Political Speech Classification

Final Project Presentation

Minwoo Yoo

December 8, 2025

1/2. Data Source and Project Rationale

A. Data and Goal

1. Source Data: U.S. Congressional Speech Record.
2. Initial Volume: 11.7 Million speeches.
3. Core Objective: Binary Classification to predict the speaker's political party.

B. Motivation

1. My Research Interest: My external doctoral work focuses on how economic policies affect the **political slant** of local U.S. newspapers.
2. Data Requirement: I need a reliable model to measure political leaning. This dataset provides the necessary **labeled political speech** to train that model.

2/2. Data Reduction Stages and Final Sample

Table: Comparison of Sample Subsets (1981–1989 Focus)

Characteristic	**Subset A**	**Subset B**	**Subset C**
Scope			**Final Training Data**
Time Period	1981 - Present	1981 - 1989	1981 - 1989
Total Size (Million)	2.27	0.81	0.39
Filtering Applied	$\geq$ 30 chars	$\geq$ 30 chars	$\geq$ **2** sents AND $\geq$ **200** chars
Purpose	Initial scope reduction	Isolate target raw data	**RoBERTa Fine-tuning**

Rationale for Data Reduction

1. Data Quality & Computational Efficiency
2. Personal Research Need
3. Exclude Uninformative Noise

Preprocessing: Text and Label Cleaning

1. Keep only speeches with valid party labels (D or R)
2. Unicode normalization (NFKC)
3. Whitespace normalization (collapse redundant spaces)
4. Apply text-length and content filtering:
   - 4.1 Minimum-length filter (applied to all datasets):
     - At least 30 characters
   - 4.2 Optional filter (used only for the final training subset):
     - At least 2 sentences
     - At least 200 characters

Model List and Training Data Overview

Table: Model Training Configurations

Model ID	Type	Time Period	Sample	Epochs
TF1	TF-IDF	1981 – 2011	Subset A	N/A
TF2	–	1981 – 1989	Subset B	–
TF3	–	–	Subset C	–
M1	RoBERTa	1981 – 2011	Subset A	1
M2	–	1981 – 1989	Subset B	–
M3	–	–	Subset C	–
M4	–	–	–	2
M5	–	–	–	3

Note:

- Subset C uses the final **0.39** Million speech sample.

Algorithm Background: TF-IDF

1. **Term Frequency (TF)**

$$\mathrm{TF}(t, d) = \frac{\text{count of term } t \text{ in document } d}{\text{total terms in } d}$$

2. **Inverse Document Frequency (IDF)**

$$\mathrm{IDF}(t) = \log\left(\frac{N}{1 + n_t}\right)$$

where N = total number of documents, n_t = number of documents containing term t.

3. **TF-IDF Weight**

$$\text{TF-IDF}(t, d) = \mathrm{TF}(t, d) \times \mathrm{IDF}(t)$$

Connection: TF-IDF to Logistic Regression

1. **TF-IDF Vectorization**
   - Converts the text corpus into a **Sparse Matrix (X)**.

2. **Logistic Regression Learning**
   - The model takes the **Sparse Matrix (X)** as input features and the party labels (**Y**) as the target.
   - **Learning:** It assigns a unique **Weight (W)** to each of the 50,000 features (words).

3. **Prediction (Inference)**
   - The model computes the final score ($\mathbf{X}_{\text{new}} \cdot \mathbf{W}$) and transforms it into the probability of being R or D.

Baseline Model Setup: TF-IDF + Logistic Regression

- **Data Focus:** Mostly 1980s speeches
- **Train / Validation / Test Split:**
  - 80 percent training, 10 percent validation, 10 percent test
- **TF-IDF Vectorizer:**
  - **Features:** 50,000 (Unigrams & Bigrams)
  - **Filtering:** min_df = 5 (removes rare noise/typos)
- **Logistic Regression:**
  - **Solver:** SAGA (optimized for sparse matrices)
  - **Class Weight:** Balanced (handles D/R class imbalance)

TF-IDF (Baseline) Results

Table: TF-IDF Classification Performance

Model ID	Time Period	Data Filter	Accuracy	F1
TF1	1981 – 2011	Minimum filter	0.650	0.649
TF2	1981 – 1989	–	0.649	0.648
TF3	–	Optional filters	0.694	0.692

Notes

- Minimum filter: $\geq$ 30 characters.
- Optional filters: $\geq$ 2 sentences AND $\geq$ 200 characters, resulting in the 0.39M sample.

Confusion Matrix: TF-IDF Baseline

Table: Confusion Matrix for TF-IDF Baseline

	Predicted D	Predicted R
Actual D	13736	6152
Actual R	5919	13592

Recall

- Democrat recall: $\frac{13736}{13736+6152} \approx 0.69$
- Republican recall: $\frac{13592}{13592+5919} \approx 0.70$

Key Point

- TF-IDF performs reasonably well and recalls both parties at a similar level.

RoBERTa Background: Core Architecture

RoBERTa is built on the **Transformer** architecture.

1. **Transformer Architecture**
   - Utilizes the **Encoder-only** side of the Transformer network.
   - Primary mechanism is **Self-Attention**.

2. **Self-Attention**
   - Allows the model to weigh the **importance of all other words** in the input sentence.
   - Result: Creates **contextualized vector embeddings**.

3. **Key Difference: Embeddings**

Model	Word Meaning	Context Dependency
RoBERTa	Dynamic (Contextualized)	Dependent
TF-IDF	Static (Fixed Weight)	Independent

RoBERTa vs. BERT: Key Optimizations

RoBERTa is an optimized version of BERT, enhancing the **Pre-training** process.

Table: Comparison of Pre-training Strategies

Optimization Factor	**Original BERT**	**RoBERTa**
Training Data Size	~16 GB	$\geq$ 160 GB
Masking Strategy	Static Masking	Dynamic Masking
NSP Task	Included	Removed

Result:

- **Superior Adaptability:** It handles new, unseen tasks better.
- **More Robust Performance:** It delivers stronger and more consistent results during fine-tuning.

Model Setup: RoBERTa Fine-Tuning

- **Dataset:**
  - Mostly 1981–1989 speeches
- **Train / Validation / Test Split:**
  - 80 percent training, 10 percent validation, 10 percent test
- **Input Representation:**
  - Tokenizer: RoBERTa (fast)
  - Max sequence length: 256
- **Training Hyperparameters:**
  - Batch size: 32
  - Learning rate: 1e-5
  - Optimizer: AdamW (weight decay = 0.01)
  - Epochs: 1–3

RoBERTa (M1–M5) Results

Table: RoBERTa Model Performance Comparison

Model	Time Period	Data Filter	Epoch	Acc	F1
M1	1981 – 2011	No	1	0.658	0.634
M2	1981 – 1989	–	–	0.662	0.659
M3	–	Yes	–	0.692	0.679
M4	–	–	2	0.715	0.699
M5	–	–	**3**	**0.729**	**0.713**

Notes

- **Epoch Selection:** Validation performance was monitored each epoch; the table reports **test-set results only**.

Confusion Matrix: RoBERTa (M5, Epoch 3)

Table: Confusion Matrix for M5

	Predicted D	Predicted R
Actual D	15507	4261
Actual R	6396	13235

Recall

- Democrat recall: $\frac{15507}{15507+4261} \approx 0.78$
- Republican recall: $\frac{13235}{13235+6396} \approx 0.67$

Key Point

- The model performs well overall,
- Misclassification is asymmetric: R $\rightarrow$ D errors are more common than D $\rightarrow$ R.

Key Observations and Model Comparison

Table: Performance Comparison: Baseline vs. RoBERTa Models

Model ID	Model Type	Optional Filter	Epoch	Acc	F1
TF3	**TF-IDF**	**Yes**	–	0.694	0.692
M2	RoBERTa	No	1	0.662	0.659
M3	–	Yes	1	0.692	0.679
M4	–	–	2	0.715	0.699
M5	–	–	**3**	**0.729**	**0.713**

- **Filtering is Critical:** Performance of the filtered baseline (**TF3 Acc 0.694**) exceeds the unfiltered RoBERTa (**M2 Acc 0.662**).
- **Initial Gain:** Performance jumped dramatically from M3 (Epoch 1, Acc 0.692) to M4 (Epoch 2, Acc 0.715), highlighting rapid learning during early fine-tuning.

Conclusion

Main Takeaways

- High-quality filtering (2 sentences, 200 chars) significantly improves model performance.
- RoBERTa fine-tuning outperforms the TF-IDF baseline, showing strong gains in accuracy and recall.

Limitation

- Only a limited set of hyperparameters was explored.
- Key choices such as `max_length`, alternative filtering rules, and additional preprocessing settings were not systematically tuned.

Future Work

- Existing economics research often relies on a Bigram-based Multinomial Logit (MNL) model.
- Next step: introduce robustness checks using modern NLP methods.
    1. Compare results using TF-IDF and RoBERTa.
- Goals:
    1. Assess whether findings depend on the Bigram–MNL framework.
    2. Examine whether more accurate partisanship measures improve research quality.

Thank you!