Overview of Facebook Data Collection through Web Crawler

S. S. Wangikar
Dept. of CS & IT
Dr. B.A.M. University,
Aurangabad (M.S), India

P. P. Chouthmal
Dept. of CS & IT
Dr. B.A.M. University,
Aurangabad (M.S), India

P. V. Kothare
Dept. of CS & IT
Dr. B.A.M. University,
Aurangabad (M.S), India

S. N. Deshmukh
Dept. of CS & IT
Dr. B.A.M. University,
Aurangabad (M.S), India

Abstract

Online social networks, such as Facebook, twitter, are utilized by many people. These networks allow users to publish details about themselves and to connect to their friends. In this network some of the information is private or public. By using the various algorithms we can predict information of users. In this paper, we explore how to get social networking data to predict information. We constructed a spider that crawls & indexes FACBOOK. The general research challenge is to build up a well-structured database that suits well to the given research question and that is cost efficient at the same time. In this paper we focus on crawler programs that proved to be an effective tool of data base building in very different problem settings. Firstly we explain how crawler programs for current profile work. In future work we design the crawler for the current profile and related to groups.

1. Introduction

Social networks, such as Facebook, are online applications that allow their users to connect their friends or for other communication purpose by using the various types of the links. This social networking is basically used over the millions of users in colleges, industry campuses & used by professional as well as non professional peoples. Facebook is a general use social networking, so individual details their information on this network. This information is arranged uniformly and aggregated into one place, there are bound to risks privacy. Users list their favourite activities, books & movies. Users may submit their data without being aware of that it may share with advertisers. Sometimes third party may build a database of Facebook data to sell. Intruders may steal passwords, or entire database from Facebook.

Facebook is one of the foremost social networking websites, with over million users spanning in college campuses [1]. With this much detailed information arranged uniformly and aggregated into one place, there are bound to be risks to privacy. In economic and social sciences it is crucial to test theoretical models against reliable and big enough databases. The general research challenge is to build up a well structured database that suits well to the given research question and that is cost efficient at the same time. In this paper we focus on crawler programs that proved to be an effective tool of data base building in very different problem settings our goal was to protect that data. We constructed a spider that "crawls" and indexes Facebook. We constructed a threat model that attempted to address all possible categories of privacy failures.

1.1 The Web Crawler Methodology Techniques

Web crawler programs are as old as the world wide web [2].They are short software codes sometimes also called as bots, ants or worms written with the objective to download web pages, extract hyperlinks, create a list of its URLs and add them to a local database[3]. Their most widely spread applications are search engines in which form they are familiar to all internet users [4]. One of the first publications of how effective search crawling and indexing can be on the web was published by Google owners in 1998 as a widely referred academic paper on crawler programming [5].Building a web crawler, first of all requires the knowledge of how exactly the users browse a website, and what happens during this process from an information processing point of view. Based on this, the simple idea behind programming a crawler is to (i) imitate the actions of a user visiting a webpage, (ii) extract the information we need, and (iii) repeat these steps. Using this idea we identified three conceptually different type of crawling method each of which is suitable to collect data for different type of analysis and related problem statement.

The first and, we might say, the simplest is when the data we are interested can be found on one site at a well defined place or technically record and its value depends on time. For instance this is the logic of storing and presenting market data on some aggregate websites, or showing price information about particular products in a web store or personal information about employees. In these cases, the data structure is static so the crawler has always visit the same page and same record and periodically download its actual value into a database.

The second type of crawler is more complex in a sense that it imitates a user who is visiting a list of websites one after the other and downloads data, usually different records from each of these sites which after the completion of the

visits is grouped, sorted, and analyzed according to the problem or research design.
The third, and most complex, type of search when the browsing sequence is not predetermined neither by its sequence of websites nor the actually numbers of them. This happens typically when we explore a domain of a topic, for instance collect information about graduate programs at universities, looking for a particular service, or try to find out the size and connection structure of a community on the net. Since this most complicated case entails both previous once technically and in this present paper we use an example for crawler illustration from this type we describe it in more details.

2. Social Networking and Facebook

Users share a variety of information about themselves on their Facebook profiles, including photos, contact information, and tastes in movies and books. They list their friends including friends at other schools. The site is often used to obtain contact information, to match names to faces, and to browse for entertainment [1]. Facebook was founded in 2004 by Mark Zuckerburg, then a Harvard undergraduate. The site is unique among social networking sites in that it is focused around universities Facebook is actually a collection of sites, each focused on one of 2,000 individual colleges. Users need an@college.edu email address to sign up for a particular college's account, and their privileges on the site are largely limited to browsing the profiles of students of that college. Over the last two years, Facebook has become fixture at campuses nationwide, and Facebook evolved from a hobby to a full-time job for Zuckerburg and his friends. In May 2005, Facebook received $13 million dollars in venture funding. Facebook sells targeted advertising to users of its site, and parterres with firms such as Apple and JetBlue to assist in marketing their products to college students [9].

2.1 Information that Facebook stores

First-party information All data fields on Facebook may be left blank, aside from name, e-mail address, and user status (one of: Alumnus/Alumna, Faculty, Grad Student, Staff, Student, and Summer Student). A minimal Facebook profile will only tell a user's name, date of joining, school, status, and e-mail address. Any information posted beyond these basic fields is posted by the will of the end user. Although the required amount of information for a Facebook account is minimal, the total amount of information a user can post is quite large. User-configurable setting on Facebook can be divided into eight basic categories: profile, friends, photos, groups, events, messages, account settings, and privacy settings. For the purposes of this paper, we will investigate profiles, friends, and privacy settings. Profile information is divided into six basic categories: Basic, Contact Info, Personal, Professional, Courses, and Picture. All six of these categories allow a user to post personally identifiable information to the service. Users can enter information about their home towns, their current residences and other contact information, personal interests, job information, and a descriptive photograph. We will investigate the amount and kind of information a typical user at a given school is able to see, and look for trends. A major goal of Facebook is to allow users to interact with each other online. Users define each other as friends through the service, creating a visible connection. Third-party information two current features of Facebook have to do with third parties associating information with a user's profile. The Wall allows other users a bulletin board of sorts on a user's profile page. Other users can leave notes, birthday wishes, and personal messages. The "My Photos" service allows users to upload, store and view photos. Users can append metadata to the photographs that allows other users to see who is in the photographs, and where in the photograph they are located. These tags can be cross-linked to user profiles, and searched from a search dialog. The only recourse a user has against an unwelcome Facebook photo posted by someone else, aside from asking them to remove it, is to manually remove the metadata tag of their name, individually, from each photograph. Users may disable others' access to their Wall, but not to the Photos feature. My Privacy Facebook's privacy features give users a good deal of exibility in who is allowed to see their information. By default, all other users at a user's school are allowed to see any information a user posts to the service. The privacy settings page allows a user to specify who can see them in searches, who can see their profile, who can see their contact info, and which fields other users can see. In addition, the privacy settings page allows users to block specific people from seeing their profile. As per the usage agreement, a user can request Facebook to not share information with third parties, though the method of specifying this is not located on the privacy settings page.

3. Fundamentals of a Web Crawler

Despite the numerous applications for Web crawlers, at the core they are all fundamentally the same. Following is the process by which Web crawlers work:
1. Download the Web page.
2. Parse through the downloaded page and retrieve all the links.
3. For each link retrieved, repeat the process.
Now let's look at each step of the process in more detail.
In the first step, a Web crawler takes a URL and downloads the page from the Internet at the given URL. Oftentimes the downloaded page is saved to a file on disk or put in a database.
Saving the page allows the crawler [10] or other software to go back later and manipulate the page, be it for indexing words (as in the case with a search engine) or for archiving the page for use by an automated archiver.
In the second step, a Web crawler parses through the downloaded page and retrieves the links to other pages. Each link in the page is defined with an HTML anchor tag similar to the one shown here : <A HREF="http://www.host.com/directory/file.html">Link</A> After the crawler has retrieved the links from the page, each link is added to a list of links to be crawled. The third step of Web crawling repeats the process. All crawlers work in a recursive or loop fashion, but there are two different ways to handle it. Links can be crawled in a depth-first or breadth-first manner. Depth-first crawling

follows each possible path to its conclusion before another path is tried. It works by finding the first link on the [11] first page. It then crawls the page associated with that link, finding the first link on the new page, and so on, until the end of the path has been reached.

4. Data Collection

Our collection of data directly from Facebook served two principles. It served as a proof of concept, to demonstrate that it is possible for an individual to automatically gather large amounts of data from Facebook. The collection of data was not entirely trivial, but we were able to produce the scripts necessary to do so within 48 hours. Also, the collection of data from Facebook will provide us with a large, nearly exhaustive and statistically significant data set, from which we can draw valuable conclusions on usage trends.

4.1. Data Acquisition for Application ID

We are not the first to download user profiles from Facebook in large numbers. In the past, others have utilized Facebook's use of predictable, easy to understand URLs to automatically request information and save user information for further analysis. Our approach used the incremental profile identifier to download information in large quantities. The algorithm we used to gather this data is very straightforward:

Step1: Log in to Facebook and Create Application.
Facebook developers gives the different types application.

Figure 1. Login Window

We can also create by the graph apps. The output of this step shown in fig.1.

Step2: Crate New Application ID

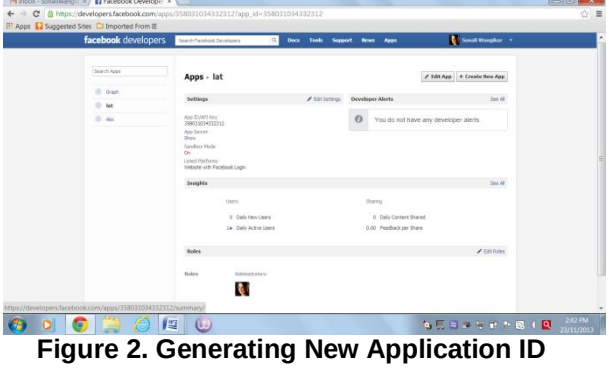

Figure 2. Generating New Application ID

In step 2 we create the new application ID. By following the step 1 we get the one new application ID for our new App Name. The output of this step shown in fig.2. *Step3: Save this Application ID.*

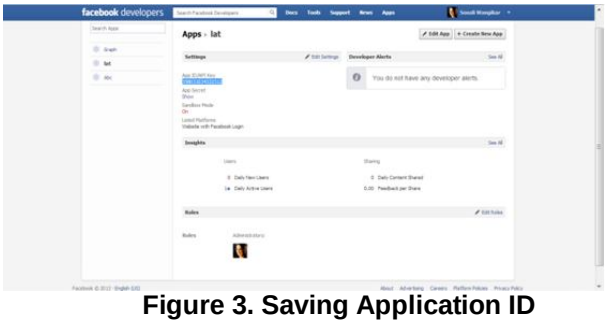

Figure 3. Saving Application ID

In Step 3 we save the New App ID in our Facebook account.
The output of this step shown in fig.3.

Step4: Access New Application ID through our Sever (Apache Server).
In Step 4 we can access this App ID through our Apache Server. First we completed install Apache Server in our system then by our crawling programming we can access the our FACBOOK account. Then save this App ID in our crawler then save that access token ID. The output of this step shown in fig.4.

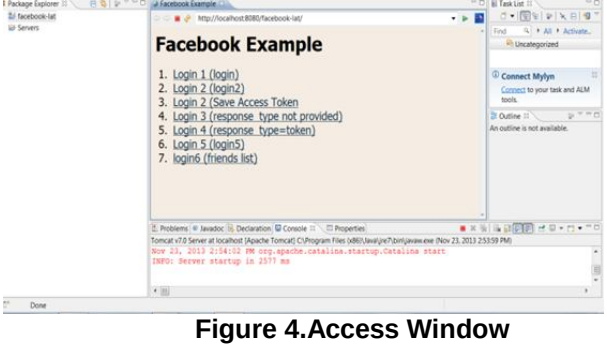

Figure 4.Access Window

Step5: Collect Your Network Data.

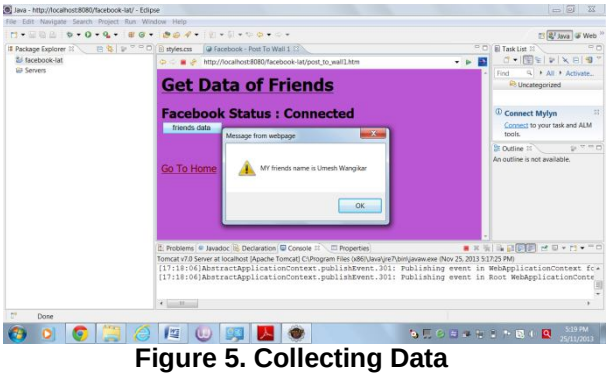

Figure 5. Collecting Data

In step5 through our crawler program we collect the friends name data. In this step we get the friends name from our Facebook account. The output of this step shown in fig. 5. To implement our algorithm, we used Apache Tomcat Server for the network downloader. In addition to implementing the above algorithm, we made pretend to be another web browser by changing its user agent We took advantage of the fact that logins and passwords are not encrypted.

5. Facebook Data Collection through Crawler

When a user wants to open e.g. Facebook.com the followings takes place on the server and on the client side:

Step 1. Open a Web Browser:
Browser is an interface to communicate with web servers by sending requests and receiving the response packages.

Step 2. Type in http://www.Facebook.com:
Browser requests the server called Facebook.com to generate and forward the source code of the front page. Sending the URL is not the only parameter the client passes to the server. Facebook.com also receives the:
Protocol: what is the "language" of the request?
Address: where to send back the answer?
User agent: who are we? (Operating system, browser type, language)
Referrer: what was the previous page we've viewed?
Additional parameters: to control the response (get, post, and cookie).

Step 3. Receive the Response:
The server according to the request, responses: the *status code* (page not found, access denied, ok…), the *content type* (html, image, video…), the *character encoding* and the *content code* itself. From this information the browser builds and visualizes the webpage. The crawler has to follow the above steps in order to properly receive a site and extract the desired information. There are several scripting environments that are able to fulfil this requirement (PHP, JAVA, .NET…), so without any further specifications:

1. Generally the crawler has to have a connection function to reach a webpage.

2. The connection is parameterized with the request data (who we are, what we want, how long the program should wait for the answer, what is the maximum number of redirections…)

3. Has to be able to receive the response, understand the status code, turn the source code to textual information, and with setting the proper character encoding, understand it (store in the memory and decode)

Using the above techniques, the crawler is ready to download and process a web page let us do the same and repeat it. Naturally, the repetition requires some further work, but it's depending on the final purpose as well. In the followings, we present a situation where one domain (Facebook.com) is being inspected through thousands of subpages.

1. First of all we need to login to Facebook.com with the crawler program as most of the data are available only to authenticated users. When a client signs in, the previous process is repeated except the second step where some additional parameters are set. These parameters are derived from the login form (mainly the username and the password) and from cookies, provided by the page. Both of them have to be attached and copied.

2. After successful login, we select a certain person to analyze his profile. So after all, it is possible to extract the information we need (using connection stream analysis) and store them.

3. Not only one profile, but its connections are also important. At this step, the crawler saves all the friends, belonging to the previous person. The database stores only the user ID-s and Group IDs, the connections (in [parent, child] form), as the profile URLs can be recovered from them.

4. For a given set of connections the crawler uses systematic breadth-first-search procedure to continue the process from the second step. It means we analyze the profiles first and not the connections. As long as there are unanalyzed parents, there is no child process happens.

5.1 BFS Crawler

The architecture of this crawler includes an agent that executes data extraction tasks and a FIFO queue, named To Be Visited, in which are stored, in order of appearance, profiles of users to be visited. The own of HTTP requests sent by the BFS crawler is described as follows: first, the agent contacts the Facebook server, providing identification required for the authentication through cookies. Once logged in, the agent starts crawling pages, visiting the friend list page of the seed profile (the logged in user) and extracts her friend list; friends user-IDs are enquired in a to be visited FIFO queue and, cyclically, they are visited in order to retrieve their friend list We started this process from a single seed, and stopped its execution after 7 days of crawling, obtaining a partial sample of the Facebook graph and contains information down to the third sub-level of friendships (friend lists of friends of friends have been acquired).

5.2 Limitations

One notable limitation we met during the data mining process is due to the technical precautionary adopted by Facebook to avoid a high traffic through their platform. In details, once the agent requests for a friend-list Web page, the server responds with a list of at most 400 friends. If the list should contain more than 400 friends, it is shortened to 400 instead. This limitation can be avoided adopting different techniques of data mining, for example exploiting platforms of Web data extraction which scrape information directly from the Web page, simulating the behaviour of a

human user, through the interface of the browser itself. Even though, the computational overhead of a similar process is too high to be adopted for a large-scale crawling task, unless a commercial platform for Web data extraction is employed.
On a smaller scale, we already faced the problem of sampling Facebook adopting this approach [13].

6. Conclusion and Future Work

The discussion of experimental results follows. The analysis of collected datasets has been designed. We strongly advise all Facebook users to restrict access to their profiles, to not post information of illegal or policy-violating actions to their profiles, and to be cautious with the information they make available. This lasting change will only come with execution time and groups of the users .By crawler designing we can have the database of Facebook. In this database we can have the different user ID and group ID. Using this ID's we can performed the various data mining techniques. In the future work we performed the various classification algorithms on that data set. We explain how crawler programs for current profile work. In future work we design the crawler for the current profile and related to groups.

7. References

[1] Terremark Worldwide, Inc."Facebook Expands Operations at Terremark's NAP West Facility" Tuesday November 1, 8:30 am ET.

[2] Risvik,K.M. and Michelsen,R. Search Engines and Web Dynamics. Computer Networks, Vol. 39, pp. 289–302, June 2002.

[3] Chakrabarti,S.Mining the web: Analysis of hypertext and semi structured data. New York: Morgan Kaufmann-2003.

[4] Pant G.,Srinivasan P.Menczer F.,"Crawling the Web", in Levene M., Poulovassilis A. Web Dynamics: Adapting to Change in Content, Size, Topology and Use, Springer, pp. 153–178-2004.

[5] Brin,S.,Page L. The anatomy of a large-scale hypertextual Web search engine, Computer networks and ISDN systems, 30 (1-7). 107–117-1998.

[6] Konrad, Rachel. Associated Press. February 24, 2005, "Burned by ChoicePoint breach, potential ID theft victims face a lifetime of vigilance."

[7] Facebook Privacy Policy, available online at http://www.Facebook.com/policy.php.

[8] MySpace Terms of Service, available online at http://viewmorepics.myspace.com/misc/terms.html.

[9] Marshall, Matt and Anna Tong. "Palo Alto, Calif.-based Facebook brings social networking".

[10] B. Bamba, L. Liu, J. Caverlee, V. Padliya, M. Srivatsa, T. Bansal, M. Palekar, J. Patrao, S. Li, and A. Singh, "DSphere: A source-centric approach to crawling, indexing and searching the world wide web," in Proceedings of the 23rd International Conference on Data Engineering, 2007.

[11] Z. Bar-Yossef and M. Gurevich, "Random sampling from a search engine's index," in Proceedings of the 15th *International World Wide Web Conference*, 2006.

[12] Friendster Terms of Service, available online at http://www.friendster.com/info/tos.php.

[13] S. Catanese, P. De Meo, E. Ferrara, and G. Fiumara. Analyzing the Facebook Friendship Graph. In Proceedings of the 1st Workshop on Mining the Future Internet, pages 14 19, 2010.

[14] Facebook code www.restfb.com