

CS 297 Report

KEYWORD SEARCH IN SOCIAL NETWORKS

by

Vijeth Patil

(vijeth.patil@gmail.com)

Fall 2011

Advisor: Dr. Chris Pollett

Department of Computer Science

San Jose State University

TABLE OF CONTENTS

1.0 Introduction.....	1
2.0 Deliverables.....	2
2.1 Deliverable 1.....	2
2.2 Deliverable 2.....	3
2.3 Deliverable 3.....	6
2.4 Deliverable 4.....	8
3.0 Conclusion.....	10
4.0 References.....	11

1.0 INTRODUCTION

The idea of this project is to access data from social networking websites and create a search engine of the collected data. As social networks grow, more and more important data is accumulated inside social networks making it more essential to provide search functionality of this data. The most widely used social networking sites are Facebook, Twitter and Del.i.cio.us.

In this semester, I worked on various experiments for accessing data from social network websites and index the data collected into a search engine. This report provides a brief description of the work done in CS297 Preparation for Master's writing project. The work was divided into four deliverables: The First deliverable was to understand Yioop!, an Open-Source search engine. The goal of the second deliverable was to access user's data from the social networking site Twitter. This task was accomplished by writing a Twitter application that accesses and posts' tweet.

The purpose of the third deliverable was to be able access user's data (posts) from Facebook. This was accomplished by creating a non-trivial Facebook application that pulls out user's posts, images and other information.

The Final deliverable was to access user's bookmarks data from Delicious.com and indexes the data. The task was achieved by writing an Archive Bundle Iterator for data to be indexed into Yioop!. We now describe each of these deliverables in more details. This will be followed by a conclusion and future work.

2.0 DELIVERABLES

2.1 DELIVERABLE 1: Epub Text Processor for Yioop!

Yioop! is an open-source search engine software written in PHP and uses MySQL and SQLite for the storage of data [2]. It allows users to index their own collection of websites or webpages.

The goal of this deliverable was to get a better understanding of Yioop!. The task was to write an ePub text processor to get the summary of data from the standard ebook publishing format *.epub*.

In-order to get the summary from Epub files, I wrote a EpubProcessor class that extends TextProcessor class. The *.epub* files are structured in the form of Zipped XML format. I used XML parsing to get the important summary from the files of a *.epub* book.

2.2 DELIVERABLE 2: Twitter Application

Description:

In this experiment, I created a non-trivial Twitter app to randomly access latest tweets from the twitter users and post random tweet among them into user's profile after every few seconds. The goal of this experiment was to access twitter user's data.

Implementation:

Building Twitter Applications: Twitter gives access to its enormous corpus of data via Twitter APIs [3]. A Twitter application will interact with Twitter APIs to access the data. In-order to access a users' private data, a Twitter application must be authorized by the user. Twitter supports OAuth authentication [5] where users are not required to share their passwords with third party application in order to authorize them on behalf to access data.

I used EpiTwitter PHP API for Twitter to access the user's data from application. You Each Twitter application has a unique consumer key and a consumer secret.

```
/* Consumer key from twitter */  
$consumer_key = 'GCdmIQH6jhtmLUypg82g';
```

```
/* Consumer Secret from twitter */  
$consumer_secret = 'jhgjdfgfgjhj76jggjhxxxjhxxx';
```

Create an EpiTwitter Object with the consumer key and consumer secret provided by the app.

```
$twitterObj = new EpiTwitter($consumer_key, $consumer_secret);
$oauth_token = $_GET['oauth_token'];
```

Getting the access token using the Oauth_token

```
$twitterObj->setToken($_GET['oauth_token']);
$token = $twitterObj->getAccessToken();
```

Accessing specific data:

To access user's friend IDs:

```
$statusInfo= $twitterObj->get_friendsIds(array('screen_name' =>
$username));
```

To update status of the user:

```
$update_status = $twitterObj->post_statusesUpdate(array('status'
=> $msg));
```

Callback URL: A callback URL is a web address an API needs to send information for use in a different web site while responding to the request.

Screenshots:



Fig 2. Twitter Authorization Dialog

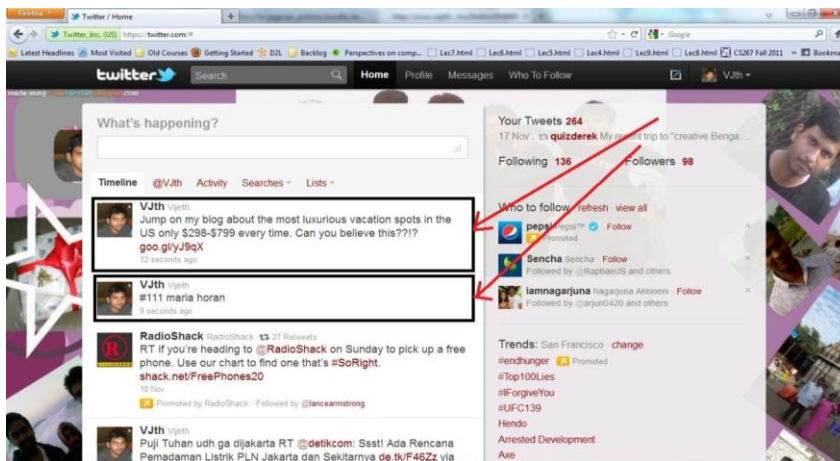


Fig 3. Tweets posted by Twitter Application (sotweet)

2.3 DELIVERABLE 3: Facebook Application

Application Description:

In this deliverable, I created a non-trivial Facebook application that displays the user's friends activities such as their updates, user likes, user's recent photos and user's friends using this application.

Apps on Facebook are loaded into a Canvas Page. A Canvas Page is quite literally a blank canvas within Facebook on which to run your app. User's data in Facebook is accessed via the APIs of Facebook [4]. In-order to access these API's, an application should be authorized by the user. The Facebook also supports OAuth authentication as in Twitter. Every Facebook app has it's own unique Application ID and Application Secret. It also needs a Callback URL to redirect users after authentication. Your app should get a Access token that enables it to access the user's information and take actions on their behalf. I used FBUtils APIs and Facebook Query Language(FQL) to query to specify the required user data. FQL enables you to use a SQL-style interface to query the data exposed by the Graph API. It provides for some advanced features not available in the Graph API, including batching multiple queries into a single call.

```
/* App ID and App Secret for Facebook application */  
$app_id = '143515066545620';  
$application_secret = '643d00914b59f832acd9f3939693301';
```

```

/* Initialize the application instance with the appId and appSecret */

$facebook = new Facebook(array(
    'appId' => $app_id,
    'secret' => $application_secret,
));

```

To Find the user's likes:

```

$likes = array_values(
    idx(FBUtils::fetchFromFBGraph("me/likes?access_token=$token&limit=4"), 'data')
);

```

To get the user's friends list:

```

$me = $facebook->api('/me/friends');

```

Using FQL query to find which of your friends are also using this application:

```

$app_using_friends = FBUtils::fql(
"SELECT uid, name, is_app_user, pic_square FROM user WHERE uid in
(SELECT uid2 FROM friend WHERE uid1 = me()) AND
is_app_user = 1", $token);

```

For getting friend's status using FQL

```

$app_friends_status = FBUtils::fql(
"SELECT uid, pic_square, name, status FROM user WHERE uid IN
(SELECT uid2 FROM friend WHERE uid1 =me())", $token );

```

To get the user's recent photos:

```
$photos = array_values (
  idx($raw = FBUtils::
  fetchFromFBGraph("me/photos?access_token=$token&limit=16"),
  'data');
```

Screenshots:

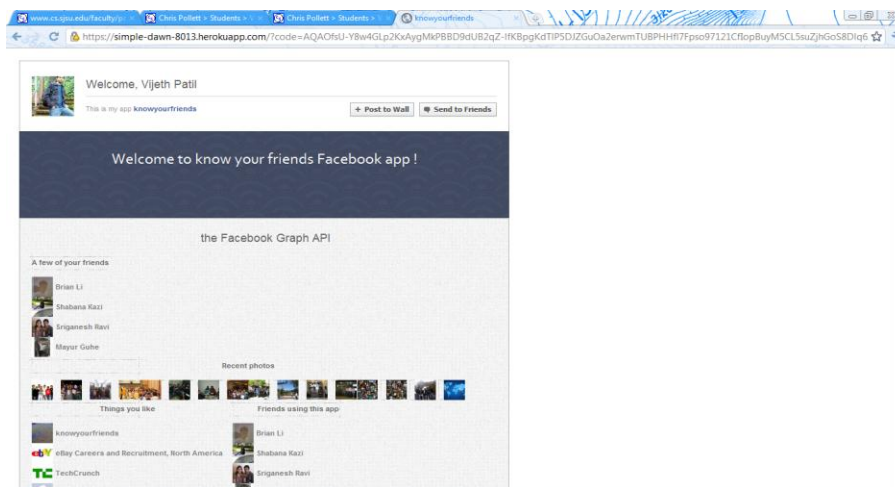


Fig 5. Facebook Application (knowyourfriends)

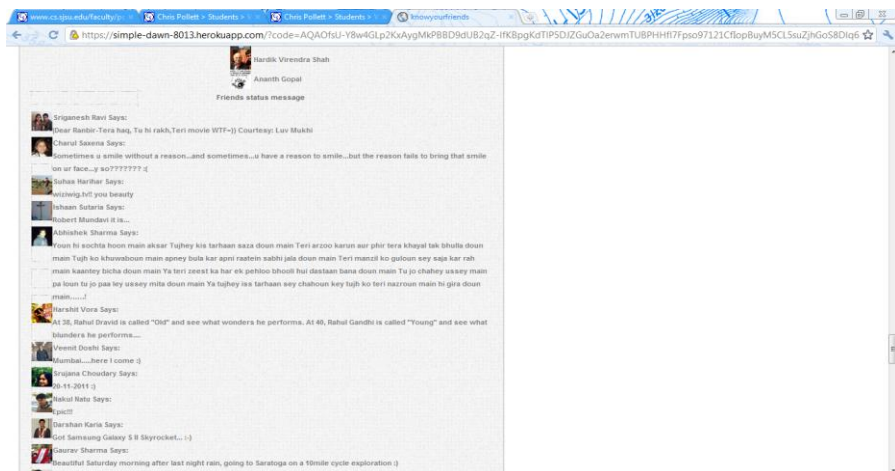


Fig 6. Facebook Application (knowyourfriends)

2.4 DELIVERABLE 4: Del.icio.us Feeds

In this deliverable, I will index the delicious bookmarks into Yioop Open-source search engine software. Delicious is the world's leading social bookmarking service for storing, sharing, and discovering web bookmarks. Users can tag each of their bookmarks and also add a comment on each of them. Delicious provides a simple REST -like API. All of the requests start with the format:

```
https://api.del.icio.us/v1/
```

/v1 APIs require https requests and supports HTTP Auth.

To get all the posts from a user, you must use the following curl request format:

```
$ curl https://user:passwd@api.del.icio.us/v1/posts/all
```

where, user and passwd represent username and password to your delicious account.

The data is returned in xml format. An iterator 'Delicious_Archive_Bundle_Iterator' is written in PHP to iterate through the data and convert to a format indexable by Yioop.

```
$site = "https://username:password@delicious.com/v1/posts/all";  
$xmlstring = FetchUrl::getPage($site);
```

Sample Data Returned from Delicious:

```

<posts tag="" total="7" user="vijethpatil">
<post description="Earth-like planet found in 'habitable' zone – USATODAY.com"
extended="" hash="dd02d11ef4635896194ba4cb198f1973"
href="http://www.usatoday.com/tech/science/space/story/2011-12-05/nasa-finds-planet-
that-could-sustain-life/51656310/1" private="no" shared="yes" tag="Earth nasa"
time="2011-12-05T21:57:06Z"/>
<post description="How will the Curiosity lab work when it reaches Mars?"
extended="sdfa" hash="f601d90ddfb37ffc844a22741b03b91b"
href="http://www.telegraph.co.uk/science/space/8917766/How-will-the-Curiosity-lab-
work-when-it-reaches-Mars.html#.TtFCMLNkgQ.delicious" private="no" shared="yes"
tag="nasa mars curiosity life extraterrestrial" time="2011-11-26T19:47:14Z"/>
</posts>

```

Screenshot:

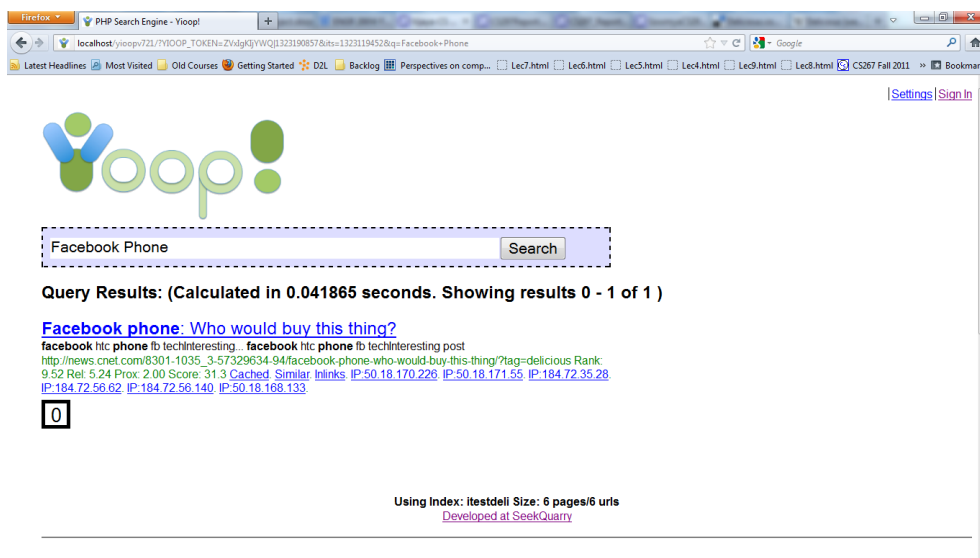


Fig 7. Delicious results in Yioop

4. CONCLUSION

In this semester, I worked on the accessing content from three different social networking websites Facebook, Twitter, Delicious. Each of them are different in their way of providing social interaction and the way of organizing social content. I created an application for both Twitter and Facebook, which uses their APIs to communicate with the users' data. I got a good understanding on different Authorization protocols they use to provide users' data with access control authenticated by user. I also got the content of social bookmarking site delicious user to be indexed into Yioop! Open-source search engine. I also worked on writing ePub text processor and Delicious ArchiveBundleIterator for Yioop!. This gave me good understanding of how to extend Yioop to index other forms of data and will help in modeling social network content to index into search engine.

In the next semester, I will be working on creating a search engine of the data collected from the social networking sites with the understanding what I have got from each of the deliverables in CS297. I will be working on integrating the data access from Facebook and Twitter with permanent tokens that can be used to access user's data any number of times with just one time authorization dialog.

5. REFERENCES

- [1] Bjørklund, T.A., Götz, M., & Gehrke, J., (2010, June 06). Search in Social Networks with Access Control. *Proceedings of the 2nd International Workshop on Keyword Search on Structured Data*, pp 4-4.
- [2] *Yioop! Documentation v 0.78*. Retrieved on Dec 5, 2011, from web site: <http://www.seekquarry.com/>
- [3] *Twitter Documentation*. Retrieved on Dec 5, 2011, from web site: <http://www.dev.twitter.com/docs>
- [4] *Facebook Application Authentication and Overview*. Retrieved on Dec 5, 2011 from web site: <http://developers.facebook.com/docs/authentication/>
- [5] *The OAuth 2.0 Authorization Protocol*. Retrieved on Dec 5, 2011, from web site: <http://tools.ietf.org/html/draft-ietf-oauth-v2-22>