International Journal of Computer Science & Communication (ISSN: 0973-7391)
Volume 10 • Issue 2 pp. 171-179 March 2019 - Sept 2019 www.csjournals.com

Social Network with Web Crawler & Cluster

Meenu Yadav, Dr. Gundeep Tanwar, Anil Wadhwa
Department of Computer Science and Engineering,
RPS Group of Institutions, Balana, Mohindergarh, Haryana, India

Abstract: Online social networks, such as Facebook, Twitter, Yahoo!, Google+ are utilized by many people. These networks allow users to publish details about themselves and to connect to their friends. Some of the information revealed inside these networks is meant to be private or public. Yet it is possible to use learning algorithms on released data to predict private or public information and use the classification algorithm on the collected data. In this paper, we explore how to get social networking data to predict information. We then devise possible classification techniques that could be used in various situations. Then, we explore the effectiveness of these techniques and attempt to use. We collect the different information from the users groups. On which we concluded the classification of that data. By using the various algorithms we can predict information of users. Crawler programs for current profile work. We constructed a spider that crawls & indexes FACBOOK. In this paper we focus on crawler programs that proved to be an effective tool of data base. In this paper we elaborate the use of data mining technique to help retailers to identify user profile for a retail store and improved. The aim is to judge the accuracy of different data mining algorithms on various data sets. The performance analysis depends on many factors encompassing test mode, different nature of data sets, and size of data set.

Keywords: Social network analysis, social networking data, data mining.

I. INTRODUCTION

The social network is a type of network that link individuals through the Social networking sites. There are many on-line social internet sites such as Instagram, Facebook, and Twitter etc. fashionable sites on the web. A user will have multiple social networking sites but there is a problem to manage and track their content/contact or alternative social activities that is formed by individual person everyday across many networks. The 'social data' (which may be posts, status, photos, links, tweets, scraps etc that is a part of updates in social network service) might cause important info overload to users. This type of knowledge overload referred to as ''network overload". Aggregation tools area unit in situ that has users to consolidate messages, track social information across networks.

Social Network Aggregator may be a wise resolution of drawback and problems in social world. Social network aggregators consolidate that user isn't needed to login to every web site and perform same group action. A user performs the group action at one web site and also the info is synchronized to any or all of the social networks that the user specifies. It's the method of collecting/aggregating information across multiple social network services. According to 2009 study of 11,000 users reported that the mainstream of MySpace, LinkedIn, and Twitter users also have Facebook accounts. There is plan is to arrange and maintain the data retrieval method for a user for multiple social network

TYPES OF SOCIAL NETWORK AGGREGATORS

Many types of social network aggregators available now days to manage the profiles at a single point. These provide specialized features to maintain the profile wisely. Some of them are listed below:

XeeMe: XeeMe is a type of application. It manage entire social identity, identify of new networks, people and nature of their presence. It offers the user the possibility of organizing all social networks at a single point and

shares their social presence with one URL with friends, customers, partners and people. It has a long number of supported networks and the user has a point of reference about his presence value and network relevance. Through the application the user can discover new networks or people who are in other networks and offer the possibility of connecting with them.

SocConnect: SocConnect is a web browser that collect social data from different Social Networks .It allows users to create personal profile for their social data. SocConnect has also features of personalized suggestion of friends that may be good for the user.It also provide machine learning techniques. Users can combine and cluster friends and Network. Based on extensive estimation, it provides a set of user settings that can provide the best performance on personalized recommendation.

Hootsuite: Hootsuite is a collection of person for businesses and organizations. It provide a web dashboard that executes across multiple social networks. The user must log in the social networks if he wants to use and give some permission to network. After that the user can see the updates, publish in one or more networks at the same time, marketing promotions, identify new users and distribute targeted messages.It provides many functionality which are given below:

- Scheduling
- Files of audio,video and etc.
- Bookmark functionality
- Integration
- Popularity of Hoot Suite
- No need to additional patch/software.

Flock: Flock is a web tool. It provides management tools but doesn't require you to provide authentication for social networking and also provide other Web 2.0 services. , It is a complete Web browser.

It integrates multimedia services such as Myspace, Facebook, Twitter etc. When signing in facebook or twitter, Flock can get status updates and friend's updates. In addition, Flock can also search in Twitter to update multiple services at once. Other features are:

- Sharing of posts, links, photos and videos.
- An email client
- A media bar showing pictures and videos.
- A reader and editor of blogs

People Aggregator: People Aggregator is basically a social identity hub. It runs only on Linux. The site is also a Digital Identity Hub, meaning that users will be able to utilize a single sign in, using, for example, their Flickr or Blogger ID; profile data can also be imported/ exported without losing any content. People Aggregator will allow users to create blogs, media galleries, podcasts, blogcasts, and forums.

WAYS OF INTEGRATING SOCIAL NETWORKS

A lot of research has been done in the area of integrating social networks from the mid 1990s.

There are four main ways by which we can integrate the social network [3] [4]:

- Content Aggregators
- Comparison Analytics
- Relationship Aggregation

- Process Aggregation

1 **Content Aggregators**: Content aggregators collect social data from multiple sources of specific topic and provide analytics based on relationships across multiple data sources or networks. Content aggregators analyzes features set content and correlate them to user data (e.g. preferences, interests), based on a user model derived by analyzing the previous actions on data by the user.

2 **Comparison Analytics**: A comparative result is evaluated on the basis of user specified criteria

3 **Relationship Aggregation**: It Provides a delta of relationships between a user and company services/ information sources with which the user has a business relationship.

4 **Process Aggregation**: Business Process which require coordination across a variety of services/ information and managed and a common point of contact is provided.

DIFFERENENCE IN SOCIAL NETWORK AGGREGATORS

There are various solutions available to integrate the social network but no one has tried to integrate the information available within multiple social networks.

Flock can get updates from friends, status updates and photos submitted at Multiple Social networks

SocConnect users can creates a personalized social and semantic contexts for their social data. Users can combine and cluster friends.

HootSuite aggregates organizations and businesses to collaboratively execute promotions across multiple social networks

XeeMe Organizes Social presence, discovers new network and people. It organizes the entire social presence of the user, determine new networks and people and develop their presence and influence.

But none of the aggregator has mined the multiple social networks and extracted some useful information after collecting data from different Social Networks.

II.LITERATURE SURVEY

In this paper we studied content recommendation on Twitter to better direct user attention. In a modular approach, we explored three separate dimensions in designing such a recommender: content sources, topic interest models for users, and social voting. We implemented 12 recommendation engines in the design space we formulated, and deployed them to a recommender service on the web to gather feedback from real Twitter users. The best performing algorithm improved the percentage of interesting content to 72% from a baseline of 33%. We conclude this work by discussing the implications of our recommender design and how our design can generalize to other information streams.[1]

In this paper, we explore whether these students are using Facebook to find new people in their offline communities or to learn more about people they initially meet offline
Large numbers of college students have become avoid Facebook users in a short period of time. Our data suggest that users are largely using Facebook to learn more about people they meet offline, and are less likely to use the site to initiate new connections.[2]

In this paper, we will show how Semantic Web technologies and especially the FOAF and SIOC vocabularies can be used to model user information and user-generated content in a machine-readable way. Thus, we will see how data and network information can be reused among various services and applications, at almost zero-cost for developers of such tools.[3]

A literature search was carried out in July 2012. The multidisciplinary search engine ISI Web of Science was used that stores articles from more than 250 disciplines and covers over 12,000 of the highest impact journals worldwide, including Open Access journals and over 150,000 conference proceedings. Only peer-reviewed papers were considered, conference abstracts were not included in the literature review.

The following search terms as topics were used:

(A)"social networking" (B), "online communities" (C), "social media" (D), "computer-mediated communication"(E), "discussion boards" (F), "networking sites" "social network sites" (G), website (H), "social computing" (I), "Virtual community" (J), "Web 2.0" (K), and "user-generated content", each search term combined with the terms "older adults", elder*, senior*, and "age differences".

In this paper, we review inherent concepts and properties of social networks and highlight major analytical evaluation criteria, which are used to identify key findings that reveal degrees of benefits and shortcomings of social networks. We also discuss some proposed solutions related to decentralized social networks in the context of business implications as well as their effects on privacy, identity and trust issues.[3].Social networks are playing an important role in personal as well as corporate environments. However, perceived issues and evolving challenges may hinder further expansion of social networks to meet new opportunities

In this section we will present a lot of information about web crawlers and their literature review. In the first place, we will mention some historical data about them and we will give them their definition in order to make this diplomatic work more understandable, we will describe the existing crawling policies and analyze them. Crawling policies relate to how the crawler chooses the next page to visit, and logically, the policy that each crawler follows plays a catalytic role in its effectiveness.

On January 27, 1994, Brian Pinkerton, a student of Computer Science and Engineering at Washington University, began to build a Web Crawler in his spare time. Initially, WebCrawler was a desktop application rather than a Web service as it is today. WebCrawler was first applied to a list of 25 Urls on March 15, 1994. **WebCrawler Timeline [16]**

April 20, 1994 WebCrawler is broadcast live on the Web with a database of more than 4,000 Web sites. After about one and a half months, Brian Pinkerton, WebCrawler announced, a group where news sites were announced on this site. **WebCrawler Timeline [16]**

1,000,000 **November 14, 1994** WebCrawler serves a one-million-dollar queue.

December 1, 1994 WebCrawler acquires two sponsors: DealerNet and Starwave. Both companies have provided money to help continue WebCrawler. WebCrawler was fully supported by an advertisement that took place on October 3, 1995, but still retained a strict separation between the ad and the search results. **WebCrawler Timeline [16]**

June 1, 1995, America Online acquired WebCrawler. At this time, AOL had less than a million users, and no ability to access the Web. They believed that AOL sources could help materialize most of WebCrawler's future status. **WebCrawler Timeline [16]**

On September 4, 1995, Spidey was born. At the first changes to WebCrawler's design, they turned to a new image where Spidey was the mascot. Spidey took many personalities over the years and explained the fun, and the carefree spirit that WebCrawler was trying to do. **WebCrawler Timeline [16]**

April 1996 WebCrawler extends its functionality beyond a simple search to include the best human Web guide: GNN Select. Formally known as the Integrated Directory of the Internet, GNN Select was the product produced by a small group of Internet savvy researchers led by Abbot Chambers. **WebCrawler Timeline [16]**

April 1, 1997 Excite acquires WebCrawler. AOL sold its WebCrawler to Excite's Mountain View. WebCrawler was initially supported by its own dedicated Excite team, but it was finally banned for WebCrawler and Excite to run in the background. **WebCrawler Timeline [16]**

2001 InfoSpace acquired WebCrawler. Excite is not going well and in bankruptcy, Inspospace acquired WebCrawler. Today Infospace runs WebCrawler as a post-search engine. And they gave Spidey a new name and made him purple. **WebCrawler Timeline [16]**

The web crawler is a program that searches the World Wide Web in a methodical and automated manner. Web crawlers - also known as robots, spiders, worms, walkers and wanderers - are as old as the Web itself . The first crawler, by **Matthew Gray Wanderer**, was written in the spring of 1993. Several papers on web crawling were presented at the first two conferences on the World Wide Web. However, at that time, the Web was much smaller than today, which made those systems not to experience "scaling" problems, as they are today.

Web crawlers are mainly used to construct a copy of the pages we've visited. This copy can then be further elaborated by a search engine that will categorize the downloaded pages to provide a very fast search. It is reasonable that all known search engines use crawlers. However, due to the competition that exists between search

engine companies, the plans of these crawlers have not been publicly described. There are two notable exceptions: Google crawler and Internet Archive crawler. Of course for both crawlers, the descriptions in the bibliography are laconic, thus preventing their ability to replicate.

The Google search engine is a distributed system that uses multiple crawling machines. The crawler consists of five operating systems running in different processes. A URL server process reads URLs from a file and sends them to multiple crawler processes. Each crawler process runs on a different machine, is a single thread and uses asynchronous I / O to receive data from 300 servers in parallel. Crawlers transfer downloaded pages to a StoreServe process, which compresses the pages and stores them in the disk. The pages are then read from the disk by an indexer process, which extracts links from HTML pages and stores them in a different file on the disk. A resolver URL process reads the link file, finds the absolute url of each one, stores it, and then reads it from the URL server. Typically, three to four crawlers are used, i.e. the entire system requires four with eight crawlers.

Internet Archive also uses multiple machines to crawl the Web. Each crawler process is assigned more than 64 sites to crawl, and no sites are assigned to more than one crawler. Each single-threaded process reads queues from the disk by URLs that correspond to the sites assigned to it. It then uses asynchronous I / O to extract parallel pages from these queues. Each time a page is downloaded, the crawler extracts the links it contains. If a link refers to the page it was found in, it is added to the appropriate queue, otherwise it is stored on the disk. Periodically, a batch process combines these URLs stored in the disk by deleting their double impressions [24].

Finally a web crawler is a kind of bot or software agent. Generally, it starts with a list of Urls to visit. As he visits each of these URLs, he finds all his links and adds them to the list of urls he will visit. It searches the Web repeatedly for a set of policies that we describe below.

There are two important web features that make web crawling very difficult:

a) its large volume; and
b) b) its frequency of change,

as a huge percentage of pages are added, changed and subtracted every day. Also, network speed has improved less than processing speeds and storage capabilities.

The large volume of the web means that the crawler can download only a percentage of pages within a given time, making it necessary to prioritize the download pages.

Additionally, the high frequency of web page changes results in new pages added to a site or existing pages being refreshed or deleted until the crawler has downloaded the last pages of that site.

A crawler must carefully select the pages to be visited at each step.The behavior of a web crawler is the result of a combination of policies:

• A selection policy that determines which pages will be downloaded.

• A re-visit policy that determines when to look for page changes.

• A politeness policy that determines how to avoid overloading Web sites.

• A parallelization policy that defines how to work distributed web crawlers.

Given the current web size, even large search engines only cover part of the publicly available hardware. A study by Lawrence and Giles (Lawrence and Giles, 2000) showed that no search engine classifies more than 16% of the web.

As a crawler only downloads part of the pages, it is highly desirable that this part contains the most relevant pages and not just a random sample of the web.

Cho and others made the first study on the strategies for crawling planning. The total of their data was 180,000 pages that were crawled by the stanford.edu domain. On this set, simulations were crawling different strategies. The layout metrics used were: breadth-first, backlink-count and Pagerank calculation (more information about Pagerank in the notes). One of the results of the study was that if the crawler wants to crawling pages with a high Pagerank then the partial Pagerank strategy is the best, followed by the breadth-first and last backlink-count strategy. However, these results are for a single domain.

Najork and Wiener (Najork and Wiener, 2001) built a real crawler on 328 million pages using a breadth-first layout. They found that a breadth-first crawl finds pages with high Pagerank at the beginning of the crawling process (but did not compare this strategy with others). The explanation given by the authors for this result is that "the most important pages have many links from many hosts and these links will be found quickly, regardless of the host or on which page the crawler was placed on."

Abiteboul (Abitebout et al, 2003) designed a crawling strategy based on an algorithm called OPIC (On-Page Page Importance Computation). In the OPIC, each page is given an initial amount of "cash" that is shared equally on the pages it points to. It's similar to Pagerank's calculation, but it's quicker and it's only one step. An OPIC-based crawler first downloads the pages with the highest amounts of "cash". Experiments were made on 100,000 pages that were a composite graph. However, there was no comparison with other real-world strategies or experiments.

Boldi and others (Boldi **et al., 2004)** used simulation on Web subsets of 40 million from .it domain and 100 million pages from WebBase crawl, testing the breadth-first layout against random layout and an all-out strategy. The most effective strategy was breadth-first, but the random layout also produced surprisingly good results. This was the approach used by **Pinkerton** in 1994 to a crawler developed and others **Diligenti** et al., 2000 suggested using the entire content of the visited pages to determine the similarity between the question and the pages that have not yet been visited. One problem is that the WebBase crawl is against the crawler used to collect the data. They also showed that poor Pagerank calculations made on some Web signatures during crawling can reach the actual Pagerank.

The main problem with focused crawling is that we would like to be able to predict the relevancy of the content of a page to the question before "download" the page. A possible prediction index is the set of single page links. Focused crawling is primarily dependent on the number of links we're looking for and is usually based on a Web search engine .

III. PROPOSED METHODOLOGY

The proposed network will mine the profiles of a user existing at multiple social sites and provide the information after combining the different profiles of their friends and their activities. We propose to develop an aggregator that will integrate several social web sites together and responds to a user's query; extracting the relevant data as specified in Query written in a natural language from multiple social networks and presenting data appropriately as result; thereby helping users who belong to multiple networks manage diverse profiles. For example: "Friends who studied from Advanced Institute of Technology and Management passed in 2016". The resultant will be user's all friends who studied from Advanced Institute of Technology and Management passed in 2016 and are there in the profiles of his/her existing multiple networks. The Proposed aggregator will maintain several accounts at one place and extracts the relevant publicly available data. It will also offer a better improvement over keyword Searching. It will perform a combine search about their friend's likes, places, pictures, events, applications, things they love or care about etc.. The Search will be based on both the content of the user and their friends' profiles.

For this approach, we propose a social web where people can come and interact with each other. They can connect with Friends, share thoughts, photos, media, events and knowledge. They can also search for the information available at their several accounts of social networking sites in a natural language. We will develop a crawler which will extract the information of the user from its several profiles existing at different networking sites. It will thus create a graph of his/her network and cluster the events accordingly and store it in a database. User's query will be divided into the tokens and these keywords will be given to query processor to search from the database and thus the result is submitted to the user. The resultant will be user's all information that exists in the profile of his/her multiple networks.

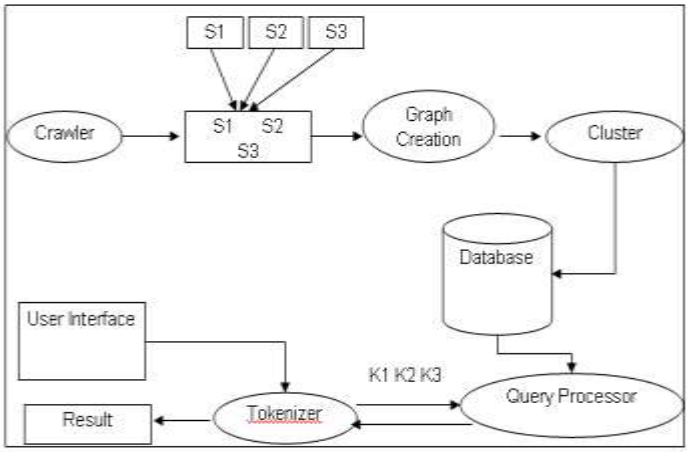

Figure 1 Proposed Model of Social Network Aggregator

Document clustering (or text clustering) is the application of cluster analysis to textual documents. It has applications in automatic document organization, topic extraction and fast information retrieval or filtering. Document clustering organizes documents into different groups called as clusters, where the documents in each cluster share some common properties according to defined similarity measure. The fast and high-quality document clustering algorithms play an important role in helping users to effectively navigate, summarize, and organize the information.

IV. RESULT AND ANALYSIS

The main idea of this approach is to classify sentences to a hierarchical Clustering which captures the theme of the sentence and then calculate a similarity measure between the sentence and the document that it belongs to. Our approach uses IF -IDF using Document Clustering. The proposed approach involves web fetching approach, Document Segmentation, Huge amount of work has been done in this area and many algorithms have been proposed. But there is still need of improvement to increase accuracy and efficiency of the summary obtained for the document clustering and information retrieval. In this dissertation, a novel approach for the information retrieval is presented. In the above scheme will be proposed early document cluttering algorithm for selecting pages, then it's fairly easy to avoid bootstraps of the infinite loop kind.

V.CONCLUSION

'Most existing summarizers work in an extractive fashion, selecting portions of the input documents (e.g. sentences) that are believed to be more salient. Non-extractive summarization includes dynamic reformulation of the extracted content, involving a deeper understanding of the input text, and is therefore limited to small domains. Query-based summaries are produced in reference to a user query while generic summaries attempt to identify salient information in text without the context of a query .Consequently the query based summarization is best into its aspects as to be dealt with Document Clustering and TF-IDF for accurate and effective summarization results.

VI. FUTURE SCOPE

For the future work the same can be implemented using parallel computing with fully automated scenarios with better performance on the part of the time and speed using map-reduce and pre-processing step can be omitted as such. Even using certain ontology parameters the respective query can correlate the document relational or features patters in specialized manner and can defined automation for the same for prompt document summarization.

REFRENCES

1. Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. Short and tweet: experiments on recommending content from information streams. In CHI '10: Proceedings of the 28th international conference on Human factors in computing systems, pages 1185{1194, New York, NY, USA, 2010. ACM
2. Cli Lampe, Nicole Ellison, and Charles Steineld. A face (book) in the crowd: Social searching vs. social browsing. In Proceedings of the ACM Special Interest Group on Computer-Supported Cooperative Work, 2006.
3. Petter Brandtzaeg and Jan Heim. Why People Use Social Networking Sites.
4. In Ant A. Ozok and Panayiotis Zaphiris, editors, Online Communities and Social Computing, volume 5621, chapter 16, pages 143-152. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
5. Uldis Bojars, Alexandre Passant, John Breslin, and Stefan Decker. Social network and data portability using semantic web technologies. In Proceedings of the Workshop on Social Aspects of the Web, 2008.
6. Francesca Carmagnola, Fabiana Vernero, and Pierluigi Grillo. Sonars: A social networks-based algorithm for social recommender systems. In Proceddings of the 17th International Conference on User Modeling, Adaptation, and Personalization, 2009
7. Michael Chisari, The future of social networking. In Proceedings of the W3C Workshop on the Future of Social Networking, 2009
8. 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.
9. Chen, J., Nairn, R., Nelson, L., Bernstein, M., & Chi, E. (2010). Short and tweet: Experiments on recommending content from information streams. In CHI '10: Proceedings of the 28th international conference on human factors in computing systems (pp. 1185–1194).
10. .Rohani, Vala Ali; Ow Siew Hock (2010). "On Social Network Web Sites: Definition, Features, Architectures and Analysis Tools".Journal of Advancesin Computer Research (2): 41-53. Retrieved 2011-06-11
11. Resnick, P., Lacovou, N., Suchak, M., Bergstrom, P., Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In Proceedings of the ACM conference on computer supported cooperative work.
12. The Internet Archive. Http://www.archive.org/
13. Web crawler. Http://en.wikipedia.org/wiki/Web_crawler
14. Larbin Multi-purpose web crawler Http://larbin.sourceforge.net/index-eng.html
15. WebSPHINX: A Personal, Customizable Web Crawler Http://www.cs.cmu.edu/~rcm/websphinx
16. The Stanford WebBase Project Http://www-diglib.stanford.edu/~testbed/doc2/WebBase/