Hierarchical Query Task Clustering using Search Logs

Vivek Bhojawala[1], Pinal Patel[2]

[1,2] Government Engineering College Gandhinagar, India

ABSTRACT

In this paper we have enhanced Query Task Clustering algorithm for to give additional feature such as URL recommendation and next sub-task based suggestion on top of existing query suggestion model. Hierarchical Query Task Clustering algorithms breaks user search task into sub tasks and captures relation between sub-tasks. It uses click through bipartite graph to generate query similarity without using any explicit judgements. It also uses query features such as average dwell time for query, total no of URLs clicks and no of unique URLs clicked to give more accurate results. Experimental results show that it captures more deep user search task information and performs as good as Query task clustering for query suggestion and opens the gates for URL recommendation framework which is input to search personalizing.

Keyword : -, *URL recommendation, Query task clustering, Hierarchical Query Task Clustering (HQTC)*

1. INTRODUCTION

Search engines provides information based on user query. When user searches on the search engine clicking URLs search engine creates record of users search history at servers called as search logs. It contains information such as search query, query terms, session id, date and time of search, URL clicked, at which time in session URL clicked, list of URLs presented to users etc. Search logs are implicit feedback of user and by accurately interpreting it we can improve user search experience [3]. Search logs can be analyzed at following different levels 1) Query level: each query is treated as an individual unrelated to other and captures the clicks performed by user for each query. 2) Session level: consecutive query in certain timeout (30 min) are captured as a unit of user interaction with search engine. Session has following limitation 1) all queries in same session are not similar. 2) Single search session may have parallel search task each of which are not related to each other. 3) Single search task may span multiple session. User information need is not satisfied within single search session; user generally resumes the search task left in previous session in to next session [8]. 3)Task level: Task captures unambiguous single information need of user. A task may span multiple search session or a task can be performed parallel by as user within same session [8]. In this paper we consider Task level analysis of search logs.

For an example we consider the task of purchasing mobile online. For that user might follow one more of following sub-tasks:

1. Visit the Home Page of well-known Mobile manufacturer's website: Suppose user is a Samsung brand fan he/she will definitely visit Samsung mobile home page for latest and upcoming product news and offers.
2. Visit Technological Blogs: If user is very much interested in the technological news he/she will regularly visit some well-known blogs. Form that user finds out latest model launched by all the companies and their hands on.
3. Visit Review websites: On review sites user finds vendor neutral information about products. Review site also contains functionality of comparing the features of products from several manufacturers.
4. Visit Price Compare sites: After choosing a particular mobile phone user searches for lowest price or best deal for a particular mobile on price compare site.
5. Visit particular e-commerce website: Finally, user will visit website of best e-commerce vendor and perform the transaction of buying the mobile.

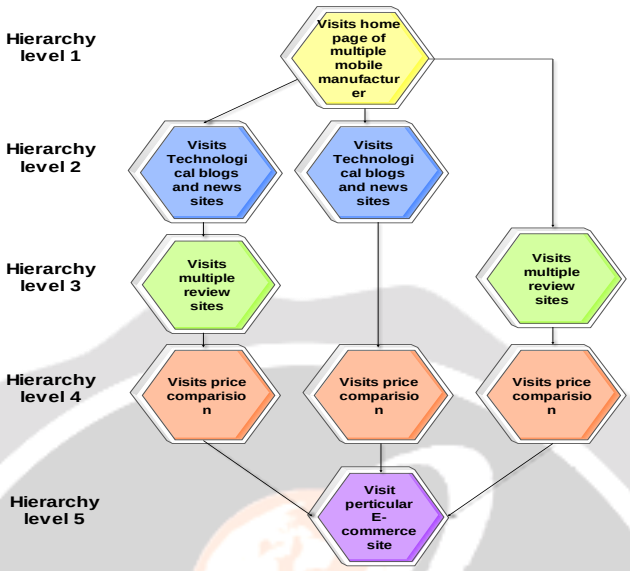

Fig 1: An example of user search task in Hierarchy of sub-tasks

As shown in figure 1 based on user's search history task of buying a mobile phone is divided into 5 subtasks. Each subtask has a specific meaning to task. User can start from any of the hierarchy of the cluster based on user experience and preferences. For each subtasks algorithm captures the query raised by the user as well as clicks performed on particular URL. When a new user searches for the same information, user query is mapped into one of the cluster of hierarchy and query suggestion and URL recommendation is produced.

2. RELATED WORK

The relationship between user search query and information goal is explained by Eric Horvitz [7] how user search information needs during the search session and how queries are inter-related to each other to fulfil single information need. Features representing user behavior explained by Eric Brill [19] can be query text, browsing and click-through feature. When manipulating with search result using user search history how user responds to new personalized search results is explained by Teevan and Jaime [22].

For computing query similarity from search logs Jun Xu [11] explained constructing query similarity bipartite graph using clicked URLs for queries. Similarity between queries and URLs can be identified using Euclidean distance. Gui-Rong [14] explained co-visited method for calculating query similarity. If users access same URL using different queries, then those queries are counted as similar in co-visited method. Learning the query intent as explained by Alex Acero [12] is a semi supervised learning method to analyses user search query.

User's search interest can be classified in to short term interests and long term interests. Short term search interests are limited to particular search session or among n number of subsequent queries where as long term search interest can span multiple search session to entire user search history. Predicting short-term interests using activity-based search context explained by Paul N. Bennett [10] uses past queries in the session as a context and builds intents from related queries inside the context and based on current query it predicts user search interest. Long term search history to improve search accuracy explained by Xuehua Shen [9] identifies user interest areas form the search history and URLs extracted by the users. Based on identified search interests web search is personalized.

Query task clustering algorithm explained by Li-wei He and Yalou Huang [8] provides task level identification of user search interest. Based on observation that consecutive queries are more likely to belong to same task as compare to non-consecutive it tries to check for query similarity and if it finds similar then those queries are added to particular user search task. For making query suggestion out of co-occurrence, log likelihood and random walk was used to measure performance out of all random walk gives best performance. Using Popular URLs to enhance

web search explained by Ryen W. [21] compared query suggestion with query destination and session destination and results shown that query destination as suggestion performs best.

3. DATASET

To implement the Hierarchical Query task clustering search logs with maximum user interaction is required. For the research purpose publicly available anonymized dataset of Yandex search engine is used. Yandex is Russian Internet Company working largely on search engine. Yandex has 50.5 million visitors per day and processes 150 million searches per day. It also provides other services such as image and video search, mail, language translator and its own Yandex browser. In October 2013 Yandex published personalized web search Challenge on Kaggle.com.

4. EXPERIMENTAL DESIGN

A Single user task may contain two or more sub-tasks which may be related to each other or not. Query Task Clustering algorithm [8] measures similarity between query based on temporal features such as time spend and query word similarity analysis between two queries. If query is not similar it is not included into the task. A single user task may involve queries which are not related to each other but still it makes valuable addition as a sub-task to atomic user task.

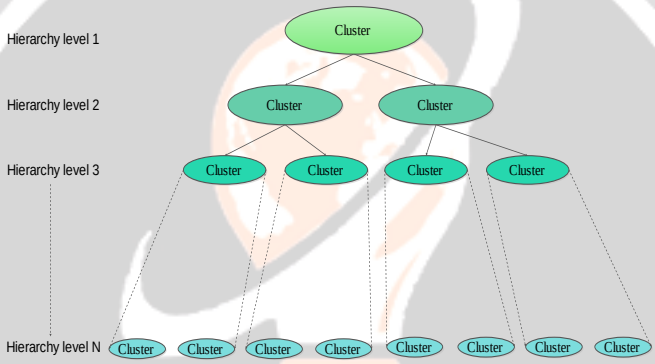

Fig 2: Framework of Hierarchical Query Task Clustering

Sub-tasks are modelled as a level of hierarchy in a cluster so that important sub task is not missed to capture a single search task. Each sub-task acquires hierarchy level based on order in which search task is performed. Cluster at the root hierarchy contains generalized information needs so it consists of most of query and clicked URLs. As the hierarchy level progress user information needs becomes more specific so it separates subtask into separate clusters and at the leaf cluster it contains most specific information needs.

1. URL recommendation using HQTC:

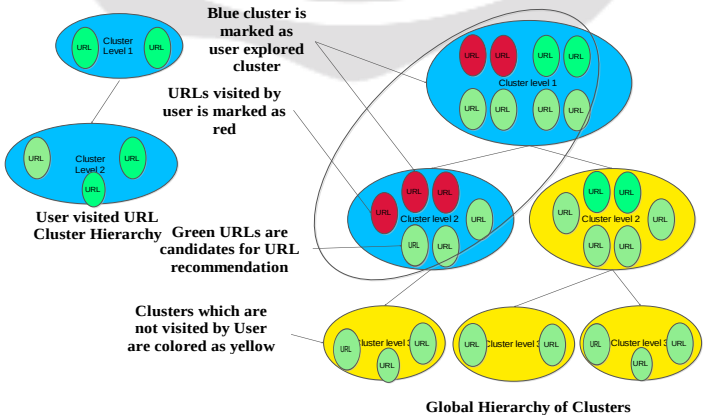

Fig 3: URL recommendation using HQTC

URL recommendation enhancing existing knowledge of a user. World Wide Web is dynamic and search topics are also dynamic. No user can find all information about a particular topic at one place or in search sessions. It is likely that user may miss some part of an information he/she was looking for during current search session. URL recommendation maps URLs visited by particular user to the hierarchical query task cluster and find out most URL that will have distinctive information regarding current user task. As shown in figure 3 User start his/ her task of searching. A single user task is divided into multiple subtasks. Each subtasks contains URLs visited by the users.

In first step algorithm finds cluster of sub-tasks formed by user and maps those cluster in to global hierarchy of task cluster. Now URL recommendation algorithm maps URLs visited by user in to the Global Cluster of tasks and based on that new URL is recommended to user.

2. Hierarchical Query Task Clustering and user search trails:

Every user of search engine has different background, knowledge level of the domain, searching skills. Some user are experts in finding information on web through their searching skills and some may face difficulties in finding information in new domain. When expert user finds information it reaches complete set of URLs required to complete the task. When a user searching for information and he/she is new to the domain than user can follow the trails of an expert user of that domain. Web search does not end at just finding information only once, user interested in particular domain will re-find same information again and again creates trails of finding information. This trails can be useful for other user to find new information in an interesting way.

As shown in figure 4 user starts its search trails form the cluster level 1 and finishes task when required information is found at cluster level 3. Purple node is starting node and red node as end node, all other connected URL green nodes are intermediary node. When user visits same trails repeatedly or many user follows the same trails it becomes interesting for other user to know those trails to enhance their existing knowledge. We can also provide the query raised by user in subtasks along with URLs visited to provide complete insight in to task. Task searching provides an interesting way to search an information without submitting multiple queries and all required information is provided at single place.

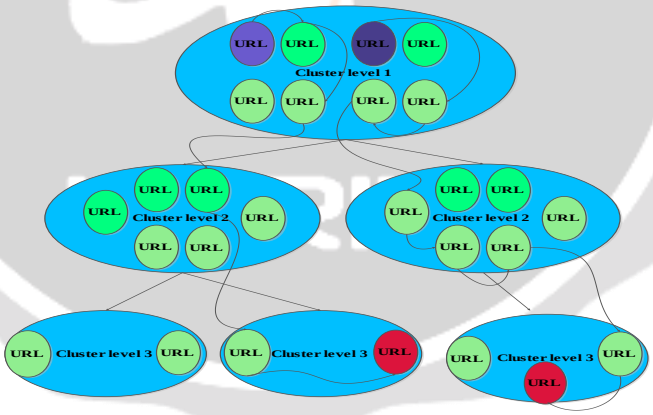

Fig 4: User search trails in HQTC

3. Query feature extraction:

Click through bipartite only gives you count of click for particular URLs that is not enough to produce accurate search results. so for hierarchical query task clustering we add total no click on URL for query, average dwell time and total no of unique URL accessed and list of those URLs.

4. Hierarchical Query Task Clustering (HQTC):

For given query from query similarity and query feature it will sort list of URLs which occurred in user search session after base query. And form list it will try to check query similarity from root cluster to leaf cluster if similarity is found query will be added to that particular cluster otherwise new cluster will be created and query will be added to that cluster.

5. EXPERIMENTAL RESULTS

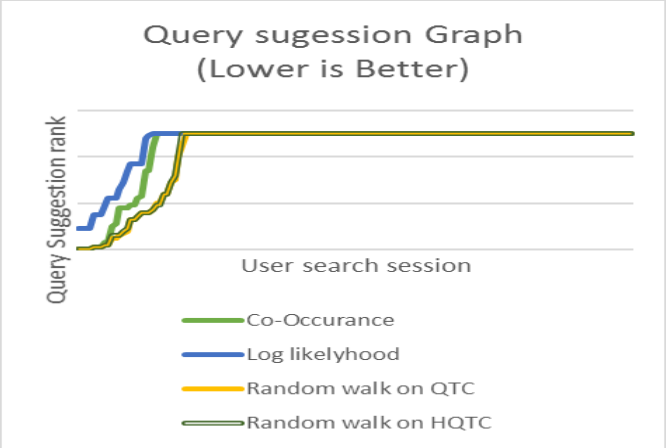

Fig 5: Query suggestion performance of different methods

As shown in figure performance of Query task clustering is compared with Hierarchical query task clustering for query suggestion using Random walk suggested in base paper [8]. Graph shows that on some occasions Query task clustering algorithm performed well where on other Hierarchical query task gave more accurate results but overall performance deviation was very small.

5. CONCLUSION

Hierarchical Query Task Clustering captures more knowledge of user search task without degrading the performance of query suggestion. Using HQTC we can also provide URL recommendation and sub task recommendation.

6. REFERENCES

[1] Broder, Andrei. "A taxonomy of web search." In ACM Sigir forum, vol. 36, no. 2, pp. 3-10. ACM, 2002.

[2] Mei, Qiaozhu, and Kenneth Church. "Entropy of search logs: how hard is search? with personalization? withbackoff?." In Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 45-54. ACM, 2008.

[3] Joachims, Thorsten, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. "Accurately interpreting clickthrough data as implicit feedback." InProceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 154-161. ACM, 2005.

[4] Radlinski, Filip, MadhuKurup, and Thorsten Joachims. "How does clickthrough data reflect retrieval quality?." In Proceedings of the 17th ACM conference on Information and knowledge management, pp. 43-52. ACM, 2008.

[5] Chapelle, Olivier, and Ya Zhang. "A dynamic bayesian network click model for web search ranking." In Proceedings of the 18th international conference on World wide web, pp. 1-10. ACM, 2009.

[6] Guo, Fan, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. "Click chain model in web search." In Proceedings of the 18th international conference on World wide web, pp. 11-20. ACM, 2009.

[7] Downey, Doug, Susan Dumais, Dan Liebling, and Eric Horvitz. "Understanding the relationship between searchers' queries and information goals." InProceedings of the 17th ACM conference on Information and knowledge management, pp. 449-458. ACM, 2008.

[8] Liao, Zhen, Yang Song, Li-wei He, and Yalou Huang. "Evaluating the effectiveness of search task trails." In Proceedings of the 21st international conference on World Wide Web, pp. 489-498. ACM, 2012.

[9] Tan, Bin, Xuehua Shen, and ChengXiangZhai. "Mining long-term search history to improve search accuracy." In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 718-723. ACM, 2006.

[10] White, Ryen W., Paul N. Bennett, and Susan T. Dumais. "Predicting short-term interests using activity-based search context." In Proceedings of the 19th ACM international conference on Information and knowledge management, pp. 1009-1018. ACM, 2010.

[11] Wu, Wei, Hang Li, and Jun Xu. "Learning query and document similarities from click-through bipartite graph with metadata." In Proceedings of the sixth ACM international conference on Web search and data mining, pp. 687-696. ACM, 2013.

[12] Li, Xiao, Ye-Yi Wang, and Alex Acero. "Learning query intent from regularized click graphs." In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 339-346. ACM, 2008.

[13] Bilenko, Mikhail, and Ryen W. White. "Mining the search trails of surfing crowds: identifying relevant websites from user activity." In Proceedings of the 17th international conference on World Wide Web, pp. 51-60. ACM, 2008.

[14] Xue, Gui-Rong, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, WenSi Xi, and WeiGuo Fan. "Optimizing web search using web click-through data." InProceedings of the thirteenth ACM international conference on Information and knowledge management, pp. 118-126. ACM, 2004.

[15] Hölscher, Christoph, and Gerhard Strube. "Web search behavior of Internet experts and newbies." Computer networks 33, no. 1 (2000): 337-346.

[16] Xiang, Biao, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. "Context-aware ranking in web search." In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp. 451-458. ACM, 2010.

[17] Shen, Xuehua, Bin Tan, and ChengXiangZhai. "Context-sensitive information retrieval using implicit feedback." In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 43-50. ACM, 2005.

[18] Cao, Huanhuan, Daxin Jiang, Jian Pei, Enhong Chen, and Hang Li. "Towards context-aware search by learning a very large variable length hidden markov model from search logs." In Proceedings of the 18th international conference on World wide web, pp. 191-200. ACM, 2009.

[19] Agichtein, Eugene, Eric Brill, Susan Dumais, and Robert Ragno. "Learning user interaction models for predicting web search result preferences." InProceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 3-10. ACM, 2006.

[20] Agichtein, Eugene, Eric Brill, and Susan Dumais. "Improving web search ranking by incorporating user behavior information." In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 19-26. ACM, 2006.

[21] White, Ryen W., Mikhail Bilenko, and SilviuCucerzan. "Studying the use of popular destinations to enhance web search interaction." In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 159-166. ACM, 2007.

[22] Teevan, Jaime. "How people recall, recognize, and reuse search results." ACM Transactions on Information Systems (TOIS) 26, no. 4 (2008): 19.