Semantic Search in Social Networks

Project Proposal for CS8803AIA

Ajay Choudhari, Mohit Jain, Avik Sinharoy, Min Zhang.
{ajay.choudhari, mjain8, avik.sinharoy, minzhang}@gatech.edu

MOTIVATION AND OBJECTIVES

A **Social Network**[1] is a set of social entities – such as people or organizations connected by a set of relationships - such as friendship, work or common area of interest. It can be visualized as a connected graph, where each *node* represents an actor and the edges (or *ties*) represent the association or behavioral relationship between two actors. The resulting structures are often very complex and extracting relevant information from a traversal of these interconnected links is a non-trivial problem.

A typical social networking service like Facebook provides a simple lookup service to search a friend or group just based on **keyword search**. It's a problem for user if he wants to **semantically search** for a friend or a group. Consider the following two scenarios:

i. Suppose a user wants to search for "people who love *sports"* - social network sites' search answers this question by searching for the keyword "*sports*" in the profile description of the users. But if a user mentions particular instance of sports – for instance *playing basketball* - as his hobby, then the search for keyword *sports* will not match his profile, even though "playing basketball" is a sport.

ii. Another scenario where a keyword search fails completely is *querying,* i.e., if a user wants to search for people between the age group 21-25 years. Then present search won't understand the semantics of the query and will return unwanted/random results.

Clearly there is a substantial gap in the capabilities of social network search functionality and the requirements of user queries.

Our project aims to develop an application which sits on the top of a social networking website and **provides semantic search capabilities** rather than **simple keyword matching.**

RELATED WORK

Current directions of web growth focus like *web services* and the *semantic web* focus on creating a web of distributed machine understandable data. TAP[2] provides an application framework upon which the semantic search is built. The paper described two implemented Semantic Search systems which are based on the denotation of the search query, augment traditional search results with relevant data aggregated from distributed sources. Lots of effort has been done in semantic knowledge extraction by search engine in web search like [3], [4] but a similar semantics-search oriented work has not been implemented in the sphere of social networks.

PROPOSED WORK

A. COMPLETE ARCHITECTURE FOR SEMANTIC SEARCH

The application proposed would perform a peer to peer search. We propose to build application as an 'implicit' mutual contract between the peers i.e., by adding our application into their profile, a peer agrees on allowing the application to access the profile and networks of the peer. In return application would enable the peer to search in the network of other peers belonging to different network and who has added the application too. The application will run on a central server which keeps track of peers which are active. Whenever a peer wants to search something, it would search not only in his network but also in all other network it is connected to through some peers. This facility is currently not available in the existing system. So as a whole, system can be viewed as centralized peer to peer network. This system would not maintain any kind of indexes for searching. All the searches would be done by indexing 'on the fly'. For enhancing performance we would be caching all queries and responses locally on server so that we can use them for our future searches.

As mentioned before, current search in social network consists of keyword match only and not based on the semantics of the query. Sometimes overcoming the semantic search, we have to do a heavy general search by specifying all possible terms. For example, for finding the "persons interested in sports", we can enter all the related terms like "swimming", "playing", "running" etc. In our proposed architecture we will handle such situations and users need not to specify all the possible sports categories.

A general approach for this semantic search is to cluster all semantically related terms. This can be done in the two ways:

- Using a semantic dictionary like WordNet to find the co-relation between the different terms.
- The semantic relationship can be addressed by Latent Semantic Analysis (LSA). This takes to collect huge number of documents, possibly from some personal blogs, and

convert such document to a term-document matrix with respect to the give term. LSA could calculate the semantic relations by the singular value decomposition (SVD) of this term-document matrix. Each pair of terms would be given a score according to their semantic relations. The higher the score, the stronger the semantic relation is. By LSA, we could give each pair of terms a semantic distance (score). *Based on the semantic distance, we could apply some clustering method, for example, k-means or k-center to cluster the terms.*

If a user wants to do semantic search for some keywords, we could augment the keyword sets by incorporating all the keywords in its clusters. For example, assume 'swimming', 'running' and 'sport' are in one cluster, when doing semantic search, it is equivalent if using either keyword, the system would automatically using all the keywords in this cluster to do search.

There is the issue of addressing the level of relevance in semantic matches. For example, if we have three sports, 'swimming', 'walking' and 'running'. If a user searches for 'running', we could either return all these three sports or just 'walking' and 'running' based on the different understanding of relevance since 'walking' and 'running' could be partitioned into a subcategory 'track' and obviously 'swimming' is not. We propose to explore methods of handling such issues with mechanisms like using a different number of clusters in k-means or k-center methods.

B. SCOPE OF THE PROJECT

Our implementation will basically deal with the development of a middleware for semantic search in the social network. This project has two major components:

i. *Finding the semantics of the search,*
ii. *Searching the query with the registered users to our application.*

Firstly, we will crawl the profile-description of the users who have registered with our application (the super node) and index their contents. The registered users will be represented as a node in graph and this graph will keep on expanding with the registration of the new users. We can't crawl the unregistered users because of the restrictions imposed by the social networks (i.e., only a friend can see the profile-description of a user and not everybody). If now a user submits a query then our "semantic extractor" will extract the semantics of the query and will search for it in the registered user network. For our initial implementation we will restrict our search to 6-7 hops inside the network and also restrict our semantic search to certain fields in the profile.

PLAN OF ACTION

Our tentative plan of action for realizing the project goals. It is based primarily on a division of the project as per the architectural components of the system.

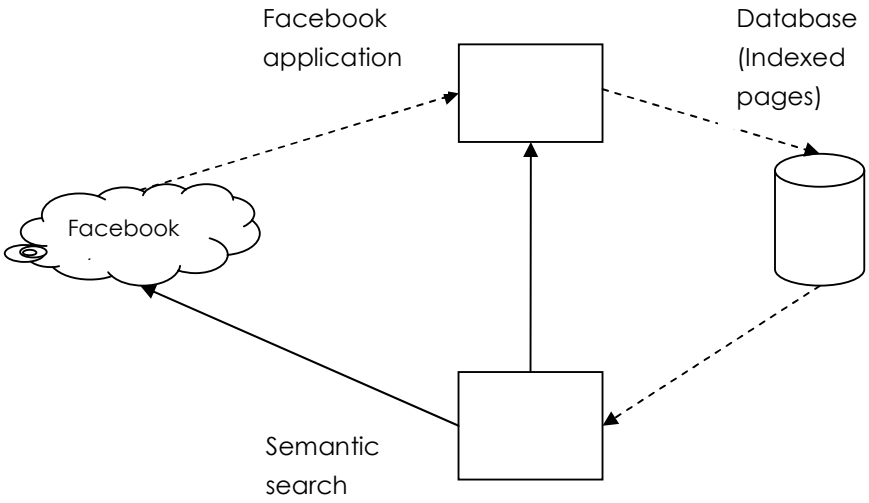


Figure 1 Tentative application architecture

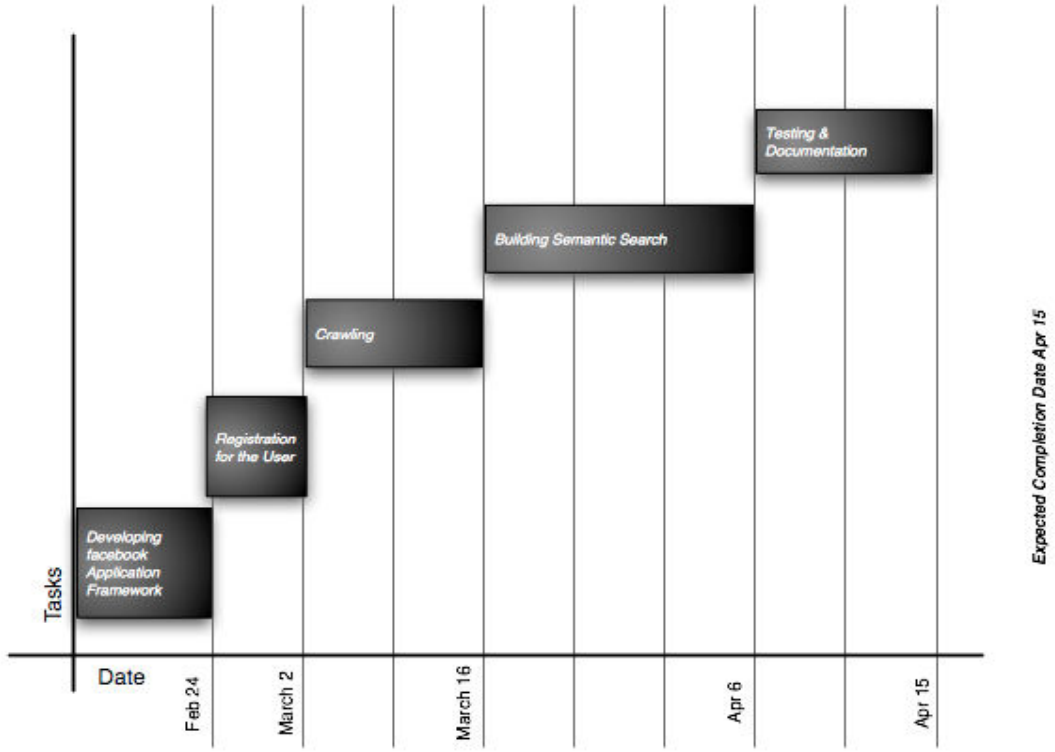
i. Develope a Facebook application as framework for the search tool. The application will need to provide user signup and search features as well as allow the search engine and backend to 'crawl' and extract user profile information.

ii. Build a network of registered users for the application. In order to test the application we need a reasonably populated graph of user nodes built up. The application sign-up procedure will include an implicit contract to make certain sections of user profile information available to our application.

iii. Build the search service at the server. The backend provides the following services:
- Maintain a graph of the system with registered users as nodes and their peer relationships as edges.
- Service search requests by initiating crawls over these graphs.
- Extract the mined user profile information from the profile pages and provide to the semantic search component
- Provide search results from the semantic search component to the user interface.

iv. Develop semantic search feature: This component is the core of our project idea. We propose to explore the semantic search by building domain knowledge in a prototype and later extending it to cover more semantic terms.
- Mining large document stores that have content similar to the contents of social network profiles. Personal blogs are a possible source we are considering.
- Building semantic keyword indices for these terms in a specific domain/field.
- Use this database for semantic search in user profiles from Facebook.

Tentative schedule



System Requirements:
Mainly we will need some space on CoC server to run our application and the Facebook API, which is available freely as open source.

EVALUATION AND TESTING METHOD

We will present our application as a plug in into the Facebook and try to test our system on Facebook after growing quite a satisfactory network of registered users. We will compare the search results in the present system to the results from our application.

BIBLIOGRAPHY

[1]. Mohsen Jamali, Hassan Abolhassani; Different Aspects of Social Network Analysis; *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*.

[2]. R. Guha, Rob McCool and Eric Miller; Semantic Search; World Wide Web Conference, May 20-24, 2003, Budapest, Hungary

[3].Bahadorreza Ofoghi, John Yearwood, Ranadhir Ghosh; A semantic approach to boost passage retrieval effectiveness for question answering. *Proceedings of the 29th Australasian Computer Science Conference* - Volume 48.

[4]. Fuchun Peng, Nawaaz Ahmed, Xin Li, Yumao Lu; Context sensitive stemming for web search; *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval.*