were likely to be the most expensive and time-consuming elements in a real-world setting (i.e., at scale), these were fundamental pieces of the improved performance we saw from fine-tuning.

Although not in the scope of this project due to time constraints, Retrieval Augmented Generation (RAG) could be useful for improving our policy-focused LLM. RAG offers the opportunity to leverage external information for the language model to access and retrieve information from. In the case of policy, a potential use case would be creating a database of all current legislation pertaining to AI policy. Doing so would give individuals both in the public and private sectors a platform to ask questions pertaining to what specific legislation exists today and the ability to easily cite specific pieces of legislation. In addition to using RAG, we might also experiment in the future with additional pre-trained models in search of better performance.

References

- Llama 2 on Hugging Face: https://huggingface.co/meta-llama/Llama-2-7b-hf
- Grading Conversational Responses Of Chatbots: https://arxiv.org/pdf/2303.12038.pdf
- Rouge 2.0: Updated And Improved Measures For Evaluation Of Summarization Tasks: https://arxiv.org/pdf/1803.01937.pdf
- Data sources:
    - United States Department of State - Artificial Intelligence (AI): Link
    - National Conference of State Legislatures - Artificial Intelligence 2023 Legislation: Link
    - The White House - Executive Order on AI: Link
    - The White House Office of Science and Technology Policy - AI Bill of Rights: Link
    - Department of Homeland Security - AI at DHS: Link
    - Cybersecurity and Infrastructure Security Agency - AI: Link
    - National Institute of Standards and Technology - AI Standards: Link
    - General Services Administration - AI Community of Practice: Link
    - National Institute of Standards and Technology - AI Research and Innovation: Link
    - Department of Homeland Security - DHS New AI Policies: Link
    - RAND Corporation - AI Activities for DoD: Link
    - Brookings Institution - AI Regulation: Link
    - CSIS - AI Regulation Outcomes: Link
    - Council on Foreign Relations - AI and Robotics: Link
    - RAND Corporation - AI Policymaking: Link
    - Brookings Institution - International Cooperation on AI: Link
    - CSIS - Japan's AI Regulation: Link
    - RAND Corporation - AI's Impact on Nations: Link
    - Brookings Institution - AI Bill of Rights: Link
    - CSIS - Trustworthy AI and Global Governance: Link
    - United Nations - Resource Guide on AI Strategies: Link
    - World Bank - AI Policy: Link
    - OECD - An Overview of National AI Strategies and Policies: Link
    - World Trade Organization (WTO) - The Promise of TradeTech: Link
    - United Nations - AI Advisory Body Interim Report: Link
    - World Bank - AI FAQs: Link
    - OECD - AI Policy Resources: Link
    - United Nations - Draft Resource Guide on AI Strategies: Link
    - World Bank - AI Interpretations: Link
    - United Nations - Unite Paper on Ethical AI: Link
    - MIT AI Policy Forum: Link
    - Public Policy Forum - AI Policy Compass: Link
    - TechPolicy.Press - AI Orders and Summits and Forums: Link
    - The Forum for Cooperation on Artificial Intelligence: Link

- Partnership on AI’s Policy Forum: Link
- CNAS AI Governance Forum: Link
- Technology Association of Iowa - 2024 AI Public Policy Forum & Legislative Reception: Link
- OCEANIS - Global AI Standards Repository: Link
- GAIEC Repository - Institute for Ethics in Artificial Intelligence: Link
- AI RESEARCH PROGRAM REPOSITORY: Link
- Data.gov - Results for "AI": Link