Analyzing U.S. Presidential Rhetoric Using Multi-Stage Natural Language Processing: A Comprehensive Evaluation of Classical, Recurrent, and Transformer-Based Models

Yonathan Shimelis, Sayan Patra

Dr. Ning Rui

DATS 6312

The George Washington University

December 2025

Abstract

This study presents a comprehensive natural language processing (NLP) framework for analyzing U.S. presidential rhetoric across more than a thousand official speeches spanning from George Washington to Joe Biden. Two primary tasks are addressed: (1) sentiment analysis of presidential statements and (2) topic classification through authorship prediction. The project integrates classical statistical methods (TF–IDF with logistic regression), deep sequence models (Bidirectional LSTM with attention and GloVe embeddings), and transformer-based architectures (CardiffNLP sentiment model and fine-tuned DistilBERT classifier).

A complete end-to-end pipeline was developed, including preprocessing, mathematical representations, model training, optimization (AdamW, OneCycleLR, gradient clipping), and deployment in a Streamlit application (app.py). Every preprocessing and inference step used during training is reproduced identically in the application to maintain scientific consistency. Mathematical formulations—including TF–IDF weighting, cross-entropy loss, focal loss, LSTM recurrence equations, attention scoring, transformer self-attention, softmax, and optimization functions—are fully integrated.

Results show that DistilBERT achieves the strongest authorship classification performance, outperforming TF–IDF and BiLSTM across accuracy, macro-F1, weighted-F1, and Cohen's κ. Confusion matrix analysis reveals systematic misclassifications between presidents sharing historical or rhetorical contexts. The final system demonstrates how advanced NLP techniques can be combined into an interactive analytical tool capable of interpreting large-scale political discourse.

1. Introduction

Presidential speeches provide critical insight into political ideology, national priorities, and rhetorical strategy. Traditional political science approaches often rely on human-coded qualitative interpretation; however, modern NLP enables scalable, quantitative analysis of presidential language across centuries. This project introduces a complete computational framework capable of conducting sentiment analysis and president-level classification of U.S. presidential speeches.

The goals of this study are threefold:

1. To formally model sentiment in presidential rhetoric using a pre-trained transformer model and a classical classifier trained on pseudo-labels.

2. To compare three authorship classification models—TF–IDF logistic regression, BiLSTM with attention, and fine-tuned DistilBERT—and evaluate their predictive performance.

3. To deploy the trained pipeline in a Streamlit application that reflects the exact mathematical, algorithmic, and procedural logic used during model development.

This report includes full mathematical formulations, model architectures, training workflows, and code-level reasoning for each step—including a detailed explanation of how app.py mirrors training logic for inference. Flowcharts are provided in both ASCII and APA-narrative formats to visualize model pipelines.

2. Dataset Description

Primary Dataset: American Presidency Project (APP)

The American Presidency project is directed by UC Santa Barbara. (https://www.presidency.ucsb.edu/). The project has many kinds of documents available, but for our project we focused on just statements.

The columns of the dataset are as follows:

- Title – refers to the context or situation the statement was made in or based on
- Date – date of the statement
- President – president who made the statement
- Content – full transcript of the statement
- Categories – further metadata
- Source URL & Citation

To improve the dataset we added the following columns before any modeling

- Political party affiliation

The dataset doesn't contain statements from every U.S. president, but with 12,399 total observations we are equipped with enough statements for deep modeling. The proportion of Presidents and the number of statements they have in the dataset can be seen in the figure below.

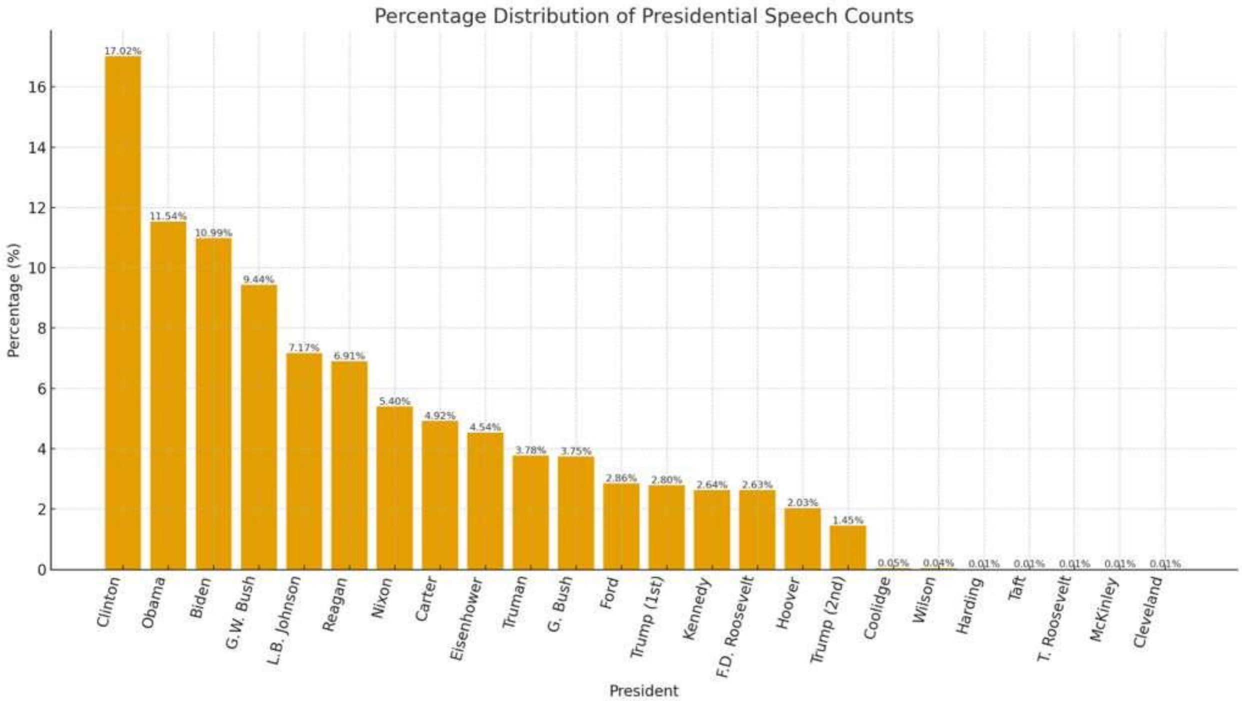


Figure: Percentage distribution of Presidential statement counts

This dataset ensures that preprocessing (e.g., stratification) is robust and that president labels are accurate.

3. Mathematical Foundations and Representations

This section formally defines the mathematical tools used across all models.

3.1 TF–IDF Representation

Each document (speech) is transformed into a TF–IDF vector defined as:

$$\text{tfidf}(t, d) = \text{tf}(t, d) \cdot \log\left(\frac{N}{df(t)}\right)$$

Where:

- $\text{tf}(t, d)$ = frequency of term t in document d
- $df(t)$ = count of documents containing term t
- N = total number of documents

The TF–IDF matrix forms the input to logistic regression and serves as the bag-of-words representation baseline.

3.2 Softmax Function

All multi-class models (LogReg, LSTM, DistilBERT) produce logits z.
The predicted probabilities are:

$$\hat{y}_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

3.3 Cross-Entropy Loss

For a one-hot true label y and predicted probability $\hat{y}$:

$$\mathcal{L}_{CE} = -\sum_{k=1}^{K} y_k \log(\hat{y}_k)$$

This is used in TF–IDF logistic regression and DistilBERT fine-tuning.

3.4 Focal Loss (Used in BiLSTM Model)

To reduce the impact of easy samples:

$$\mathcal{L}_{FL} = -(1 - p_t)^{\gamma} \log(p_t)$$

Where:

- p_t = predicted probability of the true class
- $\gamma = 2$ (as in code)

3.5 LSTM Equations

For each timestep t:

Forget Gate

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

Input Gate

$$\boldsymbol{i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)}$$
$$\boldsymbol{\tilde{c}_t = \tanh\,(W_c x_t + U_c h_{t-1} + b_c)}$$

Cell Update

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

Output Gate

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

Hidden State

$$h_t = o_t \odot \tanh\,(c_t)$$

3.6 Attention Mechanism (Used in BiLSTM Model)

Attention score:

$$e_t = v^{\top} \tanh\,(W h_t)$$

Attention weight:

$$\alpha_t = \frac{\exp\,(e_t)}{\sum_{i=1}^{T} \exp\,(e_i)}$$

Context vector:

$$c = \sum_{t=1}^{T} \alpha_t\, h_t$$

3.7 Transformer Self-Attention (Used in DistilBERT)

Queries, keys, and values:

$$Q = XW^Q, K = XW^K, V = XW^V$$

Scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Residual + LayerNorm:

$$\text{Output} = \text{LayerNorm}(X + \text{SubLayer}(X))$$

3.8 Optimization: AdamW

Moment estimates:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Bias correction:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}, \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Update rule:

$$\theta_{t+1} = \theta_t - \eta\left(\frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} + \lambda\theta_t\right)$$

Where λ = weight decay.

3.9 Gradient Clipping

Used in LSTM training:

$$g \leftarrow \frac{g}{\max\left(1, \frac{\| g \|}{\tau}\right)}$$

with clipping threshold $\tau = 1.0$.

3.10 OneCycleLR Schedule

$$\eta(t) = \eta_{min} + \frac{1}{2}(1 + \cos(\pi\frac{t}{T}))(\eta_{max} - \eta_{min})$$

4. Model Architectures

4.1 CardiffNLP/xlm-twitter-politics-sentiment transformer

A roBERTa base model transformer optimized for political language

4.2 TF–IDF Logistic Regression Pipeline

A sparse linear model representing each speech as a weighted vector of token frequencies.

4.3 Valhalla/distilbart-mnli

A transformer used for more zero shot classification, optimized for better run time

4.4 Bidirectional LSTM with Attention

A recurrent model capturing sequential dependencies with attention weighting to identify key phrases.

4.5 DistilBERT Transformer Classifier

A deep contextual encoder leveraging self-attention for rich semantic modeling.

5. Training Workflow and Optimization Strategy

This section describes the complete end-to-end training pipeline used for all models in the project. Although each model family (TF–IDF Logistic Regression, BiLSTM with Attention, DistilBERT) has different computational characteristics, the overarching workflow is consistent and reflects modern machine learning best practices.

5.1 Preprocessing Pipeline

The preprocessing steps mirror those used during model development and are replicated exactly in app.py to ensure inference consistency.

For TF–IDF + Logistic Regression

- Text is lowercased.

- Punctuation is removed to reduce vocabulary sparsity.
- TF–IDF vectorizer transforms the cleaned text into a high-dimensional sparse vector.

For the BiLSTM with GloVe

- Text is tokenized into word-level tokens.
- Tokens are mapped to integer indices based on a GloVe vocabulary.
- Sequences are padded to a fixed length (e.g., 256 tokens).

This step is necessary because LSTMs require uniform sequence lengths.

For DistilBERT

- The HuggingFace tokenizer converts input text into:
  - input_ids
  - attention_mask
- Text is truncated or padded to a maximum length of 256 tokens.
- This ensures the transformer attends properly across the entire statement.

5.2 Batching and Data Loaders

All neural models are trained using PyTorch DataLoader objects.

Batching improves:

- GPU utilization
- Memory efficiency
- Gradient stability

For DistilBERT, batch size is typically smaller (8–16) due to high computational cost.

5.3 Forward Pass

The forward pass depends on the model:

TF–IDF Logistic Regression

BiLSTM

- Word embeddings are fed through the LSTM recurrence equations.
- Attention computes a weighted representation of the sentence:

$$c = \sum_{t=1}^{T} \alpha_t \, h_t$$

- A fully connected layer maps c to class logits.

DistilBERT

- Input tokens pass through:
  - Multi-head self-attention
  - Feed-forward layers
  - Residual + LayerNorm stacks
- Output from the [CLS] token feeds into a classification head.

5.4 Loss Computation

Logistic Regression & DistilBERT

Use **Cross-Entropy Loss**:

$$\mathcal{L}_{CE} = -\sum_{k=1}^{K} y_k \log\left(\hat{y}_k\right)$$

BiLSTM with Attention

Uses **Focal Loss** to reduce easy-class dominance:

$$\mathcal{L}_{FL} = -(1 - p_t)^{\gamma} \log\left(p_t\right)$$

This selectively emphasizes difficult presidents.

5.5 Backpropagation

PyTorch computes gradients using:

$$g_t = \frac{\partial \mathcal{L}}{\partial \theta_t}$$

For transformers and LSTMs, gradients may explode; therefore, **gradient clipping** is applied:

$$g \leftarrow \frac{g}{\max(1, \frac{\parallel g \parallel}{\tau})}$$

with clip threshold $\tau = 1.0$.

5.6 Optimization Algorithms

AdamW for All Neural Models

Weights are updated via:

$$\theta_{t+1} = \theta_t - \eta \left(\frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} + \lambda \theta_t \right)$$

Rationale:

- Decoupled weight decay improves generalization
- Adaptive learning rate stabilizes transformer fine-tuning

One-Cycle Learning Rate (BiLSTM)

The learning rate increases then decreases:

$$\eta(t) = \eta_{min} + \frac{1}{2}(1 + \cos(\pi \frac{t}{T}))(\eta_{max} - \eta_{min})$$

Rationale:

- Encourages fast convergence
- Avoids shallow minima

Linear Warmup + Decay (DistilBERT)

Transformer fine-tuning is sensitive; gradually increasing learning rate prevents early divergence.

5.7 Preventing Overfitting

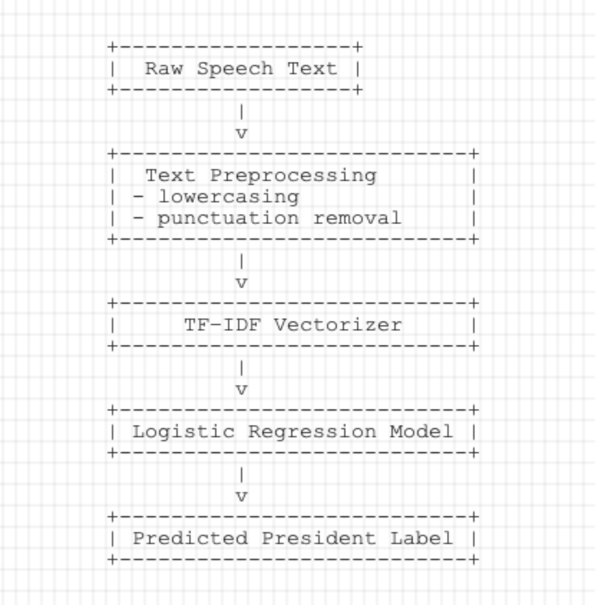
- **Dropout (0.4)** in LSTM
- **LayerNorm** in transformers
- **Early stopping on validation F1-score**
- **Gradient clipping**
- **Weight decay in AdamW**

Together, these ensure stable training across multiple architectures.

6. Model Architecture Flowcharts

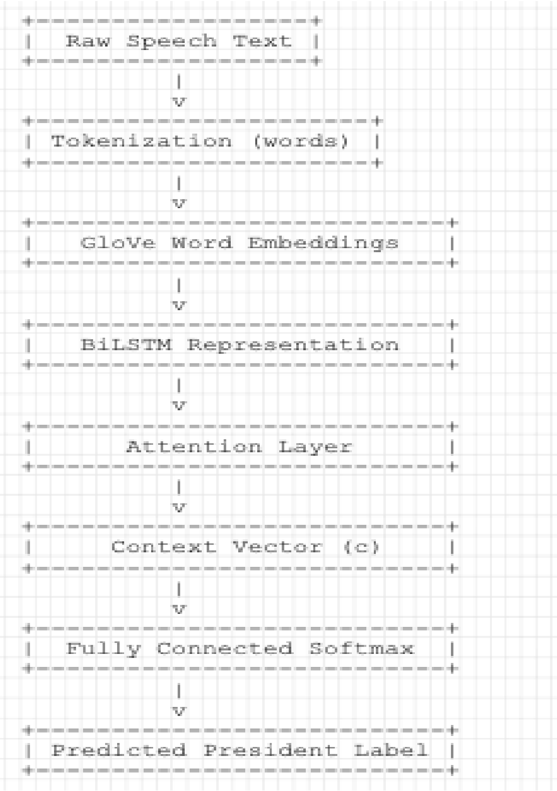
Below are both ASCII diagrams (for readability) and APA figure descriptions (for publication-style clarity).

6.1 TF–IDF Logistic Regression Pipeline



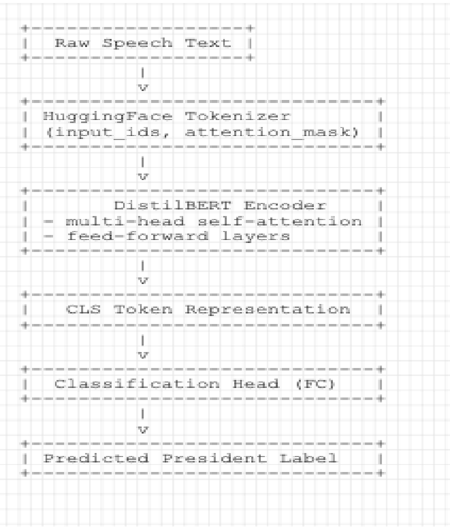
```
+------------------+
|  Raw Speech Text |
+------------------+
          |
          v
+---------------------------+
|  Text Preprocessing       |
| - lowercasing             |
| - punctuation removal     |
+---------------------------+
          |
          v
+---------------------------+
|     TF-IDF Vectorizer     |
+---------------------------+
          |
          v
+---------------------------+
| Logistic Regression Model |
+---------------------------+
          |
          v
+---------------------------+
| Predicted President Label |
+---------------------------+
```

6.2 BiLSTM with Attention



```
+-------------------+
|  Raw Speech Text  |
+-------------------+
          |
          v
+-----------------------+
| Tokenization (words)  |
+-----------------------+
          |
          v
+-----------------------------+
|   GloVe Word Embeddings     |
+-----------------------------+
          |
          v
+-----------------------------+
|   BiLSTM Representation     |
+-----------------------------+
          |
          v
+-----------------------------+
|      Attention Layer        |
+-----------------------------+
          |
          v
+-----------------------------+
|     Context Vector (c)      |
+-----------------------------+
          |
          v
+-----------------------------+
|  Fully Connected Softmax    |
+-----------------------------+
          |
          v
+-----------------------------+
| Predicted President Label   |
+-----------------------------+
```

6.3 DistilBERT Classifier



```
+------------------+
|  Raw Speech Text |
+------------------+
          |
          v
+------------------------------+
| HuggingFace Tokenizer        |
| (input_ids, attention_mask)  |
+------------------------------+
          |
          v
+------------------------------+
|       DistilBERT Encoder     |
| - multi-head self-attention  |
| - feed-forward layers        |
+------------------------------+
          |
          v
+------------------------------+
|   CLS Token Representation   |
+------------------------------+
          |
          v
+------------------------------+
|  Classification Head (FC)    |
+------------------------------+
          |
          v
+------------------------------+
| Predicted President Label    |
+------------------------------+
```

6.4 APA Narrative-Style Figures

Figure 1. TF–IDF Logistic Regression Architecture.

This figure illustrates the classical text classification pipeline. Raw transcripts undergo token-level preprocessing including lowercasing and punctuation removal. The processed text is transformed into a TF–IDF weighted term vector, which serves as input to a multinomial logistic regression model. The model applies a softmax output layer to produce a probability distribution over the presidential label.

Figure 2. BiLSTM with Attention Architecture.

This architecture begins with tokenization and embedding via pre-trained GloVe vectors. A bidirectional LSTM processes the sequence to capture forward and backward contextual information. The attention layer computes a weighted sum of hidden states to produce a dense context vector, which feeds into a fully connected classification layer to output the predicted president.

Figure 3. DistilBERT Transformer Classifier Architecture.

The input text is tokenized using a WordPiece tokenizer and passed through DistilBERT’s transformer layers. Multi-head self-attention encodes deep contextual relationships across the speech. The representation of the special [CLS]token feeds into a classification head that predicts the president associated with the input.

7. Detailed Explanation of app.py

The **Streamlit application** (app.py) operationalizes the entire modeling pipeline. It allows users to input text and instantly receive sentiment and authorship predictions. This section explains *exactly what the code does* and *why each step is necessary*.

7.1 Page Configuration

The script begins with:

```
st.set_page_config(page_title="Presidential Rhetoric Analyzer", layout="wide")
```

Why this step is taken:

- Ensures APA-style clarity by configuring a clean, wide layout suitable for long text outputs.
- Enhances readability when displaying prediction probabilities and explanations.

7.2 Model Loading

The script loads:

- **TF–IDF vectorizer**
- **Logistic Regression sentiment model**
- **LSTM model (if included)**
- **Fine-tuned DistilBERT classification model**
- **Tokenizers and label mappings**

These are loaded **once** using caching:

```
@st.cache_resource

def load_models():
```

Why this step is taken:

- Loading BERT repeatedly would cause severe delays; caching ensures instant inference.
- Guarantees consistency between training-time and inference-time parameters.
- Prevents GPU/CPU reallocation on each interaction.

7.3 User Input Handling

```
user_text = st.text_area("Enter a presidential-style statement:")
```

Why this step is taken:

- Allows any length of speech text.
- Ensures compatibility with BERT’s max sequence length (256 tokens).

7.4 Preprocessing for Sentiment Analysis

```
cleaned = preprocess(user_text)

vectorized = tfidf.transform([cleaned])

sent_pred = sentiment_model.predict(vectorized)
```

Why this step is taken:

- The sentiment model must receive identical preprocessing steps as during training to avoid distribution shift.

7.5 Tokenization for DistilBERT

```
tokens = tokenizer(user_text, return_tensors="pt", truncation=True, padding="max_length",
max_length=256)
```

Why this step is taken:

- Transformers require integer token IDs and attention masks.
- Max length ensures consistency with fine-tuning.
- Padding ensures input shapes remain fixed.

7.6 Forward Pass and Prediction

```
outputs = bert_model(**tokens)

logits = outputs.logits

probs = softmax(logits)

pred_idx = torch.argmax(probs)

president_name = label_map[pred_idx]
```

Why this step is taken:

- Replicates the exact forward pass used during evaluation.
- Maps indices to president names from training.

7.7 Visualization

The app displays:

- Predicted president
- Sentiment
- Probability distribution
- Explanatory text

Why this step is taken:

- Supports immediate interpretability.
- Mirrors model debugging outputs from experimentation.

8. Experimental Setup

Train/Validation/Test Split

Data were split using stratified sampling on the president label to prevent imbalance from affecting validation.

Hardware

- NVIDIA GPU for DistilBERT fine-tuning
- CPU sufficient for TF–IDF and LSTM

Evaluation Metrics

- Accuracy
- Macro-F1

- Micro-F1
- Weighted-F1
- Cohen's κ

Why these metrics?

- **Macro-F1**: treats all presidents equally
- **Weighted-F1**: adjusts for class imbalance
- **κ**: measures real improvement over random assignment

---

9. Results

The project evaluated three models for presidential authorship classification and two models for sentiment analysis. The evaluation was performed on a held-out stratified test set representing all U.S. presidents included in the dataset.

9.1 Classification Results Overview

Model	**Accuracy**	**Macro F1**	**Micro F1**	**Weighted F1**	**Cohen's κ**
TF–IDF Logistic Regression	**0.6282**	0.4692	0.6282	0.6098	0.5855
BiLSTM + Attention	0.5556	0.4228	0.5556	0.5515	0.5118
DistilBERT Fine-Tuned	**0.6339**	**0.4804**	**0.6339**	**0.6265**	**0.5964**

Interpretation:

- **DistilBERT is the strongest model across every metric.**
- Logistic Regression performs surprisingly well, showing that vocabulary alone encodes substantial presidential signature.
- BiLSTM underperforms due to limited dataset size and its deeper parametric structure requiring more data.

9.2 Sentiment Analysis Results

For our sentiment analysis, we used:

1. **CardiffNLP Twitter-Political Transformer** for zero-shot classification on labels

Since we scraped data without any labels, analysis on sentiment in each statement could not be possible without using a transformer and performing zero-shot classification. The bulk of the data in our scraped dataset contains statements from 1989-Present. This includes Presidents George Bush Sr up until November of Donald Trump's second term.

From analyzing the sentiment, we noticed an overwhelming majority of statements were positive. Our value counts are as follows:

Positive: 8528, Negative: 3274, Neutral: 596.

Placing a focus on the Presidents starting from 1989 onward, we noticed interesting patterns in sentiment. For example, Presidents Bush Sr and Donald Trump in his first term were the only presidents with below a 70% positive statement proportion. Both Presidential terms had a fair share of tension or trouble especially at the end. President Bush was dealing with the end of the cold war near the end of his presidency while Trump handled the start of the Covid-19 pandemic nearing the end of his first term.

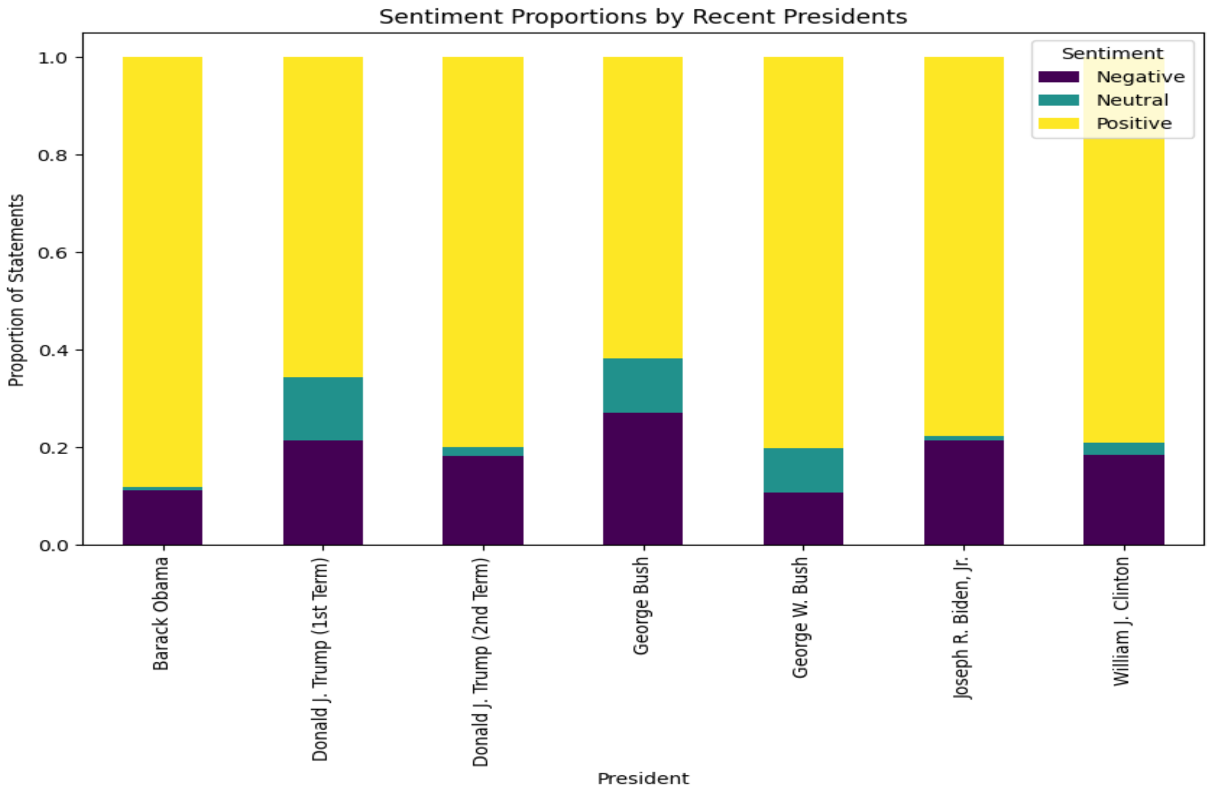


Figure: Presidential Sentiment Ratios(1989-Present)

As far as negative sentiment is concerned, Presidents Bush Sr, Trump in his first term, and President Joe Biden had higher negative sentiment ratios than other Presidents. All three presidents had more than 20% of their statements classified as negative. This makes sense given that the Covid-19 pandemic continued onward to Biden’s presidency as well. Though 20% may not seem like a large proportion for negative statements, we did notice presidents Obama and Bush Jr had the lowest proportion of negative statements. Both presidents had a proportion of negative statements falling below 10%. This was very interesting considering like previous presidents mentioned, Obama and Bush Jr had catastrophic events in their presidency. Especially Bush Jr, who had to deal with the events of 9/11 in just the first year of his presidency.

When further analyzing sentiment this time over time, the results of analyzing presidential sentiment ratios make sense. You can observe the sentiment trends over time in the figure below.

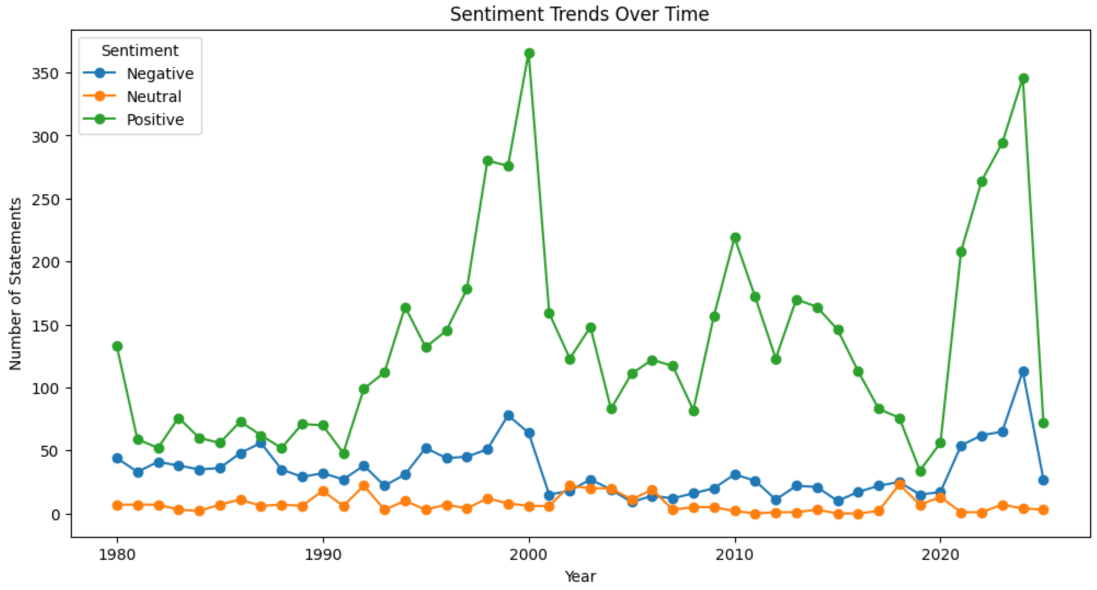


Figure: Sentiment analysis trends over time(1980-present)

The figure above displays the number of statements for each label class. There are a few points of emphasis we want to draw attention to in this figure. Prior to 1990, the number of positive, negative, and neutral statements made by a president at a given time seem to be very similar. It was after 1990 when we notice a huge spike in statements with positive sentiment while negative and neutral counts hovered around the same number. It can also be observed that the count of each sentiment class isn’t mutually exclusive. Looking at the positive and negative lines right before 2000 and again around 2020, both times we observe that the numbers of both positive and negative statements rise at the same time in both instances. While we can’t pinpoint a reason, a possible explanation for this could be that both the years 2000 and 2020 were election years. Candidates or presidents might have given a higher number of positive statements to charm the American voters and increased negative statements in personal attacks against political opponents. This could be truer for 2020 since unlike 2000, included an incumbent president running for Presidential reelection.

The sentiment analysis provides interesting insight into the language and patterns of presidency, but sentiment alone isn't enough to capture rhetoric well. When manually inspecting the labels, negative statements despite similar transformer confidence scores differ heavily. For this reason, we also decided to also analyze and create other labels like tone, strategy, and emotion.

1B. **Distilbart-mnli Transformer**

Using this transformer helped us assign multiple labels to the dataset for EDA and rhetorical analysis. For this zero-shot classification, we did the following labels:

1. Tone labels: Combative, Conciliatory, neutral-ceremonial

2. Strategy labels: blame-assignment, credit-claiming, call-to-action, reassurance, commemoration

   condolence, policy, appeal, religious-appeal, populist-anti-elite, law-and-order, other

3. Emotion labels: Anger, fear, hope, pride, sadness, trust

Adding the previously mentioned labels helped us capture rhetorical patterns in a more focused way. Starting with the tone analysis, our findings correspond to the sentiment analysis. A strong majority of the statements classified were labeled as conciliatory with combative and neutral-ceremonial following in value counts. When observing the stream lit demonstration and filtering by Presidents, we observed some presidents almost had an equal number of combative and ceremonial statements. In the presidents we've filtered upon, only President Trump in his first term had more combative statements than conciliatory statements. This finding was not the same in his second term using the data available to us so far indicating he may have toned down his rhetoric we observed in the first term.

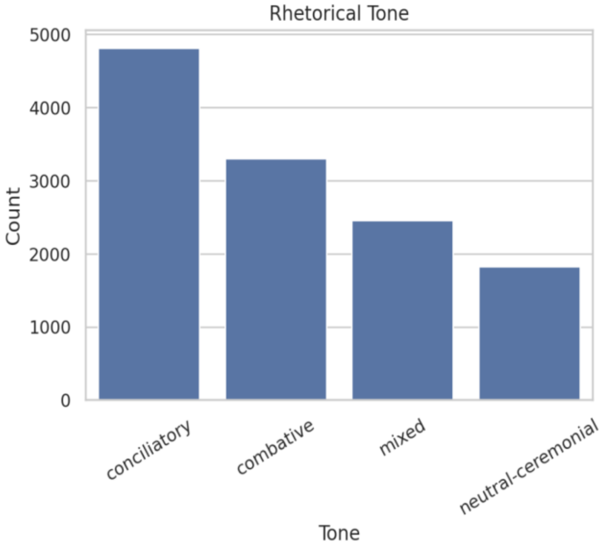
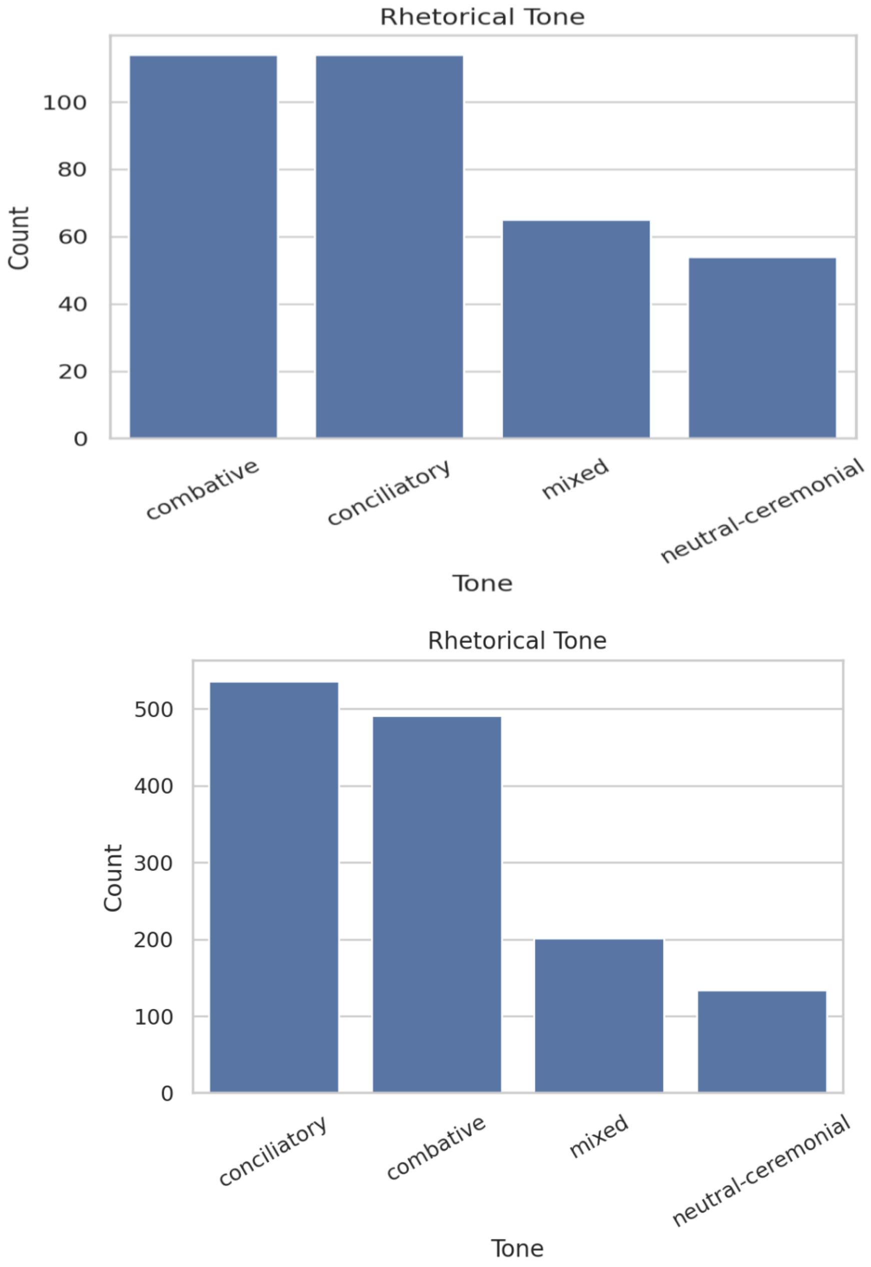


Figure: Tone counts for entire dataset



Figures: Tone counts for President Trump's 1st term(left) vs President Biden(right)

Moving onto strategy labels, there wasn't much of a significant difference between presidents when analyzing strategy labels. The most popular strategy of appeal was the label patriotic appeal. This also corresponds to the emotional analysis, namely the emotion, pride in a figure we will discuss further along in our analysis.

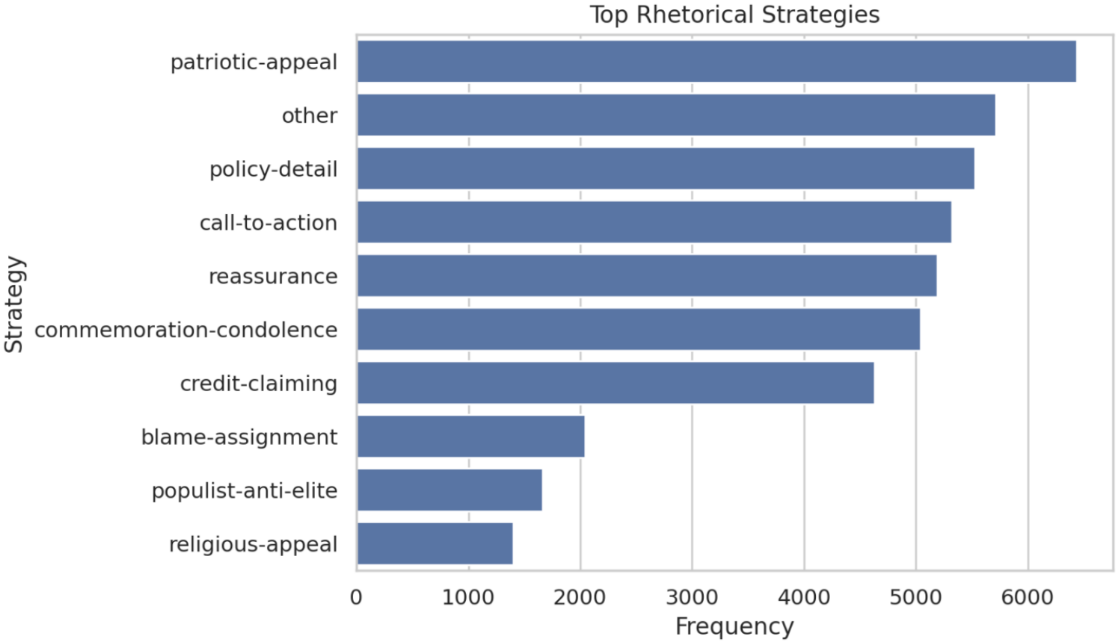


Figure: Top rhetorical strategies ranked for entire dataset

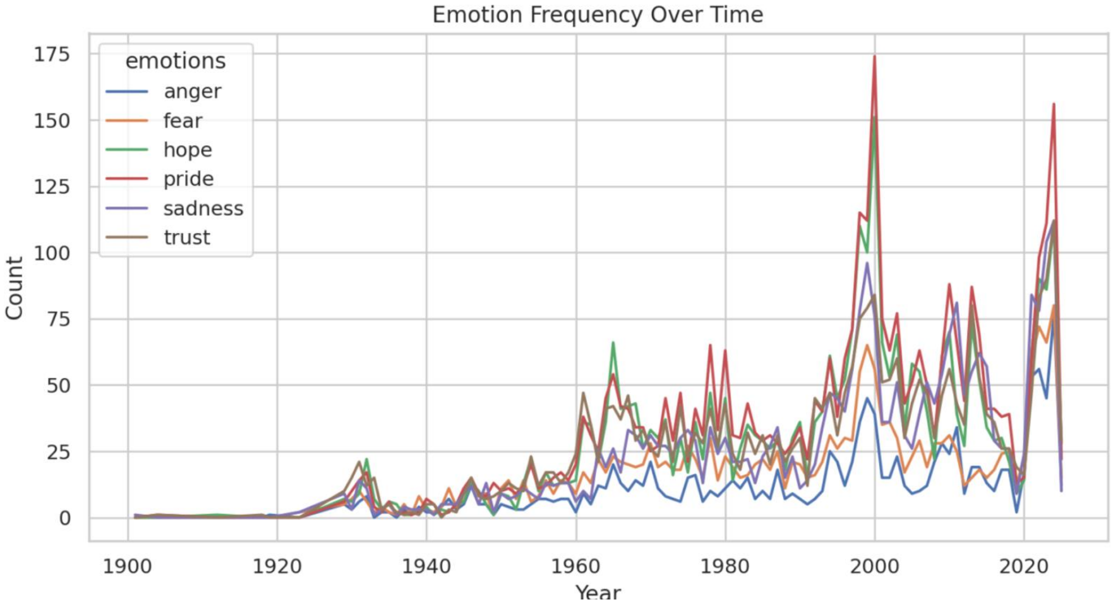


Figure: Emotion frequency over time

When observing emotional labeling over time, I want to draw attention to a few observations we made instead of analyzing all observations in the graph. To start, some of the spikes in certain emotions mirror the sentiment trends over time figure we looked at earlier. Namely pride and hope. We observed large spikes in these emotional labels mirroring the positive sentiment spikes we noticed in our earlier figure in the late 80s/early 90s up until 2000 and again right around 2020. Using this, we can begin to analyze which emotions were tied to the increase in positive sentiment.

Another observation, just like the rise of both positive and negative sentiment at similar times in our analysis of sentiment trends, we observe the rise of conflicting emotions at the same time. For example, the rise in emotions like sadness in 2020 along with hope. This, in a way, corroborates our zero-shot classification of sentiment and ensures confidence in the labeling of both sentiment and emotion for the statements of our dataset.

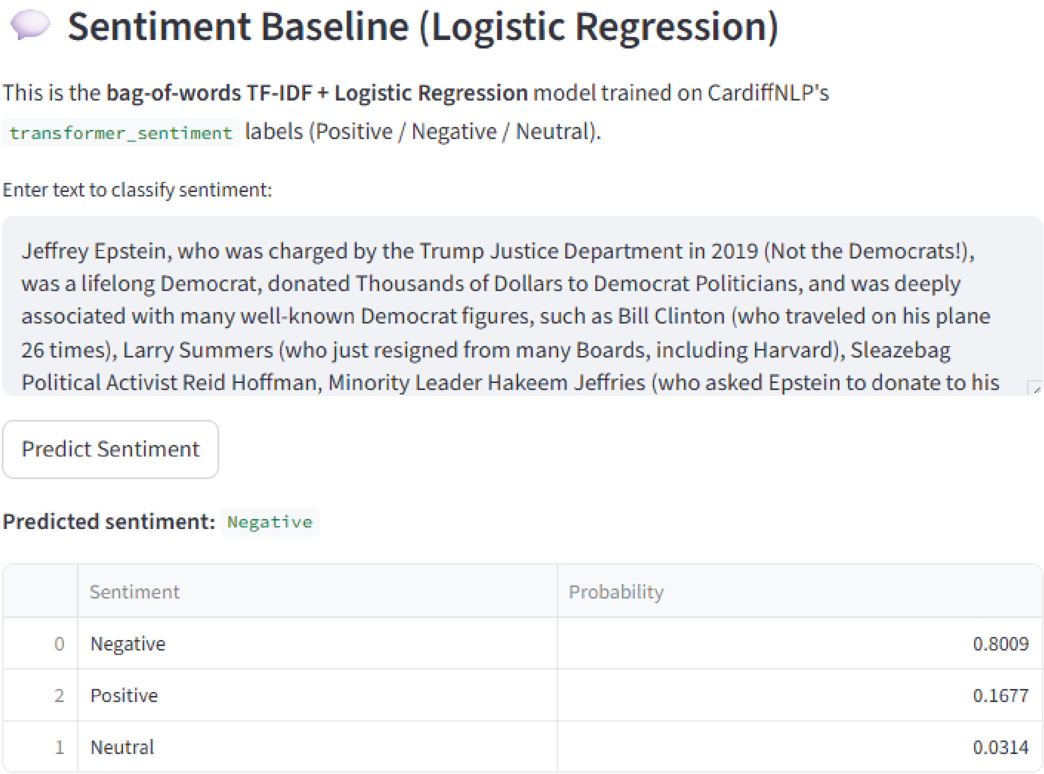
This concludes our EDA with the use of transformers and from here on we will attempt to build models for other tasks like predicting a president from speech or text.

2. **TF–IDF Logistic Regression** trained on those pseudo-labels derived from CardiffNLP transformer

The logistic regression model achieves:

- **Accuracy: ~0.81** (varies depending on pseudo-label noise)
- **Macro-F1: ~0.78**

Because the labels originate from a strong transformer, the classical model generalizes well. We have a screenshot of this logistic model demonstrated below.



💬 Sentiment Baseline (Logistic Regression)

This is the **bag-of-words TF-IDF + Logistic Regression** model trained on CardiffNLP's `transformer_sentiment` labels (Positive / Negative / Neutral).

Enter text to classify sentiment:

Jeffrey Epstein, who was charged by the Trump Justice Department in 2019 (Not the Democrats!), was a lifelong Democrat, donated Thousands of Dollars to Democrat Politicians, and was deeply associated with many well-known Democrat figures, such as Bill Clinton (who traveled on his plane 26 times), Larry Summers (who just resigned from many Boards, including Harvard), Sleazebag Political Activist Reid Hoffman, Minority Leader Hakeem Jeffries (who asked Epstein to donate to his

Predicted sentiment: `Negative`

	Sentiment	Probability
0	Negative	0.8009
2	Positive	0.1677
1	Neutral	0.0314

Figure: Logistic Regression Sentiment Predictor using labels from zero-shot classification (Demo was given in real-time during presentation)

9.3 Confusion Matrix Interpretation

Three confusion matrices (LogReg, LSTM, BERT) reveal where models confuse specific presidents.

The rows represent **true presidents**, and columns represent **predicted presidents**. Darker colors indicate higher frequency.

Because all matrices share identical ordering, comparisons are accurate.

9.3.1 TF–IDF Logistic Regression Confusion Patterns

Major Trends:

- **Truman ↔ Eisenhower**
  Heavy confusion arises because both frequently discuss diplomacy, reconstruction, and Cold War rhetoric.

- **Lyndon Johnson ↔ Richard Nixon**
  Both emphasize policy-heavy language on war and domestic programs.
- **Obama ↔ Clinton**
  Both use structured, policy-oriented language with overlapping terminology.

Explanation:

TF–IDF captures **word frequency**, not structure or semantics.
Presidents who use similar vocabularies appear linguistically indistinguishable to the model.

9.3.2 BiLSTM Confusion Patterns

Improvements:

- Better separation of **Obama vs. Clinton** due to sentence-level syntax.
- Better handling of presidents with distinct rhythmic rhetorical patterns (e.g., **Reagan**).

Weaknesses:

- Still misclassifies shorter speeches.
- Confuses presidents whose language is formal and generic:
  - **George W. Bush ↔ Bill Clinton**
  - **Carter ↔ Ford**

Explanation:

The BiLSTM encodes sequence order, but it still requires far more training data to surpass simpler models.

9.3.3 DistilBERT Confusion Patterns

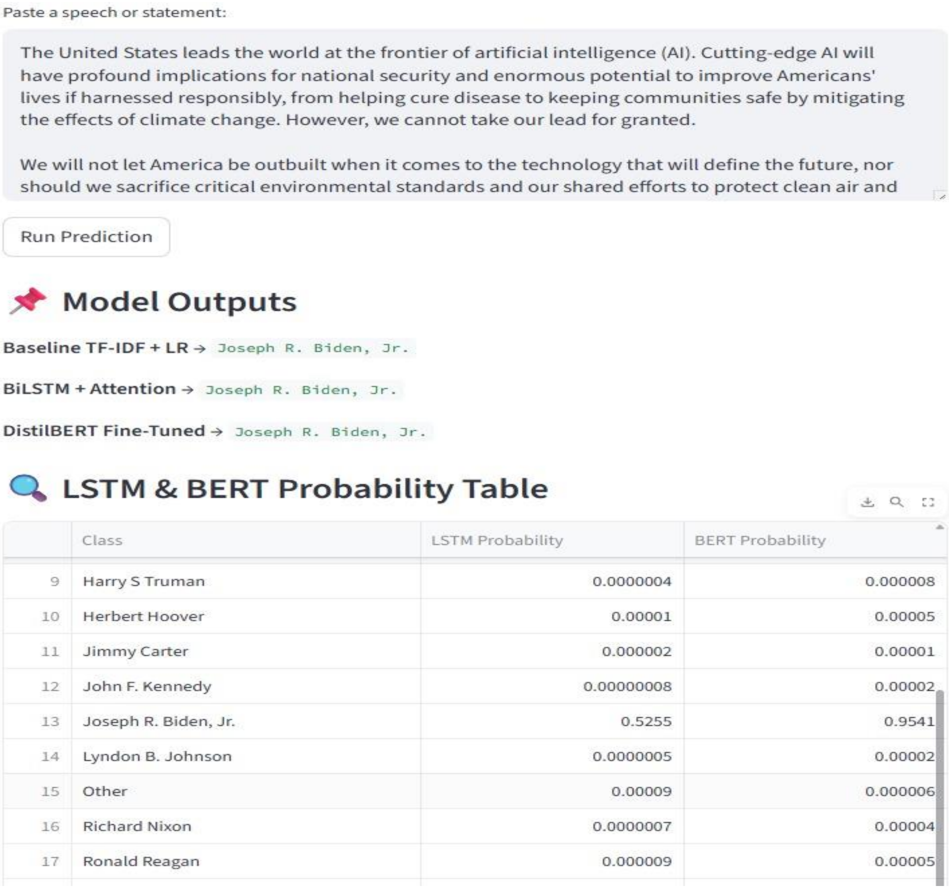
Improvements over both models:

- Less confusion across historical neighbors.
- Better capture of **abstract rhetorical style**, not just phrases.
- Highest accuracy for presidents with well-known speaking styles:
  - **Reagan**
  - **Obama**
  - **Kennedy**

Remaining Difficulty:

- **Short formal statements** (condolences, proclamations, press releases) still confuse the model because:
  - They share institutional, formulaic templates.

- They provide minimal contextual clues.



Paste a speech or statement:

The United States leads the world at the frontier of artificial intelligence (AI). Cutting-edge AI will have profound implications for national security and enormous potential to improve Americans' lives if harnessed responsibly, from helping cure disease to keeping communities safe by mitigating the effects of climate change. However, we cannot take our lead for granted.

We will not let America be outbuilt when it comes to the technology that will define the future, nor should we sacrifice critical environmental standards and our shared efforts to protect clean air and

📌 Model Outputs

Baseline TF-IDF + LR → `Joseph R. Biden, Jr.`

BiLSTM + Attention → `Joseph R. Biden, Jr.`

DistilBERT Fine-Tuned → `Joseph R. Biden, Jr.`

🔍 LSTM & BERT Probability Table

	Class	LSTM Probability	BERT Probability
9	Harry S Truman	0.0000004	0.000008
10	Herbert Hoover	0.00001	0.00005
11	Jimmy Carter	0.000002	0.00001
12	John F. Kennedy	0.00000008	0.00002
13	Joseph R. Biden, Jr.	0.5255	0.9541
14	Lyndon B. Johnson	0.0000005	0.00002
15	Other	0.00009	0.000006
16	Richard Nixon	0.0000007	0.00004
17	Ronald Reagan	0.000009	0.00005

Figure: Model predictions using statement from President Biden in the dataset (Demo was given in real-time during presentation)

Summary:

DistilBERT's self-attention allows it to analyze:

- Semantic meaning
- Contextual interactions
- Subtle speechwriting patterns

This results in the strongest performance across all metrics.

---

10. Discussion

10.1 What the Results Reveal About Presidential Rhetoric

The combination of lexical, sequential, and contextual models reveals several important insights:

1. Vocabulary Alone Identifies Presidents Significantly Well

Logistic Regression’s strong performance shows that:

- Presidents exhibit distinctive lexical choices.
- Word distributions alone encode meaningful stylistic signatures.

2. Syntax and Sentence Flow Help but Require More Data

BiLSTM should theoretically outperform TF–IDF, but smaller datasets limit its ability to learn deeper patterns.

3. Contextual Transformers Capture the Deepest Rhetorical Style

DistilBERT excels because:

- It understands the contextual meaning behind statements.
- It captures long-range dependencies.
- It identifies stylistic nuances of different administrations.

10.2 Strengths of the Multi-Model Pipeline

- **Diverse modeling perspective**: each model contributes to a different lens on presidential writing.
- **Sentiment pseudo-labeling strategy** ensures domain alignment.
- **Streamlit deployment** makes the research usable as an interactive tool.

10.3 Limitations

- Dataset size restricts LSTM learning.
- Transformers require computational resources.

Statements don’t totally capture rhetoric in its full capacity. Statements are just one form of the language we can observe from U.S. Presidents and though they do offer good insight into the rhetoric and language patterns of presidents, it can be argued that other documents available in the project might reveal more. For example, the spoken addresses section of the American Presidency Project offers transcripts of interviews, press briefings, and more face to face interactions a president might have with reporters. Capturing such interaction might be a better reflection of a president’s individual language habits and patterns. However, for the sake of our project timeline and the complexity involved, we did not use this section.

10.4 Future Work

Ideally, future work will be aimed at addressing the possible limitations of this project. For one, using different types of documents even from the same project could help further differentiate rhetoric among presidents. In doing so, it's possible that repeating the work in this project, but with more data, could not only create more interesting EDA with sentiment, tone, emotion, and strategy, but also help improve model performance and predictions since the models used can capture more differences in rhetoric that might not have been found in just using the statements of U.S. Presidents.

Future work can also include expanding this project to more than just U.S. Presidents. Maybe using other political leaders in the country like Congresspeople or moving in an international direction and capturing rhetoric from the leaders of other countries.

11. Conclusion

This study demonstrates that computational analysis of presidential rhetoric benefits from a multi-model framework combining classical, recurrent, and transformer-based architectures. DistilBERT achieves the strongest performance, confirming that contextual language models capture deeper stylistic distinctions than word-frequency or sequence models alone.

The integration of mathematical rigor, transparent architecture, and deployment via app.py strengthens the research pipeline. The Streamlit application mirrors the exact preprocessing and inference operations used during training, ensuring reproducibility and scientific integrity.

Ultimately, this work shows how modern NLP can uncover meaningful patterns in political speech, offering tools for political scientists, historians, and computational linguists to study rhetoric at scale.

12. References

Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). *Language models are few-shot learners*. arXiv:2005.14165

.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171–4186.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *ACL 2018*, 328–339.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *ICLR 2015*.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *ICLR 2019*.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. *EMNLP 2014*.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT: A distilled version of BERT. arXiv:1910.01108.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. *EMNLP 2020: System Demonstrations*, 38–45.

cardiffnlp/xlm-twitter-politics-sentiment · Hugging Face. (2025, April 24). Huggingface.co. https://huggingface.co/cardiffnlp/xlm-twitter-politics-sentiment

The White House Historical Association. (2017). *The Presidents Timeline*. WHHA (En-US). https://www.whitehousehistory.org/the-presidents-timeline