EECE 437 Project Proposal:

**Extracting the timelines and related events of news stories from online data**

Team members: Karl Hayek, Alexandre Megarbane, Elie Tamer, Tarek Tohme

Sub-team 1: Alexandre Megarbane & Tarek Tohme

Sub-team 2: Karl Hayek & Elie Tamer

Description:

News stories are often short-lived hypes that disappear from mainstream coverage after their attention-gathering potential has been used up. This project aims at visualizing news stories as timelines that show their major events, then using these timelines to cross-reference events that seem to be strongly correlated to gain a better temporal and relational perspective on the news. The user can then access these timelines through a search query on a website. For example, a user enters a news story: “Syria war”. The search results display a timeline of the Syrian war’s major events, and a set of related events and stories.

We plan to do this by first scanning archival news articles, news articles currently being published, and twitter hashtags/google adwords for trending topics. This is the “fetching” component of our project. The information is fetched periodically (e.g. every 10 minutes) from multiple sources. Every source has a particular API (Internet archive, twitter, news channels RSS feed etc.) that we can use to access articles, and we can use Python for web scraping if needed. This component outputs text files with article content (without photos), title, subtitles, date, location if available.

This information is then fed to an indexing component, which generates event objects from temporal information in the files. Each event is assigned three labels: event title, event date and story title. We can do this using textrazor, a NLP API that uses machine learning to extract information from text it is given (from keywords, word frequencies, links). The labeled events are then filtered by relevance (only the most recurring ones are kept) and stored in a database. Note that an article can belong to multiple timelines (e.g. article on Trump and his Wall could be placed in a “US/Mexico border conflicts” timeline, as well as in a “Trump Presidency” timeline), and that there could be multiple events mentioned in an article.

The database component stores a list of timeline objects, each belonging to a story, and a list of event objects indexed chronologically. Event objects contain links to the articles that reference them, and timeline objects contain pointers to event objects. By searching for a story, relevant timelines are returned as a result, and the user can then see each major event of the story in the timeline, and access the articles that mentioned it In addition, if events are part of other related stories (have many events in common for example, or similar titles), the stories are suggested to the user.

After they've been labeled and sorted, we can use a correlation metric between events to judge if they’re related in the news. This metric could incorporate their co-occurrences in timelines, the amount of search results they generate when searched together on google, or the similarity of their titles.