CS6742: NLP and Social Interaction, Spring 2011
2/3/2011: Assignment A3, acquiring data from Twitter

Introduction In assignments A3 and A4 we are going to take a hands-on approach to the question, "what triggers a specific type of response in a conversation?", where a "specific type of response" can be, for example, an indication of amusement or of contempt.

Assignment A3 is about getting the necessary conversational data for this task, as a means of getting experience with language-data collection. Specifically, we will work with Twitter.

You should work alone on this assignment. (The reasons are that we would like to get maximal variety in responses and perspectives in the solutions submitted and that we would like everyone to get practice with this kind of activity.)

Make sure to read the **important points** listed at the end of this handout before starting to work on this assignment.

In this document, the blue URLs should be live (clickable).

Due dates: **10pm Wednesday Feb 9th** and **10pm Monday Feb 14th** (details explained below).

Task 1 Start by defining the response type you wish to target and by selecting indicators for that respective type of response. For example, if your target is "amusement", potential indicators could be ":)", "LOL", or "haha".[1] There are multiple slang and acronym dictionaries available online that could provide inspiration for type of response and indicators (e.g., http://www.abbreviations.com/acronyms/CHAT).

Collect a set of at least 2000 tweets that are replies to some other user and that contain an appropriate indicator, using repeated calls to the Search API, which is documented at both of the following URLs:

http://dev.twitter.com/doc/get/search
http://apiwiki.twitter.com/w/page/22554756/Twitter-Search-API-Method:-search

and by automatically filtering the results to make sure the tweets you selected do indeed contain good indicators for the type of response you choose to target.

The easiest way to get an initial feeling for how requests work is through a browser. For example, the following link should return information about 15 tweets containing the word "language":

http://search.twitter.com/search.json?q=language

Hints:

- One possible high-level approach (among others) to this task is to use the `rpp` parameter to obtain 100 tweets per request, then use the `max_id` parameter to retrieve the "next" 100 tweets.
- Make use of the "lang" and "to_user_id" parameters.
- The precise way you eventually end up sending requests to the Twitter APIs for this assignment will depend on the programming language you use (or the library you may chose to employ; see the "Important points" section below).
- Note that the type of indicators that you can use is limited by the type of requests you can make (for example, one can not easily search for syntactic patterns).

[1] We welcome other response types, but you may use "amusement".

Write the 2000 tweets to a plaintext file named `A3task1output.txt`, one tweet per line (please use UNIX line-breaks throughout the assignment)[2], each line formatted as:

id⟨TAB⟩text

That is, the tweet id and the text of the tweet should be separated by a tab character.

Also create a short plaintext or pdf file `A3task1description.txt (or .pdf)` describing: the response type you chose to collect, the respective indicators, the filtering you applied to the tweets returned by the API (if any). Also, an optional question: was there any type of response or any indicators that you would have liked to use and were unable to because of the Search API limitations?

Submit both files to the course CMS, http://cms.csuglab.cornell.edu, under assignment "A3, milestone 1 (a3.i)". Due **10pm Wednesday Feb 9th**. (These are *not* the only files due then, see below.)

Task 2 For each of the 2000 reply tweets gathered in Task 1, retrieve the text of the respective initial tweets (i.e., the tweets to which your reply tweets are replies to) using the Twitter REST API:

http://dev.twitter.com/doc/get/statuses/show/:id

(Note that "status" is another term for "tweet".)

Note that in some cases the initial tweet can not be retrieved; ignore those cases. If at the end of this process you get fewer than than 500 initiator-tweet pairs[3], go back to Task 1.

Hint: use the tweet id to connect Tasks 1 and 2.

Write the results in two plaintext files:

- `A3task2initial.txt` should contain all the initiator tweets, one tweet per line in the format: id⟨TAB⟩text
- `A3task2reply.txt` should contain all the reply tweets in the form: id⟨TAB⟩text

Each line in the second file should be a reply to the corresponding line in the first file (e.g., line 321 in A3task2reply.txt is a reply to line 321 in A3task2initial.txt).

A *sample* of these two files (named `A3task2initial_sample.txt` and `A3task2reply_sample.txt`, 30 lines each) must be submitted the course CMS, http://cms.csuglab.cornell.edu, under Assignment "A3, milestone 1 (a3.i)", before **10pm Wednesday Feb 9th**. The *complete* files (500 lines or more) are due **10pm Monday Feb 14th**, also through CMS, under Assignment "A3, milestone 2 (a3.ii)". This will give you 5 more days to collect the rest of the data (and given the rate limitations explained below, you are likely to need these days).

Important points to keep in mind

(a) Rate limitations: http://dev.twitter.com/pages/rate-limiting
Not complying with the rate limitations will result in your IP getting blacklisted and thus making it impossible for you to finish this assignment on time. Note that both the Search API and REST API have rate limitations. While for the REST API there is a clear limit (one can hit the API 150 times per hour) and an easy way to track your limit status: http://api.twitter.com/1/account/rate_limit_status.json (which should be called frequently by your code), that is not the case for the Search API. To play it safe, do not hit the Search API more than 150 times per hour (which should be enough since 150 hits should get you 15000 results).

[2] One utility for such conversion: https://ccrma.stanford.edu/~craig/utility/flip/

[3] 30 pairs are sufficient for the first due date.

(b) When interacting with the API, use exception clauses. Many things can go wrong. When handling the exceptions, keep (a) in mind.

(c) Design your data gathering so that it can be easily restarted, without losing what was already collected.

(d) For the most popular programing languages there are many Twitter Libraries that can be used to send requests to the API: http://dev.twitter.com/pages/libraries .

(e) To collect as many examples as possible, have the data processing running for as long as possible between the two deadlines. More data will represent an advantage in A4.