INFO 256 Fall 2013 — Final Project Proposal

Khoa Tran

December 10, 2013

1 Idea

A Facebook Topics Extraction and Similarity system, which would allow users to identify important topics from their news feed and/or facebook groups and find similarities among posts.

How I got this idea: Facebook Groups use "exact-match" search, which might not be particularly helpful when a user wants to search for a certain topic but doesn't know the exact term. I often found myself trying different terms like "declaring", "minimum GPA", etc. to look for any posts related to the new major declaration policy implemented starting Spring 2014 for the Computer Science major. A simple TFIDF model with Cosine Similarity could make a huge difference in finding useful information for most Facebook users based on the post content.

2 API Usage

The Facebook Graph API currently allows data from[1]:

- All public posts
- People
- Pages
- Groups

which would be most of the data required for this project. For example, using python-facebook, I could get all of my friends by executing

[1] `https://developers.facebook.com/docs/reference/api/search/`

```
>>> graph = facebook.GraphAPI(oauth_access_token)
>>> profile = graph.get_object("me")
>>> friends = graph.get_connections("me", "friends")
```

and from there, retrieve their public posts for analysis.

I have not been able to find any information about how much data the API allows me to access, though I don't think there is any hard cap, since most Facebook apps constantly access users' information and post to their walls 24/7.

3 Project Goals

- To identify important keywords from a user's news feed and/or facebook groups, and classify them into different categories. NLP Concepts applied: Stemming, Lemmatization, POS Tagging, Chunking/Noun-phrase Extraction, Classfication.
- To find similar posts related to a user's interest. NLP Concepts applied: TFIDF, Cosine Similarity, perhaps Latent Semantic Analysis.
- Unlike Twitter, which has a 140-letter limit, most Facebook posts are often spelled out in full. I believe this fact would greatly help any stemming/lemmatizing algorithms applied on Facebook posts. This would make sentiment analysis a great addition to the project, though it is not as important as the two main goals above.
- Finally, visualize the search result from Facebook and from the NLP approach in this project side by side, and compare and contrast them.

4 Evaluation

I plan to compare and contrast the current search functionality that Facebook has, versus the NLP approach that I decide to tackle in this project. In the end, I hope to display the searches side by side and conclude whether the NLP approach yield better result.

An example search using the Facebook API would look like: `https://graph.facebook.com/search?q=waterm`
In this project, I plan to retrieve the data (perhaps saving locally first before moving on to real-time data) and process the search result in the backend. It might not be as fast as Facebook search, but comparing the results would be interesting. The two kinds of search would then be compared and displayed side by side.

This would be especially interesting when applying to students searching unofficial university Facebook groups, since oftentimes Facebook search functionality for groups can be

frustrating and doesn't exactly yield the expected result. The answer to several questions posed by new students (like GPA cutoff or major requirement) can almost always be found from a search, so hopefully this would provide students a better alternative to look for helpful information from a group page.

5 Roles

Unfortunately, there were some last-minute changes among previous team members, and I will be tackling this final project by myself. I recognize it as a challenging and ambitious task, but by no means detrimental. By the end of this project, I hope to develop a stronger understanding of many NLP concepts covered in this course, and to have a better idea of working with real-world, messy data like Facebook posts.

A non-exhaustive list of tasks includes: create a Facebook token to access users' posts, build a robust chunker to extract important noun phrases, use the Bag of words and TFIDF model to create a large matrix of documents for computing similarity, and visualize and output the result in some format, hopefully all in a single web application, for the final presentation.

6 Resources

To first acquire users' posts, the Facebook Graph API would be very essential. First, I plan to cache some data for locally use using Python's cPickle, and then allow users to log in to the app with their Facebook accounts for live contents. After that, some important NLP tools I would like to try out include:

- NLTK (`http://nltk.org/`)
- Pattern (`http://www.clips.ua.ac.be/pattern`), which has some great built-in TFIDF functionalities
- TextBlob (`https://textblob.readthedocs.org/en/latest/`), which includes simple noun phrase extraction and sentiment analysis
- scikit-learn (`http://scikit-learn.org/stable/`), for machine learning and classification tasks

For visualizing the output, Matplotlib is without a doubt the first choice. I would also want to look into NetworkX for creating similarity graph, and Pandas for displaying and visualizing Facebook posts in a time series.

7 Schedule

- **Week of Nov 3**: Since the project proposal is due this week, it's not recommended that we start implementing immediately without first receiving feedback. To that end, I plan to simply create a Facebook token and play around with their API for accessing users' posts, since this part has very little to do with the NLP-part of the project.
- **Week of Nov 10**: Cache the data locally, and build a chunker to extract important noun phrases from sample Facebook posts. Also find features for the classification task.
- **Week of Nov 17:** Carry out the actual classification implementation, and try out the TFIDF/Cosine Similarity algorithms to detect similarities among posts.
- **Week of Nov 24**: Finish up with similarity measurement, and try out different algorithms like Jaccard Similarity or LSA to see if there's any improvement
- **Week of Dec 1**: Sync the engine with live contents from Facebook, and build a simple interface that would allow users to compare Facebook search with the NLP approach. Begin typing up the final report.
- **Week of Dec 8**: Wrap up everything, ready for the awesome final project demonstration, and submit the report!