EECE 437 Team Project Progress Report

Subteam: Karl Hayek, Elie Tamer

April 14th, 2018

1 Project description

Newstime is a website where you can visualize news stories on a timeline of events, and get easy access to a variety of news sources for a story. A user types keywords related to a news story in a search bar, for example facebook cambridge analytica scandal, and gets an interactive timeline summarizing the major events of the story, each with a list of news articles that reference the event. Heres a brief sketch of how it works: a fetcher stage periodically performs a custom web search to obtain urls of news articles from a list of major journals (including new york times, bbc, euronews, al jazeera, etc.) after a coarse preliminary selection of what articles to collect. These urls are forwarded to a labeler stage that extracts topics and entities referenced in the articles. Then, a classifier stage uses the extracted keywords to decide what story the article belongs to, and finally it is either discarded or sent to a noSQL database where it can be searched easily. When a user inputs a keyword, a search through the database is performed and the relevant articles are returned and displayed on the page.

2 Work done

The goal of our subteam is to develop the website where we can access the news timelines and search through them. To this end we developped a web application using Node.js and ExpressJS, and used MongoDB (which is NoSQL) for the database. Previously we had no experience in making websites with Node.js so most of our work till now was learning and getting familiar with web development with Node.js, including building non-relational databases with MongoDB (which we connect to our Node application using the module Mongoose).

In the database there are two Collections (equivalent to tables in SQL): Timelines and Articles. A timeline document can have zero or more articles, and an article document can belong in one or more timelines. This is a many-to-many relationship between documents of the two collections, and to represent it we can simply just include in the Timeline model an array of IDs pertaining to the documents that are in this timeline. Then, when we want to access a Timeline object, we use MongoDB's populate which populates this timeline with the articles that it has (using their IDs).

We used mLab to store our production database online (using a free account), and have deployed our application online on Heroku. You can access it on newstime437.herokuapp.com (it may take a few seconds to load because we're using a free account and the web app is not cached for long periods of time).

3 Work left

- Incorporate searching across timelines (and maybe article labels)
- Integrate the labeler component that the other subteam is working on: this includes periodically running the labeler on the server on automatically fetched article links and updating the database. To achieve this we can output the labler/classifier results in a JSON format and have a script read this JSON file and update the database accordingly
- Make the UI for displaying a specific timeline more appealing (for example a graph scrollable through time where events/articles are displayed according to their dates)