ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 11, November 2014

QUERY SUGGESTION AND RECOMMENDATION USING BIPARTITE GRAPH AND K-MEANS CLUSTERING

Dr.E.S.SAMUNDEESWARI[1], BRINDHA S[2]

Associate Professor, Department of Computer Science, Vellalar College for Women (Autonomous), Erode, India[1]

Vellalar College for Women (Autonomous), Erode, India[2]

Abstract: With the diverse and explosive growth of web information, retrieving, organizing and utilizing 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. The second challenge is to take into account the personalization feature. As the exponential explosion of various contents generated on the Web, Recommendation techniques have become increasingly indispensable. Innumerable different kinds of recommendations are made on the Web every day, including movies, music, images, books recommendations, query suggestions, tags recommendations, etc., The proposed work carries out query suggestion and recommends query and URL's using Bipartite graph and K-means clustering.

Keywords: Recommendation, Clustering, Query suggestion.

INTRODUCTION

Query suggestion is a technique widely employed by commercial search engines to provide related queries to users' information need. It plays an important role in improving the usability of search engines. Writing queries is never easy, because usually queries are short (one or two words on average) and words are ambiguous. To make the problem even more complicated, different search engines may respond differently to the same query. Therefore, there is no standard or optimal way to issue queries to search engines, and it is well recognized that query formulation is a bottleneck issue in the usability of search engines. Recently, most commercial search engines such as Google, Yahoo!, Live Search, Ask, and Baidu provide *query suggestions* to improve usability. Recommender system can use data mining techniques for making recommendations using knowledge learnt from the action and attributes of users. There are many different approaches to creating recommender systems, but the most common systems fall into two broad classes: *collaborative-filtering* systems [1] and *content-based* systems. Content-based systems make recommendations based on an item similarity measure, based on item features. On the other hand, Collaborative-filtering systems use patterns in user ratings to make recommendations. Both types of recommender systems require significant data resources. Recommendations for users are computed by finding items that are similar to other items the user has liked. Because the relationships between items are relatively static, item-based algorithms may be able to provide the same quality as the user-based algorithms with less online computation.

PROBLEM DEFINITION

Queries submitted to a Web search engine are usually short and ambiguous. Currently, most search engines respond to a user's query by using the bag-of-words model, which matches keywords between the query and Web documents but ignores contexts and users' preferences. Thus, many irrelevant results are returned by the conventional search engines. Thereby redundant search pages and irrelevant web links are eliminated. This proposed work considers abbreviated terms for query as users may feel comfortable if they could get search results for abbreviated terms also. Graphical analysis is included for viewing search result quality.

METHODOLOGY

Query suggestion refers to the process of suggesting related queries to search engine users. This framework first retrieves a set of query candidates from search engine logs using random walk and other techniques. Then re-rank the suggested queries in the order which maximizes the diversification function that measures the difference between the original search results and the results from suggested queries A set of relevant queries are suggested to a user on the search engine result page (SERP) after the user submitted a query. If the user is not satisfied with the results shown on the page, user may choose to click on one of the suggested queries to refine the search.

Generating alternative queries, also known as query suggestion, has long been proved useful to help a user explore and express his information need. In many scenarios, such suggestions can be generated from a large scale graph of queries and other accessory information, such as the click through. Empirical experiments on a

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 11, November 2014

large scale query log of a commercial search engine and a scientific literature collection show that hitting time is effective to generate semantically consistent query suggestions. Without involvement of twisted heuristics or heavy tuning of parameters, this method clearly captures the semantic consistency between the suggested query and the original query. Despite its simplicity, query suggestion has few benefits,

1. The suggestions generated with the proposed algorithm are semantically similar to the original query.
2. The suggestions generated do not have to occur with the original query.
3. This approach boosts the long tail queries as suggestions.
4. This provides a natural treatment for personalized query suggestions.

In this work, the keywords can be abbreviated. In the search box, the abbreviated word is given by the user. The search engine searches the related keyword as well as the queries. A single abbreviation may be mapped to multiple keywords. For instance, 'IP', an abbreviated keyword may mean Internet protocol, Instructor pilot, International paper, Industrial park, Industrial policy, Information processing, Installation plan etc., This list should be very helpful for the new user who has no knowledge about the exact term.

The keyword should be sorted based on the hit count and also click through by other users. The abbreviated keyword and the related URLs should be stored in database. When the user is searching a new keyword for the same abbreviation, it is also added in the database. The abbreviated and expansion of words are kept in a database table, fetched and displayed during query typing.

By intuition, one can say that two queries are very similar if they link to a lot of similar URLs and two URLs are very similar if they are clicked as a result of several similar queries. The similarities between URLs are calculated, and then the similarities for queries are computed based on the similarities of URLs using clustering.

Suppose two query phrases display many similar URLs, and then if one query is typed, the second query phrase is also suggested. The figure 1 shows the suggestion of queries for an abbreviation.

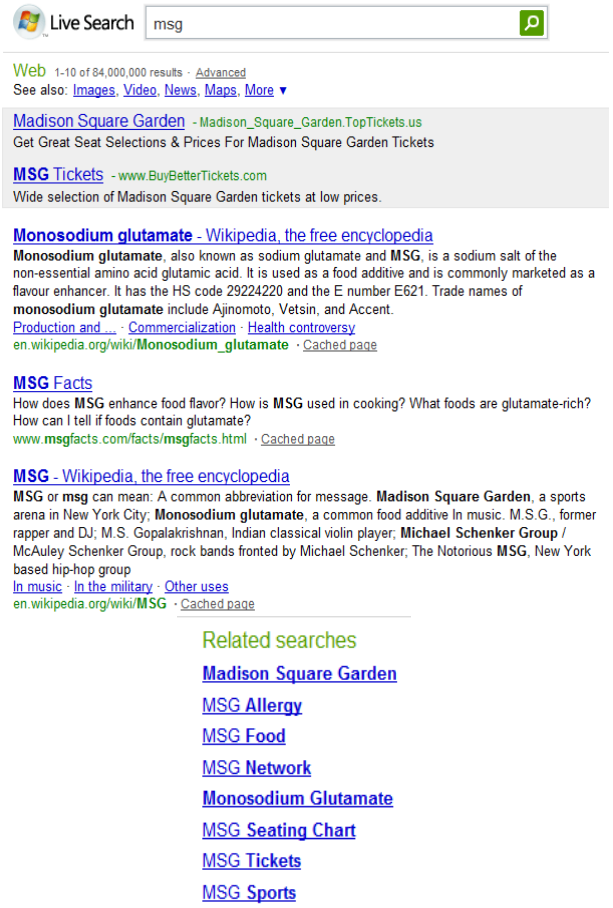

Fig.1: Query suggestion for abbreviation.

GRAPH CONSTRUCTION

A graph is denoted by G(V,E) where V is the vertex set and E is the edge set of the graph. A graph G(V,E) is bipartite with two vertex classes X and Y if V = X U Y with $X \wedge Y = \Phi$ and each edge in E has one endpoint in X and one endpoint in Y. A bipartite graph is constructed using the details of query and the related URL clicked.

QUERY-URL BIPARTITE GRAPH.

For the query-URL bipartite graph, consider an undirected bipartite graph. Eql {[q,l]} there is an edge from q and l is the set of all edges. The edge (q,l) exists if and only if a user u, clicked a URL l after issuing a query q. The values on the edges specify how many times a query is clicked on a URL.

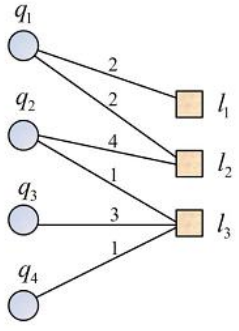

Fig.2 Query-URL bipartite graph

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 11, November 2014

QUERY/URL	U1	U2	U3
Q1	2	2	0
Q2	0	4	1
Q3	0	0	3
Q4	0	0	1

Table.1 Query-URL

CONVERTED QUERY-URL BIPARTITE GRAPH.

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.

A converted bipartite graph G= $(V + UV *, E)$ consists of query set V+ and URL set V*. The two directed edges are weighted using the method introduced .

Given a query q in v+, a sub graph is constructed by using depth-first search in G. The search stops when the number of queries is larger than a predefined number.

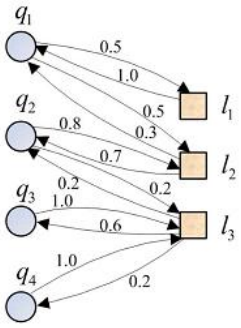

Fig.3 Converted Query-URL

PERSONALIZE THE BIPARTITIE GRAPH

Conventional approach to query recommendation has been focused on query-term based analysis over the user access logs. In this work, utilizing the connectivity of a query-URL bipartite graph to recommend relevant queries can significantly improve the accuracy and effectiveness of the conventional systems. The affinity graph consists of only queries as its vertices and its edges are weighted according to a query-URL vector based similarity (distance) measure.

A novel rank mechanism is employed for ordering the related queries based on the merging distances of a hierarchical agglomerative clustering. This ranking algorithm works with both native ranking that uses the query-URL similarity measure directly, and the single-linkage based ranking method. The experimental results from two query collections demonstrate the effectiveness and feasibility of this method. Each record consists of a user's query to a search engine along with the URL which the user selected from among the candidates offered by the search engine. By viewing this dataset as a bipartite graph, with the vertices on one side corresponding to queries and on the other side to URLs, one can apply an agglomerative clustering algorithm to the graph's vertices to identify related queries and URLs. One noteworthy feature of the proposed algorithm is that it is "content-ignorant"---the algorithm makes no use of the actual content of the queries or URLs, but only how they co-occur within the click through data.

In this work, predicting user's interest in a query is based on various sorts of information including query, user information and interactions between query and URLs clicked. User's visit time is also added in the database and also the number of times the user entered into that particular keyword, as well as visit count. The user's requirements and most visiting keywords and URL's are listed out.

CLUSTERING

Adaptive user clustering and profiling is essential to be able to accurately predict user actions. The results of clustering and personalization projects compare several distance measures used in clustering. A new measure to assess the quality of clustering independent of the distance measure used in the clustering-algorithm. The technique of clustering individual user sessions is seen as a method to aid in collaborative filtering. Most of these approaches concentrate on either user or url clustering, but not on employing both. Likewise, they concentrate more on the closing/profile creation methods themselves. In this methods employing both user as well as URL closing information for better completeness and relevancy of the recommended set. Several distance measures commonly employed in the literature and evaluate some others. This quantity does not depend on the distance metrics invoked by the clustering algorithm; it can be computed directly from the original data.

Accordingly a comprehensive strategy would consist of the following steps:

(1) Cluster the URLs as well as the users.

(2) For each user, maintain a list of the url clusters. When the user logs-in, first retrieve links from the url clusters he/she is close to.

(3) The last step is to search the user clusters to find what other urls "like minded" users have visited.

COMMON DISTANCE MEASURES

Distance measure will determine how the similarity of two elements is calculated and it will influence the shape of the clusters. The Euclidean distance (also called 2-norm distance) is given by:

$$d(x,y) = \sum_{i=0}^{p}(xi - yi)$$

x_n is a vector representing the n^{th} data point and y_j is the geometric centroid of the data points in d. Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group where K is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

USER WITH HIT COUNT BASED RESULT

In user wise URLs, group of users who have common interest in visiting similar URLs. In user wise URL, many users are using similar URLs. For this, users and URLs are in the form of matrix format and using hit count, the

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 11, November 2014

distance can be identified. Using Euclidean distance method, the minimum distance can be identified.

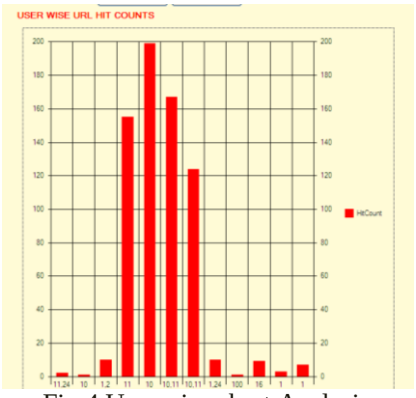

Fig.4 User wise chart Analysis

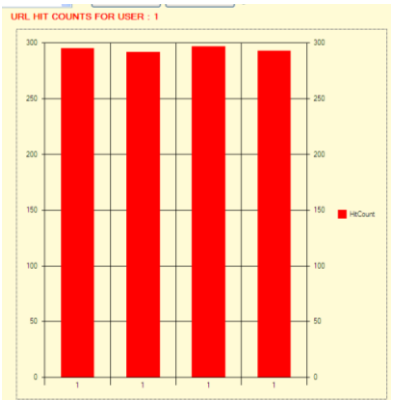

Fig.5 Selected User chart Analysis

KEYWORD WISE URL HIT COUNT

Keyword wise URL cluster is a group of keywords used for visiting similar URLs. Single keywords having multiple or different kinds of URLs and various keywords is also having multiple URLs. It can be clustered using k-means clustering using Euclidean distance method to find out the minimum distance and clustering themselves.

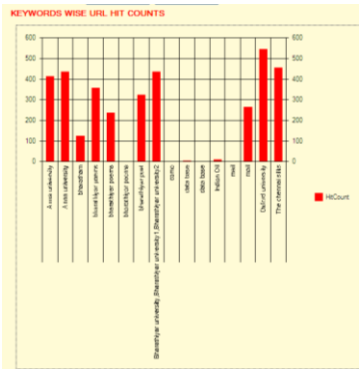

Fig.6 Keyword wise chart Analysis

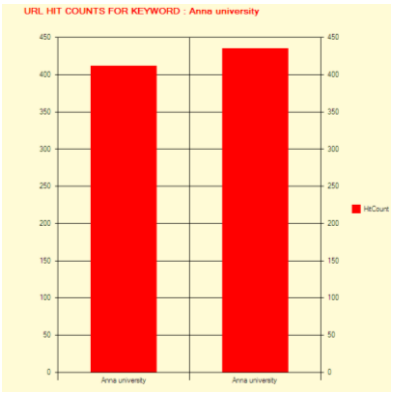

Fig.7 Selected Keyword chart Analysis

Based on the above clustering results, filtering can be done based on userid or keyword filtering and also filtering can be done both with the userid and keyword, the clustering results should be displayed in the Recommendation to the user which URLs and keywords are highly utilized by other users.

CONCLUSION

Currently, most search engines respond to a user's query using the bag-of-words model, which matches keywords in the query with web documents. The experimental results demonstrate the usefulness of bipartite graph. It provides latent semantically relevant results to the original information need. This model provides personalized recommendations for users. Similarity measure between two queries and abbreviation based query recommendation are also considered. This approach is effective and feasible in helping users optimally represent their information needs by keyword queries. The user-URL based clustering and keyword-URL based clustering is generated to recommend the URL to the new user. The bipartite graph and k-means clustering are complement in nature in that query suggestion and recommendation is made more effectively.

REFERENCES

[1] Andri Toma, Radu Constantinescu, "Recommendation system based on the clustering of frequent sets".AIKED'09 Proceedings of the 8th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases. Pp World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA 2009.

[2] Anna Huang "Similarity Measures for text document clustering". The University of Waikato, Hamilton, New Zealand 2008. http://web.ist.utl.pt/ acardoso/datasets".

[3] Badrul Sarubar, George Karypis, Joseph Konstan and John Riedl "Item-Based Collaborative Filtering Recommendation Algorithm". University of Minnesota, Minneapolis, MN 55455.

[4] Bamshad Mobasher, Robert Cooley, Jaideep Srivastava "Creating Adaptive Web Sites Through Usage-Based Clustering of URLs". KDEX '99 Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchangepp.07-11-1999.

[5] Beeferman, D., and Berger . A . et al." 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.

[6] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. STOC, pages 587–596, 2011.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

[7] Denis Xavier Charles, Max Chickering, Nikhil R. Devanur, Kamal Jain, and Manan Sanghi. Fast algorithms for finding matchings in lopsided bipartite graphs with applications to display ads. ACM Conference on Electronic Commerce, pages 121–128, 2010.

[8] Harold N. Gabow. Using euler partitions to edge color bipartite multigraphs. International Journal of Parallel Programming, 5(4):345–355, 1976.

[9] H.N. Gabow and O. Kariv. Algorithms for edge coloring bipartite graphs and multigraphs. SIAM J. Comput., 11(1):117–129, 1982.

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

[11] R. Cole and J.E. Hopcroft. On edge coloring bipartite graphs. SIAM J. Comput., 11(3):540–546, 1982.

[12] R. Cole, K. Ost, and S. Schirra. Edge-coloring bipartite multigraphs in O(E log D) time. Combinatorica, 21(1):5–12, 2001.

[13]Wen, J - R.., Nie .J - Y., Zhang . H - J., et al. "Clustering user queries of a search engine". In WWW'01 : Proceedings of the 10th international conference on world wide web pages 162-168, New York, NY, USA,2001. ACM Press.