EE239AS Project 2

Crawling and Data Collection on The Web

Winter 2015

Introduction: The first step in any big data project is data collection. The nature of data could vary in different applications; it could be text, image, audio, etc. and ranging from authors' data in academic publications to discussions and interactions in an online forum on political issues. Depending on how hard it is to get data, data collection could be the most time-consuming and challenging part of a project. Crawling[1] is a very popular way of collecting data from websites and essential to a lot of applications. Search engines like Google and Yahoo all implement a crawler to systematically find desired information on the web.

Twitter[2] is a good example of an online platform where people can share updates and thoughts and discuss them with others. Any user on Twitter can make posts -called "tweets"- that mainly contain short pieces of text limited to 140 characters. All tweets are public, but each user has some followers that will see the tweets made by the user in their feeds. Others can read and mark a tweet as their favorite, or re-tweet them to let it reach to more people through their own followers.

In this project, we crawl Twitter to collect useful data on an event, from the public opinion. There are several approaches to crawling and the order you navigate through data; here we organize data based on topics (hashtags) and timestamps. Tweets are often marked with a tag that starts with a hash sign, e.g. #Oscars2015 or #UCLA. These hashtags specify an event or subject that the tweet is related to. One can then search for a hashtag on twitter and find tweets related to a specific matter.

1) Reserve a slot of hashtags and timeframe from the Google Spreadsheet[3]. The hashtags are all related to the 2015 Super Bowl. For one of the hashtags in the slot, write a code that searches for the hashtag and saves the top 5 results returned. Use the Topsy API[4] and save the "tweet" field of the results in a txt file named `top_tweets.txt`. If you load the JSON response of Topsy API in a Python object named `ret`, the tweets are stored as a list in `ret['response']['results']['list']`. Write each tweet item of the list in a single line of the file. Your file should look like the sample uploaded in the footnote link[5] which shows the results of a search for #Oscars2015. (If you open the file in an editor you might see messy text, because the lines are very long. You can open the file in Python and see how each one of its 5 lines look). In your report, write the tweet text, the user posting the tweet, and the posting date for all these 5 tweets.

---

[1] http://en.wikipedia.org/wiki/Web_crawler
[2] https://twitter.com
[3] https://docs.google.com/spreadsheets/d/1gK-li-gsKhm42CliDmbKfTK9t3sK2eiv5OjsQJaStZY/edit?usp=sharing
[4] http://api.topsy.com/doc/
[5] https://www.dropbox.com/s/xoub8mzmtokseyr/tweets.txt?dl=0

2) Write a script that extracts all tweets containing any of the hashtags in your slot, and are posted in the specified timeframe (Note that the times are all written in Pacific Standard Time). Store all tweets in a file named `tweets.txt` in the same format as the file in part 1. Note that the maximum number of results the API can return for a search is 500 (which you have to manually set it to do). Therefore, to cover all tweets in the entire timeframe, you should make several queries to the API for small time windows, possibly as short as a few seconds, that contain less tweets posted in them than this maximum number of results. If a query returns the maximum number of results, you should divide the query time window and run them again so that each query returns less number of results than the maximum limit. Remember to also have the parameter field `include_metrics=1` in your query, as you will need the metrics for next parts.

Store a log of all your searches in a file named `search_log.txt` in the following format for a query in each line:

```
[query_string]    From: [start_date]    To: [end_date]    No. of Results:[results_number]
```

For example:

```
#SuperBowl      From: 2015-01-31  15:45:30      To: 2015-01-31  15:45:35      No. Of Results: 167
#SuperBowl      From: 2015-01-31  15:45:35      To: 2015-01-31  15:45:45      No. Of Results: 135
#superbowlads   From: 2015-01-31  15:45:35      To: 2015-01-31  15:45:45      No. Of Results: 135
....
```

3) Find the total number of tweets for each hashtag in your timeframe and plot these numbers (e.g. a bar graph that shows the number of tweets for each hashtag). For the most popular hashtag in your results, plot the rate of the number of tweets containing it in time, over your given timeframe. This should be a plot of "number of tweets per second" over time.

4) For every unique tweet in your data, find how many retweets it has. Plot the number of tweets retweeted k times, over number of retweets k, i.e. for each value of $k = 1,2,3, \dots$, show how many tweets exist that have k retweets. Draw this plot in linear scales once, and another time in log-log scale. Can you fit a line to any of these plots?

5) For the first and second most popular hashtags, see if there is a correlation in their trends. To see this, at each point t in time, find the number of tweets per second for each of the two hashtags. Then using these pairs of values obtained in different sampling times t, plot the time rate for hashtag 1 tweets versus time rate for hashtag 2 tweets.

6) Write a parser that is able to read one line of your data in tweets.txt and either output or print out the tweet's post date, text, number of retweets, and the user posting it.

Submission: Please submit a zip file containing your codes, the 3 output files (top_tweets.txt, tweets.txt, search_log.txt) and a short report to ee239as.winter2015@gmail.com . If your zip file is too large to email, you can upload it on Dropbox (or Box at ucla.box.com) and send the link to this email. You can use any programming language but there will be samples of Python code given to you for crawling and parsing that you can use and modify. If you had any questions you can send an email to the same address.