EECE 437 Team Project Progress Report

Tarek Tohme, Alexandre Megarbane

April 11th, 2018

1 Overview of the project

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. Here's 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 returned and displayed on the site's page.

2 Progress so far

We wrote a labeler class that uses the Textrazor API to extract some information from a news article from a given url. The labeler returns two lists of keywords: topics and entities. Entities are nouns that appear in the text and that refer to a place, person, date, or other relevant object. Topics are "general themes that aren't explicitly mentioned in the document" (as described in the Textrazor tutorial page). The two lists of keywords are outputted to a file as follows:

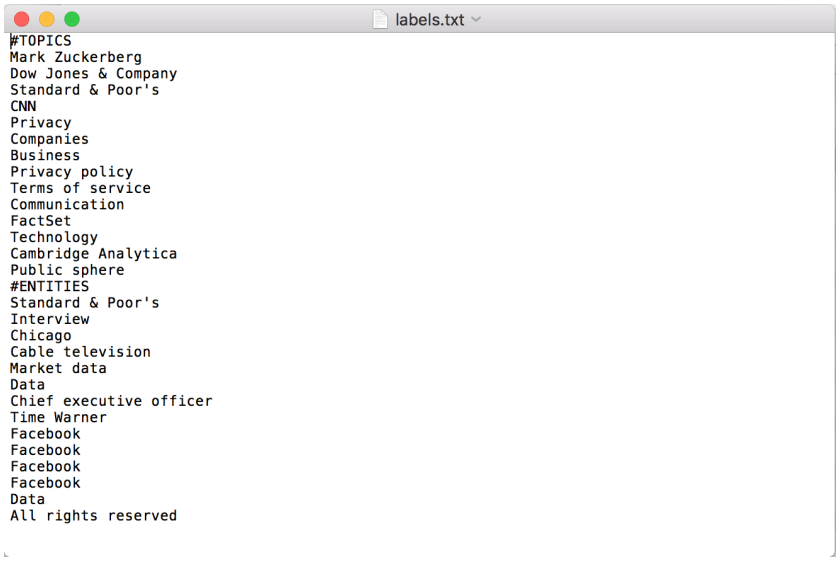

Figure 1: Labeler sample output

We also made some design changes from the first proposal:

* Instead of finding out which story an article belongs to by analyzing the text's meaning in depth, we found out that comparing the keywords returned by the labeler was enough to judge whether two articles belong to the same story or not, without knowing what story it is. A "similarity metric" compares keywords from two lists, and counts the number of common occurrences. Weights are assigned to each kind of keyword (proper noun, common noun, date, location) that appears in both lists, and a score is computed from the comparison. This way, the database is populated by comparing articles to each other, not by giving them a fixed "story" tag.

* We divided the labeling and classifying tasks into two separate components, adjusting to the way the textrazor API works.

3 Roadmap

The other subteam is responsible for the database and interface, so on our part, we still need to write a fetcher component and a classifier component. We plan to do the fetcher using google custom search API, with which we'd search for articles from select news websites. The classifier will be a set of functions that compare two or more articles according to the similarity metric mentioned above. It decides whether they belong to the same news story or not, and sends them to the database.