Political Atlas

“Map to Politics”

Khinshan Khan, Adeebur Rahman, Brian Cheung, Talha Rahman

Capstone Proposal 02/22/2021

Pitch

The 411

Pitch

Our *vision* is to improve political understanding of the American electorate.

Our *target audience* are voters looking to better understand political rhetoric.

Solve the *problem* of ambiguous & potentially deceitful political speech.

Our *strategy* is bettering political transparency through textual tone analysis & vocal tone analysis.

Our *goal* is having a growing user base => increased voter turnout.

Understanding the problems

01 Voters don't normally have the time to go through all content associated with a politician.

02 Politicians often lie or, at best, are misleading with how they deliver information.

03 Voters may not have the ability to critically evaluate all statements made by a politician.

Project Objective

- Better understanding of government
- Create a website to allow voters to view politicians' speeches in-depth
- Speeches are analyzed for different kinds of tones & emotions

Target Audience

Users interested in:

- a better understanding of political rhetoric
- how speeches are delivered
- linguistics or data visualization

Plan

What will we do?

Understanding our Workflow

01 Get video & transcript data

02 Analyze data

03 Present data in a clear & meaningful manner

Large Group Breakout

DATA SCRAPING

DATA ANALYSIS

DATA ENDPOINTS

WEBSITE

DISCORD BOT

The Stack

What technologies are we *actually* using?

Tech Stack

DATA SCRAPING

Python

DATABASE

SQLite3

DATA ANALYSIS

Python

IBM Watson

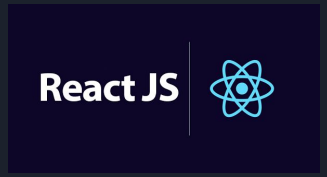
DeepAffects

Tech Stack

BACKEND

Python

WEBSITE



Gatsby.js

Material UI

d3.js

DISCORD BOT

Discord.js

Typescript

Plan In Depth

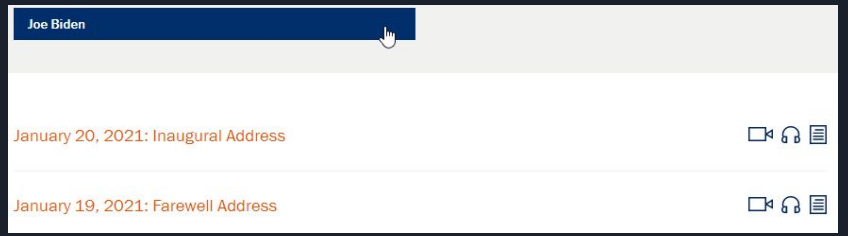
How will we do what we do?

Data Scraping

Where does the data come from?

Where does the data come from?

- Scrape the Miller Center website for presidential speeches using BeautifulSoup in Python.
- Store videos & transcripts in our database
- MVP: Scrape only once.
  - Stretch goal: setup pubsub & queue to continuously download multiple videos & transcripts.
- Locally, maintain a database table which will contain:
  - Correspondence of speech name
  - Politician name
  - video + transcript location path
  - likely with a uuid per speech



Backup Plan

- Backup speeches from youtube using https://github.com/ytdl-org/youtube-dl/

Data Analysis

What APIs will our analysis use and how?

Where does the data analysis come from?

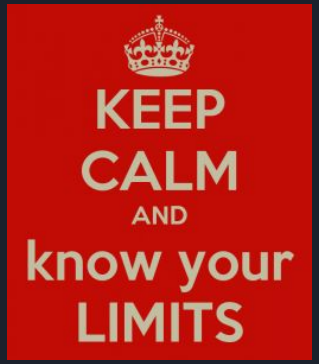
A couple of APIs we're looking into:

- IBM Watson
  - Fetches analysis based on text

- Deep Affects
  - Fetches analysis based on video

DeepAffects

Potential Issues with APIs



- IBM Watson
  - >128 KB of total input content & >1000 individual sentences
  - Analyzes the first 1000 sentences for document-level analysis & only the first 100 sentences for sentence-level analysis
  - Lite: 2,500 API calls per month at no cost
  - Paid plan: up to 250,000 calls at $0.0088 per call for each month
- Deep Affects
  - Free audio processing for 200 minutes of data (unsure if per month)
  - 5 requests per minute
  - 100 requests per day

How will we store the analyses?

- MVP: Go through our database created with the scraping “once”.
- Potentially setup a timer due to API rate limits. Keeping a queue of data to be processed.
  - Stretch goal: additional data would add onto the queue, letting the analysis continue automatically as new speeches are added.
- Locally, maintain a database table with a correspondence of
  - speech name
  - politician name
  - video + transcript location path

Data Endpoints

How will our frontend know what the data is?

How do we expose our analyses?

Set up a python web server using Flask & run queries to our backend with endpoints

Endpoints:

- /list
- /metainfo?id=
- /analysis/transcript?id=
- /analysis/video?id=

Stretch Analysis!

Stretch Endpoints:

- /analysis/transcript?id=
  - Get an array (potentially enumerated lists) of analyses of the transcript with its corresponding text
- /analysis/video?id=
  - Get an array (potentially enumerated lists) of analyses of the video with its corresponding timestamps
- /lookup?partialname=&politican=&area=?
  - Attempts to find speeches based on other meta information than id

Website

What *is* our frontend?

How do we showcase our data?

We plan on having 3 pages for MVP purposes:

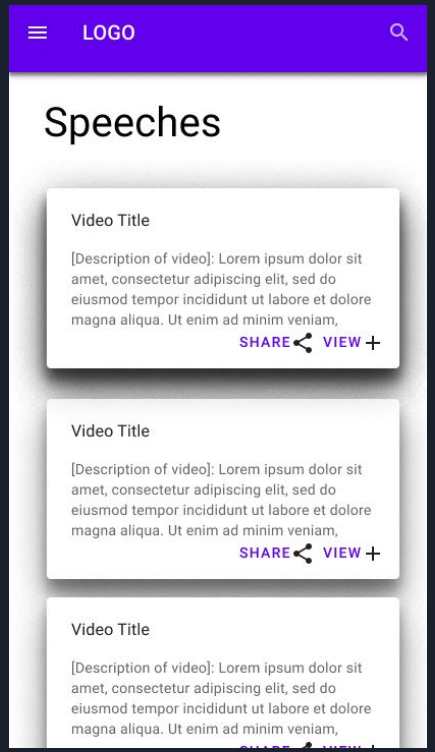
- /
  - Lists all from most recent -> oldest
  - Look into adding pagination to this as our list grows
- /speech?id=
  - Serve video with transcripts and data visualizations
- /about
  - Show information about us & the project
  - Explain why we're *trustworthy* and list our sources

Stretch Pages!

If the backend can support it, we also want to implement:

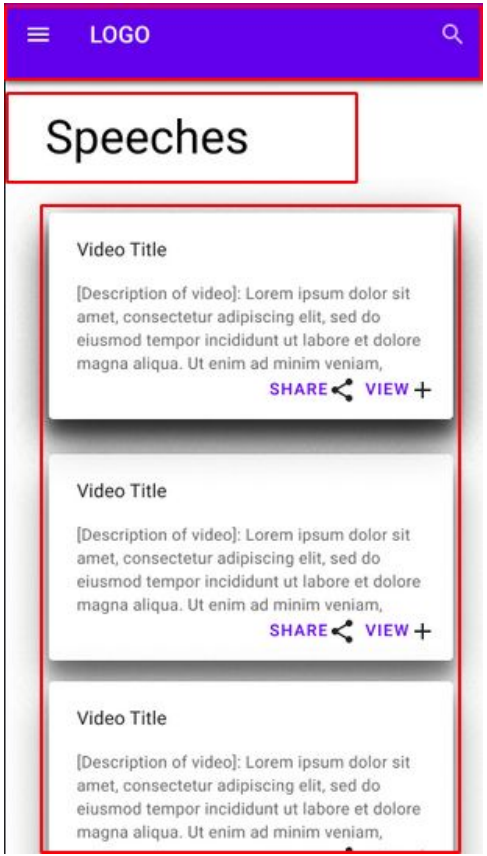
- /search?
  - User selects attributes they want to filter the list by
  - Keep state management of dynamic search and dynamic results
  - We display results similar to homepage
- /feedback
  - User can give feedback about the website (does it lack accessibility? Is it unintuitive?)
  - User can make requests
    - New speeches
    - Different visualizations to include
    - Bugs/ site requests

Mockup

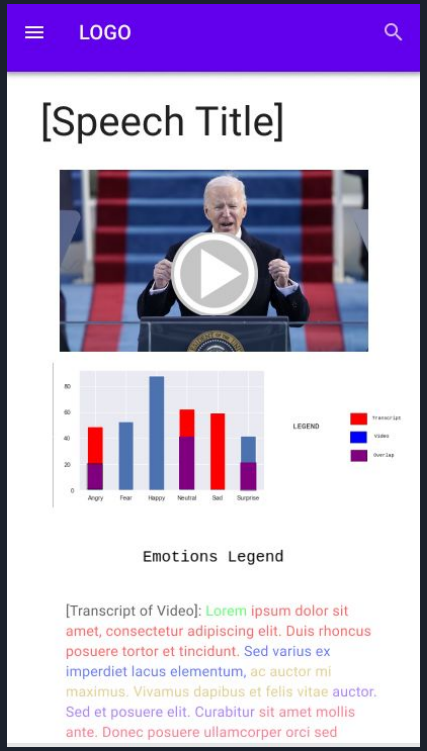

LOGO
Speeches
Video Title
[Description of video]: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
SHARE
VIEW
Video Title
[Description of video]: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
SHARE
VIEW
Video Title
[Description of video]: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,

Components

- Navbar
- Page title
- Listing
  - Shows video names
  - Shows video description

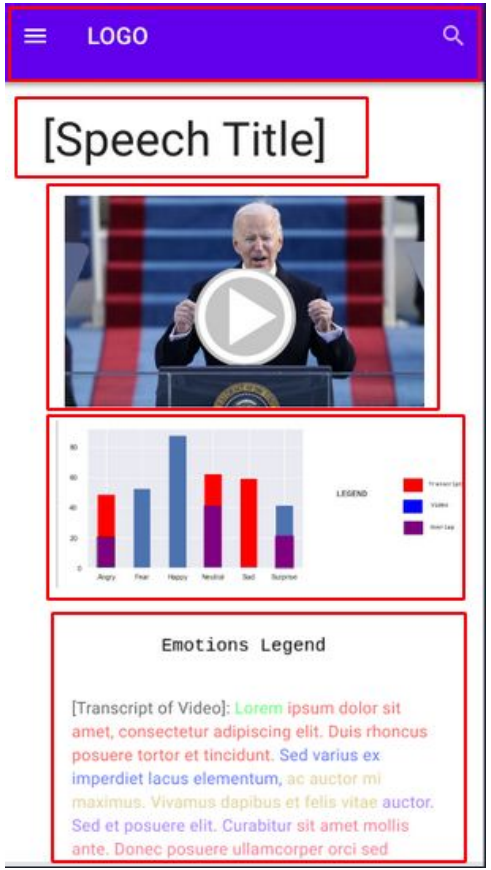


Mockup


LOGO
[Speech Title]
LEGEND
Angry
Fear
Happy
Neutral
Sad
Surprise
Emotions Legend
[Transcript of Video]: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis rhoncus posuere tortor et tincidunt. Sed varius ex imperdiet lacus elementum, ac auctor mi maximus. Vivamus dapibus et felis vitae auctor. Sed et posuere elit. Curabitur sit amet mollis ante. Donec posuere ullamcorper orci sed

Components

- Navbar
- Page title
- Video player
- Graph
  - Video vs Transcription
  - Legend
- Transcription
  - Emotions visualized
  - Legend



Discord Bot

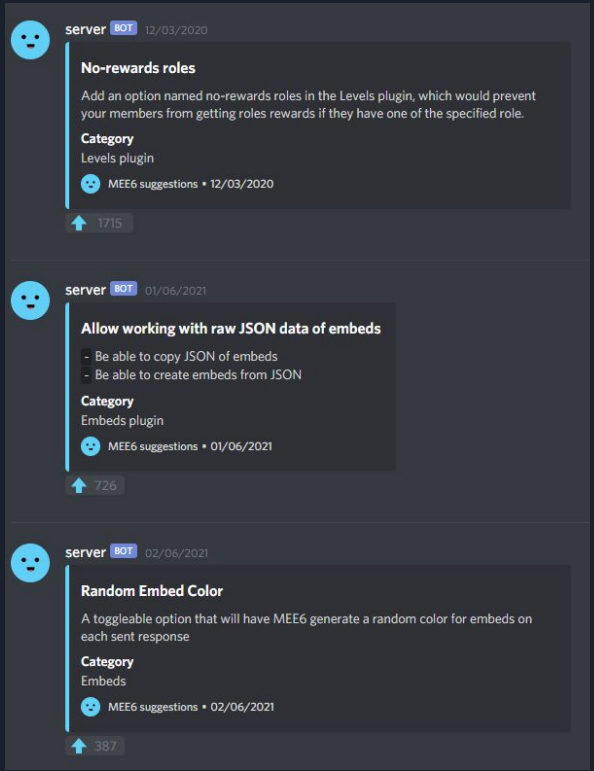
Why do we even need one?

...what is it? Why?

Use Discord bot to filter out suggestions made from users

Devs can see which suggestion is more liked and should be implemented first

Additional data/metrics could be posted to discord as well



Work Distribution

Who's responsible for what?

Responsibilities

Khinshan Khan
- Design (Layout, Mock UI, Color Scheme)
- Frontend (React.js + Material UI)
- Setting up Raw Data Database

Adeebur Rahman
- Data Visualization (D3.js)
- Expose Backend Data via Endpoints (Python + Flask)
- Scrape Data

Brian Cheung
- Data Analysis with APIs
- Setting up Analysis Database
- Assist with Data Visualization

Talha Rahman
- Coordinate Contact for APIs
- Data Analysis with APIs
- Assist with exposing Backend Data

Q & A

Questions? Concerns? Declarations or outcries?

Credits!

All logos belong to their respective owners.

- Pitch: https://www.dreamstime.com/music-notes-stave-eps-vector-illustration-music-notes-stave-classic-black-white-colors-vector-music-notes-image119141167
- Clipboard: https://www.pinterest.com/search/pins/?rs=ac&len=2&q=clipboard%20clipart&eq=clipboard%20cli&etslf=8443&term_meta[]=clipboard%7Cautocomplete%7C0&term_meta[]=clipart%7Cautocomplete%7C0
- Stack: https://dribbble.com/softradix
- Blueprint: https://www.pinterest.com/pin/694821048752426167/
- Data scraping: https://blog.impresivhealth.com/tackling-sdoh-one-question-at-a-time
- Data analysis: https://s4be.cochrane.org/blog/tag/statistics/
- Keep calm meme: https://www.keepcalmandcarryon.com/creator/
- Rest logo: https://restfulapi.net/
- Mockups: Figma with Material Design Tool Kit
- Discord Picture: MEE6
- Work Distribution: https://www.quill.com/blog/tutorials/how-to-ensure-fair-distribution-of-the-workload.html

Fin

Thank you!