A Novel Approach for Query Suggestions for Personalizing the Web

R. Lokeshkumar, M. Shanmugapriya, P. Sengottuvelan

Abstract*—Web recommender systems predict the needs of web users and provide them with recommendations to personalize their pages. Such systems had been expected to have a bright future, especially in ecommerce and E-learning environments. However, although they have been intensively explored in the Web Mining and Machine learning fields, and there have been some commercialized systems, the quality of the recommendation and the user satisfaction of such systems are still not conclusive. In this paper we proposed a more robust approach that leverages search query logs for automatically identifying query groups for a number of different users and record the query logs and their respective sessions. The system uses query reformulation and click graphs which contain useful information on user behavior when searching online. Such information can be used effectively for the task of organizing user search histories into query groups. The proposed technique finds value in combining with keyword semantic similarity and filtering which applies knowledge gained from these query groups in various applications such as providing query suggestions for web personalization by favoring the ranking of search results.

Index Terms*— Web Mining, Collaborative filtering, Personalization, Ranking pages, recommended systems.

I. INTRODUCTION

As the size and richness of information on the Web grows, so does the variety and the complexity of tasks that users try to accomplish online. Users are no longer content with issuing simple navigational queries. Various studies on query logs (e.g., Yahoo's [1] and AltaVista's [2]) reveal that only about 20% of queries are navigational. The rest are informational or transactional in nature. This is because users now pursue much broader informational and task-oriented goals such as arranging for future travel, managing their finances, or planning their purchase decisions. However, the primary means of accessing information online is still through keyword queries to a search engine. One important step towards enabling services and features that can help users during their complex search quests online is the capability to identify and group related queries together. Recently, some of the major search engines have introduced a new "Search History" feature, which allows users to track their online search by recording their queries and clicks. This history includes a sequence of the queries displayed in reverse chronological order together with their corresponding clicks. Due to the advancement in computing and communication technologies especially in social networking sites like you tube, face book etc. enables people to get together and share information in innovative ways. This includes collaboration, communication and collective intelligence aiming in user interaction. This behavior analysis and patterns are to predict user behavior in heterogeneous (distinctive relations) social network. Distinctive connection limits the effectiveness of the prediction and provides an efficient approach to user behavior clustering in social network website; latent social dimensions are extracted based on network topology to capture the potential affiliations of users in the social network and the extracted social dimensions represent how each actor is involved in diverse affiliations and can be treated as features of actors for subsequent discriminative learning. The proposed (dictionary learning) algorithm which we organize user search histories and learning of user behavioral patterns in social media hence ranking can be performed.

Manuscript Received on November, 2013.

Prof. R. Lokeshkumar, Department of Information Technology, Bannari Amman Institute of Technology, Sathyamangalam, Erode Dist, Tamil Nadu, India

Prof. M. Shanmugapriya, Department of Information Technology, MP Nachimuthu M.Jaganathan Engineering College, Erode Dist, Tamil Nadu, India.

Dr. P. Sengottuvelan, Department of Information Technology, Bannari Amman Institute of Technology, Sathyamangalam, Erode Dist, Tamil Nadu, India.

II. RELATED WORK

Review Stage Recommender systems can be built in three ways: content based filtering, collaborative filtering, and hybrid systems. Content-based recommender systems, sometimes called information filtering systems, use behavioral user data for a single user in order to try to infer the types of item attributes that the user is interested in. Collaborative filtering compares one user's behavior against a database of other users behaviors in order to identify items that like-minded users are interested in. Even though content-based recommender systems are efficient in filtering out unwanted information and generating recommendations for a user from massive information, they find few if any coincidental discoveries.

On the other hand, collaborative filtering systems enable serendipitous discoveries by using historical user data. Collaborative filtering algorithms range from simple nearest neighbor methods [5, 17] to more complex machine learning based methods such as graph based methods [1, 15], linear algebra based methods [4,,7] and probabilistic methods. A few variations of filter both based algorithms [11, 12] and hybrid methods that combine content and a collaborative filtering have also been proposed to attack the so-called cold-start problem. Tapestry [9] is one of the earliest recommender systems. In this system, each user records their opinions (annotations) of documents they read, and these annotations are accessed by others' filters. GroupLens , Ringo and Video Recommender [10] are the earliest fully automatic recommender systems, which provide recommendations of news, music, and movies.

PHOAKS (People Helping One Another Know Stu) [20] crawls web messages and extracts recommendations from them rather than using users' explicit ratings. GroupLens has also developed a movie recommender system called MovieLens5 . Fab [2] is the first hybrid recommender system, which uses a combination of content-based and collaborative filtering techniques for web recommendations. Tango [6] provides online news recommendations and Jester [10] provides recommendations of jokes. Page et al. [19] and Kleinberg [16] first proposed a new concept of document relevance, often called document authority, and developed the PageRank and HITS algorithms, respectively, for better precision in web search. Both algorithms analyze the link structure of the Web to calculate document authorities. Haveliwala [12] proposed topic sensitive PageRank, which generates multiple document authorities biased to each specific topic for better document ranking. Note that our approach is different from general web search engines since we use user ratings rather than link structure for generating item authorities. Also, our approach is different from topic-sensitive PageRank since we provide personalized item authorities for each user rather than topic-biased item authorities. Also, our approach is different from recommender systems since it uses predictions of items as a ranking function for information search rather than generating recommendation

III. METHODOLOGY

A. Search Logger

In this method we develop a meta search engine as the backend search engines to ensure a broad topical coverage of the search results. The meta search engine collects click through data from the users and performs personalized ranking of the search results based on the learnt profiles of the users. Only top 100 results are returned to the user for every query posted by the user. The users are given the tasks to find results that are relevant to their interests. The clicked results are stored in the click through database and are treated as positive samples in training. The click through data, the extracted content concepts, and the extracted location concepts are used to create OMF profiles.

B. Web user Sessions

Understanding the user activity especially their interest can be analyzed using Web user session in clustering which conveys all the necessary information in the web. It also shows the time taken for the user to have an user interface .beginning of the user session is when the accessing of application is done by the user and when the process ends when the user quits the application. The session of activity is indicated by identifying each user with that a user unique IP address spends on a Web site during a specified period of time is called a user session.

The number of user sessions on a site is used in measuring the amount of traffic a Web site gets. The site administrator determines what the time frame of a user session will be (e.g., 30 minutes). If the visitor comes back to the site within that time period, it is still considered one user session because any number of visits within those 30 minutes will only count as one session. If the visitor returns to the site after the allotted time period has expired, say an hour from the initial visit, then it is counted as a separate user session. Contrast with unique visitor, hit, click-through and page view, which are all other ways that site administrators measure the amount of traffic a Web site gets. Identifying user-session plays an important role both in Internet traffic characterization and in the proper dimensioning of network resources. The typical behavior describes the informal definition of a user session can be obtained by user running a Web browser: an activity period (session), when the user browses the Web, alternates with a silent period over which the user is not active on the Internet. This activity period, named session in this paper, may comprise several TCP connections, opened by the same host toward possibly different servers

C. Update Checker

The Updater checks at specific time intervals for unsettled data collections. For each such collection, the Updater performs a number of database updating tasks:

- It adds the user key and his metadata on the Users-Attributes table or replaces already existing records for that specific user.
- It checks for not yet discovered URLs and adds them to the Distinct URLs table.
- It parses the URLs, their titles and semantic tags and adds possible new keywords to the Inverted Index table. The Inverted Index table maps each word with a unique number, a word-ID that is used as an identifier for all URLs that are related to this word.
- It adds all distinct connections between word-IDs, URLs and users to the Bucket tables. In Buckets, we actually store a detailed decomposition of how words, URLs and users interact with each other. For each word, URL and user connection, a value is also stored that is later used from the Users Rank module to order pages. Currently, Searches gives higher values to keywords that are found to URL titles (semantic tags). The system uses a number of different Buckets to keep them reasonable in size.

D. Bucket Creation for Personalization

For each query posed, an dictionary is created to have an semantic similarity for each queries that have been posed by the user with assigned user id. Once the similarity is found, only the related pages get stored in the dictionary in order to provide memory optimization. Finally based on the threshold, the pages get aligned by the means of relativity. Moreover the related URLs clicked by the user gets stored in the respective search logger.

E. Behavioral Analysis

To predict the behavioral pattern of the users in social networking site, the numbers of users in the particular social media followed by their attributes are taken into account. Behavioral features like network bandwidth, message count, pair behavior. The attributes of the users can be

- The contact network between the users in the social media.
- The number of shared friends between two users in the social media.
- The number of shared subscriptions between two users.
- The number of shared subscribers between two users.
- The number of shared favorite videos.

F. Pattern Discovery

Pattern discovery can be done using forward subset select based regression for the input matrix. The matrix can be obtained by greedy pursuit and convex relaxation. First is Greedy pursuit involves matching and orthogonal pursuit by selecting one atom per iteration. In Contrast, an stage wise orthogonal matching pursuit is proposed for selection of more than one atom per iteration. Second is convex relaxation which is slow compared to former one. The one approach used is basis pursuit to measure the sparseness. Another is least angle regression to overcome the computational complexity. To improve the performance further, the least angle regression and stage wise orthogonal pursuit are combined which emerge as new approach called stage wise least angle regression. This new approach is faster than the previous approach (stage wise orthogonal pursuit).

G. Ranking

When a user poses a query, the Search Query Analyzer parses and analyzes it. For each non trivial word, the Analyzer finds the relative bucket and word-ID from the inverted index table. The Users Rank module produces an ordered list of the relative URLs based on their aggregated values. When we work with multiple word queries, we internally get a result set for each word inside the query. The result sets are then merged by multiplying their original single word score and by giving a disproportional benefit to results that are presented in multiple set pairs. This way single word matches are not excluded from the results but better matches combined with a good Users Rank are more probable to occur at the first spots. It is possible to augment the search query with restrictions about the participants.

IV. IMPLEMENTATION RESULTS

A. Web user Session

Understanding the user activity especially their interest can be analyzed using Web user session in clustering which conveys all the necessary information in the web. It also shows the time taken for the user to have an user interface Beginning of the user session is when the accessing of application is done by the user and when the process ends when the user quits the application. The session of activity is indicated by identifying each user with that a user unique IP address spends on a Web site during a specified period of time is called a user session.

The number of user sessions on a site is used in measuring the amount of traffic a Web site gets. The site administrator determines what the time frame of a user session will be (e.g., 30 minutes). If the visitor comes back to the site within that time period, it is still considered one user session because any number of visits within those 30 minutes will only count as one session. If the visitor returns to the site after the allotted time period has expired, say an hour from the initial visit, then it is counted as a separate user session.

Contrast with unique visitor, hit, click-through and page view, which are all other ways that site administrators measure the amount of traffic a Web site gets. Identifying

user-session plays an important role both in Internet traffic characterization and in the proper dimensioning of network resources. The typical behavior describes the informal definition of a user session can be obtained by user running a Web browser: an activity period (session), when the user browses the Web, alternates with a silent period over which the user is not active on the Internet. This activity period, named session in this paper, may comprise several TCP connections, opened by the same host toward possibly different servers.

B. Click through Streaming

Search engine stores the click through data in triplet

Form (q, r, c) where

Q-query

R-ranking presented to the user

C- Clicked links or URLs.

Once the user posts the query, each query gets assigned with an unique user id. This gets stored in the query log along with query word and the presented ranking after an effective relevance measure Two Events based on link(before creating dataset) level0 and level 1.

Steps in Click through Streaming

1) Collecting Dataset

2) Dataset (Query id, Query, Time of Query, Item Rank and Clicked URL)

C. Recording Click through Streaming

Initial step in this proposed algorithm is based on recording the user clicks and storing the recorded links in data dictionary (log file). Various clicks over several URLs are recorded which have been already termed as Search History. Especially each query is assigned with an unique user id, the ranking is based on the search results that are retrieved by multi users. Efficient Retrieval makes the Search more efficient and flexible and hence the quality is improved and finally file search and ranking is done in an user friendly.

D. Ranking SVM in Co-Training Framework:

SVM takes input as a click though data i.e., the items in the search result by the user and ranking the link based on the search history of the user. As the first step in ranking, the algorithm initially categorizes the obtained input as labeled data set, containing the scanned datum and unlabelled data set, containing un-scanned datum. After then, labeled data set gets augmented with unlabelled data set. To automatically optimize the retrieval quality of search engines using click through data. a good information retrieval system should present relevant documents high in the ranking, with less relevant documents following below. The previous approach to learning retrieval functions requires training data generated from relevance judgments by experts. This makes an difficult and expensive task.

The Users Rank module produces an ordered list of the relative URLs based on their aggregated values. When we work with multiple word queries,

we internally get a result set for each word inside the query. The result sets are then merged by multiplying their original single word score and by giving a disproportional benefit to results that are presented in multiple set pairs. This way single word matches are not excluded from the results but better matches combined with a good Users Rank are more probable to occur at the first spots. It is possible to augment the search query with restrictions about the participants. This makes an efficient computation cost for analyzing search results based on the user preference and parsing is done on the specified source file It parses the URLs, their titles and semantic tags and adds possible new keywords to the Inverted Index table. The Inverted Index table maps each word with a unique number, a word-ID that is used as an identifier for all URLs that are related to this word. It adds all distinct connections between word-IDs, URLs and users to the Bucket tables. In Buckets, we actually store a detailed decomposition of how words, URLs and users interact with each other. For each word, URL and user connection, a value is also stored that is later used from the Users Rank module to order pages. Currently, Searches gives higher values to keywords that are found to URL titles (semantic tags). The system uses a number of different Buckets to keep them reasonable in size.

E. Parsing Process

The number of user sessions on a site is used in measuring the amount of traffic a Web site gets. The site administrator determines what the time frame of a user session will be (e.g., 30 minutes).

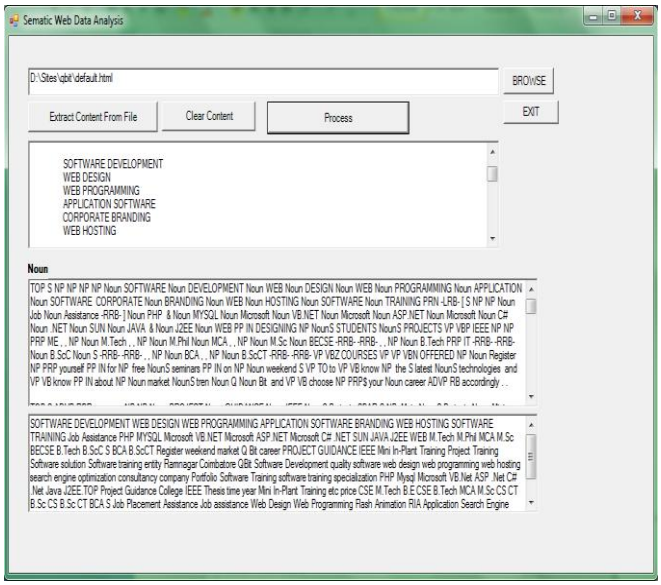

Fig. 1. Parsing Process

If the visitor comes back to the site within that time period, it is still considered one user session because any number of visits within those 30 minutes will only count as one session. If the visitor returns to the site after the allotted time period has expired, say an hour from the initial visit, then it is counted as a separate user session.

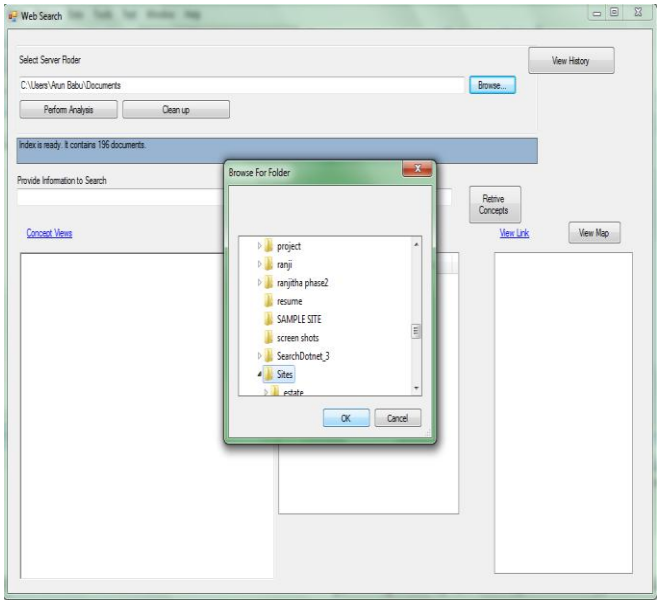

Fig. 2. Searching from Source.

Contrast with unique visitor, hit, click-through and page view, which are all other ways that site administrators measure the amount of traffic a Web site gets. Identifying user-session plays an important role both in Internet traffic characterization and in the proper dimensioning of network resources.

F. Search Results

Once the parsing gets done, the registered users can pose their query and various links that are perfectly related to the posed query gets displayed. The result page shows the various URL related to search and their ranking based on various user search histories.

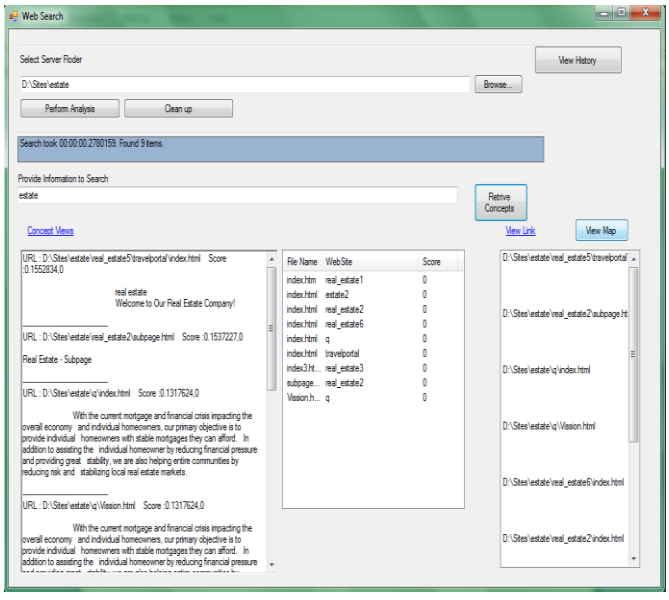

Fig. 3. Searching after Parsing.

REFERENCES

1. G J. Teevan, E. Adar, R. Jones, and M. A. S. Potts, "Information reretrieval: repeat queries in yahoo's logs," in SIGIR. New York, NY, USA: ACM, 2007.
2. A. Broder, "A taxonomy of web search," SIGIR Forum, vol. 36, no. 2, pp. 3–10, 2002.
3. A. Spink, M. Park, B. J. Jansen, and J. Pedersen, "Multitasking during Web search sessions," Information Processing and Manage- ment, vol. 42, no. 1, pp. 264–275, 2006.
4. R. Jones and K. L. Klinkner, "Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs," in CIKM, 2008.
5. P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna, "The query-flow graph: Model and applications," in CIKM, 2008.
6. D. Beeferman and A. Berger, "Agglomerative clustering of a search engine query log," in KDD, 2000.
7. R. Baeza-Yates and A. Tiberi, "Extracting semantic relations from query logs," in KDD, 2007.
8. J. Han and M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
9. W. Barbakh and C. Fyfe, "Online clustering algorithms," International Jthenal of Neural Systems, vol. 18, no. 3, pp. 185–194, 2008.
10. M. Berry and M. Browne, Eds., Lecture Notes in Data Mining. World Scientific Publishing Company, 2006.
11. V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," Soviet Physics Doklady, vol. 10, p. 707, 1966.
12. M. Sahami and T. D. Heilman, "A web-based kernel function for measuring the similarity of short text snippets," in WWW '06: Proceedings of the 15th international conference on World Wide Web. New York, NY, USA: ACM, 2006, pp. 377–386.
13. J.-R. Wen, J.-Y. Nie, and H.-J. Zhang, "Query clustering using user logs," ACM Transactions in Information Systems, vol. 20, no. 1, pp. 59–81, 2002.
14. A. Fuxman, P. Tsaparas, K. Achan, and R. Agrawal, "Using the wisdom of the crowds for keyword generation," in WWW, 2008.
15. K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova, "Monte carlo methods in PageRank computation: When one iteration is sufficient," SIAM Jthenal on Numerical Analysis, vol. 45, no. 2, pp. 890–904, 2007.
16. L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," in Technical report, Stanford University, 1998.
17. P. Boldi, M. Santini, and S. Vigna, "Pagerank as a function of the damping factor," in WWW, 2005.
18. T. H. Haveliwala, "Topic-sensitive PageRank," in WWW, 2002.
19. W. M. Rand, "Objective criteria for the evaluation of clustering methods," Jthenal of the American Statistical Association, vol. 66, no. 336, pp. 846–850, 1971.
20. D. D. Wackerly, W. M. III, and R. L. Scheaffer, Mathematical Statistics with Applications, sixth edition ed. Duxbury Advanced Series, 2002.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

AUTHORS PROFILE

Mr. R. Lokeshkumar, received B.E Degree in Computer Science & Engineering Anna University, Chennai in 2006 and M.Tech Degree in Information Technology from Faculty of Engineering and Technology, Anna University, Coimbatore in 2009. Currently he is working as Assistant Professor in the Department of IT, Bannari Amman Institute of Technology, Sathyamangalam. He is doing part time research in Data Mining at Anna University. His current research focuses on Data Mining, Data Base Systems, Web mining, Adhoc networks. He is a member of ISTE.

Mrs. M. Shanmugapriya, received M.Sc Degree in Mathematics from Bharathiyar University, Coimbatore 2006 and M.Tech Degree in Information Technology from Faculty of Engineering and Technology, Anna University in 2009. Currently she is working as Assistant Professor in the Department of IT, M.P.Nachimuthu M Jaganathan Engineering College, Erode District, INDIA. She is doing part time research in Data Mining at Anna University. Her current research focuses on Data Mining.

Dr. P. Sengottuvelan, received M.Sc., Degree in Computer Technology from Periyar University, Salem in 2001 and Master of Philosophy in Computer Science from Bharathiar University, Coimbatore in 2003 and M.E. degree in Computer Science & Engineering from Anna University, Chennai in 2004. He also received his Ph.D in degree in Computer Science & Engineering Vinayaka Missions University, Salem in 2010. Since 2004, he has been the Faculty in the Department of IT, BIT, Sathyamanagalam. His current research focuses on Concurrent Engineering, Multi Agent System networks, Constraint Management Agents He is member of IACSIT, ACEEE, IAENG and Life Member of FUWA and ISTE.