Query Click and Text Similarity Graph for Query Suggestions

D. Sejal[1](✉), K.G. Shailesh[1], V. Tejaswi[2], Dinesh Anvekar[3], K.R. Venugopal[1], S.S. Iyengar[4], and L.M. Patnaik[5]

[1] Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore-1, India
sej_nim@yahoo.co.in
[2] National Institute of Technology, Surathkal, Karnataka, India
[3] Nitte Meenakshi Institute of Technology, Bangalore, India
[4] Florida International University, Miami, USA
[5] Indian Institute of Science, Bangalore, India

Abstract. Query suggestion is an important feature of the search engine with the explosive and diverse growth of web contents. Different kind of suggestions like query, image, movies, music and book etc. are used every day. Various types of data sources are used for the suggestions. If we model the data into various kinds of graphs then we can build a general method for any suggestions. In this paper, we have proposed a general method for query suggestion by combining two graphs: (1) query click graph which captures the relationship between queries frequently clicked on common URLs and (2) query text similarity graph which finds the similarity between two queries using Jaccard similarity. The proposed method provides literally as well as semantically relevant queries for users' need. Simulation results show that the proposed algorithm outperforms heat diffusion method by providing more number of relevant queries. It can be used for recommendation tasks like query, image, and product suggestion.

Keywords: Image suggestion · Query suggestion · Query relevance · Recommendation

1 Introduction

Exponential growth of information on the Web is a challenging task for the search engines to meet the need of the users. How organising and utilizing the Web information effectively and efficiently has become more and more critical. To get any information from web, the user issues queries, follows some links in web snippets, clicks on advertisement, and spends some time on pages. If the user is not satisfied with the information which he has received from the clicked page, he reformulates his query. In order to enhance the user experience, it is a common practice in a search engine to provide some types of query suggestion.

P. Perner (Ed.): MLDM 2015, LNAI 9166, pp. 328–341, 2015.
DOI: 10.1007/978-3-319-21024-7_22

On the Web, query suggestion is a technique to recommend a list of relevant queries to users' input by mining correlated queries from the previous knowledge. The simple way to suggest a query is spelling correction. In this paper, our interest is to suggest more elaborate forms of queries. For example, if the user submits a query *Java*, then user may be prompted to other queries like *Java for windows*, *Java 32 bit* or *Java download*, and also a related concept like *sun micro system*.

Basically, query suggestions on commercial sites like flipkart.com, Myntra.com etc., suggests products based on collaborative filtering [1,2]. Collaborative filtering is a method for automatic predictions about the interests of a user by collecting rating information or preferences from many users. Therefore collaborative filtering algorithm needs to build product-user matrix which determines relationship between user preferences to that product. The constraint with this approach is that in most of the cases, rating data are not available. However on the web, search log is always available to us. This is used to retrieve information about how people search information on the web and how they rephrase their query.

Motivation: There are several challenges to design suggestion framework on the web. First, most of the time users tend to submit short queries with only one to three terms. Short queries are more likely to be ambiguous. We observe that 9.82 % of web queries contain one term, 27.31 % of web queries contain two terms, and 26.99 % of web queries contain three terms. Second, in most of the cases, the users do not have enough knowledge about the topic they are searching for, and they are not able to clearly phrase the query words. Then, users have to rephrase the query words and rephrase their queries frequently. So, it is necessary to solve the above mentioned problems for query suggestion to satisfy users' information need and to increase search engine usability. Different types of data sources are used for suggestion on the web. In most cases, these data sources can be converted to graphs. We can solve many suggestion problems by designing a general graph suggestion approach.

Contribution: In this paper, we propose a generic method for the query suggestions on the web by using query relevance directed graph generated from a search log. We have constructed query click graph by capturing the relationship between queries frequently clicked on common URLs. Then, we have constructed query text similarity graph using Jaccard similarity between queries. Finally, we have combined query click graph and query text similarity graph to construct query relevance graph. This method has several advantages. It is a general method which can be used in many suggestion tasks such as query, image, and product suggestion. It can provide semantically relevant results to meet the original users' need and is scalable to a large dataset.

Organization: This paper is organized as follows: We have reviewed various query suggestion techniques under Sect. 2. Section 3 describes the Background Work. Section 4 presents Query Relevance model and algorithm. Section 5 discusses experiment results, query suggestion results comparison, and efficiency analysis. Finally, conclusions are presented in Sect. 6.

2 Related Work

It is a great challenge for any search engine to understand users' search intention. Various techniques have been studied extensively in the past decade to improve performance and quality of query suggestions. In this section, we have reviewed several papers related to query suggestion, query expansion and query term suggestion methods.

Udo et al. [3] have presented a systematic study on different query modification methods applied to query log collected on a local Web site. They have distinguished methods that derive query suggestions from previously submitted queries using logs or from actual documents. Mohamed et al. [4] have proposed a novel location aware recommender system (LARS) that uses location based rating to produce recommendations. LARS produces recommendations using taxonomy of three types of location based ratings within a single framework: (1) Spatial ratings for non-spatial items, (2) non-spatial ratings for spatial items, and (3) spatial ratings for spatial items.

Yang et al. [5] have proposed a new user friendly patent search paradigm, which can help users to find relevant patents more easily, and improve their search experience. They have developed three techniques, error correction, topic-based query suggestions, and query expansion, to make patent search more user-friendly. Brian et al. [6] have proposed a method for improving content-based audio similarity by learning from sample of collaborative filter data. First, a method for deriving item similarity is developed from a sample of collaborative filter data. Then, the sample similarity is used to train an optimal distance metric over audio descriptors. The resulting distance metric can then be applied to previously unseen data for which collaborative filter data is unavailable.

Gao et al. [7] have presented their work to cross-lingual query suggestion, i.e., for a input query in one language, it suggests similar or relevant queries in another language. Support Vector Machine (SVM) regression algorithm is used to learn the cross-lingual term similarity function. Aris et al. [8] have developed a framework which models the querying behaviour of users for query recommendation by a query flow graph. A sequence of queries submitted by a user can be seen as a path on this graph. It is based on the use of a Markov chain random walk over the query-flow graph. As user clicked information is not considered to create a graph, this approach results in lower accuracy.

Hossein et al. [9] have presented a new query recommendation technique based on identifying orthogonal queries in an ad-hoc query answer cache. This approach requires no training, is computationally efficient, and can be easily integrated into any search engine with an answer cache. Rodrygo et al. [10] have proposed a ranking approach for producing effective query suggestion. A structured representation of candidate suggestions is generated from related query with common clicks and common sessions. This representation helps to overcome data sparsity for long-tail queries.

Term Suggestion is a method in which, as the user types in queries letter by letter, suggest the terms that are topically coherent with the query. In [11,12], authors have recommended queries based on terms of the queries. A user can modify a part of the query by adding terms after the query, deleting terms within the query or modifying terms to new terms.

In [13–18] authors have used snippet information of clicked URLs or search results returned from a query for query recommendation in different ways. These methods are not general and the extensibility is very low.

Adam [19] has described a clustering-by-directions (CBD) algorithm for interactive query expansion. When a user executes a query, the algorithm shows potential directions in which the search can be continued. The CBD algorithm first selects different directions, and afterward, it determines how the user can move in each direction. Pawan et al. [20] have provided theoretical analysis of a parametric query vector, which is assumed to represent the information needs of the user. A global query expansion model is derived based on the lexical association between terms.

Huanhuan et al. [21] have proposed a novel contexts aware query suggestion approach, which considers immediately preceding queries in query log as context in query suggestion. Qiaozhu et al. [22] have proposed a novel query suggestion algorithm ranking queries with the hitting time on large scale bipartite graph. Every query is connected with a number of URLs, on which the users have clicked when submitting query to a search engine. The weights on the edges present the number of times, the users used this query to access this URL.

Hao et al. [23] have presented a method to suggest both semantically relevant and diverse queries to web users. The proposed approach is based on Markov random walk and hitting time analysis on query-URL bipartite graph. Guo et al. [24] have proposed a method to recommend queries in a structured way for satisfying both search and exploratory interests of users. A Query-URL-Tag tripartite graph is obtained by connecting query logs to social annotation data through URLs. A random walk on the graph and hitting time is employed to rank possible recommendations with respect to the given query. Then, the top recommendations are grouped into clusters with label and social tags. This approach satisfies users' interest and significantly enhances users' click behaviour on recommendations.

Yang et al. [25] have introduced the concept of diversifying the content of the search result from suggested queries while keeping the suggestion relevant. First, query suggestion candidate is generated by applying random walk with restart (RWR) model to query-click log. Zhu et al. [26] have proposed query expansion method based on query log mining. This method extracts correlations among queries by analysing the common documents selected by a user. Then, expansion terms are selected by analysing relation between queries and documents from the past queries. Nick et al. [27] have applied a Markov random walk model on click graph to produce a probabilistic ranking of documents for a given query from a large click log. Click graph is a bipartite graph, with two types of nodes: queries and documents. An edge connects a query and document if the user has clicked for that query-document pair.

In [21–27], authors have used query-URLs clicked information to construct query-URL bipartite graph for query suggestion in various ways. There are two major problems with query-URL based query suggestion: (1) the number of common clicks on URLs for different queries is limited. (2) Although two queries may lead to the same URLs clicking, they may still be irrelevant because they may point to totally different contents of the web document. These methods ignore the rich information between two queries relevance. Our approach in this work is to consider relevance information between two queries to enhance user suggestion.

3 Background

Hao et al. [28] have proposed query suggestions based on heat diffusion method on directed query-URL bipartite graph. An undirected bipartite graph is considered where, B_{ql} = (V_{ql}, E_{ql}); V_{ql} = (Q ∪ L), Q = $\{q_1, q_2, \ldots q_n\}$ and L = $\{l_1, l_2, \ldots l_p\}$. $E_{ql} = \{(q_i, l_j)$, there is an edge from q_i to $l_j\}$ is the set of all edges. The edge (q_i, l_j) exists if and only if a user u_i clicked a URL l_k after issuing a query q_j. The weight on the edges is calculated by number of times a query is clicked on a URL.

This undirected bipartite graph cannot accurately interpret the relationship between queries and URLs. Hence, they are converted into directed query-URL bipartite graph. In this converted graph, every undirected edge 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. The weight on a directed URL-query edge is normalized by the number of times that the URL is clicked. Query suggestion algorithm is applied on the converted graph.

A converted bipartite graph G = (V ∪ U , E) consists of query set V and URL set U. For 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. Then, heat diffusion process is applied on the sub-graph. Top-k queries are suggested with largest heat value. This method outperforms SimRank, Forward random walk and backward random walk.

Heasoo et al. [29] has developed online query grouping method by generating graph which combines the relationship between queries frequently issued together by users, and the relationship between queries frequently leading to clicks on similar URLs. Related query clusters are generated by Monte Carlo random walk simulation method for a given query.

4 Query Relevance Model and Algorithm

4.1 Problem Definition

Given a user input query q and search log of search engine, we convert search log into a graph, where nodes represent queries and edges represent relationship between queries. The objective is to provide semantically relevant query suggestions to meet the original users' need.

4.2 Assumptions

It is assumed that the user is online and enters input query with less than six terms.

4.3 Query Relevance Model

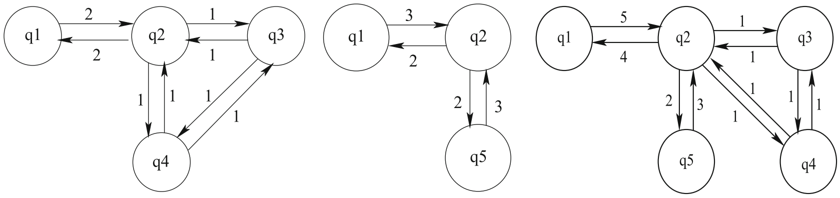
The query relevance model captures the relevant queries from user search log. This model constructs query relevance graph by combining query click graph which captures the relationship between (i) queries frequently clicked on common URLs and (ii) query text similarity graph using Jaccard similarity between queries.

Query Click Graph. Relevant queries from the search logs can be obtained by considering those queries that stimulate the users to click on the same set of URLs. For example, the queries *solar system* and *planet* are not textually similar, but they are relevant. This information can be achieved by analyzing common clicked URLs on queries in the search log.

Consider a URL-Query undirected bipartite graph, $BG_{qu} = (V_{qu}, E_{qu})$, where V_{qu} = Q ∪ U, Q=$\{q_1,q_2,...q_m\}$ and U=$\{u_1,u_2,...u_n\}$. E_{qu} is the set of all edges. The edge (q_i,u_j) exists if and only if user has clicked a URL u_j after issuing query q_i. Often, the user issues query and by mistake clicks on some URL, which has no relation. In order to reduce noise and outliers, those edges which have only one click between query and URL are removed.

From BG_{qu} Query Click directed Graph, QC=(V_q,E) is constructed, where V_q are queries and E is a directed edge from q_i to q_j which exists if and only if there is atleast one common URL u_k, that both q_i and q_j link in BG_{qu}.

The weight of edge (q_i,q_j) in QC, $w_c(q_i,q_j)$ is calculated by counting occurrence of the pair (q_i,q_j) in URL-Query group. A URL-Query group $UQ_i = \{Q_i \in$ Q$\}$ is a set of queries Q_i generated by unique URL u_i clicked by user for different queries. Figure 1a shows Query click graph.



(a) Query Click Graph

(b) Query Text Similarity Graph

(c) Query Relevance Graph

Fig. 1. (a) Query click graph; (b) Query text similarity graph; (c) Query relevance graph

Query Text Similarity Graph. It is not necessary for the user to always click the same URL for different queries. In such a case, we may not get two or more queries having common URL, but queries may be relevant since they share common words. For example, queries *cloud computing* and *cloud computing books* shares common words. To get these relevant queries, query text similarity graph is constructed.

For this graph, query text similarity by Jaccard coefficient J_c is calculated. Jaccard coefficient is defined as the fraction of common words between two queries as given in Eq. 1.

$$J_c(q_i, q_j) = \frac{words(q_i) \cap words(q_j)}{words(q_i) \cup words(q_j)} \tag{1}$$

Query Text Similarity directed graph is defined as QG_{ts}=(V_q,E), where V_q is distinct queries in search log and E an edge between q_i to q_j exists if $J_c(q_i,q_j)>0.6$. The weight of edge (q_i,q_j) in QG_{ts}, $w_{ts}(q_i,q_j)$ is calculated by counting the occurrence of q_j in the search log with respect to q_i. Figure 1b shows the example of Query Text Similarity Graph.

In Fig. 1b, q_1, q_2 and q_5 are distinct queries in search log. The Jaccard value of q_1 and q_2, is assumed to be greater than 0.6, therefore q_1 and q_2 are included as nodes in the graph. The weight of q_1 to q_2 is 3 as occurrence of q_2 is 3 with respect to q_1 in the search log and weight of q_2 to q_1 is 2 as occurrence of q_1 is 2 with respect to q_2 in the search log. Similarly, Jaccard value of q_2 and q_5 is greater than 0.6. The weight between q_2 and q_5 is calculated using the same procedure as q_1 and q_2.

Query Relevance Graph. The query click graph QC and query text similarity graph QG_{ts} captures two important properties of relevant queries. In order to utilize both the properties, the query click graph and query text similarity graph are combined into a single graph, Query Relevance Graph QRG = (V_q,E), where V_q is set of queries either from QC or QG_{ts} and E an edge between q_i to q_j exists either from QC or QG_{ts}. The weight of edge (q_i,q_j) in QRG is calculated as follows : $w_r(q_i,q_j) = w_c(q_i,q_j) + w_{ts}(q_i,q_j)$. Figure 1c represents query relevance graph from the combined graphs of Fig. 1a and b.

The query relevance graph is normalized by Eq. 2.

$$Normalizedw_r(q_i, q_j) = \frac{w_r(q_i, q_j)}{\sum_{n=1}^{k} w_r(q_i, q_n)} \tag{2}$$

4.4 Query Suggestion Algorithm

The Query Suggestion algorithm is shown as Algorithm 1. Given a search log of search engine, query relevance graph is constructed. Then, given a user input query, depth first search algorithm is used to suggests queries to meet the original users' need.

Algorithm 1. Query Suggestion Algorithm

```
Input : Input query $q$
Output: Top-5 Suggested Queries
begin
    Construct query relevance graph $G=(V,E)$ using the method shown in the
    previous section. The directed edges $E$ are weighted by normalization.
    Given a query $q$, apply depth first search method on query relevance graph's
    nodes $V$.
    The first Top-5 results are the suggested queries.
```

5 Experiments

5.1 Data Collection

Publicly available America On-line(AOL) search engine data [30] is used to construct query suggestion graph. Nearly 3813395 click through information with 1293620 unique queries and 400694 unique URLs have been considered.

The Web user activities can be obtained by click through data records. The users' interest and latent semantic relationship between users and queries as well as queries and clicked URLs can be retrieved. Each line of click through data contains information about anonymous user ID number, the query issued by the user, the time at which the query was submitted for search, the rank of the item on which they clicked and the domain portion of the URL in the clicked result. Each line in the data represents one of the two types of events: a query that was not followed by the user clicking on a result item and a click through an item in the result list returned from a query.

In this paper, we have used the relationship of queries and URLs for construction of query click graph and query to query relationship for construction of query text similarity graph. We have ignored user ID, rank and time information of click through data.

5.2 Data Cleaning

Click through data is the raw data recorded by search engine that contains a lot of noise which affects the efficiency of query suggestion algorithm. Hence, data is filtered by keeping well formatted, frequent, English queries i.e., queries which only contain characters *a, b .. z* and space. After cleaning, data is reduced by 58.45 %. Nearly 490866 distinct queries and 334224 distinct URLs have been used in this experiments.

As discussed previously, most of the time, user tend to submit short queries with only one, two or three terms and therefore filtered data is obtained by keeping queries less than six terms. Further, data is reduced by 6.20 %, resulting in 448299 distinct queries and 318644 distinct URLs. Query suggestion results are generated by including all queries and by including only those queries which have less than six terms. We observe that both the results are similar.

5.3 Varying of Parameter-Jaccard Coefficient

As discussed in Sect. 4, Jaccard coefficient(J_c) is defined as the fraction of common words between two queries. Query suggestion results are generated by varying the value of J_c greater than 0.5 and 0.6. Finally, it is observed from the results, that when the optimal value for Jaccard coefficient is greater than 0.6, it yields more related queries.

When the query text similarity graph is constructed with J_c value greater than 0.5, queries which are only literally similar are obtained from the generated query relevance graph. For example, for the query *java*, queries *new one campaign commercial* and *one campaign commercial* are also obtained which are not relevant for the given input query. For the query *bank of america*, queries *bank of america banking centers*, *bank of america online banking* and *bank of america banking on line* are obtained which are only literally similar.

When the value of J_c is greater than 0.6, it results in those queries which are not only literally similar but also latent semantically relevant queries. For input query *java*, queries *java sun systems* and *sun java* are obtained which are latent semantically relevant queries.

If the value of J_c is equal to 0.5, then irrelevant queries are being selected. For example, consider two queries *California state university* and *California state polls* in which the value of J_c is 0.5, and both the queries are not relevant. Even when the value of J_c is less than 0.5, words matching between the two queries reduces, and hence relevant queries are not obtained.

5.4 Query Suggestion Results

We have displayed the Top-5 suggestion results of Heat Diffusion algorithm and our algorithm in Table 1. For Heat Diffusion Algorithm the numeric values shows the heat value for that query. For our algorithm, numeric value shows the normalized value of query in the query relevance graph.

5.5 Performance Analysis

From the results shown in Table 1, it is observed that our query relevance model is suggesting literally similar queries as well as latent semantically relevant queries. As discussed earlier, most of the time, the user tends to submit short queries with only one, two or three terms. Therefore, 60 test queries with one, two or three terms have been considered and different topics for input queries, such as Health, Shopping, Computer, Art have been covered in our experiments.

For example, given an input query *java*, the algorithm suggests *java download*, and *download java script*, which are literally similar queries. While a *sun microsystems* query suggests the company name of the java platform. This query is latent semantic to the input query *java*. Similarly for the input query *free music*, the query suggestion results are *free music downloads*, *music downloads*, *shareware music downloads*, *broadway midi files* and *midi files*. Midi file is the Musical Instrument Digital Interface protocol for music. Here it is observed that

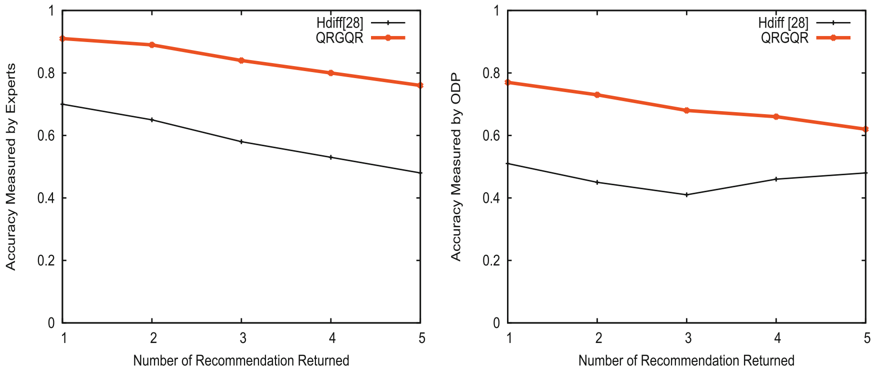
Table 1. Query suggestion results comparison

Sr.No	Query	Heat diffusion algorithm	Our algorithm
1	Java	Word whomp game 0.9044	Download java 0.012
			Download java script 0.087
			Java download 0.32
			Sun microsystems 0.0455
			Sun microsystems colorado 0.0625
2	Fireworks	Phantom fireworks 0.8651	Firecrackers 0.25
			Phantom fireworks 0.25
			How to make a firework 0.25
			Phantomfireworks 0.25
3	Free music	Sad songs 0.9174	Free music downloads 0.2523
		Bored 0.9075	Music downloads 0.0588
		Humorous pictures 0.9051	Shareware music downloads 0.007
		Free music downloads 0.8962	Broadway midi files 0.0159
		Funny quotes 0.8928	Midi files 0.5
4	Wedding	Wedding channel 0.8653	Bridal shows nj 0.0714
			Wedding channel 0.0119
			Wedding dresses 0.125
			Lavender wedding dresses 0.0287
			Tea length wedding dress 1.0

the resulting queries are literally similar and latent semantically to the given input query.

We have compared the performance of our algorithm against Heat Diffusion algorithm (Hdiff) [28]. In the heat diffusion algorithm, the query-URL bi-partite graph is constructed from the search log. For a given input query, a sub graph is constructed by using depth-first search on bi-partite graph and then the diffusion process is applied on the sub graph to the ranked result. The top-5 queries based on heat value are used as suggested results. The graph construction method is the major difference between our proposed model and heat diffusion model. In heat diffusion model, query-URL bi-partite graph is used. While in proposed model URL-query relation and query text similarity relation is used to construct query-query graph i.e., query relevance graph.

To evaluate the quality of semantic relation is not easy in query suggestion as the queries taken as input is generated by users, and there are no linguistic resources available. In this paper, we have evaluated quality of semantic relation manually by three human experts and automatic evaluation based on Open Directory Project (ODP) database. We have adopted the method used in [31] to evaluate the quality of query recommendation results.



(a) Experts

(b) ODP

Fig. 2. Accuracy comparison

In manual evaluation, three research students are asked to rate the query suggestion results. We have asked them to evaluate relevance between testing queries and suggested results in the range of 0 to 1, in which 0 means totally irrelevant and 1 means totally relevant. The average value of rating results is shown in Fig. 2a. It is observed that the accuracy of our algorithm increases by 25.2 % in comparison with Hdiff.

In automatic evaluation, ODP database is used; also known as *dmoz* human edited directories of the web. When a user types a query in ODP, besides site matches, we can also find categories matches in the form of paths between directories and these categories are ordered by relevance. For example, the query *solar system* would provide the category *Science : Astronomy : Solar System*, while one of the results for *planets* would be *Science: Astronomy : Solar System : Planets*. Here, ":" is used to separate different categories. Hence, to measure the relation between two queries, we use a notion of similarity between the corresponding categories as provided by ODP and measure the similarity between two categories D and D_1 as the length of their longest common prefix CP(D,D_1) divided by the length of the longest path between D and D_1. More precisely, denoting the length of a path with $|D|$, this similarity is defined as $sim(D,D_1) = |\text{CP}(D,D_1)|/max\{|D|, |D_1|\}$. The similarity between the two queries above mentioned is 3/4. Since they share the path *Science : Astronomy : Solar System* and the longest path is made of the four directories. We have evaluated the similarity between two queries by measuring the similarity between the most similar categories of the two queries among the top five answers provided by ODP.

Accuracy comparison measured by ODP is shown in Fig. 2b, from which it is observed that the accuracy of our algorithm increases by 23.2 % in comparison with Hdiff. From the above two evaluation process it can be concluded that our proposed query suggestions algorithm is efficient. The major difference between our proposed model and heat diffusion model is the method of graph construction. The query relevance graph can identify richer query relevance information as there are more available paths to follow in the graph. Hence our proposed algorithm outperforms the heat diffusion algorithm.

5.6 Efficiency Analysis

Experiments have been conducted on 8 GB memory and intel Xeon(R) CPU E31220 @ 3.10 GHz Quad Core processor workstation. Dataset used for both Hdiff and our method are same. Though our algorithm gives more number of relevant queries than heat diffusion method, the retrieval time is same. The computation time for the query suggestion of both Hdiff and our method is around 0.01 seconds.

6 Conclusions

In this paper, we have proposed a general method for query suggestions by combining query click graph which captures the relationship between queries frequently clicked on common URLs and query text similarity graph which finds the similarity between two queries using Jaccard similarity. The proposed method provides literally as well as semantically relevant queries for users' need. Simulations performed on America On-line (AOL) search data and as compared with Heat diffusion method, have shown that our method outperforms Heat diffusion method [28] by providing more relevant queries for a given input query with almost the same computation time as Heat diffusion method. It can be used for many recommendation tasks like query, image, and product suggestions. The query click graph has relationship between query and URLs, and hence the our algorithm can also be used to suggest URLs.

References

1. Das, A., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: WWW 2007: The Proceedings of 16^{th} International Conference on World Wide Web, pp. 271–280 (2007)
2. Ma, H., King, I., Lyu, M.R.: Effective missing data prediction for collaborative filtering. In: SIGIR 2007: The Proceedings of 30^{th} International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 39–46 (2007)
3. Kruschwitz, U., Lungley, D., Albakour, M.-D., Song, D.: Deriving query suggestions for site search. J. Am. Soc. Inf. Sci. Technol. **64**(10), 1975–1994 (2013)
4. Sarwat, M., Levandoski, J.J., Eldawy, A.: LARS*: an efficient and scalable location-aware recommender system. IEEE Trans. Knowl. Data Eng. **26**(6), 1384–1399 (2014)
5. Yang Cao, J., Fan, J., Li, G.: A user-friendly patent search paradigm. IEEE Trans. Knowl. Data Eng. **25**(6), 1439–1443 (2013)
6. McFee, B., Barrington, L., Lanckriet, G.: Learning content similarity for music recommendation. IEEE Trans. Audio Speech Lang. Process. **20**(8), 2207–2218 (2012)

7. Gao, W., Niu, C., Nie, J.-Y., Zhou, M., Hu, J., Wong, K.-F., Hon, H.-W.: Cross-lingual query suggestion using query logs of different languages. In: SIGIR 2007: The Proceedings of 30^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 463–470 (2007)
8. Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A.: An optimization framework for query recommendation. In: WSDM 2010: The Proceedings of 3^{rd} ACM International Conference on Web search and Data Mining, pp. 161–170 (2010)
9. Vahabi, H., Ackerman, M., Baeza-Yates, D.L.R., Lopez-Ortiz, A.: Orthogonal query recommendation. In: RecSys 2013: The Proceedings of the 7^{th} ACM Conference on Recommender System, pp. 33–40 (2013)
10. Santos, R.L.T., Macdonald, C., Ounis, I.: Learning to rank query suggestions for adhoc and diversity search. ACM J. Inf. Ret. **16**(4), 429–451 (2013)
11. Song, Y., Zhou, D., He, L.: Query suggestion by constructing term-transition. In: WSDM 2012: The Proceedings of 5^{th} ACM International Conference on Web Search and Data Mining, pp. 353–362 (2012)
12. Fan, J., Wu, H., Li, G., Zhou, L.: Suggesting topic based query terms as you type. In: The Proceedings of 12^{th} International Asia-Pacific Web Conference, pp. 61–67 (2010)
13. Liu, Y., Miao, J., Zhang, M., Ma, S., Liyun, R.: How do users describe their information need : query recommendation based on snippet click model. Int. J. Expert Syst. Appl. **38**(11), 13874–13856 (2011)
14. Sharma, S., Mangla, N.: Obtaining personalized and accurate query suggestion by using agglomerative clustering algorithm and P-QC method. Int. J. Eng. Res. Technol. **1**(5), 1–8 (2012)
15. Narawit, W., Chantamune, S., Boonbrahm, S.: Interactive query suggestion in Thai library automation system. In: The Proceedings of 10^{th} International IEEE Conference on Computer Science and Software Engineering, pp. 76–81 (2013)
16. Leung, K.W.-T., Ng, W., Lee, D.L.: Personalized concept-based clustering of search engine queries. IEEE Trans. Knowl. Data Eng. **20**(11), 1505–1518 (2008)
17. Chen, Y., Zhang, Y.-Q.: A personalized query suggestion agent based on query-concept bipartite graphs and concept relation trees. Int. J. Adv. Intell. Paradigms **1**(4), 398–417 (2009)
18. Kim, Y., Seo, J., Croft, W.B., Smith, D.A.: Automatic suggestion of phrasal-concept queries for literature search. Int. J. Inf. Process. Manage. **50**(4), 568–583 (2014)
19. Kaczmarek, A.L.: Interactive query expansion with the use of clustering-by-directions algorithm. IEEE Trans. Ind. Electron. **58**(8), 3168–3173 (2011)
20. Goyal, P., Behera, L., McGinnity, T.M.: Query representation through lexical association for information retrieval. IEEE Trans. Knowl. Data Eng. **24**(12), 2260–2273 (2012)
21. Cao, H., Jiang, D., Pei, J., He, Q., Lian, Z., Chen, E., Li, H.: Context-aware query suggestion by mining click-through and session data. In: KDD 2008: The Proceedings of 14^{th} International ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 875–883 (2008)
22. Mei, Q., Zhou, D., Church, K.: Query suggestion using hitting time. In: CIKM 2008: The Proceedings of 17^{th} ACM Conference on Information and Knowledge Management, pp. 469–477 (2008)
23. Ma, H., Lyu, M.R., King, I.: Diversifying query suggestion results. In: AAAI 2010: The Proceedings of 24^{th} AAAI International Conference on Artificial Intelligence, pp. 1399–1404 (2010)

24. Guo, J., Cheng, X., Xu, G., Shen, H.-W.: A structured approach to query recommendation with social annotation data. In: CKIM 2010: The Proceedings of 19^{th} ACM International Conference on Information and Knowledge Management, pp. 619–628 (2010)
25. Song, Y., Zhou, D., He, L.: Post ranking query suggestion by diversifying search results. In: SIGIR 2011: The Proceedings of 34^{th} International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 815–824 (2011)
26. Kunpeng, Z., Xiaolong, W., Yuanchao, L.: A new query expansion method based on query logs mining. Int. J. Asian Lang. Process. **19**(1), 1–12 (2009)
27. Craswell, N., Szummer, M.: Random walks on the click graph. In: SIGIR 2007: The Proceedings of 30^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 239–246 (2007)
28. Ma, H., King, I., Lyu, M.R.T.: Mining Web graphs for recommendations. IEEE Trans. Knowl. Data Eng. **24**(6), 1051–1064 (2012)
29. Hwang, H., Lauw, H.W., Getoor, L., Ntoulas, A.: Organizing user search histories. IEEE Trans. Knowl. Data Eng. **24**(5), 912–925 (2012)
30. Pass, G., Chowdhury, A., Torgenson, C.: A picture of search. In: The Proceedings of 1^{th} International Conference on Scalable Information Systems, June 2006
31. Baeza-Yates, R., Tiberi, A.: Extracting semantic relations from query logs. In: KDD 2007: The Proceedings of 13^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 76–85 (2007)