CS51 FINAL PROJECT DRAFT SPECIFICATION

ArOund

The Team

Andrew Malek
amalek@college.harvard.edu
Ivan Cisneros
ivanalexiscisneros@college.harvard.edu
Marc Abousleiman
marcabousleiman@college.harvard.edu

TF: ***Winnie Wu***

Brief Overview

With ArOund, our goal is to create an efficient representation of current events that allows the user to easily identify news stories based on their location and "importance". Many of us receive news updates from different sources in a linear fashion. What ArOund aims to achieve is geographical browsing of important news, giving the user a dynamic access to information. We plan on crawling Twitter, looking for tweets from reputable news sources that fulfill certain criteria, in order to establish an efficient and sophisticated ranking of news events. Ultimately, it would be ideal if we were able to represent our algorithm in an iOS map app.

Feature List

Core Features:

Crawler: *Basic Twitter Crawler implementation*
Graph: *(limited to US). Insert tweet into appropriate 2-3 Tree representing the relevant city*
PageRank*: Importance rank of event determined solely by time*

Cool Extensions

iOS app interface: *iPad map app to visualize our results*
Crawler: *Add filtering algorithm for irrelevant words*
Graph: *Recognize that several distinct tweets are part of the same event and sort accordingly. Scaling up to countries and continents (not only US).*
PageRank: *Add more complicated factors to determine overall rank, (e.g. # of reputable sources tweeting about event, identifying keywords, etc.)*

Technical Specification

Modularization:

- ***Crawler* -** Will crawl Twitter looking for event tweets that fit certain criteria (keywords, location, twitter sources, etc.)
- ***Graph*** – Will create sets of the crawler's results, specifically intersecting sets of similar key words and sources.
- ***PageRank algorithm*** - Will use the graph module to rank events based on prespecified criteria (e.g. the number and quality of sources tweeting about an event, temporal relevance, use of certain buzzwords which indicate importance). The ranking algorithm will not only help in deciding which events to actually display on the visual interface, but also how the events will be displayed; the map markers will vary in color and size depending on the importance ranking.
- ***iOS App interface* -** Map will be responsible for displaying locational information and the features related to location (such as radius size and its correlation to the importance of the event).

Module Subsections

Crawler

- ***Server Cron Job*** - Will repeatedly execute a python script (probably every 5 seconds), therefore updating our graph by adding new nodes representing new world events, if not updating the score of preexisting events nodes.
- ***Twitter Streaming API*** (https://dev.twitter.com/docs/api/streaming) - Crawler will make various HTTP requests to Twitter and extract the relevant information.
- ***Keywords Dictionary* -** Will be the used by the crawler to create the HTTP requests
- ***Tweet Filtering* -** Filters irrelevant words (prepositions, very common words, etc) before storing the event in memory as a list of keywords.

Graph

- ***Balanced 2-3 "City" Trees* -** Every node of the 2-3 tree will contain the tuple *(city_name, (lat, long), pointer_to_event_tree_in this city)*
- ***Tweet to Event Conversion* -** tweets are converted to events (by comparing tweet keywords obtained from the Crawler to pre-existing events' information and deciding whether the tweet belongs to an existing event or is a new one), then added to the appropriate tree.

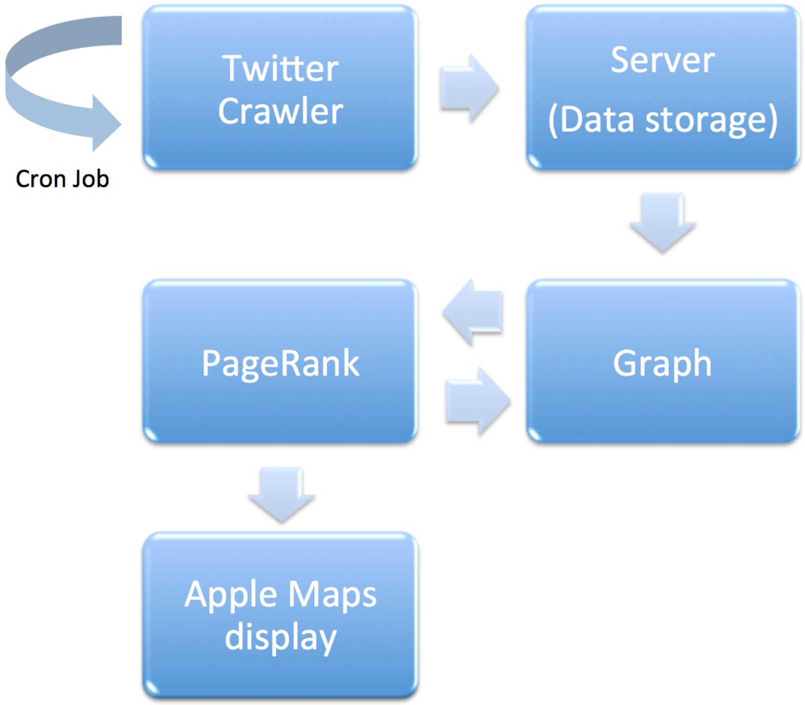
PageRank

- Intersecting "events" trees that are pointed to by the cities represented in the map view.
- Will use the data stored in the event object to rank events by order of "importance". Events will be identified or joined together by keywords and locational data via graph module.
- Importance rank is determined by keywords used in event description ("destruction", for example, will be given a higher ranking score than "celebrity"), as well as by the number of quality sources that tweet about this event (CNN and Al-Jazeera tweeting about the same event will give it more legitimacy, and thus a higher ranking).
- Event objects are matched with the relevant tweets, which will then be displayed on the map next to the locational marker.

iOS App interface

- ***PageRank* -** App obtains results from PageRank and displays properly sized and properly colored annotations on the map.
- ***MKMapView* -** Objective C's built-in delegate for displaying maps on iPhone and iPad devices.

Process Flow



Next Steps

- Getting a better grasp of Python / Objective C
- Setting up git to work efficiently as a group
- Work Distribution will tentatively be:
    - Marc: Twitter Crawler & iOS app
    - Andrew: Graph Structure & iOS app
    - Ivan: PageRank algorithm & help with other sections