Volume 5, Issue 9, September 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Web Graph Based Recommendations Identification for Query Suggestion with Efficient Search Reranking

S. Indhumathi, G.ijaybaskar

Research Scholar, Assistant. Professor,
PGP College of Arts & Science, Dept of CS & Applications,
Tamil Nadu, India

Abstract- With the diverse and explosive growth of Web information, how to organize and utilize the information effectively and efficiently has become more and more critical. The first challenge is that it is not easy to recommend latent semantically relevant results to users. Take Query Suggestion as an example; there are several outstanding issues that can potentially degrade the quality of the recommendations, which merit investigation and the ambiguity which commonly exists in the natural language. Queries containing ambiguous terms may confuse the algorithms which do not satisfy the information needs of users. The second challenge is how to take into account the personalization feature. This research aims at providing a general framework on mining Web graphs for recommendations, 1) A novel diffusion method is proposed which propagates similarities between different nodes and generates recommendations; 2) then it is illustrated how to generalize different recommendation problems into the graph diffusion framework.

The proposed framework can be utilized in many recommendation tasks on the World Wide Web, including query suggestions, tag recommendations, expert finding, image recommendations, image annotations, etc. In this study, abbreviation based Query suggestion is also considered. In addition, search engine results comparison is also considered. Personalized recommendations are given importance. Previous search query words are also taken into query suggestion calculation.

Keywords- web usage mining, Graph construction, Query suggestion, web queries, and bi-partite graph.

I. INTRODUCTION

The system is aimed at solving the recommendation problem and proposed a general framework for the recommendations on the Web. This framework is built upon the heat diffusion on both undirected graphs and directed graphs, and has several advantages. It is a general method, which can be utilized to many recommendation tasks on the Web and it can provide latent semantically relevant results to the original information need. This model provides a natural treatment for personalized recommendations. The designed recommendation algorithm is scalable to very large data sets. The system introduced a graph diffusion model for recommendation. It shows how to convert different Web data sources into correct graphs in the models; it conducts several experiments on query suggestions since Query Suggestion is a technique widely employed by commercial search engines to provide related queries to users' information need. Graph construction and Query Suggestion algorithm is used to provide related queries. Graph construction and Query suggestion algorithm is implemented. Similarity measure between two queries is also considered.

Abbreviation based query suggestion is also considered. Personalized recommendations are given importance. Previous search query words are also taken into query suggestion calculation.

The proposed methodology provides the following: Expansions of abbreviated query are considered and new search mechanism satisfies the user's information needs. And also unwanted Web page links are eliminated to better extent.

II. PROBLEM FORMULATION

The related query-link and top most URLs does not identified. The redundant search page and irrelevant web links are not avoided. The visit count value does not increased. The recommendations of query phrase with related link does not to be categorized. The query which is not link to the related web links. The search page does not contains the abbreviated word suggestion. The visit count value dose not display in the search page. The query word does not match the URLs. The related information's are does not display for the required query word.

The existing system is aimed at solving the recommendation problem and proposed a general framework for the recommendations on the Web. This framework is built upon the heat diffusion on both undirected graphs and directed graphs, and has several advantages. It is a general method, which can be utilized to many recommendation tasks on the Web. It can provide latent semantically relevant results to the original information need. This model provides a natural treatment for personalized recommendations. The designed recommendation algorithm is scalable to very large data sets.

The existing system introduced a graph diffusion model for recommendation. It shows how to convert different Web data sources into correct graphs in the models; It conducts several experiments on query suggestions since Query

Suggestion is a technique widely employed by commercial search engines to provide related queries to users' information need. Graph construction and Query Suggestion algorithm is used to provide related queries.

- Although the query suggestion is effective, the result is not compared with other search engine outputs.
- Redundant search pages and irrelevant web links are not eliminated.
- Expansions of abbreviated query such as (OS for Operation System) are not considered.
- Chart based analysis is not provided for viewing search result quality.

III. QUERY SUGGESTION ALGORITHM

Query expansion that is embedded into a top-k query processor with candidate pruning. Traditional query expansion methods select expansion terms whose thematic similarity to the original query terms is above some specified threshold, thus generating a disjunctive query with much higher dimensionality. A priority queue is used for maintaining result candidates, the pruning of candidates is based on Fagin's family of top-k algorithms, and optionally probabilistic estimators of candidate scores can be used for additional pruning.

All methods aim to generate additional query terms that are "semantically" or statistically related to the original query terms, often producing queries with more than 50 or 100 terms and appropriately chosen weights. Given the additional uncertainty induced by the expansion terms, such queries are usually considered as disjunctive queries Here the query suggestion algorithm is used to evaluate the enhanced result in web mining, this concept is used to process the query suggestion algorithm for query by query search, improved query by query search and abbreviation based query suggestion. These information extracted in the basis of top k-query result hit rate. Each and every link click process, the hit rate will be increased for the purpose of further reference. The user can give abbreviated word in the search control to avoiding the time consumption.

IV. OVERVIEW OF THE SYSTEM

Query Suggestion

Query Suggestion is a technique widely employed by commercial search engines to provide related queries to users' information need. In this section, user demonstrates how the method can benefit the query suggestion, and how to mine latent semantically similar queries based on the users' information need.

Data Collection

The query suggestion graph based on the click through data of the AOL search engine. In total, this data set spans 3 months from 01 March, 2006 to 31 May, 2006. There are a total of 19,442,629 lines of click through information, 4,802,520 unique queries, and 1,606,326 unique URLs. Click through data record the activities of Web users, which reflect their interests and the latent semantic relationships between users and queries as well as queries and clicked Web documents. Each line of click through data contains the following information: a user ID (u), a query (q) issued by the user, a URL (l) on which the user clicked, the rank (r) of that URL, and the time (t) at which the query was submitted for search. From a statistical point of view, the query word set corresponding to a number of Web pages contains human knowledge on how the pages are related to their issued queries.

Thus, in this research, we utilize the relationships of queries and Web pages for the construction of the bipartite graph containing two types of vertices hq; li. The information regarding user ID, rank and calendar time is ignored. This data set is the raw data recorded by the search engine, and contains a lot of noise which will potentially affect the effectiveness of our query suggestion algorithm. Hence, they conduct a similar method employed to clean up the raw data. We filter the data by only keeping those frequent, well formatted, English queries (queries which only contain characters "a,", "b,"…,"z," and space). After cleaning and removing duplicates, we get totally 2,019,265 unique queries and 915,771 unique URLs in our data collection. After the construction of the query-URL bipartite graph using this data collection procedure.

Graph Construction

The bipartite graph extracted from the click through data into the diffusion processes since this bipartite graph is an undirected graph, and cannot accurately interpret the relationships between queries and URLs. Hence, we convert this bipartite graph into Fig. 2b. In this converted graph, every undirected edge in the original bipartite graph is converted into two directed edges. The weight on a directed query-URL edge is normalized by the number of times that the query is issued, while the weight on a directed URL-query edge is normalized by the number of times that the URL is clicked.

Query Suggestion Results

The query suggestions result is generated. It is based on the intuition that two queries are very similar if they link to a lot of similar URLs. On the other hand, two URLs are very similar if they are clicked as a result of several similar queries. Based on this intuition, in this module, the similarities between URLs are calculated, and then the similarities for queries are computed based on the similarities of URLs. Suppose two query phrases display the many similar URLs, and then if one query is types, the second query phrase is also suggested.

Previous Search Query Words Based Query Suggestion

During the query phrase suggestion, when few words of query phrase is typed, most typed queries starting with (or containing) these words in the past are all fetched and the query phrases are displayed as suggestion.

Abbreviation Based Query Suggestion

In this module, during the query phrase suggestion, when an abbreviated word is types such as OS, then query phrases like Operating System, Open Source and the like phrases are suggested. The abbreviated and expansion words are kept in a database table, fetched and displayed during query typing.

V. PROPOSED ALGORITHM

Algorithm1: QUERY SUGGESTION BY QUERY SEARCH (QSQS)

```
Input: a set RefDocs of reference documents and a number K
Output: a set QS containing K query suggestions
// Lexical Alias Search
for all d 2 RefDocs do
    use Query Search to find a lexical alias for d, LAd, based on d’s title terms and most frequent terms
end for
// Query Suggestion Candidate Search
initialize QSC, the set of query suggestion candidates, to be empty.
for all d 2 RefDocs do
    use Query Search to find the set, QSCd, of minimal subqueries of LAd that cover d
    QSC = QSC [ QSCd
end for
// Greedy Selection of final Query Suggestions
initialize QS to be empty.
for i = 1 to K do
    add to QS the query qs 2 QSC that most increases MCC (break ties to max increase MEC)
    remove qs from QSC
end for
```

Algorithm 2: IMPROVED QUERY SELECTION BY QUERY SEARCH (IQSQS)(PREVIOUS QUERY BASED SEARCH)

```
Input: the user’s query, Q0, the set RefDocs of reference documents for Q0, a number N 20,
and a number K
Output: a set QS containing K query suggestions
// Term Selection
for all d 2 RefDocs do
    quickly find a set, Fd, of up to 20 terms in d that are likely to be useful in constructing queries with high coverage
    score each term in Fd according to its coverage when combined with Q0 to form a query
    sort Fd (highest scoring term first) and delete all but the first N terms
end for
// Query Suggestion Candidate Generation
initialize QSC, the set of query suggestion candidates, to be empty.
for all d 2 RefDocs do
    generate a set, QSCd, of queries built from terms in Fd
    QSC = QSC [ QSCd
end for
// Greedy Selection of final Query Suggestions
initialize QS to be empty.
for i = 1 to K do
    add to QS the query qs 2 QSC that most increases MCC (break ties to max increase MEC)
    remove qs from QSC
end for
```

VI. EFFICIENT SEARCH RERANKING

The first step in the software development life cycle is the identification of the problem as the success of the system depends largely on how accurately a problem is identified. At present, although the query suggestion is effective, the result is not compared with other search engine outputs. Redundant search pages and irrelevant web links are not eliminated.

Expansions of abbreviated query such as (OS for Operation System) are not considered. Chart based analysis is not provided for viewing search result quality. To overcome the problem, an application is required and it should be capable of improving the query suggestion.

The diverse and explosive growth of Web information, how to organize and utilize the information effectively and efficiently has become more and more critical. This is especially important for Web 2.0 related applications since user-generated information is more freestyle and less structured, which increases the difficulties in mining useful information from these data sources. In order to satisfy the information needs of Web users and improve the user experience in many Web applications, Recommender Systems, have been well studied in academia and widely deployed in industry.

Typically, recommender systems are based on Collaborative Filtering which is a technique that automatically predicts the interest of an active user by collecting rating information from other similar users or items. The underlying assumption of collaborative filtering is that the active user will prefer those items which other similar users. Based on this simple but effective intuition, collaborative filtering has been widely employed in some large, well-known commercial systems.

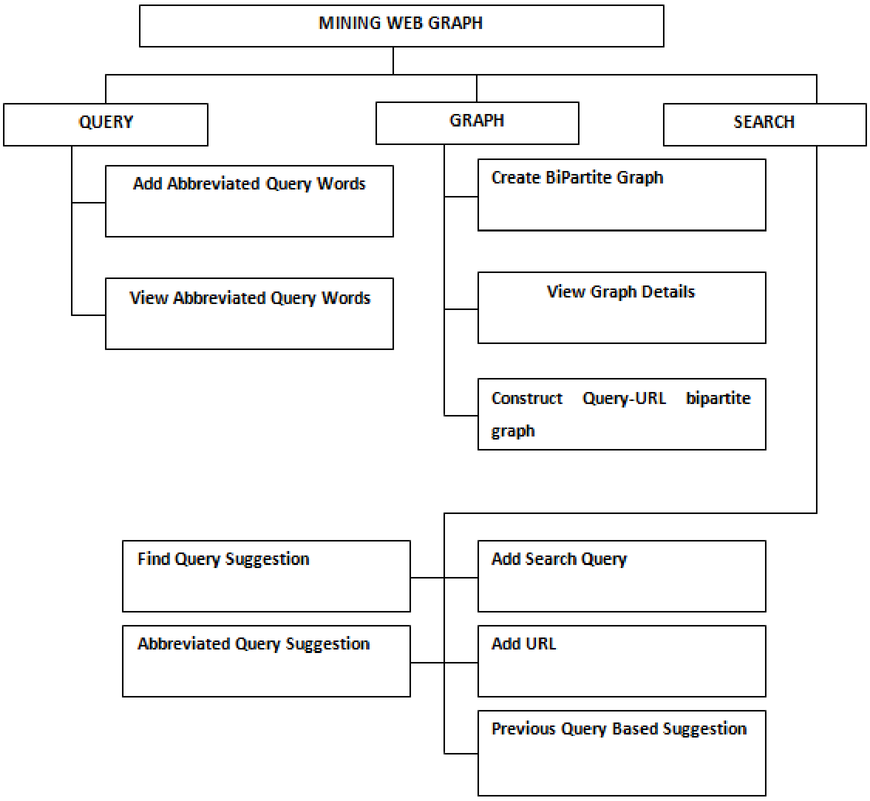
Typical collaborative filtering algorithms require a user-item rating matrix which contains user-specific rating preferences to infer users' characteristics. However, in most of the cases, rating data are always unavailable since information on the Web is less structured and more diverse. Fortunately, on the Web, no matter what types of data sources are used for recommendations, in most cases, these data sources can be modeled in the form of various types of graphs. If user can design a general graph recommendation algorithm, we can solve many recommendation problems on the Web.

However, when designing such a framework for recommendations on the Web, we still face several challenges that need to be addressed. The first challenge is that it is not easy to recommend latent semantically relevant results to users. Take Query Suggestion as an example, there are several outstanding issues that can potentially degrade the quality of the recommendations, which merit investigation. The first one is the ambiguity which commonly exists in the natural language. Queries containing ambiguous terms may confuse the algorithms which do not satisfy the information needs of users. Another consideration, as reported in is that users tend to submit short queries consisting of only one or two terms under most circumstances, and short queries are more likely to be ambiguous.

Through the analysis of a commercial search engine's query logs recorded over three months in 2006, we observe that 19.4 percent of Web queries are single term queries, and further 30.5 percent of Web queries contain only two terms. Third, in most cases, the reason why users perform a search is because they have little or even no knowledge about the topic they are searching for. In order to find satisfactory answers, users have to rephrase their queries constantly. The second challenge is how to take into account the personalization feature. Personalization is desirable for many scenarios where different users have different information needs. As an example, Amazon.com has been the early adopter of personalization technology to recommend products to shoppers on its site, based upon their previous purchases.

The adoption of personalization will not only filter out irrelevant information to a person, but also provide more specific information that is increasingly relevant to a person's interests. the Web pages are seen as ontology individuals, frequent navigational patterns are in the form of ontology instances instead of Web page addresses, and page clustering is done using semantic similarity. The result is used for generating web page recommendations to users. The recommender engine presented in this paper which is based on semantic patterns and page clustering creates a list of appropriate recommendations. The Semantic Web is just this: today's Web enriched by a formal semantics ex-pressed as ontology's that captures the meaning of pages and links in a machine-understandable form. The main idea of the Semantic Web is to enrich the current Web by machine-process able information in order to allow for semantic-based tools supporting the human user.

VII. ARCHITECTURE DIAGRAM


MINING WEB GRAPH
QUERY
GRAPH
SEARCH
Add Abbreviated Query Words
View Abbreviated Query Words
Create BiPartite Graph
View Graph Details
Construct Query-URL bipartite graph
Find Query Suggestion
Abbreviated Query Suggestion
Add Search Query
Add URL
Previous Query Based Suggestion

VIII. CONCLUSION

The existing system does not deals with the query suggestion in the query search page. The visit count suggestion is not provide for this web information's. The top most URLs does not identified. The redundant search page and irrelevant web links are not avoided. The recommendations of query phrase with related link does not to be categorized.

In this method, which can be utilized to many recommendation tasks on the Web? It can provide latent

In the proposed system, Similarity measure between two queries is also considered. Abbreviation based query suggestion is also considered. Personalized recommendations are given importance. Previous search query words are also taken into query suggestion calculation.

Expansions of abbreviated query are considered. The new search mechanism satisfies the user's information needs. Unwanted Web page links are eliminated to better extent. To suggest the users, which query more visited the same URL means it is best one of the URLs in that query. This query suggestion result displays by using visit count value.

IX. FUTURE ENHANCEMENT

The project has covered almost all the requirement. Further requirements and improvements can easily be done since the coding in mainly structured or modular in nature. Improvements can be appended by changing the existing modules or adding new modules.

Several areas to be developed in future, so the application must be upgraded for the new ones required and it is possible to modifications according to new requirements and specifications.

The project deals with the query and URLs which is stored in the database, and it will be displayed the by the use of web browser control.

The Future Analysis of this project as follows:

- In future, same project will developed in web based application. It should not require software installation.
- The bipartite graph visit count value is added automatically.
- The undirected graph visit count to be retrieved from the search page result.

REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais, "Improving Web Search Ranking by Incorporating User Behavior Information," SIGIR '07: Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 19-26, 2006.

[2] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, 1999.

[3] D. Kelly and J. Teevan, Implicit feedback for inferring user preference: A bibliography. In SIGIR Forum, 2003.

[4] T. Joachims, Optimizing Search Engines Using Clickthrough Data. In Proceedings of the ACM Conference on Knowledge Discovery and Datamining (SIGKDD), 2002

[5] S. Fox, K. Karnawat, M. Mydland, S. T. Dumais and T. White. Evaluating implicit measures to improve the search experience. In ACM Transactions on Information Systems, 2005

[6] N. Pharo, N. and K. Järvelin. The SST method: a tool for analyzing web information search processes. In Information Processing & Management, 2004

[7] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay, Accurately Interpreting Clickthrough Data as Implicit Feedback, Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR), 2005

[8] N. Craswell and M. Szummer, "Random Walks on the Click Graph," SIGIR '07: Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 239-246, 2007.

[9] T. Joachims. Optimizing search engines using clickthrough data. In KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 133–142, New York, NY, USA, 2002. ACM Press.

[10] R. Baeza-Yates, C. Hurtado, M. Mendoza, and G. Dupret. Modeling user search behavior. In LA-WEB '05: Proceedings of the Third Latin American Web Congress, page 242, Washington, DC, USA, 2005. IEEE Computer Society.

[11] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval, pages 154–161, New York, NY, USA, 2005. ACM Press.

[12] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 407–416, New York, NY, USA, 2000. ACM Press.

[13] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, and W. Fan. Optimizing web search using web click-through data. In CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management, pages 118–126, New York, NY, USA, 2004. ACM Press.

[14] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, "Query Expansion by Mining User Logs," IEEE Trans. Knowledge Data Eng., vol. 15, no. 4, pp. 829-839, July/Aug. 2003.