AP PROJECT 2 REPORT

2014061 - Madhav Varma

2014115 - Vasu Verma

The aim of the project was to create a like predictor for Facebook posts, videos, status updates etc. We utilized the Facebook's Graph API and JSON Parser to accomplish this task.

We used developer.facebook.com and graph explorer tool to mine all the fields necessary for our project. The main field was me/posts? and its further fields were as follows:-

- Type
- Message
- Story
- Id
- Create_time
- Likes
- Names

i.e. we mined the type(status, photo etc.), the message of the post, id and created time of the post and the likes on the post with the name and id of the person who liked the post.

GENERATION OF THE ACCESS TOKEN.

To access all the fields it was first necessary to generate an access token for the user. The user logs in via Facebook OAuth and the app then asks the logged in user to grant the app access to the various fields. Once the requested permissions are granted, facebook then sends the code to the servlet using which the access token in received. Using this Access Token, the user's posts and data related to it is received by implementing the Graph API.

JSON PARSING

The data was in the form of JSON data types, the data arrays were made as JSONArrays and the rest were made as JSONObjects. After nesting of JSONArrays and JSONObjects we obtained the array containing JSONObjects of likes (id and name) whose length was used as the number of likes of that post. For posts which had multiple pages of likes(i.e. >25), a function

GetPageLikes() was called which redirected to the post page having next set of likes which kept on a while loop till no next page was remaining and returned the number of additional likes to that of the original page(URL).

Data Storage And Predictor

First we have made four HashMaps, each corresponding to the four likely types of posts i.e. status, photo, link and video. These hashmaps are String→ArrayList<Integers>. For each word in the message of these posts, say status, we add the word to the hashmap and append the number of likes to the ArrayList corresponding to it. Also only words having size greater than 2 are considered.

For example:

If we have two status:-

- I am totally totally absolutely incredible. (which has 8 likes)
- I am totally incredible and fabulous. (which has 4 likes)

Our statusHashMap would be

{absolutely:[1], totally: [8,4] , incredible: [8,4], fabulous:[4] }

Now if we try to predict the likes of a new status say, “I am totally fabulous”, it would predict the like on the status in the following manner:

$$\sum_{i=1}^{n} \frac{2i}{(n+1)n}(p_{n-i+1})$$

where n = number of elements in the ArrayList<Integer>

p = Elements of the ArrayList<Integer>

But for example if no words have occurred in his previous posts, the program would simply take average of all the posts of that type and print it out as the predicted number of likes because of no relation of the new post to previous ones.

Also just for fun, the program outputs the id's of people who have recently been liking your posts along with the percentage probability of them of liking any of your post.

The code offers two functionalities:-

1. The user inputs a certain percentage that he wants to reserve for prediction purposes and the rest of the posts will be used as the data to predict. For example if user inputs 20, the likes on the recent 20% of posts will be predicted using the previous 80% posts.
2. After the user inputs a certain percentage to be used as data, he has a choice to prompt the program to predict likes on a new post generated by him on the spot.