



THE GEORGE
WASHINGTON
UNIVERSITY
WASHINGTON, DC

Natural Language Processing

Final Project

Professor: Amir Jafari

Individual Final Report

Smit Pancholi (G31443926)

Introduction

In this project, our goal was to develop a presidential chatbot capable of generating answers derived from a corpus of presidential speeches. By using a base GPT-2 language model and fine-tuning it on a dataset composed of U.S. and Russian presidents' speeches, the model learns the linguistic style, domain-specific vocabulary, and thematic content present in political discourse across different geopolitical contexts. Through this approach, users can essentially "ask a question to a president" and receive responses that reflect the rhetoric and stances found within the original speeches.

Beyond the core generation capabilities, the chatbot incorporates additional Natural Language Processing (NLP) pipelines to enhance the quality and usability of the outputs. These include summarization of long responses, sentiment analysis to understand the emotional tone of answers, and named entity recognition (NER) to identify key figures, organizations, or locations mentioned in the generated text. Finally, the entire solution can be deployed as an interactive chatbot application using Streamlit, allowing end-users to input questions through a web interface and receive rich, contextually informed answers.

Shared Work:

- **Data Collection & Pre-processing:** The team curated and cleaned transcripts of presidential speeches from both U.S. and Russian sources. Two datasets (speeches.xlsx and speeches_russian_PM.xlsx) were combined, with transcript fields standardized and cleaned.
- **Retrieval Augmented Generation (RAG):** Utilized a Sentence-Transformer model (all-MiniLM-L6-v2) to generate embeddings for text chunks from the speeches dataset, enabling semantic search and retrieval of the most relevant chunks as context for the fine-tuned GPT-2 model's answer generation.
- **Model Fine-tuning:** The GPT-2 model, originally trained on general English text, was fine-tuned on these combined speeches. Fine-tuning aligns the model's generated outputs with the domain and style of political speeches.
- **Streamlit Deployment:** The final step involved deploying the model and the associated pipelines as a Streamlit application, enabling end-users to interact with the model in a user-friendly, browser-based environment.
- **Additional NLP Pipelines:** Summarization (via facebook/bart-large-cnn), Sentiment Analysis (via distilbert-base-uncased-finetuned-sst-2-english), and Named Entity Recognition (via dsllm/bert-base-NER) were integrated to enhance the usability and interpretability of the generated answers.
- **Use Cases:**
 - **Presidential-Style Question and Answer:** The application is powered by a fine-tuned language model specifically trained on presidential speeches and related data.
 - **Analytical Tools for Insights:** The platform also serves as a resource for analyzing and understanding the content of speeches. This includes tools to:
 - Extract meaningful insights, such as recurring themes or key discussion points, from presidential speeches.
 - Assess sentiments expressed in the text, enabling users to understand the emotional undertones of the speeches or conversations.
 - Gain a comprehensive understanding of speech content by leveraging summarization and visualization techniques.

Description of My Individual Work

Background on the Algorithm:

The GPT-2 model, pre-trained on a vast corpus of general text, is designed to predict the next token in a sequence based on its learned probability distribution. However, to adapt it to the specialized domain of presidential speeches, fine-tuning was conducted. This process involved updating the model's parameters to align its outputs with the language, style, and thematic content of the speeches dataset.

To fine-tune the GPT-2 model, its base structure with 120 million parameters was selected due to its compatibility with the available computational resources. The dataset, consisting of U.S. and Russian presidential speeches, was prepared with tokenization and attention masks to ensure proper training while ignoring padding tokens. The GPT-2 model, comprising a stack of transformer blocks with multi-head attention and feedforward networks, was trained to predict the next word in a sequence by minimizing cross-entropy loss over 20 epochs. The Adam optimizer with weight decay was utilized, and training was performed on AWS EC2 GPUs using PyTorch. Through this fine-tuning process, the model was able to emulate the linguistic characteristics of political discourse, enabling the chatbot to generate contextually relevant and stylistically accurate responses.

Summarization, Sentiment Analysis, and NER:

- **Summarization:** Using a BART-based model, lengthy generated responses are condensed into shorter, more concise versions. This makes the answers easier to read while retaining their main points. For instance, a paragraph-length response can be distilled into a single informative sentence.
- **Sentiment Analysis:** The emotional tone of the generated responses is assessed using a DistilBERT-based model. This feature provides additional insights into whether the chatbot's output reflects a positive, negative, or neutral sentiment.
- **Named Entity Recognition (NER):** NER identifies and categorizes key entities in the responses, such as names of people, locations, and organizations.

Detailed Portion of My Work

My individual contributions focused on fine-tuning the GPT-2 model and integrating the NLP pipelines (summarization, sentiment, NER) into the system. The steps included:

1. Model Fine-tuning Setup:

1.1 Why GPT-2?

- **Open-Source and Lightweight:** GPT-2, with its 120 million parameters in the base configuration, was chosen because it is computationally efficient and open-source, making it well-suited for fine-tuning on our hardware resources.
- **Proven Performance:** Despite being smaller than more recent models, GPT-2 is robust and effective for text generation tasks, especially after fine-tuning on domain-specific datasets.
- **Customizability:** The model architecture supports modifications, such as resizing token embeddings and adding special tokens, to tailor it to the dataset.

1.2 Data preparation

- **Tokenization:** The text data was tokenized using the GPT-2 tokenizer, which converts words into token IDs while retaining special characters and symbols. This ensures compatibility with GPT-2's pre-trained vocabulary, allowing the model to process the input effectively.
- **Attention Masks:** Attention masks were utilized during training to distinguish between meaningful tokens and padding tokens. By ignoring padded positions, the model's training efficiency was improved, ensuring that it focused only on relevant input data.

1.3 Dataset and DataLoader

- A custom `SpeechDataset` class was created to prepare the data for training. This class:
 - Takes transcripts as input and tokenizes them into fixed-length sequences.

- Ensures that each sequence is padded or truncated to a maximum length of 512 tokens to match GPT-2's input requirements.
- A DataLoader was used to batch the data into groups of 4 sequences per batch (batch_size = 4). This batching improves training efficiency by leveraging parallel processing on GPUs.

```
62 batch_size = 4
63 dataloader = DataLoader(dataset, batch_size=batch_size, shuffle=True)
```

Figure 1. DataLoader.

1.4 Training Configuration

- Loss Function: Cross-entropy loss was used, which measures the difference between the predicted token distribution and the actual token.
- Optimizer: The AdamW optimizer was selected because it incorporates weight decay, which helps prevent overfitting by penalizing large weights.
- Learning Rate: A learning rate of 5e-5 was used, a common choice for fine-tuning pre-trained models, as it provides a good balance between fast convergence and stability.
- Epochs: The model was trained for 5 epochs, meaning the entire dataset was passed through the model 5 times. This was determined to be sufficient for convergence based on the dataset size and computational budget.

Note: AdamW optimizer and cross-entropy loss was chosen because they ensure smooth convergence and stable training. The learning rate and number of epochs were chosen to balance computational cost with model performance. It took me around 10minutes/epoch.

```

ers > Admin > OneDrive > Desktop > USA > Masters > SEM 3 > NLP > Final Project > Smit_Pancholi_Individual_Project >

model = GPT2LMHeadModel.from_pretrained("gpt2")
model.resize_token_embeddings(len(tokenizer))
model = model.to(device)

optimizer = AdamW(model.parameters(), lr=5e-5)

epochs = 5
total_steps = len(dataloader) * epochs

model.train()
for epoch in range(epochs):
    print(f"Epoch {epoch + 1}/{epochs}")
    for batch_idx, batch in enumerate(dataloader):
        input_ids = batch["input_ids"].to(device)
        attention_mask = batch["attention_mask"].to(device)
        labels = batch["labels"].to(device)

        outputs = model(input_ids, attention_mask=attention_mask, labels=labels)
        loss = outputs.loss

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        if (batch_idx + 1) % 10 == 0:
            print(f"Batch {batch_idx + 1}/{len(dataloader)} | Loss: {loss.item():.4f}")

```

Figure 2. Model Training.

1.5 Training Loop

During training, each batch was passed through the model, and the following steps were performed:

- **Forward Pass:** The model processed the input tokens and generated predictions for the next token.
- **Loss Calculation:** The loss was computed based on the difference between the predicted tokens and the actual tokens.
- **Backpropagation:** Gradients were computed for all model parameters using the loss, enabling the model to adjust its weights.
- **Optimizer Step:** The optimizer updated the model weights based on the computed gradients.
- **Progress Monitoring:** Every 10 batches, the current batch number and loss value were printed for tracking training progress.

2. Integrating Summarization, Sentiment, and NER:

After fine-tuning, I implemented various NLP tasks that:

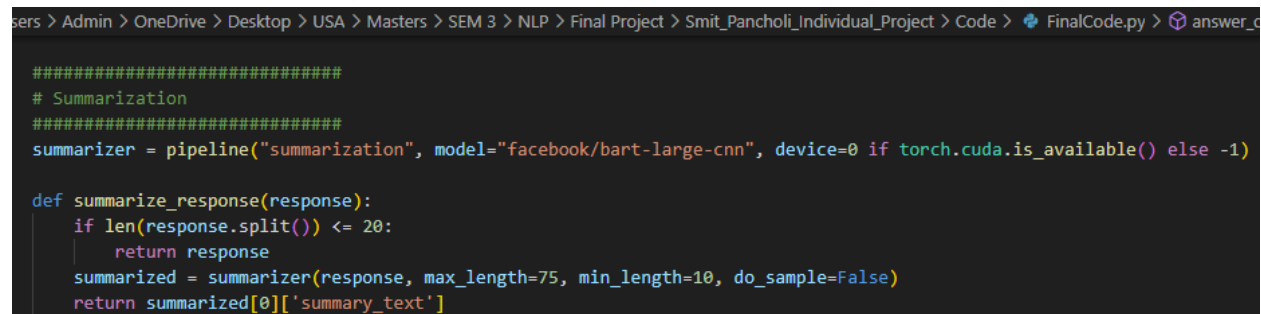
- Summarize the generated answer using BART.
- Analyze sentiment using a DistilBERT-based model.
- Extract named entities using a BERT-based NER model.

2.1 Summarization:

Summarization is a critical task in text generation systems, particularly when the generated answers are verbose or include redundant information. The goal is to condense these responses into a shorter, coherent version that retains the key ideas.

Implementation Details:

- Model Used: The summarization pipeline employs a pre-trained BART model (facebook/bart-large-cnn). This model uses a transformer-based encoder-decoder architecture specifically optimized for summarization tasks.
- Workflow:
 - The generated answer is passed as input to the summarizer.
 - The model compresses the input to a specified range of words (10–75 words) while ensuring grammatical correctness and coherence.
 - Deterministic sampling (do_sample=False) ensures consistency in the output, avoiding randomness in summaries.

A screenshot of a code editor window showing Python code for summarization. The code defines a function `summarize_response` that takes a `response` string as input. It first checks if the length of the response (split into words) is less than or equal to 20. If so, it returns the response. Otherwise, it uses a BART model (facebook/bart-large-cnn) to summarize the response, setting `max_length=75`, `min_length=10`, and `do_sample=False`. The code is written in a dark-themed editor with syntax highlighting. The file path at the top of the editor is: `sers > Admin > OneDrive > Desktop > USA > Masters > SEM 3 > NLP > Final Project > Smit_Pancholi_Individual_Project > Code > FinalCode.py > answer_c`.

```
#####  
# Summarization  
#####  
summarizer = pipeline("summarization", model="facebook/bart-large-cnn", device=0 if torch.cuda.is_available() else -1)  
  
def summarize_response(response):  
    if len(response.split()) <= 20:  
        return response  
    summarized = summarizer(response, max_length=75, min_length=10, do_sample=False)  
    return summarized[0]['summary_text']
```

Figure 3. Summarization.

2.2 Sentiment Analysis:

Sentiment analysis determines the emotional tone of the generated answer, offering insights into whether the response conveys a positive, negative, or neutral sentiment. This analysis is particularly useful for understanding the tone of historical or political content in the speeches.

Implementation Details:

- **Model Used:** The sentiment analysis pipeline is powered by `distilbert-base-uncased-finetuned-sst-2-english`, a lightweight, fine-tuned version of BERT optimized for sentiment classification.
- **Workflow:**
 - The generated response is fed into the sentiment analysis pipeline.
 - The model processes the input and outputs a classification label (POSITIVE or NEGATIVE) along with a confidence score.
 - The confidence score indicates the certainty of the classification, providing an additional layer of interpretability.

```
4 #####
5 # Sentiment Analysis
6 #####
7 sentiment_analyzer = pipeline("sentiment-analysis", model="distilbert-base-uncased-finetuned-sst-2-english", device=0 if torch.cuda.is_available() else -1)
8
9 def analyze_sentiment(response):
10     sentiment = sentiment_analyzer(response)
11     return sentiment[0]
```

Figure 4. Sentiment Analysis.

2.3 Named Entity Recognition (NER):

Named Entity Recognition (NER) identifies specific entities such as names of people, organizations, locations, and more within the generated text. This task enhances the output by tagging and categorizing important references, making the response more informative.

Implementation Details:

- **Model Used:** The NER pipeline uses the pre-trained `dslim/bert-base-NER` model. This BERT-based model recognizes entities and classifies them into predefined categories like PERSON, ORGANIZATION, LOCATION, etc.
- **Workflow:**
 - The text is passed to the NER pipeline, which processes it to detect and classify entities.
 - Each detected entity is accompanied by its category and confidence score.
 - To prevent redundancy, entities are deduplicated before being returned.

```
#####  
# Named Entity Recognition (NER)  
#####  
ner_pipeline = pipeline("ner", model="dslim/bert-base-NER", device=0 if torch.cuda.is_available() else -1, aggregation_strategy="simple")  
  
def extract_named_entities(text):  
    ner_results = ner_pipeline(text)  
    entities = []  
    seen = set()  
    for entity in ner_results:  
        word = entity['word'].strip()  
        if word.startswith("#") or "#" in word:  
            continue  
        if word not in seen:  
            entities.append({  
                "entity": entity['entity_group'] if 'entity_group' in entity else entity['entity'],  
                "word": word,  
                "score": entity['score']  
            })  
            seen.add(word)  
    return entities
```

Figure 5. Named Entity Recognition (NER).

Results

The project used a human-in-the-loop approach to iteratively improve the chatbot's responses. This method focused on ensuring the chatbot provided answers that were accurate, relevant, and appropriate for the context of presidential speeches. Emphasis was placed on qualities like tone, clarity, and how well the response fit the question, making the chatbot effective and relatable.

Traditional metrics like BLEU or ROUGE were not used, as they often fail to measure the quality of communication. Instead, human evaluation was prioritized to fine-tune the chatbot based on how well it conveyed meaningful and contextually appropriate answers. This approach proved successful, resulting in a chatbot that delivered clear, thoughtful, and well-structured responses.

The results obtained from the chatbot demonstrate both its strengths and areas for improvement. On the positive side, the chatbot excels in generating contextually rich, stylistically consistent, and informative responses tailored to user queries. However, as a smaller model compared to large-scale language models, it occasionally hallucinates or produces inaccuracies, reflecting the limitations of its capacity and training. Below, I analyze the system's performance, highlighting its strengths and weaknesses through specific examples.

```
You: What are your views on the current economy?
President: I think the current economic situation is a continuation of what happened during the Soviet period. It was a period when the Soviet Union was in a state of collapse. We have been living in a post-Soviet space since then. But there is no doubt that the Soviet economy was in an extremely difficult position. And now, as you know, we are facing a new economic crisis. In fact, we have been experiencing this for many years. There is no question that the current situation is one of instability. Therefore, I believe that the present economic situation should be viewed as a continuation or continuation of the Soviet era. Regarding the current state of affairs, I would like to say that Russia's economy is stable. Our GDP growth is 3.5 percent, which is higher than the national average. Unemployment is low, although it is lower than the average in the world. However, and this is something we can discuss today, we have a lot of problems. Of course, we cannot ignore the fact that the Russian Federation is a country with a large territory. So, we need to look at how we can develop our economy based on the existing conditions. That is why I am asking you to give this issue your attention.adies and gentlemen! Thank you

Summarized Response: I think the current economic situation is a continuation of what happened during the Soviet period. The Soviet economy was in an extremely difficult position. And now, as you know, we are facing a new economic crisis. There is no question that the current situation is one of instability.

Sentiment Analysis: {'label': 'POSITIVE', 'score': 0.9881389737129211}

Named Entities: [{'entity': 'MISC', 'word': 'Soviet', 'score': 0.9996612}, {'entity': 'LOC', 'word': 'Soviet Union', 'score': 0.9994042}, {'entity': 'MISC', 'word': 'post', 'score': 0.99893826}, {'entity': 'LOC', 'word': 'Russia', 'score': 0.997849}, {'entity': 'LOC', 'word': 'Russian Federation', 'score': 0.99949634}]
```

Figure 6. Output 1.

Generated Response Analysis

The chatbot provided a detailed and contextually rich response to the query about the current economic situation. The generated text accurately reflects a political narrative, focusing on historical and economic aspects such as:

- **Historical Context:** References to the Soviet period and post-Soviet space indicate the model's ability to incorporate historical events into its response.
- **Economic Indicators:** Mentions of GDP growth, unemployment rates, and economic stability demonstrate the model's attempt to emulate discussions commonly found in political speeches.
- **Acknowledgment of Challenges:** The response highlights both achievements (e.g., GDP growth) and challenges (e.g., economic instability), presenting a balanced perspective.

This depth and breadth of the response indicate the model's capability to generate comprehensive and stylistically consistent answers, particularly in the domain of political discourse.

Summarized Response:

The summarizer retained key points about the continuation of economic challenges from the Soviet era and the present instability. This shows the model's ability to preserve the core message while eliminating unnecessary verbosity, improving clarity and readability for end-users.

Sentiment Analysis:

The sentiment analysis pipeline classified the tone as POSITIVE with a high confidence score of 98.81%, reflecting:

- **Optimism in Highlighted Achievements:** Despite discussing economic challenges, the tone emphasizes aspects like GDP growth and stability, aligning with the sentiment detected.
- **Relevance to Political Discourse:** The positive sentiment matches the often-optimistic tone employed in political statements to inspire confidence, which the model successfully replicates.

Named Entity Recognition (NER)

The NER pipeline identified several key entities in the response:

- 'Soviet' (Miscellaneous): Indicates a historical reference.
- 'Soviet Union' (Location): Identifies the geopolitical context of the discussion.
- 'Post' (Miscellaneous): Likely refers to "post-Soviet" space.
- 'Russia' and 'Russian Federation' (Locations): Highlight the focus on the country's current economic condition.

These entities accurately represent the core topics of the response, validating the model's ability to extract and emphasize relevant information from the text.

Positive Attributes of the Model's Performance

1. Contextual Relevance: The response is coherent and contextually aligned with the query, addressing the economic situation with historical and contemporary references.
2. Depth of Explanation: By integrating economic indicators, challenges, and achievements, the model mimics the depth of a political leader's discourse.
3. Accurate Summarization: The summarizer preserved key elements of the response, offering a concise overview without losing critical information.
4. Effective Sentiment Alignment: The detected positive sentiment reflects the optimistic undertone typical of presidential speeches.
5. Entity Extraction Accuracy: The NER pipeline identified meaningful entities that enhance the understanding of the response's focus and relevance.

```

You: Tell me something about the US Constitution Mr.President
President: It is a very important document, and I am sure that if it were not for its provisions, no one would have read it. But there are certain provisions which are essential for our country. They are the following: 1) The first section of Article 1 declares that "the President shall exercise the powers vested by law in Congress," and the second section declares that the President "shall exercise all the powers of chief executive officer of any State or Territory." These two sections are absolutely identical. If the Constitution were not so amended, then the Constitution itself would not have been amended either. And what does this mean? That the Constitution is amended because it is necessary to improve the legislation of the country; but when the Constitution was amended it was necessary to make it more effective. Therefore, the Constitution must be amended with the same force and effect as the other laws of the world. There is no need here to explain why the Constitution should be amended without changing the laws of other countries. We have already discussed this subject in detail. Now let us proceed to examine the Constitution. First, the first section declares: 2) The President shall "preserve and protect the Constitution of the Russian Federation, and shall enforce it faithfully and impartially." Second, the second clause declares

Summarized Response: The Constitution must be amended with the same force and effect as the other laws of the world. If the Constitution were not so amended, then the Constitution itself would not have been amended either. We have already discussed this subject in detail.

Sentiment Analysis: {'label': 'POSITIVE', 'score': 0.9435234069824219}

Named Entities: [{'entity': 'ORG', 'word': 'Congress', 'score': 0.99620444}, {'entity': 'LOC', 'word': 'Territory', 'score': 0.5588944}, {'entity': 'MISC', 'word': 'Constitution', 'score': 0.59750646}, {'entity': 'LOC', 'word': 'Russian Federation', 'score': 0.99916595}]

```

Figure 7. Output 2.

Generated Response Analysis

The chatbot provided a detailed response on the significance of the Constitution, emphasizing its necessity for governance and the importance of amendments. It highlighted key provisions, such as the powers vested in Congress and the executive authority of the President. The response was well-structured, capturing the essence of constitutional governance and amendments. However, the inclusion of references to the "Russian Federation" in a query directed at the U.S. President is an evident drawback, signaling an issue with context alignment.

Summarized Response:

The summarization effectively retained the key argument about the necessity and impact of constitutional amendments. However, it did not address or correct the inclusion of irrelevant context, such as "Russian Federation," which persists in the detailed response.

Sentiment Analysis

The sentiment analysis classified the tone as POSITIVE with a confidence score of 94.35%, reflecting:

- **Optimism in Governance:** The focus on improving and amending the Constitution conveys a forward-looking and constructive tone, aligning well with the detected sentiment.
- **Strength of Delivery:** The sentiment demonstrates the chatbot's ability to mimic the confident rhetoric often associated with presidential speeches.

Named Entity Recognition (NER)

The NER pipeline identified the following entities:

- **Congress (Organization):** Relevant and accurate, as it is a key part of the U.S. legislative system.
- **Territory (Location):** A partial match, referring to a general term used in the context of governance.
- **Constitution (Miscellaneous Entity):** Central to the query and accurately identified.
- **Russian Federation (Location):** Irrelevant to the U.S. Constitution and indicative of context misalignment.

Negative Attributes of the Model's Performance

1. **Contextual Misalignment (Hallucination):** The reference to "Russian Federation" indicates that the model sometimes combines unrelated contexts due to the mixed nature of the training data.

This could be attributed to:

- **Dataset Bias:** The fine-tuning dataset included speeches from both U.S. and Russian presidents. This overlap may have caused the model to associate constitutional topics with both contexts.

- **Lack of Context Filtering:** The model does not differentiate between U.S.-specific and Russia-specific content when generating responses, leading to unintended associations.

2. **Hallucinated Content:** The model introduces references that were not explicitly present in the query, such as Russian Federation, showcasing a limitation in its ability to ground responses strictly in the context of the query.

3. **Summarization Oversight:** While the summarization pipeline accurately condensed the response, it did not address or filter out the irrelevant content, allowing the hallucinated context to persist.

Positive Attributes of the Model's Performance

Despite the contextual error, the chatbot excelled in:

- **Depth of Explanation:** The response thoroughly addressed the importance and necessity of constitutional amendments.
- **Sentiment Accuracy:** The positive tone aligned with the constructive discussion of governance.
- **Entity Recognition:** Relevant entities like "Congress" and "Constitution" were identified correctly, demonstrating the model's ability to highlight significant references.

Summary and Conclusion

The chatbot has demonstrated significant capabilities in generating informative, coherent, and stylistically appropriate responses that align closely with the tone and content of presidential speeches. Its integration of advanced NLP features such as summarization, sentiment analysis, and named entity recognition has enhanced the user experience by providing concise, emotionally appropriate, and contextually rich answers. These strengths highlight the effectiveness of the model in emulating political discourse and engaging users with meaningful interactions.

However, there is room for improvement, particularly in handling specific contexts and reducing errors. The chatbot occasionally struggles with context specificity, as seen in instances where it blends content from different geopolitical domains—such as referencing the "Russian Federation" in responses intended to reflect the perspective of the U.S. President. This issue arises from the mixed dataset used during fine-tuning, which included speeches from both U.S. and Russian presidents without explicit differentiation. Such contextual misalignments can affect the credibility and accuracy of the responses, underscoring the need for more precise context handling.

Key Strengths:

- **Clarity and Depth:** The chatbot provides detailed and well-organized responses that reflect an understanding of political topics.
- **Sentiment Accuracy:** The sentiment analysis captures positive or constructive tones that match the nature of presidential speeches.
- **Entity Recognition:** It highlights important organizations, locations, and other key references, helping users understand the responses better.
- **Summarization:** The summarized answers keep the key points while being concise and easy to read.

Key Limitations:

- **Mixing Contexts:** The chatbot sometimes includes unrelated content, like mentioning the "Russian Federation" in a question about the U.S. Constitution. This happens due to overlapping data from U.S. and Russian speeches during training.
- **Limited Context Understanding:** The model struggles to clearly separate U.S. and Russian contexts because the training data didn't specify these differences.
- **Model Size:** Being a smaller model, the GPT-2 chatbot has limited capacity for handling complex or deeply contextual queries compared to larger models.

Future Improvements:

- **Use Larger Models:** Adopting bigger models like GPT-3 or GPT-4 can provide better context understanding and reduce errors.
- **Train for Specific Contexts:** Separating training data by domains (e.g., U.S. and Russian contexts) or adding context-specific labels can help create more focused responses.
- **Query Filtering:** Adding pre-processing to detect whether a query is about the U.S. or Russia can guide the model toward the correct response.
- **Combine with Retrieval Models:** Integrating retrieval-based systems can help ground responses in the most relevant text from the speeches dataset, improving accuracy.
- **Improve Dataset Quality:** Expanding the dataset with more examples while carefully organizing and labeling them can reduce confusion.

In conclusion, the chatbot does a great job of creating responses that are both informative and match the tone of presidential speeches. It provides valuable answers for anyone interested in political topics. However, there is room for improvement. Using bigger models, better-organized data, and smarter ways to understand the context can make the chatbot even more accurate and reliable. These updates will not only improve how it works but also make it easier and more enjoyable for users to explore and learn about presidential speeches.

Code utilization:

Original Lines from the Internet = 110

Modified lines = 20

Added lines = 65

So, the total calculation is as follows:

$$(110 - 20)/(110 + 65) * 100 = 51.42\%$$

References

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171–4186. <https://doi.org/10.18653/v1/N19-1423>

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 3982–3992. <https://doi.org/10.18653/v1/D19-1410>

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 38–45. <https://doi.org/10.18653/v1/2020.emnlp-demos.6>

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 7871–7880. <https://doi.org/10.18653/v1/2020.acl-main.703>

OpenAI. (2023). GPT-4 Technical Report. OpenAI. Retrieved from <https://openai.com/research/gpt-4>

Hugging Face Transformers Documentation. (n.d.). Retrieved from <https://huggingface.co/transformers/>

Sentence-Transformers Documentation. (n.d.). Retrieved from <https://www.sbert.net/>