Justin Roth
CS Login: Jroth01
Email: JustinCarlRoth@gmail.com

Tufts University
COMP86 HW6 Project Proposal
Professor Jacob
TA: Tomoki Shibata

Rhetoric

A real-time speech analysis web application

Project Description

OVERVIEW

A single page web application that analyzes speech and visualizes the speaker's ideas in real time.

HOW IT WORKS

Upon loading the webpage, the app will begin reading in user audio input, and generate an ongoing text transcript that dynamically appears on the page as the user speaks. Every time there is a meaningful pause or the user stops speech detection, speech text will be analyzed for keywords, sentiments, concepts, phrases, and other ideas. The relationship between ideas will then be displayed in a network visualization graph .

PURPOSE

To incorporate a new method of interaction into an application, specifically speech recognition, and provide real-time analysis of a speaker's ideas.

GOALS

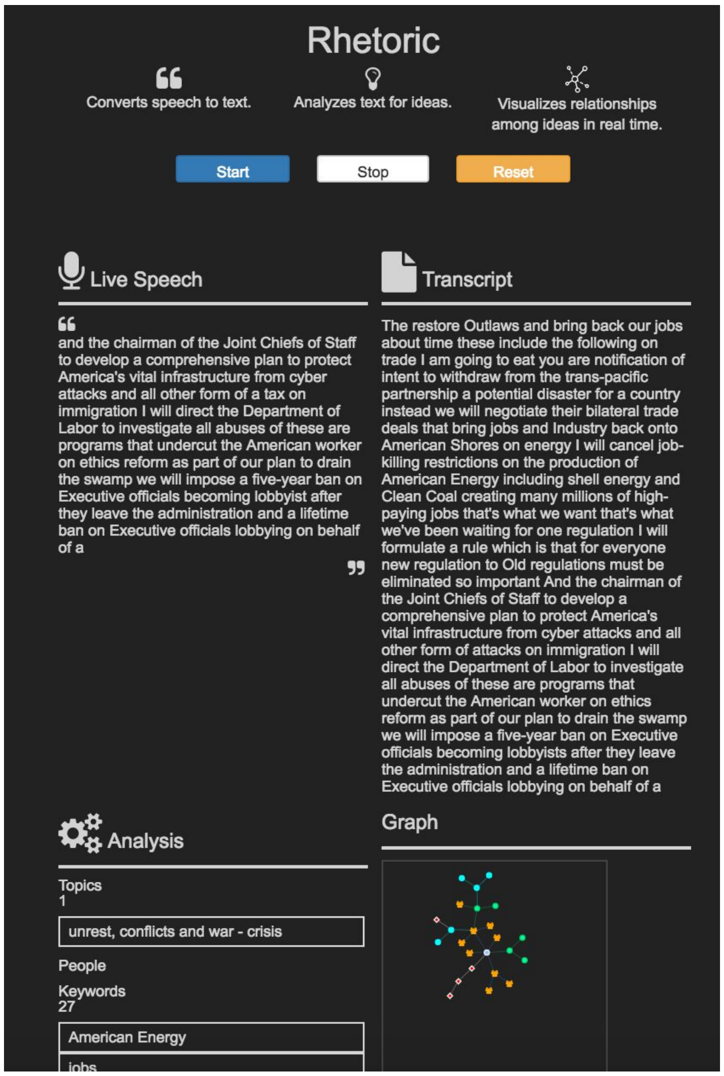
1. Convert speech to text in real time, and render an ongoing transcript in the browser using the *Webkit Speech Recognition API*.
2. Periodically analyze the transcript by making calls to the *Aylien Text Analysis API* .
3. Visualize relationships between keywords, sentiments, concepts, phrases, and other ideas in a network graph using the *VisJS* library

STRETCH GOAL: Perform the same analysis in Goals 1-3, but translating the input from one language into another language (in order for this to work, both languages must be supported by both the *Webkit Speech Recognition API,* and the *Aylien Text Analysis API)*

TARGET USERS

- Economic analysts trying to quickly identify real time market trends and sentiments about the future
- News analysts looking to distill ideas from speeches by leaders and public figures
- Lawyers who seek insight into human motivations from speech
- Medical or psychiatric professionals trying to identify patterns and relationships between described symptoms
- Public speakers who wish to add visual aid to their speeches in real time, or make sure they are properly conveying the core ideas of an overarching message

PROJECT SKETCH



Live Speech

and the chairman of the Joint Chiefs of Staff to develop a comprehensive plan to protect America's vital infrastructure from cyber attacks and all other form of a tax on immigration I will direct the Department of Labor to investigate all abuses of these are programs that undercut the American worker on ethics reform as part of our plan to drain the swamp we will impose a five-year ban on Executive officials becoming lobbyist after they leave the administration and a lifetime ban on Executive officials lobbying on behalf of a

Transcript

The restore Outlaws and bring back our jobs about time these include the following on trade I am going to eat you are notification of intent to withdraw from the trans-pacific partnership a potential disaster for a country instead we will negotiate their bilateral trade deals that bring jobs and Industry back onto American Shores on energy I will cancel job-killing restrictions on the production of American Energy including shell energy and Clean Coal creating many millions of high-paying jobs that's what we want that's what we've been waiting for one regulation I will formulate a rule which is that for everyone new regulation to Old regulations must be eliminated so important And the chairman of the Joint Chiefs of Staff to develop a comprehensive plan to protect America's vital infrastructure from cyber attacks and all other form of attacks on immigration I will direct the Department of Labor to investigate all abuses of these are programs that undercut the American worker on ethics reform as part of our plan to drain the swamp we will impose a five-year ban on Executive officials becoming lobbyists after they leave the administration and a lifetime ban on Executive officials lobbying on behalf of a

Analysis

Topics
1

unrest, conflicts and war - crisis

People

Keywords
27

American Energy

jobs

Graph

After clicking the start button, the program will initiate speech to text recognition. As text is translated, it will appear under "Live Speech". If there is a significant pause in the speech recognition or the user clicks stop, the live speech will be added to the cumulative transcript. At that point, the program will run the cumulative transcript through text analysis api, and return results. Some of these results will include topics, keywords, people, organizations, sentiments, concepts, phrases, and other ideas. The results will be shown in a more raw form on the right, and also in a visual graph on the left that allows the user to zoom and explore relationships.

TECHNICAL REQUIREMENTS

Text Editor / IDE / Debugging Tools :

- Chrome Browser + Chrome Developer Tools
- Sublime text

Programming Languages :

- Javascript
- NodeJS

Markup Languages:

- HTML / CSS

Basic Frameworks and Libraries:

- Twitter Bootstrap
- JQuery
- VisJS - a dynamic, browser based visualization library

System / Other:

- Soundflower - MacOS system extension that allows applications to pass audio to other applications (Already downloaded, installed, and working)

Advanced API & Library Usage:

Text Analysis - Aylien Text Analysis API

http://aylien.com/text-api

I have already contacted the company and obtained an API key for a free tier app, and have been able to successfuly make API calls. I plan to use the "entities", "concepts", "classify", and "hashtags" endpoints. The API provides a way to make combined calls to multiple endpoints, and receive a combined response.

Network Visualization - VisJS

http://visjs.org/index.html#download_install

Speech to text - Webkit Speech Recognition API

https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API

TECHNICAL RISKS

I’m experienced with web development, and so I’m comfortable with Javascript, HTML/ CSS, Bootstrap, NodeJS, and making API calls. However, I’m concerned about the following:

- Exceeding the API Call rate limit for Aylien Text Analysis API
    - The documentation indicates there is a rate limit for the number of calls your application can make to the API, but does not specify the exact limit for the free tier.
- Not keeping Javascript code organized into different components
    - Because this application is using 3 main APIS and dynamically rendering information onto the DOM, a challenge will be to organize code in an object oriented way such that the project is maintainable and easily debugged.

TECHNICAL FEASIBILITY TESTS

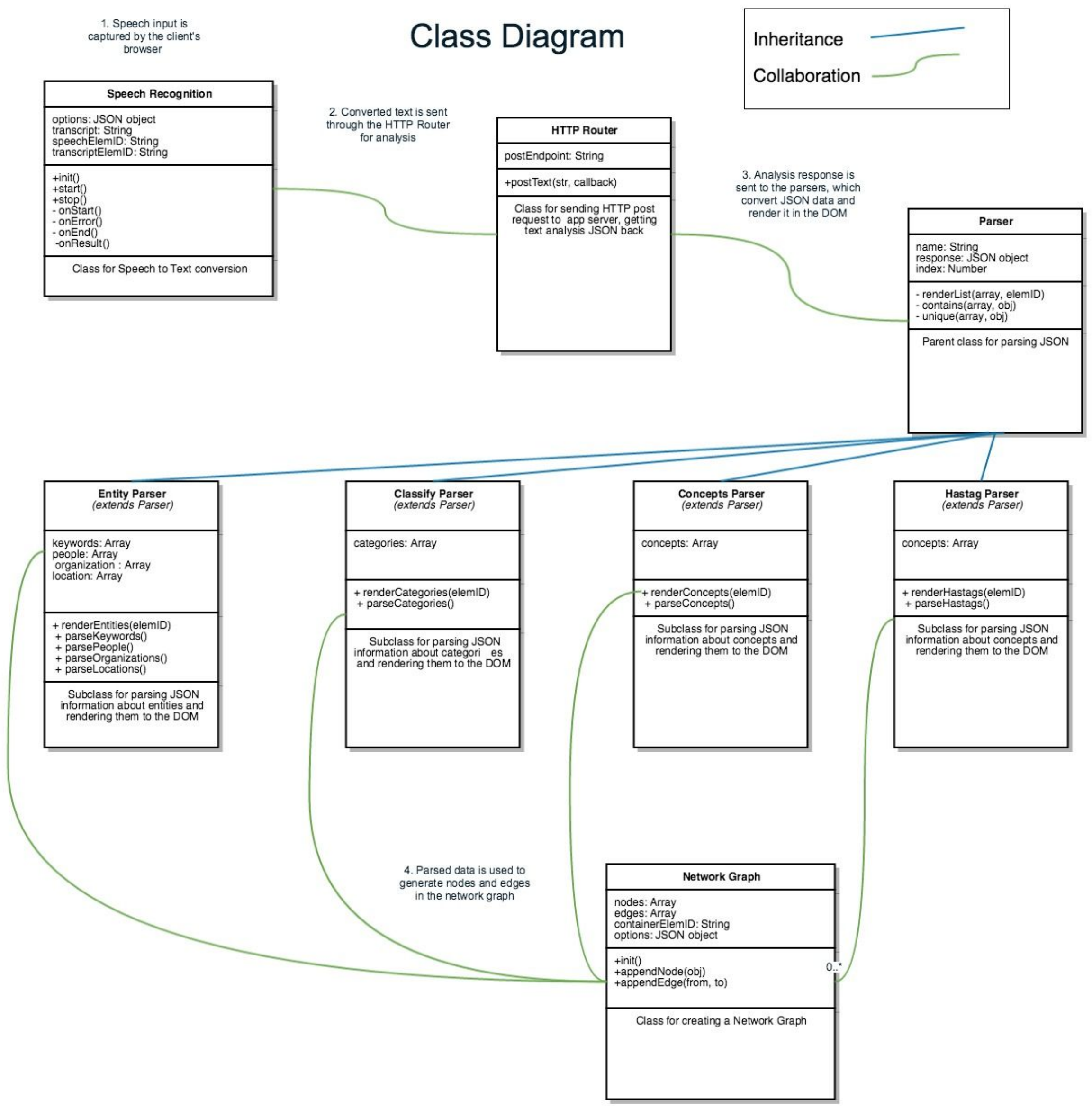
I plan to independently unit test the key features of the application to make sure it meets the following checkpoints. Initially I will perform these tests by hand because the application is dynamic and the results are indeterminate:

1. **Renders text from audio speech input, in real time.**
    a. I will play pre-recorded audio and speak into my computer’s built in microphone. The application should transcribe the audio input in real time, and dynamically render text to the webpage.
2. **Displays raw text analysis after a break or pause in speech recognition**
    a. When the speech recognition callback detects a meaningful pause in audio input, or the user hits the stop button, the application should send a post request to the text analysis API. The response should be parsed and rendered on the page in a more raw format (displayed in tables, lists, simple panels, etc)
3. **Creates and displays a network graph visualization after a break or pause in speech recognition**
    a. When the speech recognition callback detects a meaningful pause in audio input, or the user hits the stop button, the application should create a networking graph that illustrates related concepts / ideas that occur in the text.

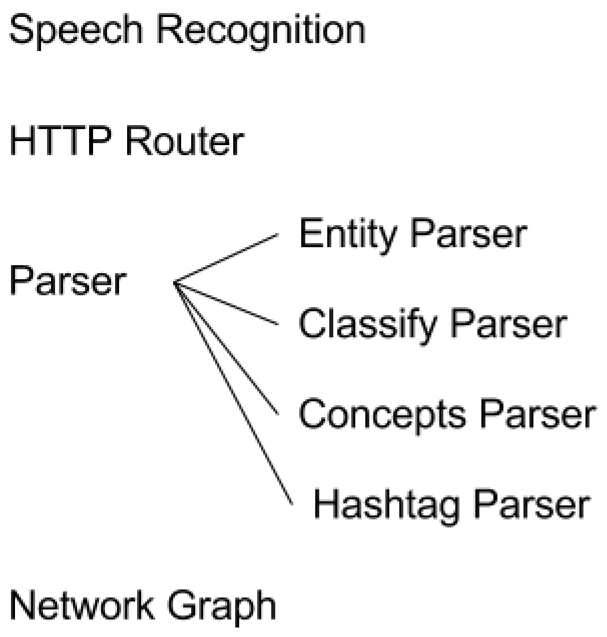
If possible, I’d like to look into ways that I could write automated tests for the three key application features.

DESIGN DOCUMENTATION

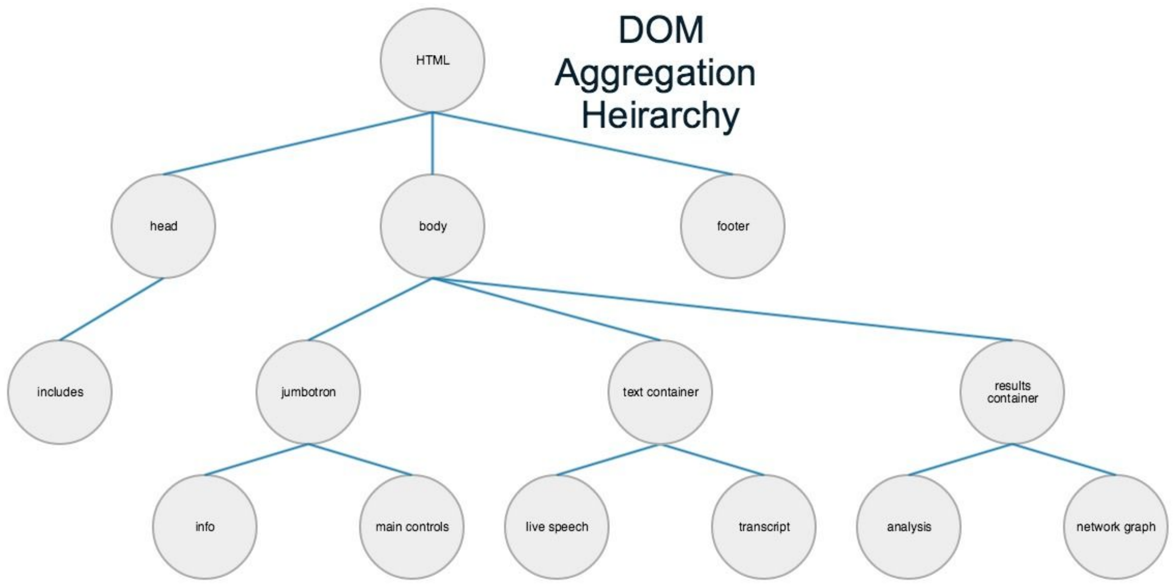
JAVASCRIPT CLASS DIAGRAM OVERVIEW (Subject to change)



JAVASCRIPT INHERITANCE HIERARCHY



HTML DOM AGGREGATION HIERARCHY



JAVASCRIPT COLLABORATION RELATIONSHIPS

Speech recognition first interprets live speech and transcribes it into text. When the api detects a meaningful pause in speech or the user triggers the callback to render the latest transcript, the transcript will be saved.

↓

HTTP Router accepts the transcript text from **Speech Recognition** and sends a post request to the app's server, which makes a call to the Aylien text analysis api, and sends the response back to the client. (Sending it to the app's server first is necessary for hiding the API key credentials).

↓

Parser classes receive the response from the **HTTP Router,** extract the data, and render it onto the HTML DOM.

↓

Network Graph accepts data from the **Parser classes**, and renders nodes onto the networking graph that illustrate relationships between data.

INFORMATION HIDING

Speech recognition: hides information related to the inner workings of the Webkit Speech Recognition API. Provides simple public methods for starting and stopping recognition, as well as public strings of live speech text and the latest transcript.

HTTP Router: hides the details of submitting an XML HTTP post request with the latest transcript text. Provides a simple public method that allows the user to define the callback function when the response is received from the Aylien text analysis api.

Parser & Parser Subclasses: hides information related to how the JSON response from the Aylien text analysis api is parsed into meaningful chunks that can be rendered onto the HTML DOM. Provides simple public methods for parsing desired information and rendering it onto the page.

Network Graph: hides information related to how the VisJS network graph is rendered onto the page. Provides public method for adding nodes and edges.