International Journal of Research in Engineering and Applied Sciences (IJREAS)
Available online at http://euroasiapub.org/journals.php
Vol. 6 Issue 11, November - 2016, pp. 1~14
ISSN(O): 2249-3905, ISSN(P) : 2349-6525 | Impact Factor: 6.573 | Thomson Reuters ID: L-5236-2015

Inferring User Search Goals with Feedback Sessions Using Clicked Documents for Related Search Recommendation

Y.Raju[1],
Associate Professor,
Geethanjali College of Engineering and Technology,
IT Department, Hyderabad, India
Dr D. Suresh Babu[2],
Professor,
Department of Computer Science,
Head, Kakatiya Government College, Kakatiya University, India
Dr.K.Anuradha[3]
Professor,
Head, GRIET, CSE Department , Hyderabad, India

Abstract

Query suggestion plays an important role in improving the usability of search engines. Although some recently proposed methods can make meaningful query suggestions by mining query patterns from search logs, none of them are context-aware they do not take into account the immediately preceding queries as context in query suggestion. Hence, the input queries are normally short and ambiguous. Query recommendation is a method to recommend web queries that are related to the user initial query which helps them to locate their required information more precisely. It also helps the search engine to return appropriate answers and meet their needs. Usually users have ambiguous keywords in their mind to represent their information need. Hence, it is not a good idea to generate relation between user query keywords for recommendations. In this paper, we have presented Related Search Recommendation (RSR) framework, which discovers keywords which are present in snippets clicked and unclicked documents in feedback session.

Pseudo documents are generated from feedback sessions which reflect what users wish to retrieve.

***Keywords*:** Pseudo Document, Recommendation, Semantic Similarity, User Feedback Session.

1 INTRODUCTION

Web data keeps expanding and is available in various data forms because of rapid growth of online advertising, publishing e-commerce and entertainment. Although web search technology provides efficient and effective information access to users, it is still a difficult task to search useful knowledge about users needs from their search queries. Users may seek disrete information on a distinct subject, hence may check out various query terms. Users may not have sufficient knowledge on a topic. Therefore adequate terms are not known to retrieve the required information. In Kato et al. Query recommendations are frequently used when (1) a initial query is an exceptional query (2) Single term query is used as input query 3) explicit queries are suggested (4) Suggestions are provided based on modification of input query (5) Various URLs has been clicked by users on the resulting search page.

Silverstein et al.derived that users' input query's average length is 2.35 terms (AltaVista search engines query log). This shows that most of the user queries are short. A short query cannot describe information need of user search and sometimes ambiguous in meaning expression.

Some authors have focused on query suggestions by considering users' previous query and click behavior. There are two major issues with query-URLs recommendations: (i) the common clicks on URLs are limited for various queries (ii) though users may click the same URLs for two different queries, they may be irrelevant as that web documents may have different contents. It is necessary to generate useful suggestions by solving these problems. It is required to discover users' information needs to organize queries with precise meaning. Users' search log provides information needs from users' click behavior. If a certain retrieved result is clicked by the user, we cannot conclude that the clicked result is completely relevant to the user query since he has not seen the full document. But the brief description of the document i.e., snippet is shown to the user and is read by the user if he decides to click that document. It can be considered that snippet reflects user's information need.

1.1 Query Logs

Query logs are auto saved data of user activities on search engines servers. It consists of user identity attributes as Session ID, IP address, Time- Stamp, Query string, Number of results on result page and results page number. A relevance click through data also saved consisting of clicked URL, associated query, posisition on results page and time- stamp attributes in the logs.

The application used in client side can be modified to handle the query and click through usage logs in the user side computer. It can be very important source for user personalization. A number of other works are investigated in to make use of past navigated log data and queries for mining click through logs data to construct efficient user personalization information. There has been some work related are described below.

a) Language Modeling Approaches
b) Collaborative Filtering Based Approaches
c) Machine Learning Based Approaches

1.2 Click through Data

Under this we can find out the user clicks based on the old line model .In the evaluation process for identifying the problem to the best of my knowledge and measures are certain. For obtaining click through data, we have developed a web application with click event handler to store the click through event link data using five different users click through activities. We have taken 5 different queries as, "*Web Personalization, Deep Web data Extraction, Data Mining for Information Service, Web Information Retrieval and Extracting Quality Data*" and integrated with Yahoo search engine for evaluation. Click-through rate (CTR) is the ratio of users who click on a specific link to the number of total users who view a page, email, or advertisement. It is commonly used to measure the success of an online advertising campaign for a particular website as well as the effectiveness of email campaigns.

Construction: The click-through rate is the number of times a click is made on the advertisement divided by the total impressions (the number of times an advertisement was served)

Click-through rate = Click-through (#) / Impressions (#)

1.3 Online advertising CTR and Email

The click-through rate of an advertisement is defined as the number of clicks on an ad divided by the number of times the ad is shown (impressions), expressed as a percentage. For example, if a banner ad is delivered 100 times (100 impressions) and receives one click, then the click-through rate for the advertisement would be 1%.

$$\mathbf{CTR} = \frac{\boldsymbol{Clicks}}{\boldsymbol{Impressions}} \times \mathbf{100}$$

An email click-through rate is defined as the number of recipients who click one or more links in an email and landed on the sender's website, blog, or other desired destination. More simply, email click-through rates represent the number of clicks that your email generated. Email click-through rate is expressed as a percentage, and calculated by dividing the number of click troughs by the number of tracked message opens. Most email marketers use this metrics along with open rate, bounce rate and other metrics, to understand the effectiveness and success of their email campaign.[1] In general there is no ideal click-through rate. This metric can vary based on the type of email sent, how frequently emails are sent, how the list of recipients is segmented, how relevant the content of the email is to the audience, and many other factors. Even time of

day can affect click-through rate. Sunday appears to generate considerably higher click-through rates on average when compared to the rest of the week.

2 Measuring Performance

We first evaluate our proposed framework offline before evaluating other approaches. Then, we evaluate our approach with other approaches and compare with different measuring metrics as Precision, Recall and Fallout Rate.

Precision (P):In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the query Precision takes all retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. This measure is called *precision at n* or *P @ n*.

$$\textbf{Precision (P) \%} = \frac{|\text{Number of appropriate and relevent results}|}{|Number\ of\ appropriate result|} \times 100$$

Recall (R) : Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved.

$$\text{Recall (R) \%} = \frac{|\text{Number of appropriate and relevent results}|}{|Number of\ relevant\ result|} \times 100$$

Fallout Percentage Rate (F):It measures error rate in the information retrieval. It is calculated as the ratio of Number of appropriate and nonrelevant results against the Number of nonrelevant results. It represented as

$$\textbf{Fallout (F) \%} = \frac{|\text{Number of appropriate and nonrelevent results}|}{|Number\ of\ nonrelevant result|} \times 100$$

3 Measuring Similarity between two Words

Miao et al., have developed query expansion method based on Rocchio's model. In this model, proximity information is modelled by proposed Proximity based Term Frequency *ptf* in the pseudo relevant documents. Expansion terms and their proximity relation with query terms are modelled by *ptf*. This model achieves better performance over position relevance model and classic Rocchio's model. Hamai et al., have discussed a transformation function to measure semantic similarity between two given words. This approach uses page counts of documents title to measure similarity. This approach outperforms similarity measures defined over snippets. Bollegala et al.,have presented an approach to calculate semantic similarity between words. Text snippets are used to obtain Lexico-syntactic patterns from a web search engine. Support vector machine is used to integrate page count based similarity score and

lexico-syntactic patterns to generate semantic similarity measure. This method performs better than Information content measures and Edge counting WordNet based methods. Li et al. have presented an approach to calculate semantic similarity between terms and multiword statement. A large web corpus is used to form an *is A* semantic network to provide contexts for the terms. The meaning of input terms is formulated by *K*-Medoids clustering algorithm and similarity is computed with *max-max* similarity function. This algorithm outperforms multi-word expressions pairs and pearson correlation coefficient on word pairs. Bollegala et al., have developed a relational model to calculate the semantic similarity between two words. Snippets of web pages are used to obtain Lexical patterns. Semantically related patterns are identified by extracted clusters from sequential pattern clustering algorithm. Mahalanobis distance is used to calculate semantic similarity between two words.

4 Query Recommendation Techniques

Song et al.,have designed query suggestion method by using users' feedbacks in the query logs. Query-URL bipartite graph is constructed for click and un-click information. Random Walk with Restart (RWR) technique is applied on both the graphs. This framework gives better performance than pseudo-relevance feedback models and random walk models. Kharitonov et al.,have focused on contextualization framework for diversifying query suggestion. This framework utilizes the user's history query, the previously clicked and skipped documents and examines query suggestions. Mean Reciprocal Rank (MRR) is used as performance evaluation metric. This framework is compared with non-diversified ranking with previous query, ranking with the previous query as a context and clicks and skips as context.Ozetem et al.,have developed an approach to learn the probability with machine learning that a user may find a relevant follow up query after executing the input query. To measure relevance of follow-up query probabilistic utility function is used which relies on the query co-occurrence. This approach shows significance improvement over Mutual Information (MI) method. Broccolo et al.,Zhang et al.,have developed an approach for query suggestion based on query search. This approach constructs an ordered set of search terms drawn from documents to create candidate query suggestions. It builds query suggestions separately for each potentially relevant document. This approach provides more relevant query suggestions for short queries as well as long queries. Gomex et al.,Liu et al.,have proposed a snippet click model for query recommendation. This model determines information need of users from search logs. The following table shows existed concepts and the comparison of closely related works with our proposed approach.

Author	Concept	Advantages	Disadvantages
Li et al.,	Suggest topically related web queries using hidden topic model	Provides better query suggestions than URL model and Comparable Results with term feature model	Training dataset need to be generated to find topic of web queries from external data source.
Zhang et al.,	Provide improved query suggestion by query search	Provides more relevant query suggestion for short queries as well as long queries compared to suggestion by query search	User feedback is not considered
Miao et al.,	Query expansion based on proximity based Rocchio's model	This model achieves better performance over position relevance model	The exact relationship between the window size factor and information of collection is not fixed
Lu et al.,	Inferring User Search goals with Feedback Sessions	User search goals can be utilized in query recommendations	Finds Personalized Search goals.
Liu et al.,	Provide query recommendation based on snippet click model	Provides more effective recommendations than Biadu and Sogou search engines	Only click information is used to create model
Our work	Recommending related search with user feedback session and semantic similarity between words	Provides Semantically related search to inputs and this approach can be extended to generate multiple related words	

Table 1: shows the comparison of closely related works with our proposed approach

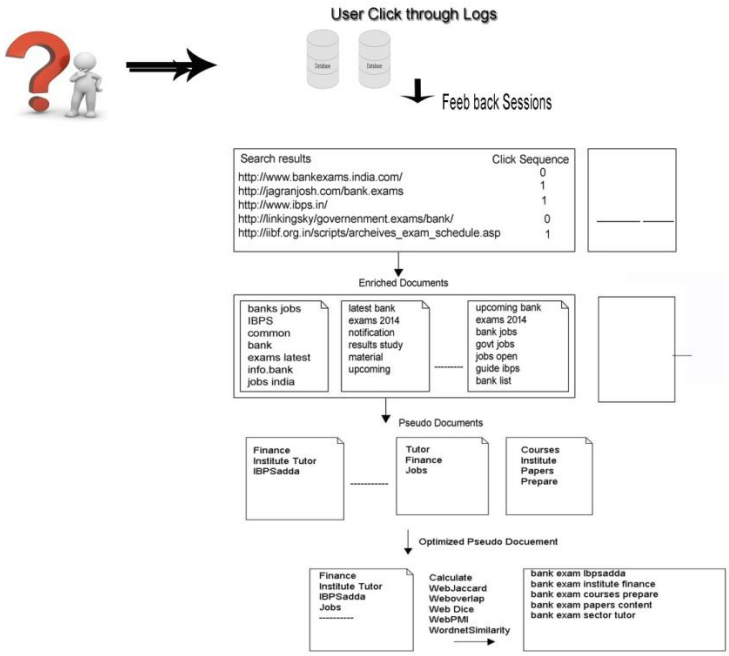

Figure 1.An example of feedback session for query *bank exam* in rectangular box

Generate Enriched Documents from Feedback Sessions: It is not suitable to use feedback sessions directly to obtain meaningful information for suggestions as it may differ for different search history and queries. Usually, users have ambiguous keywords in their minds to represent their information need. keywords for recommendations. Enriched documents are generated from feedback sessions and this enriched document is used to locate keywords that appear in snippets clicked and un-clicked documents in feedback session.

Generate Pseudo-Documents from Enriched Documents*:* For a feedback session, each URL is converted into enriched document. This document contains frequent terms that appears in clicked and un-clicked documents. For each feedback session, a Pseudo-Document is generated from its enriched documents.

Generate Optimized Pseudo-Document from Pseudo Documents: The pseudo document reflects both the relevant and irrelevant documents to the users. Optimized Pseudo-document is generated by combining all the pseudo documents for an input query.

5 CONCLUSIONS

Semantic Similarity measures between words plays an important role information retrieval, natural language processing and in various tasks on the web. In this work, a new approach is proposed in this of inferring user search goals by using the feedback session and pseudo document. In the feedback session both the clicked and the un clicked URLs ones before last click are stored. Pseudo document is made from mapping of feedback session.we have presented Related Search Recommendation (RSR) to suggest related queries to given input query by using feedback session from user click through log. Each feedback session is converted into enriched documents. Pseudo Documents are generated by combining all the enriched documents of a feedback session. Optimized Pseudo Document is generated by combining all the Pseudo Documents for a given input query, which reflects the user's information need. In future find out Semantic similarity is calculated by Web Jaccrad, Web Dice, Web PMI and Web Overlap methods for terms present in the optimized Pseudo Document. Recommendations are generated and ranked by combining query and terms for all the methods. In future Recommending related search with user feedback session and semantic similarity between words. Expecting that It Provides Semantically related search to inputs and this approach can be extended to generate multiple related words.

REFERENCES

[1] M. P. Kato, T. Sakai, and K. Tanaka, "When Do People Use Query Suggestion? A Query Suggestion Log Analysis," *Journal of Information Retrieval*, vol. 16, no. 6, pp. 725–746, December 2013.

[2] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz, "Analysis of a Very Large Web Search Engine Query Log," *In SIGIR Forum*, pp. 6–12, 1999.

[3] M. P. Kato, T. Sakai, and K. Tanaka, "When Do People Use Query Suggestion? A Query Suggestion Log Analysis," *Journal of Information Retrieval*, vol. 16, no. 6, pp. 725–746, December 2013.

[4] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz, "Analysis of a Very Large Web Search Engine Query Log," *In SIGIR Forum*, pp. 6–12, 1999.

[5] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li, "Context-Aware Query Suggestion by Mining Click-Through and Session Data," p. 875–883, 2008.

[6] Z. Kunpeng, W. Xiaolong, and L. Yuanchao, "A New Query Expansion Method based on Query Logs Mining," *International Journal on Asian Language Processing*, vol. 19, no. 1, pp. 1–12, 2009.

[7] J. Miao, J. X. Huang, and Z. Ye, "Proximity based Rocchios Model for Pseudo Relevance Feedback," pp. 535–544, August 2012.

[8] M. S. Hamani and R. Maamri, "Word Semantic Similarity based on Document's title," *In the Proceedings of 24th IEEE International Workshop on Database and Expert Systems Applications*, pp. 43–47, 2013.

[9] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring Semantic Similarity between Words using Web Search Engines," pp. 757–766, 2007.

[10] P. Li, H. Wang, K. Q. Zhu, Z. Wang, and X. Wu, "Computing Term Similarity by Large Probabilistic *isa* Knowledge," *CIKM '13 : In the Proceedings of 22nd International Conference on Information and Knowledge Management*, pp. 1401–1413, 2013.

[11] B. Danushka, M. Yutaka, and I. Mitsuru, "A Relational Model of Semantic Similarity between Words using Automatically Extracted Lexical Pattern Clusters from the Web," *EMNLP '09 : In the Proceedings of International Conference on Empirical Methods in Natural Language Processing*, pp. 803–812, 2009.

[12] D. Bollegala, Y. Matsuo, and M. Ishizuka, "A Web Search Engine-Based Approach to Measure Semantic Similarity between Words," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 977–990, July 2011.

[13] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," *IEEE Transactions on Knowledge and Data Engineering*, pp. 448–453, 1995.

[14] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pas¸ca, and A. Soroa, "A Study on Similarity and Relatedness using Distributional and Wordnet-Based Approaches," *In the Proceedings of Human*

[15] G. Hirst and D. St-Onge, "Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms," *WordNet : An Electronics Lexical Database*, pp. 305–332, 1998.

[16] Y. Song and L.-w. He, "Optimal Rare Query Suggestion with Implicit User Feedback," *WWW '10: In the Proceedings of 19th International Conference on World Wide Web*, pp. 901–910, April 2010.

[17] N. Craswell and M. Szummer, "Random Walks on the Click Graph," *SIGIR '07: In the Proceedings of 30th Annual International ACM SIGIR Conference on Research and Development in Information Retreival*, pp. 239–246, 2007.

[18] 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," pp. 118–126, 2004.

[19] E. Kharitonov, C. Macdonald, P. Serdyukov, and L. Ounic, "Intent Models for Contextualizing and Diversifying Query Suggestions," *CIKM '13:* pp. 2303–2308, 2013.

[20] U. Ozertem, O. Chapelle, P. Donmez, and E. Velipasaoglu, "Learning to Suggest: A Machine Learning Framework for Ranking Query Suggestions," pp. 25 –34, August 2012.