Influential Tweeple

INFO290-SC Social Computing – Prof. Irwin King – Fall 2011
Arthur Suermondt (arthur@ischool.berkeley.edu, 3 units)
Chulki Lee (chulkilee@ischool.berkeley.edu, 3 units)
Ram Joshi (ram@ischool.berkeley.edu, 3 units)

Progress report

Summary update

After presenting our project proposal we made some adjustments to the initial strategy based on the feedback we received. We decided to first focus on designing a good relevancy filter before attempting to tackle the more complex problem of a recommendation engine.

At this point we are ready to collect, store and analyze Twitter data for any given user. Some sample results from our crawler are available in appendix A. The data collection, parsing and storage steps, as listed in the Milestones section of the original proposal, took slightly longer than expected. We had to deal with several changes and, as this is the foundation for the remainder of the project, we wanted to make sure we had a solid crawling script.

The crawler is written in Python, leveraging several modules to access and store the data. Using OAuth authentication we access the Twitter API for the end-user to crawl their profile and timeline. We then store those results in MongoDB. The reason why we chose MongoDB is 1) it has no schema so that we don't need to determine what to store when crawling and 2) it is more scalable for storing huge amounts of data and we don't need to query over them while crawling. We use Twitter's Stream API as well as the REST API. In order to collect real-time samples, the Stream API is used. The REST API is used to collect previous tweets of users.

Plan

Using the collected data we came up with several strategies to determine a user's interests, which we could then use to determine the relevancy of individual tweets to this user.

A first step was to get a general sense of the data we had collected using our crawler, using a script to analyze these crawled tweets. A sample result is included in appendix A. Note that since we tested fetching all tweets for just a few users, most tweets came from them and the sample results are somewhat limited. An interesting observation that stood out to us was that there are no favorited tweets or retweets - we are figuring out whether this is specific to our sample dataset, a general trend, or a bug in our crawler.

We came up with 5 initial algorithm strategies to determine the user's interest profile. Depending on the accuracy of the relevancy filter we might add or remove some of these.

1. Term frequency analysis of tweets
2. User's language
3. User's location
4. Twitter profile information
5. Entity extraction on User's website and potentially tweets

The milestones section of this report has been updated to reflect the completed tasks as well as our current plan for the remaining tasks. In addition to the general table of milestones we also added a slightly more specific list of tasks (as used while implementing a first prototype). We will update our project page on the wiki with links to our code repositories and demo pages as soon as we have those available for testing.

Issues

Terminology

The Twitter terminology can often be confusing, something we noticed while developing the crawler and designing the analysis tools. We decided to come up with a list key terms used for our project to make sure we were designing and developing with using the same concepts. Naturally this list is non-exhaustive and currently features mostly terms used in the crawling / data collection stage.

Key Definitions

Please note: these definitions can use Twitter terms with a different meaning

- Unique User Stream: all of a user's own tweets
- Aggregate User Stream: all of a user's own tweets (unique user stream) + all of the tweets from the people they're following
- Twitter Stream: all of the tweets posted to Twitter
- User: a single user of Twitter identified by a unique Twitter handle
- Status: a single Tweet with a unique identifier

Tweet length

Several algorithms don't produce reliable results on the short length of a single tweet. Depending on the results as we go along we will apply some algorithms to an aggregated set of tweets to improve reliability and accuracy. An example is determining the user's language. This is hard to do and unreliable based on just a couple words of a single tweet.

Abstract

Analyzing influential sources of information in a social network or a hyperlinked corpus of documents such as the web is both a critical and interesting problem. We want to focus on identifying valuable sources of information within the Twitter ecosystem. This provides unique challenges but also presents easier methods of analyzing a network of information sources due to the distinct design of Twitter. For instance, the concept of retweeting is a unique information penetration process that also preserves the origin of the information source allowing rapid discovery of good information sources. We want to combine such Twitter specific use cases with traditional network analysis approaches to create a useful tool for recommending both Twitter-internal and external sources of information based on the interest keywords of a user.

The primary motivation of this exercise is mainly our personal frustration with separating signal from noise within Twitter and on the web in general. Users can get flooded with irrelevant updates from Twitter. We want to use the structure of the Twitter network itself to discover influential, like-minded and relevant information sources.

Our solution involves data scraping/mining and offline analysis of tweets for interest topic extraction, news origin analysis, external hyperlink analysis and use of document and source ranking algorithms.

1. Introduction and Background

Our project is based on the broader mission to enhance the Twitter user experience by making it more relevant to each individual user. For the scope of this project we will focus on several more specific problems related to this broader mission.

Inspired by Jure Leskovec's work on the diffusion patterns of blog articles and news items we aim to implement a similar application in the context of Twitter. This application allows Twitter users to quickly and effectively find the most interesting and timely Twitter sources for their particular interests. A common problem in the use of Twitter today is the increasing amount of noise, ranging from tweets from interesting sources but irrelevant to the user's interest to tweets on altogether irrelevant topics (such as bathroom use). Although most users are usually able to find and narrow down the group of Twitter users they follow to the most relevant ones, this is a slow process, involving constant adjustment and a great deal of mental effort. Phenomena such as "#FollowFriday" provide basic workarounds to solve this problem without using any kind of automated analysis[10].

1.1 Specific problems

For this project, we will focus on solving several specific problems. Twitter users face these problems daily and are currently trying to work around these issues manually. We believe that algorithms and techniques of social computing can help them, thereby greatly improving their user experience.

First, finding the most interesting users to follow. The Twitter term "following" is equivalent to connecting users in a single directional way, where the other users has the option to reciprocate this connection. These connections enable users to view a stream of information coming from all the users they are "following", it is critical to find the most interesting users to follow to avoid information overload and to expand one's network. Although a "Who to follow" feature is provided by Twitter, it does not reflect individual interest well. Furthermore, suggestions are limited to Twitter users who are relatively "close" to the current user, avoiding users that are potentially the most relevant sources but are more "hops" away. We believe a more personalized, advanced suggestion system is needed, providing users directly with the most relevant people to follow, skipping over the additional intermediary "hops".

Second, creating relevant lists of followers based on interests. Just suggesting "Who to follow" is not enough, because the "following" relationship may imply various intentions. Therefore, Twitter's list feature, which allows more contextual categorization of followed users, can be useful for grouping followed users based on various topics. Currently, lists can only be created manually, a cumbersome process subject to similar levels of mental effort needed to find good people to follow. Assisting users by automatically grouping, filtering and ranking their incoming tweet streams can greatly improve the effectiveness of using Twitter.

Third, finding relevant external sources, such as blogs or news websites. Even though it is becoming ever more common for news to break on Twitter, the best sources to follow are not always on Twitter. Regularly, news will come out through hundreds, if not thousands of Twitter users, posting a tweet based on an external source. This makes it very difficult to determine the most relevant user to follow, and it may be more useful to be able to follow the external original source directly.

2. Proposed Solution

2.1 Use cases

In order to solve the problems discussed in the previous section, we plan to build a web application drawing on information provided by back-end analyses. Our proposed solution is described using four use cases, to be implemented in the final application:

1. build interest profile of the user
2. recommend relevant users to follow (the 'sources')

3. generate "smart lists", tweets based on topics of interest
4. recommend relevant websites/blogs to follow (the 'external sources')

For these use cases to be successful, we need to collect data, extract information from them and apply various analyses on them. In the following sections, we clarify which data to collect and how to collect it, as well as laying out an initial analysis strategy. Figure 1 shows a high-level overview of the use cases and how they interact with Twitter users, sources and each other.

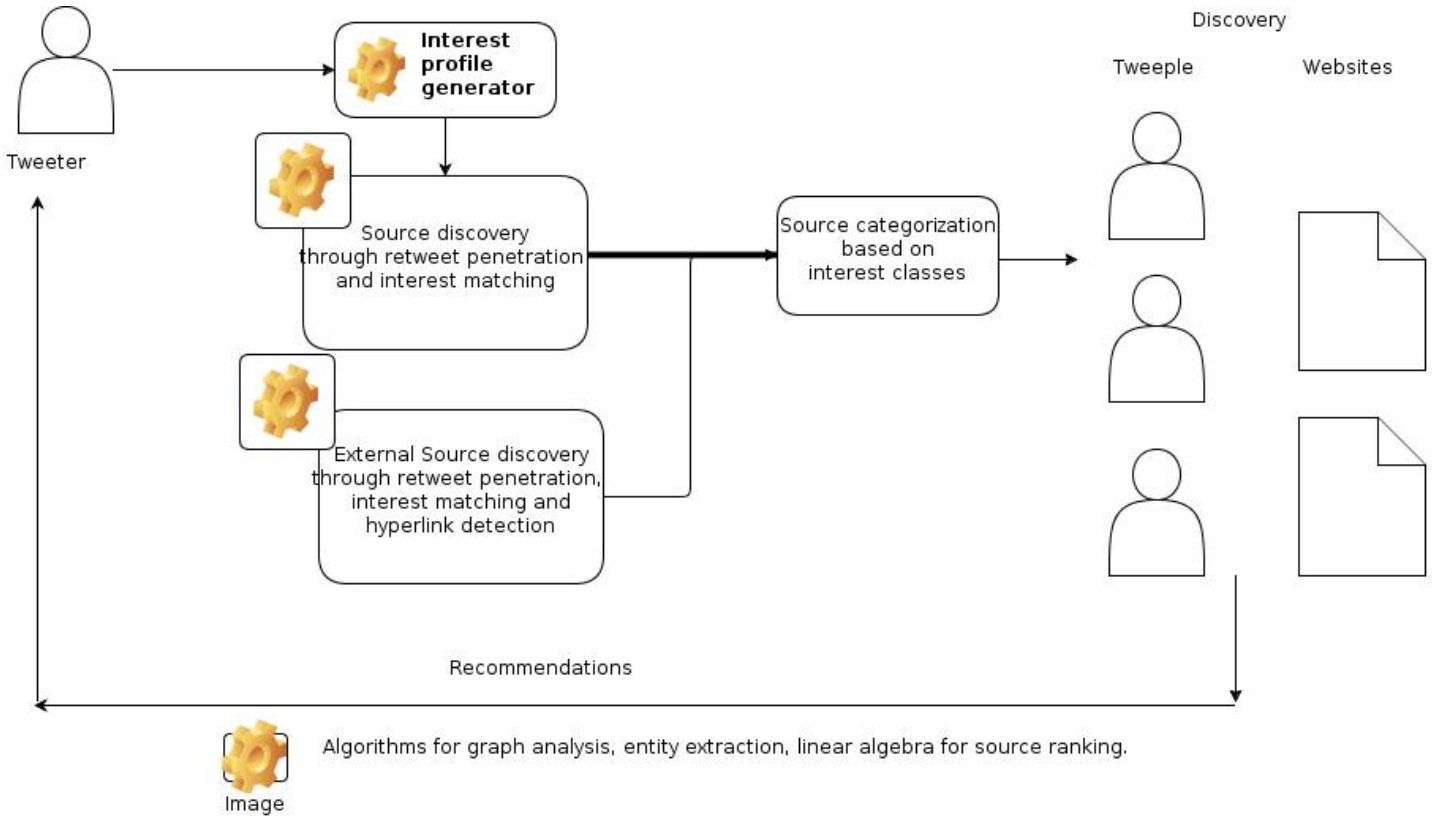

Figure 1 - high-level use case overview

2.2 Data collection

First of all, we need to determine the **interests of a user**. Identifying these interests can be done by analyzing their tweet contents, among other strategies. For example, a user follows other users, maintains lists, retweets other tweets, flags some tweets as favorite and so on. Note that such activities may not reflect interests of the user: for example, the user follows others because they have a personal relationship with the other user, not because their interests match with the user's interests. Because of such diversity in usage of features, we plan to employ a "trail and error" approach.

We then need to determine **which users and which sources provide information** about

the identified relevant topics. An initial strategy could be to traverse a series of relevant retweets in order to find the original source for that tweet. Some related issues include:

- Multiple sources: news items can be disseminated by multiple users
- External sources: news items could also have originated from an external source

In order to improve the recommendations, we need to determine **which users and sources are 'better'**. We want to use several factors for determining the quality of the source:

- timeliness
- coverage: sources have to cover the interest well (not too narrow, broad)
- selectivity: sources have to choose reasonable number of results (not too many, too small)
- influencing power: some source are more influential, potentially making them more worthwhile to follow
- originality: original sources vs. intermediary/brokers
    - find the original tweet for any retweet
    - find the original "source" rather than tweet mentioning them

A combination of these factors will be necessary to achieve an optimal result. For example, suggesting only original sources may result in losing opportunities of getting insights from the news. Some may want to have very selective results but others may not. Therefore, it is important to consider all factors and let users adjust those factors. Having the analyses algorithms learn from the users in this way will further optimize the overall experience.

2.3 Analyses / front-end

After acquiring the needed information, the application should be able to:

- compute a recommendation based on the collected data
- suggest computed recommendations and letting user adjust these suggestions
- generate "smart lists"

3. Technology

In this section, various tools for building the solution are listed. Further experimentation with tools, algorithms and technologies will be needed to decide on what to use in the final product. For an initial planning see the section 4, milestones.

Data collection

- Collect tweets using Twitter stream API
- Use tweet archive or dataset, if possible
- Scrape information from Twitter that is not available through the API
- Crawl web pages linked in tweets (external sources)

Data storage

- Store data as unstructured collection sets for NLP analysis
- Store data in a structured format to enable advanced querying

Data selection

- Filter tweets by hashtags (#) or terms
- Specific period, users
- Detect topics in tweets using NLP (140 characters are maybe too short to use NLP)

Data analysis

- NLP for processing text: slang, mistype, vocabulary problems
- TF-IDF for calculating similarity of contents
- Document/source ranking for identifying influential sources
- Filtering influential sources based on relevant interest categories

4. Milestones

As a general guideline for the planning of this project we propose the milestones as shown in table 1. While these milestones provide an initial planning we intend to use an iterative approach where features from previous milestones can be changed or improved upon in later milestones.

Completed milestones are marked in green, upcoming milestone are white.

Table 1. Project milestones, course deadlines underlined

M0, Sep 14 **Preparation**	• Build a team • Pick up a topic • 9/14: Project proposal submission
M1, Sep 23 **Data collection and system set-up**	• Survey related technologies and algorithms • Decide the scope of tweets to collect • Build a draft model for network analysis • 9/14-9/23: Project proposal review • 9/23: Project discussion and presentations
M2, October **Start data collection and parsing/storage**	• Start collecting data: tweets, linked external sources • Parse and store data • Build first prototype • 11/01: Mid-term project report submission
M3, November 15 **Interest profile, relevancy filter** **Front-end interface**	• Design & develop front-end (UI) • Implement interest profile algorithms • Filter tweets based on relevancy to interest profile
M4, November 30 **Recommendation engine** **General optimization**	• Design & develop recommendation engine • Optimize system components • Improve usability of the system
M5, Early December **Final phase**	• Finalize system • Final report • 12/2 & 12/9: Final project presentation

Specific Tasks List (for tasks up to and including milestone M3)

Crawler & Relevancy filter

1. Crawl data [python + mongodb]
    - Unique User Stream (store from which User's timeline tweet was acquired)

    - Aggregate User Stream of given User (store from which User's timeline tweet was acquired)
    - Profile text
    - Location of User (user['location'])
    - Location per Status (tweet['geo'])
    - Website of User (user['url'])
2. Determine interest profile of User
    - run term-frequency style analyses on Statuses
    - determine language of Status
    - location-based: location in user profile / geo-tagged Status
        - if > 10% of Statuses have geotag information: use in analysis
    - key terms from profile information
    - run entity style analysis on User's website (Concept mining / Terminology extraction)
        - Evri entity extraction http://api.evri.com / Alchemy entity extraction
        - Term Extraction Documentation for Yahoo! Search
3. Filter Statuses crawled in (1) by determining whether they match interest profile from (2)
    - rate Statuses with a pre-determined range of values
    - display Statuses in ranked order
4. Improve performance of above
    - possible issues: ambiguous terms, informal words, acronym
    - computational complexity

User Interface

1. add OAuth authentication interface for end-user
2. display filtered timeline for user with threshold and sorting controls

References

1. Twitter API Documentation
2. Twitter Streaming API
3. Discovering Who To Follow, Twitter Blog, 2010/07/30
4. Suggested Users, Twitter Blog, 2009/03/25
5. Tweet Preservation, Twitter Blog, 2010/04/14
6. Who to follow: Browse Interests
7. Twitter Unlocks its Valuable Web Analytics Data with Free Dashboard, ReadWriteWeb, 2011/09/13
8. Facebook Releases Smart Friend Lists to Counter Google+ Circles, ReadWriteWeb, 2011/09/13
9. Facebook Officially Unveils Smart Friend Lists, TechCrunch, 2011/09/13
10. #FollowFriday, Mashable, 2009/03/06
11. Jure Leskovec: Stanford CS / SNA professor

Appendix A. – Sample results from crawling session

```
================================================================================
Tweets
--------------------------------------------------------------------------------
- tweets: 18135
- oldest tweet at: Fri Apr 16 10:27:32 +0000 2010
- latest tweet at: Wed Sep 30 09:40:56 +0000 2009

================================================================================
Tweets metadata
--------------------------------------------------------------------------------
- tweets with geo: 113 (0.62%)
- tweets with place: 316 (1.74%)
- tweets with favorited: 0 (0.00%)
- tweets with retweeted: 0 (0.00%)
- tweets with user_id: 18135 (100.00%)

================================================================================
Tweets by language (based on user's profile)
--------------------------------------------------------------------------------
  en     11071
  es     2208
  pt     1994
None     1166
  ja     1118
  fr     159
  tr     96
  nl     92
  de     56
  ru     52

================================================================================
Tweets by guess-language
--------------------------------------------------------------------------------
None     12174
  en     1835
  pt     828
UNKNOWN     532
  ja     440
  es     413
  ko     292
  ca     178
  id     162
  it     112

================================================================================
Tweets by user_id
--------------------------------------------------------------------------------
759251.0        800
```

22117600.0 332
71044761.0 34
373609808.0 3
89054783.0 3
109740608.0 3
58098124.0 3
392259196.0 3
244344468.0 3
112917813.0 2

==========
Users

- users: 16752
- geo enabled users: 4418 (26.37%)

==========
Language by user
----
en 10934
es 2190
pt 1954
ja 1106
fr 155
tr 96
nl 89
de 54
ru 51
it 48

==========
Language of description by guess-language

None 12977
en 1312
UNKNOWN 818
pt 406
ja 376
es 254
ca 96
it 67
id 46
fr 35

==========
Events

- events: 657 (3.62 of tweets)
- delete events: 657 (100.00%)
- other events: 0 (0.00%)