Inside the Chamber:
Applying Natural Language Processing & Machine Learning to Measure Political Polarization Trends

Homa Fotros

American University
Spring 2022

Motivation & Goal

- Political Polarization, Radicalization, Diversity of Thought, Bias
- Social Media: Virtual Public Square, Echo-chambers
- Using NLP’s methods to clarify/demystify the concepts
- Obtaining objective, concrete, trackable, and improvable measures

Data Source

- Tweets sent by U.S. Senators (117th Congress)
- 100 Senators: 50 Democrats, 50 Republican
- Training: ~24,000 Tweets
  - Election Season, Sep. 1st to the election results (Nov. 10th, 2020)
- Test/Validation: ~152,000 Tweets
  - From 2020 Election to April 1st, 2022
- Labels: D (Democrat) | R (Republican)

```
dts['Orientation'].value_counts()
```

```
D    13806
R    10217
Name: Orientation, dtype: int64
```

Data Preparation

Twitter API: Academic Account

- Sign-up --> Explanation of Application/Project --> Verification & Approval
- Authentication codes

Fetching Tweets using Tweepy Interface

- A Twitter List of Senators (e.g., C-Span list) : Twitter ID
- Excluding retweets, including time stamp
- Simple Code, pagination, max-limit query
- Trick about the time-limit (no 15-mins bummer!)
- Other meta-data for each tweet

BERT

- Pre-processing: Text Cleaning (hashtags, url, emoji, ...)
- BERT model by Transformers from '*Hugging Face*'
  - Binary classification using BertForSequenceClassifcation
  - Case-based model (416 mb)
  - 12 NN layers, 768 Features
  - BERT pooler layer: Linear Classification on top of NN, activation by Tanh()
- PyTorch on Colab's GPU (Tesla T4)

BERT

- Tokenizer
- Tagging & Padding
- Fine-tuning the pre-trained BERT
- 90/10 split, 4 epochs (~1 hr 15 mins)

- Saving & Loading the Fine-tuned model

```
======== Epoch 4 / 4 ========
Training...
  Batch    40  of    676.    Elapsed: 0:01:04.
  Batch    80  of    676.    Elapsed: 0:02:09.
  Batch   120  of    676.    Elapsed: 0:03:13.
  Batch   160  of    676.    Elapsed: 0:04:18.
  Batch   200  of    676.    Elapsed: 0:05:22.
  Batch   240  of    676.    Elapsed: 0:06:26.
  Batch   280  of    676.    Elapsed: 0:07:31.
  Batch   320  of    676.    Elapsed: 0:08:35.
  Batch   360  of    676.    Elapsed: 0:09:39.
  Batch   400  of    676.    Elapsed: 0:10:44.
  Batch   440  of    676.    Elapsed: 0:11:48.
  Batch   480  of    676.    Elapsed: 0:12:53.
  Batch   520  of    676.    Elapsed: 0:13:57.
  Batch   560  of    676.    Elapsed: 0:15:01.
  Batch   600  of    676.    Elapsed: 0:16:06.
  Batch   640  of    676.    Elapsed: 0:17:10.

  Average training loss: 0.09
  Training epcoh took: 0:18:07

Running Validation...
  Accuracy: 0.87
  Validation Loss: 0.47
  Validation took: 0:00:44

Training complete!
Total training took 1:15:24 (h:mm:ss)
```

TF-IDF & SVD (LSA)

- TF-IDF
  - Computationally Cheap
  - Low attention to the context

- Uni-gram & Bi-gram, 800 Features
- Accuracy
  - Train: 0.65
  - Test: 0.59

------------- Prediction & Classification Report -------------

	precision	recall	f1-score	support
D	0.60	0.65	0.63	81147
R	0.56	0.51	0.54	71374
accuracy			0.59	152521
macro avg	0.58	0.58	0.58	152521
weighted avg	0.59	0.59	0.58	152521

Word2Vec

- Google-News Pre-trained model
- Sentences: Mean(Words)
- Feature Extraction, Train, Test: ~18 hrs
- Train Accuracy: 0.76
- Test Accuracy: 0.69

```
-------------- Prediction & Classification Report --------------
-----------------------------------------------------------------
              precision    recall  f1-score   support

           D       0.70      0.74      0.72     81147
           R       0.68      0.63      0.66     71374

    accuracy                           0.69    152521
   macro avg       0.69      0.69      0.69    152521
weighted avg       0.69      0.69      0.69    152521
```

BERT

```
print(classification_report(flat_true_labels,flat_predictions))
```

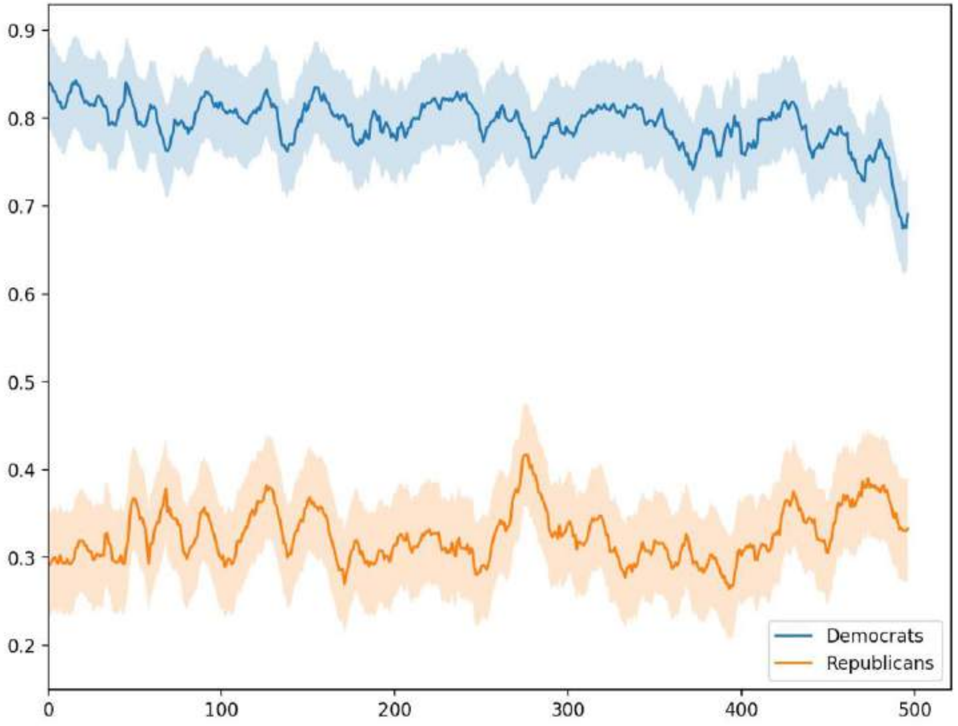
	precision	recall	f1-score	support
0	0.75	0.81	0.78	81715
1	0.77	0.70	0.73	72254
accuracy			0.76	153969
macro avg	0.76	0.76	0.76	153969
weighted avg	0.76	0.76	0.76	153969

- Validation set
- Scores:
  - Accuracy: 0.76
  - F1-Score: 0.73
  - Matthews Correlation Coefficient: 0.515 (Nice!)

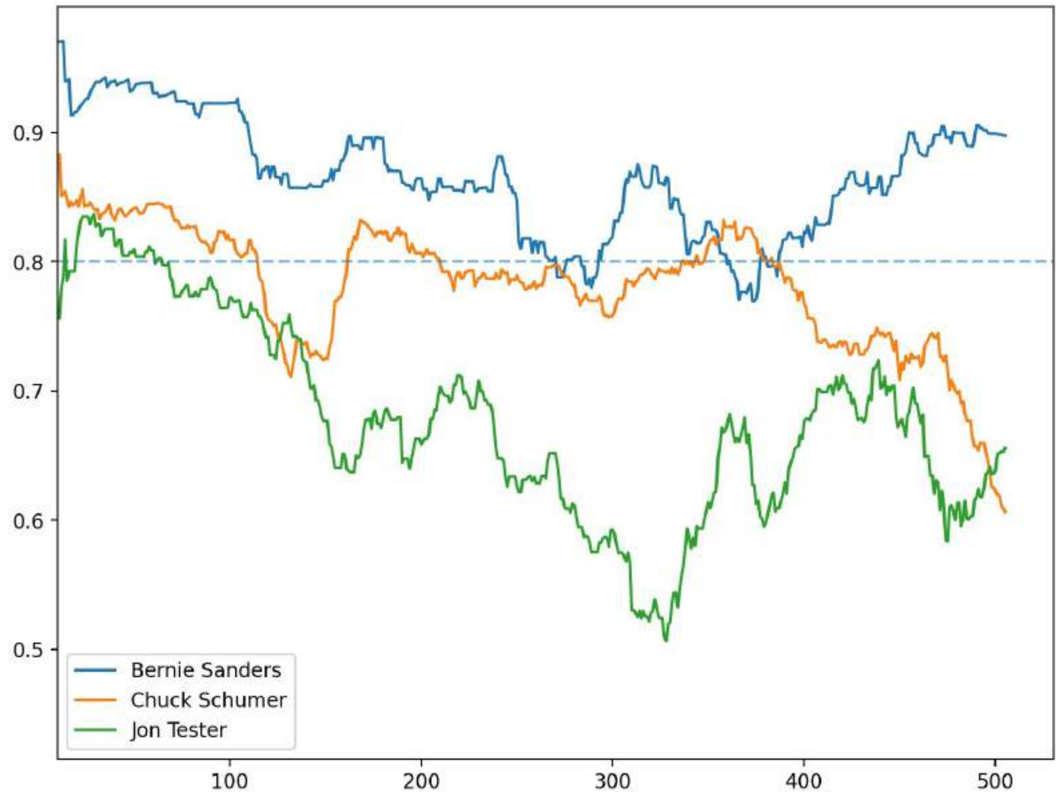
- There are a lot of mis-classified tweets, BUT >>>
  - Probability Scores for each Tweet

BERT: Findings

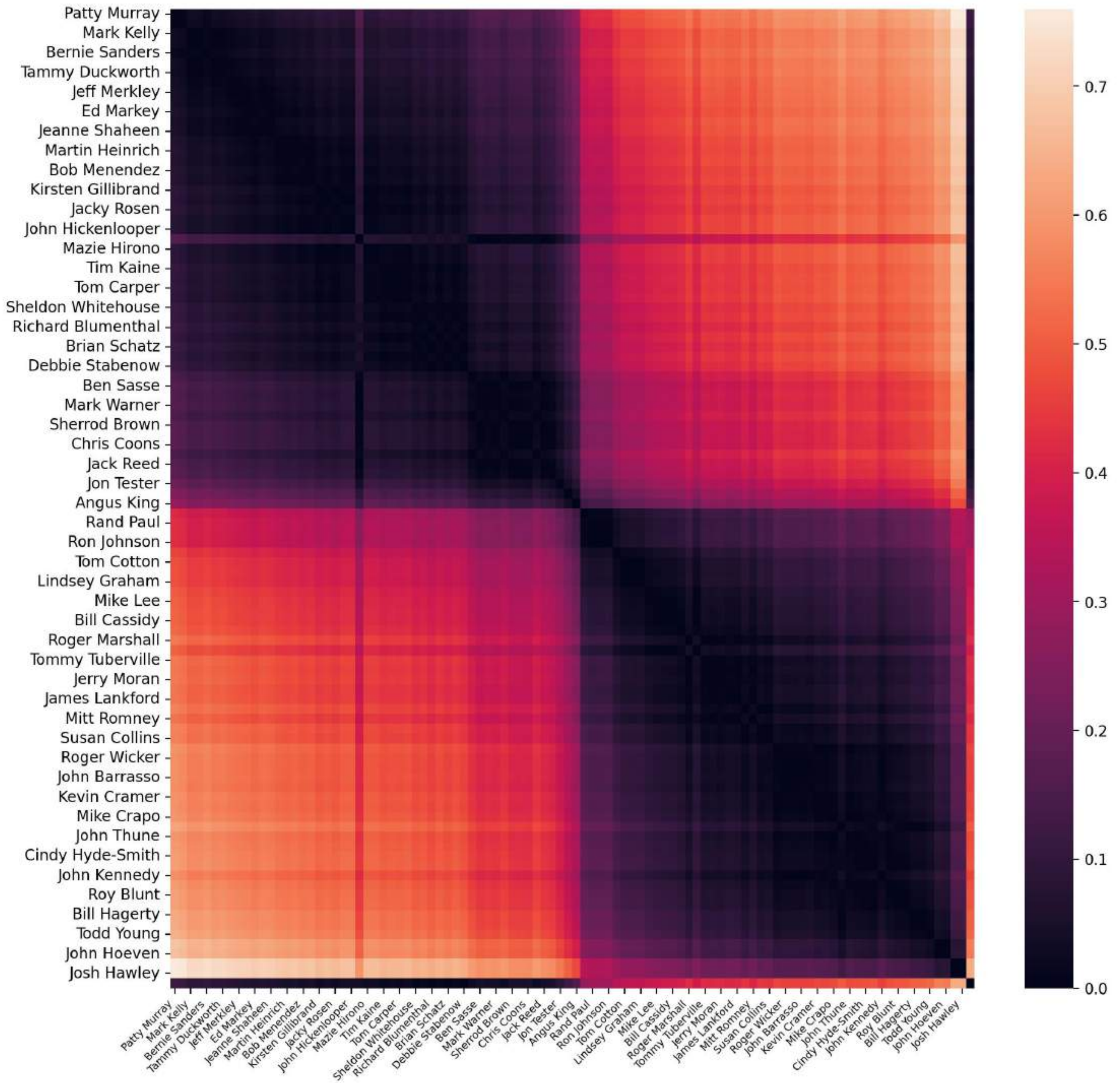
- Average Score (Democrat Direction)
  - Democrats: 0.79
  - Republicans: 0.33
- Top/Least D Tweets by a D
- Top/Least R Tweet by a R
- Polarization Gap over time

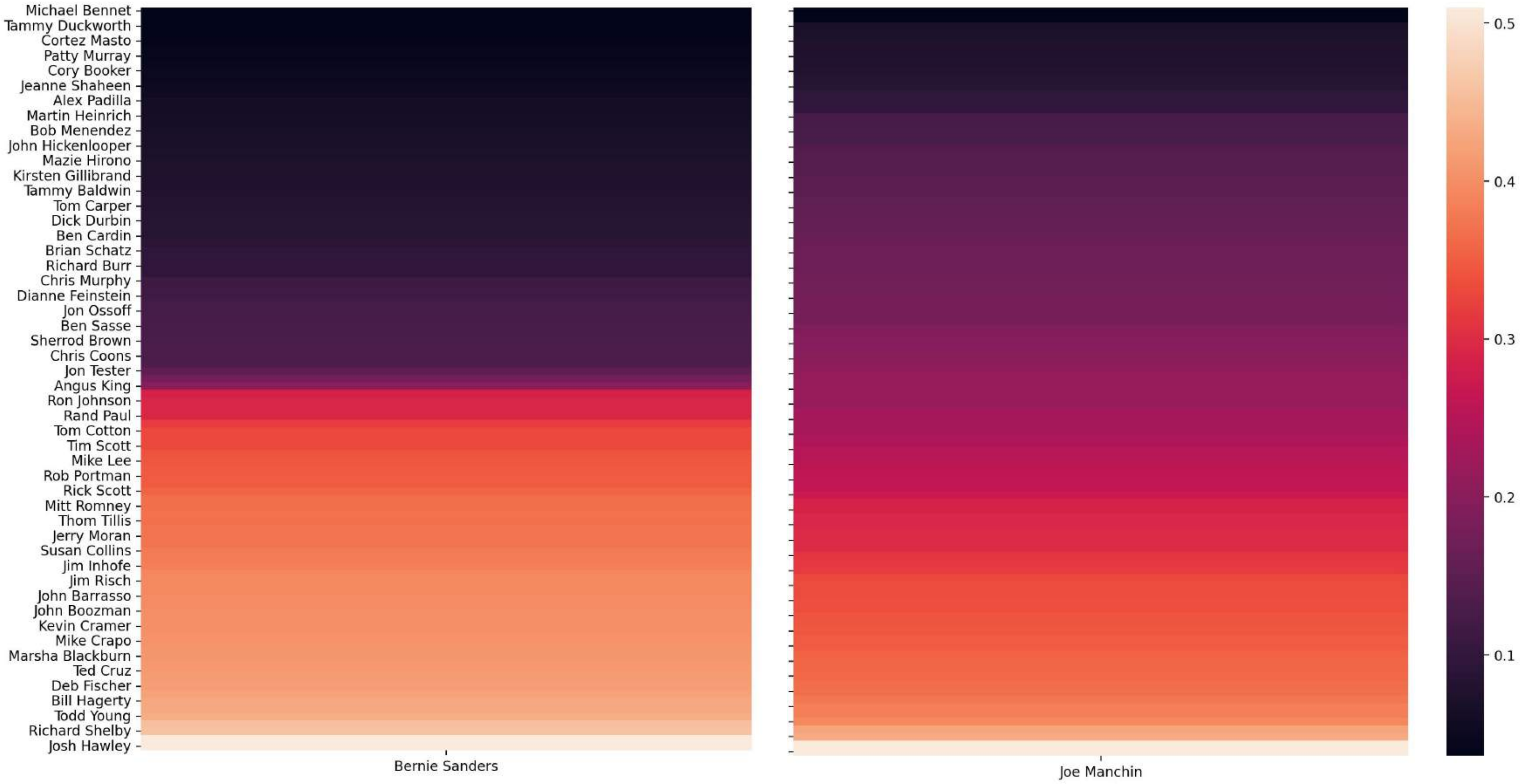


Trends, Similarity, Divergence

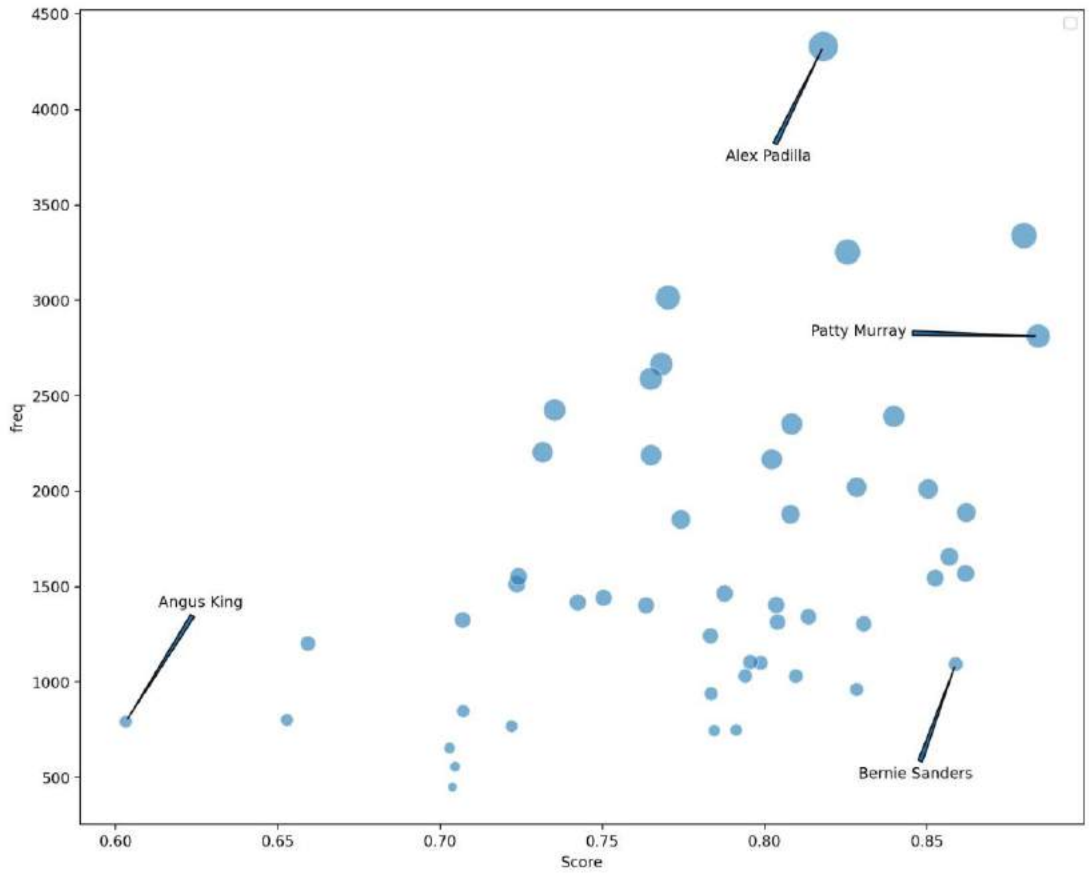


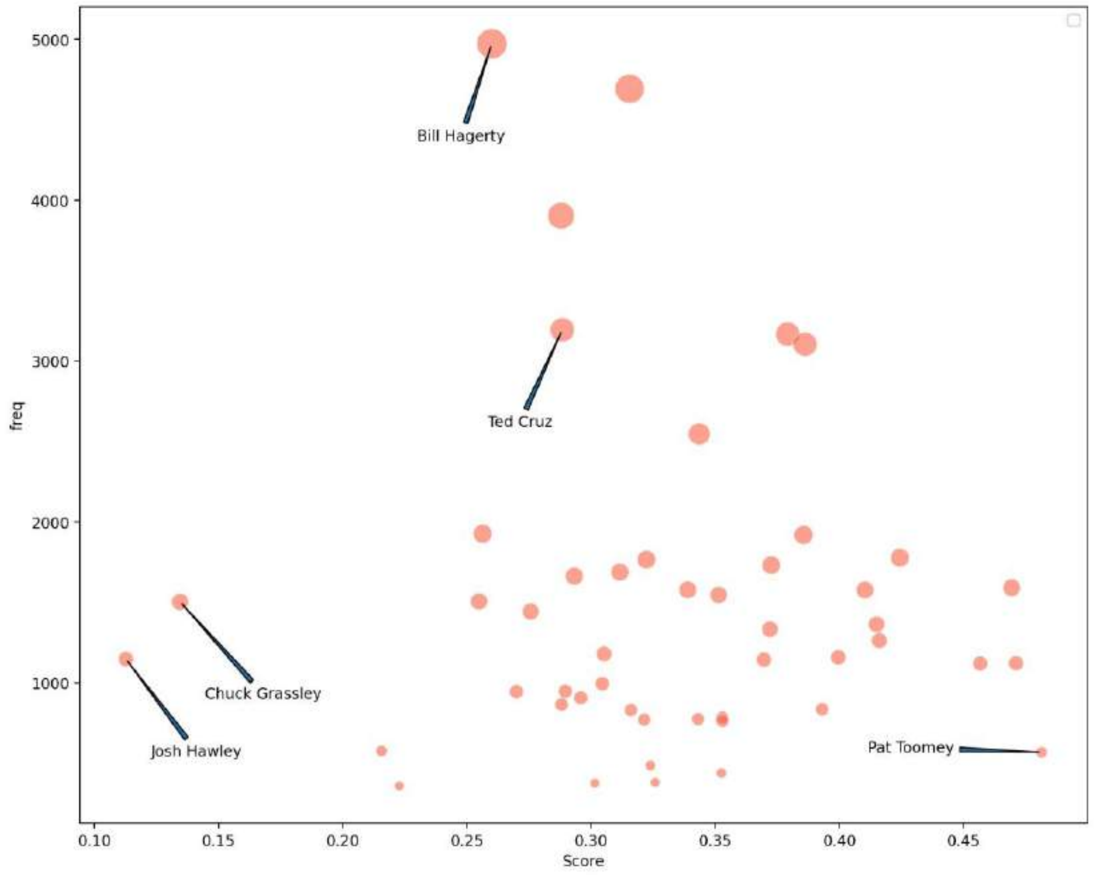
Average Day-Difference




Michael Bennet
Tammy Duckworth
Cortez Masto
Patty Murray
Cory Booker
Jeanne Shaheen
Alex Padilla
Martin Heinrich
Bob Menendez
John Hickenlooper
Mazie Hirono
Kirsten Gillibrand
Tammy Baldwin
Tom Carper
Dick Durbin
Ben Cardin
Brian Schatz
Richard Burr
Chris Murphy
Dianne Feinstein
Jon Ossoff
Ben Sasse
Sherrod Brown
Chris Coons
Jon Tester
Angus King
Ron Johnson
Rand Paul
Tom Cotton
Tim Scott
Mike Lee
Rob Portman
Rick Scott
Mitt Romney
Thom Tillis
Jerry Moran
Susan Collins
Jim Inhofe
Jim Risch
John Barrasso
John Boozman
Kevin Cramer
Mike Crapo
Marsha Blackburn
Ted Cruz
Deb Fischer
Bill Hagerty
Todd Young
Richard Shelby
Josh Hawley
Bernie Sanders
Joe Manchin
0.5
0.4
0.3
0.2
0.1

Loud voices are more extreme?





Limitations

- Impact of High-frequency tweet-senders on training/test set
- Tweets are not well-structured texts
- Computational limitations
  - Hidden & Output layers
  - Training on a larger set
- BERT’s Linear Classifier Layer

Future Works

- Expand training set (computation)
- Fine-tuning & Extracting Model’s Hidden Layers (computation)
- Comparing with CRF & other methods

- Similar Data sets & Contexts (e.g., entire congress)
- Feature Importance: What makes tweets/sentences less partisan?
- How can we better apply NLP models on tweet texts?

- Thanks!