Project Overview

After an extensive search for an interesting topic to do our final project on, we settled on working with transcripts of spoken presidential remarks and executive orders from every single U.S. president in history. This topic was unique and presented a wide range of possible objectives. It also presented a great challenge, as we had to create the dataset and deal with unforeseen problems as we started to apply natural language processing techniques to the data. Our project has two main parts: an embedding part and a low rank adaptation (LoRA) part. Both parts have the goal of mimicking presidential speeches and executive orders.

Creating our dataset

The American Presidency Project, maintained by UC Santa Barbara, has compiled over 200,000 documents sourced from every president. The site contains documents such as spoken addresses, executive orders, campaign messages, and statements. In order to keep efficiency while getting quality data, we chose to only focus on spoken addresses and executive orders. This gave us about 45,000 pieces of text that could be used to achieve our objectives. However, downloading 45,000 documents and processing them is not that simple, especially when they are not readily available for download. We used a web scraping tool to pull down each directory page, grab all links to necessary documents on that directory page, and then loop through those pages. Once our scraper visited a document page, it would grab the author of the document, the date the document was written, and the complete text of the document.

The data we scraped was collected into a tab-separated value (TSV) file with three columns: Author, Date, and Text. We decided to use the TSV format due to our familiarity with processing TSV files and the ease with which we could put the web scraper output into TSV format. Data cleaning was relatively straightforward as the scraper handled most of the exceptions up front. However, there were instances where the scraper would pull the same document twice, but this was easily fixed through standard command line text processing.

For use with LoRA, simple text processing commands were used to grab all of the unique authors, essentially creating a list of presidents, and collect them in a text file. At the end of this process, our dataset consisted of two TSV files, one containing 10,000 executive orders and one containing 35,000 spoken records and addresses, and one text file containing all unique authors. An excerpt from our dataset on executive orders is depicted below:

Harry S Truman	April 17, 1945	Whereas after investigation I find and proclaim that the…
Harry S Truman	April 13, 1945	By virtue of and pursuant to the authority vested in me…
Harry S Truman	April 13, 1945	By virtue of the authority vested in me by section 1753…
Franklin D. Roosevelt	April 11, 1945	By virtue of the authority vested in me by section…
Franklin D. Roosevelt	April 10, 1945	Whereas after investigation I find and proclaim…
Franklin D. Roosevelt	April 04, 1945	By virtue of the authority vested in me by the…
Franklin D. Roosevelt	April 03, 1945	By virtue of the authority vested in me by the…
Franklin D. Roosevelt	March 23, 1945	By virtue of the authority vested in me by the…

Creating the speech embeddings

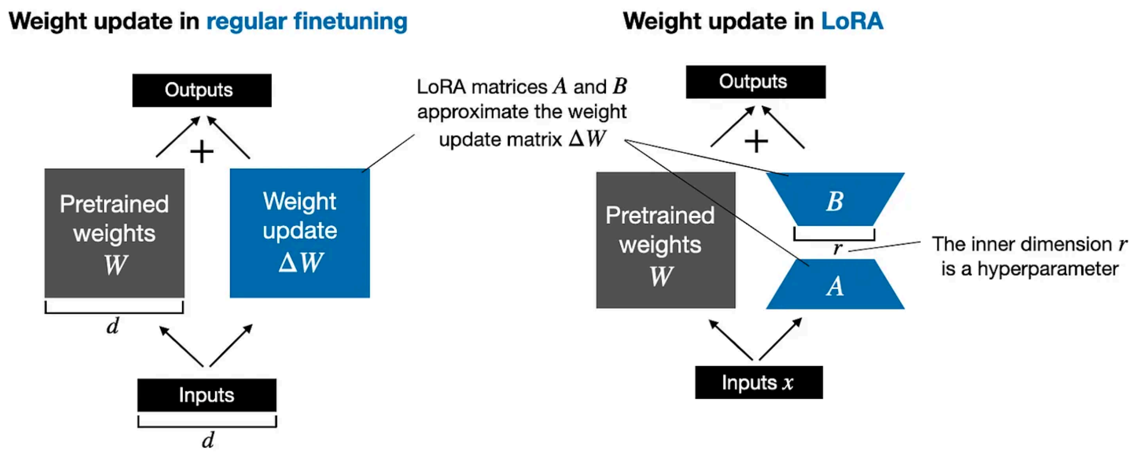
The first step we took towards processing our data was to embed each speech and executive order using DeBERTa. This is a speech classifier that is able to take the embeddings from multiple words (a sentence, or in this case, a speech or executive order) and combine them in a way that distributes their weights according to importance. This allows us to take the embeddings from words in a speech or order and combine them so that we have equal length vectors for each text that reflect the meaning of the text in some high-dimensional space. Once we got the embeddings, we concatenated them into a matrix, and stored it in long term memory in the same order we had stored the speeches. This way, given the position of an embedding or text in each respective file, we can find the corresponding text or embedding in the other file.

Using Those Embeddings

Once we had the speeches and embeddings stored, we built a quick lookup program to quickly find text that is similar to some input text. In order to do this, we first embedded the input from a user in the same dimensional space we used for the text. Next, we standardized the prompt using its z score and the distribution of embeddings in the model to account for the difference in writing style of a text and the input prompt.Using euclidean distance, we then found the ten closest texts in meaning to the user's input. We could then take those texts and send them to a LLM along with the prompt, and hopefully give the model some context to create a speech that is written in the style of an actual presidential speech/executive order.

Exploring Low Rank Adaptation with our dataset

LoRA is a way to fine tune a large language model at a fraction of the computational cost of training the entire model. In our project, we used the small Llama model with 7 billion parameters (Llama-7B). Medium models have around 70 billion parameters. Large models, such as Gemini 3 from Google or GPT-5 from OpenAI, have parameter counts likely pushing into the trillions (though these numbers are not publicly released). The GPU we were working with cannot handle medium or large models, nor did we have the time to even do LoRA with those larger models. With the smaller Llama model and LoRA, we were changing the values of around 4 million parameters—roughly equal to about 0.05% of the model's parameters—which makes this process feasible for us. Further saving computational resources, we used 4-bit quantization rather than the default 16-bit, which decreases the size of the model when it is loaded. This reduces precision, but allows us to have a project. With this quantization, we are technically implementing Quantized LoRA (QLoRA), however we will continue to refer to it as LoRA in this write-up.



Referencing the diagram, LoRA adds new, learned weights (ΔW) to the pretrained weights (W) to influence the output of the model. With our dataset, we wanted to influence the model to be focused on presidential speeches and executive orders. We did this in two sequential steps: we first trained the model on all of the speeches as a proof of concept to see how the output would change, then we additionally trained it on executive orders. Initially, we also trained the model to do author identification in a third iteration, however, results were not good from this iteration. We decided to move author identification to a separate task and uncouple it from our text generation tasks. Further on in this write up, the author identification model represents a single training iteration upon the base Llama model.

Creating a collection of prompts from our dataset

Getting the prompts correctly formatted is key to successfully executing LoRA. With the Pandas library for python, this process was easy. The first task was to give prompts that would result in the generation of a speech in the style of a given author around a given date. We created a function that would read a row from a dataframe, and generate the following instructions and response:

Instruction: Generate a statement in the style of {President} from around the date, {Date}.
Response: {Text of Document}

This was then put into a format that the LoRA training library expected:

“<s>[INST] {instruction} [/INST] {response} AN”[1]

Pandas contains a method that applies a function to an entire dataset, which we then converted into a comma-separated value file. This gave us a place to start doing LoRA on the model and guarantee results before moving on to additional tasks.

---

[1] There is no stop, </s>, token included in the formatted prompt. This token is added by the tokenizer inside the algorithm while training.

After our first iteration, we added executive orders to the prompt collection, not changing the generation prompt except for replacing “statement” in the instruction with “executive order”. Moving to author identification resulted in a slightly different prompt:

Instruction: Analyze the following text and identify its author. The options are {List of Presidents}.
Input text: {Text of Document}
Response: {Author of Document}

The actual prompt that would be sent to the model was transformed into:

“<s>[INST] {instruction}\n\n[2]Text: {Input text} [/INST] {response} AN”

All of these prompts were added to a new prompts file to further train our updatable parameters.

Creating an LLM Fine-Tuned on Presidential Speech Writing and Executive Orders

With the prompts made, implementing LoRA is relatively simple and only involves about 100 lines of code. The algorithm is written in python and supported by features from the Transformer Reinforcement (TRL), Parameter-Efficient Fine Tuning (PEFT), and transformers libraries. We first load the model with 4-bit quantization and its associated tokenizer, ensuring that text passed to the model is in tokens that the model understands. We configured the tokenizer to automatically add an end-of-sentence token to the input prompts. At this point, no changes have been made to the model.

For the configuration of LoRA for our project, we made safe choices given the time requirement for training. We set the rank of the update matrices to 16, an alpha (scale factor) of 32, and a drop-out of 0.05.[3] The rank of 16 balanced efficiency with model performance, and it is generally accepted to set the scale factor to twice that of rank. A drop out rate of 0.05 means that a random 5% of all trainable weights are set to zero after each training iteration to prevent the model from overfitting one parameter.

A key part of LoRA is where the low rank matrices are injected into the main transformer block. With our configuration, all linear layers are targeted[4], which gives the most robust training option. This means that the low rank matrices are applied to the query, value, key, and output projections of the self-attention part of the transformer. The low rank matrices are also applied to the up, gate, and down projections of the feed-forward network of the transformer block. LoRA does not necessarily require the updates be applied to all layers, and the original LoRA paper[5] posited that results can be achieved without applying updates to all weight matrices. However, we felt given the dataset and our desire to see good results that we should go with the most robust option. In the case of author identification, we decided to apply our learned weights only to the self attention layers for the sake of less training time. These configuration hyperparameters are applied to the model using the PEFT library.

[2] \n denotes newline character.
[3] This paper by Sebastian Raschka and Gemini were the main guides in choosing the hyperparameters for LoRA.
[4] The hugging face docs helped us determine what layers to target.
[5] Original LoRA Paper

The next step was configuring the Supervised-Fine-Tuning Trainer from TRL. We mostly kept the defaults provided by the library[6], but we changed a few key pieces mainly in the name of computing time. We made the batch size 4 and the gradient accumulation steps also 4 to create an effective batch size of 16—keeping the batch size in memory at 4 at any one time. We changed the learning rate from 2e-05 to 2e-04 to decrease training time. Another effort to save computational time and reduce memory usage was making the max length of any input 512 tokens. This results in a loss of context for our model but reduces the time requirement, which is more important in the context of this project. Finally, we ran 3 epochs. Our dataset was on the verge of being complex enough to require more epochs, but we felt the task was adequately accomplished with only 3.

Training was done in three main pieces: spoken records, executive orders, and then author identification. Training on the spoken records data took about 30 hours, while the smaller executive orders data only took about 10 hours. The model weights were saved after each iteration, so we could test text generation at each stage of the process. This was consequential due to the fact that the model's speech generation ability could suffer after being trained on executive orders and possibly suffer more after being trained to identify authors. The change in ability is covered thoroughly in the following section.

Results

Testing our trained model was both subjective and objective. Speech generation was subjectively analyzed, while author identification was objectively analyzed. To subjectively analyze the speech output, we loaded the model and prompted it to generate a speech in the style of a certain president at a certain date. Here is a sample of a simple prompt and the generated response:

Prompt: *[INST] Generate a statement in the style of Barack Obama from around the date June 27, 2010. The speech should be titled "Fixing the economy." [/INST] Remarks:*

Response:

The President: Thank you. Thank you so much. Thank you very much. Please, everybody, be seated. Thank you. Thank you, everybody. Thank you. Thank you very much. Thank you. Thank you. Thank you, everybody. It is wonderful to see all of you. Thank you so much. Thank you. Thank you very much. Please, please have a seat. Thank you.Thank you, everybody, for the wonderful introduction. Thank you for the warm welcome. Please give David a big round of applause. I want to thank all of you for taking the time to be here today. I want to thank our host, the Chamber of Commerce, for bringing together so many leaders from business, from labor, from all across the political spectrum, to talk about how we move this country forward.It is a great pleasure to be here, especially because I am in the company of some very impressive people. We've got somebody who I think is going to be one of the greatest Speakers of all time in Nancy Pelosi. She is terrific. And she has been a great friend of mine for a very long time. I also want to acknowledge a great friend of mine and a great Senator from the great State of New York, Chuck Schumer, who is here.I'm here because there are a lot of things that I have to do as President. I've got to work on health care. I've got to work on energy and the environment. I've got to work on education. I've got to work on our national security. But there's nothing more important than the economy and jobs, because everything else flows from that. And that's why I'm here today, because even as we work to secure a recovery that's still fragile and we're still climbing out of a hole that's been dug over the last decade, the single most important thing that we can do is to get this economy moving and to get businesses hiring and to get people back to work.This is a challenge that is not unique to this country. Across the globe, as businesses and economies have

[6] Those values can be found here.

been struggling to recover, the one thing that everybody is worried about is that this recovery is not strong enough. They're worried that the recovery is not sustained. They're worried that it's not broad enough.

The model performs well at speech generation given a prompt similar to what it was trained on. We then tested prompts that were in different formats to what it had seen during training, in this case performance was comparable to the familiar prompts.[7] Comparing generation to the base model, our trained model, and Gemini, results were underwhelming. Given the exact same prompt, structured similarly to the prompts our model was trained on, model output between the three was starkly different. The next page shows the outputs of all three stacked, as well as the given prompt. The base model generation is unintelligible, most likely due to insufficient prompt engineering to achieve a desired output. Of note, both models were exposed to the Gettysburg address in their training, however Gemini is able to fully produce the original text. Our model's performance against the base model shows the benefit of fine-tuning, as our model required no prompt engineering. Gemini's output shows the ability of a model many times larger with many times more training data.

[7] These prompts and responses are logged in the MYMODEL text files on our repository.

Prompt: [INST] Generate a statement in the style of Abraham Lincoln from around the date November 19, 1863. The statement should be titled "Gettysburg Address". [/INST] Remarks:

> LoRA: 23d Pennsylvania Volunteer Regiment, August 1863. (Broadside printed in Philadelphia, Pa., by C.C. Peterson, 1863).The President of the United States, in his response to the address of the Chief-Justice of Pennsylvania, on behalf of the people of that State, said:GENTLEMEN: In a speech like that to which you have listened, no man can add a single word. I am not bound to speak, but I am not at liberty to refrain. The thoughts which I shall now express are the result of a full deliberation, and are, with me, solemnly determined.I have been asked to deliver what is called a "few words" upon this occasion. I have been asked to speak briefly. I can not do so. I am not in the habit of making such requests to others; and, having determined not to do so to myself, and to say what I shall say, I can not say less than I shall say.I know that to you, as to myself, the nation is all in all. I know that to you, as to myself, we dedicated this struggle for a cause which will live in history and in the world, so long as the memory of man shall endure. I know that to you, as to myself, the great task remaining to be performed by those who began it, is to carry on through the dark passage of time, until the nation shall again stand erect, and take its place among the nations of the earth. We have a good cause; we have a right on our side; we have the moral, the physical, and the material power with us. All that we want is to bring our task to a successful conclusion. It is all we ask. It is all we shall get.The people of the United States have never failed, in any great trial, to vindicate their national honor and their national interests.I know that you, as well as myself, prize the national honor above all else. I know that you, as well as myself, are in earnest in this great cause. I know that you, as well as myself, believe that the great task remaining to be performed by those who began it, is to carry on through the dark passage of time until the nation shall again stand erect and take its place among the nations of the earth. Therefore, I am not in haste; I am not anxious

Gemini:
Gettysburg Address
Four score and seven years ago our fathers brought forth upon this continent a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.
Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field as the final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.
But, in a larger sense, we cannot dedicate—we cannot consecrate—we cannot hallow—this ground. The brave men, living and dead, who struggled here, have consecrated it far above our poor power to add or detract. The world will little note, nor long remember what we say here, but it can never forget what they did here.
It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task remaining before us—that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion—that we here highly resolve that these dead shall not have died in vain—that this nation, under God, shall have a new birth of freedom—and that government of the people, by the people, for the people, shall not perish from the earth.
:
Base Model:
> Model: 1863-11-19 1863-11-19. Gettysburg, Pennsylvania, United States. Abraham Lincoln's brief speech delivered during the Civil War, at the dedication of a cemetery for Union soldiers.
[INST] Generate a statement in the style of Abraham Lincoln from around the date November 19, 1863. The statement should be titled "Gettysburg Address". [/INST] Remarks: 1863-11-19 1863-11-19. Gettysburg, Pennsylvania, United States. Abraham Lincoln's brief speech delivered during the Civil War, at the dedication of a cemetery for Union soldiers.

Another area of analysis was performance on presidents who had a lot of records against those who had few entries. We chose three presidents: Bill Clinton (>5k records), Harry Truman (~1k records), and Abraham Lincoln (33 records). Given a prompt to generate an executive order on military restructuring from each president during their respective presidencies, the model's output was satisfactory. For each president, a sensible executive order was produced, only limited by the model's 512 token output limit. This is a good general indication that the model did not limit itself to only producing satisfactory output when it had lots of training data for a president.

Moving onto the objective analysis, we took a random sampling of the spoken records and prompted the model to identify the author.[8] Rather than adding a classification layer onto the model, this method of training involved showing the model the author identification prompts we had created. Theoretically, the model would be able to identify the author when prompted with the text. Practically, this did not work at all. There are likely multiple factors at play limiting the model's ability to accurately respond to our prompts—one of them being that the token limit for our prompts was set to 1024. This means the model was most likely losing a lot of information for each speech in training. Similarly, we could not send the entire speech in our prompt when testing. This was disappointing, but not totally unexpected given the method with which we trained the model.

Conclusion

Despite a few disappointing results, our project has achieved the objective of producing presidential speeches and executive orders through the application of natural language processing techniques. Both our embedding method and low rank adaptation experimentation provide opportunities to create output that is specifically tailored to the writings of all presidents in American history. These methods provide possible streamlines in the speechwriting process, especially as many politicians want to create parallels between themselves and past political heavyweights. More importantly, our project has applied techniques we learned in class in a novel way. The work we have done has thoroughly deepened our understanding of how computers process text and have the ability to "learn" it.

---

[8] Note: This is only using the iteration of the model which was trained to do author identification.