ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

Survey on Search Recommendation with Mining Query Facets

Hemangi P. Patil, Prof. M.A.Bhandari.

M.E. Student, Dept. of Computer Engineering, GHRIET, Wagholi, Maharashtra, India

Assistant Professor, Dept. of Computer Engineering, GHRIET, Wagholi, Maharashtra, India

ABSTRACT: Web search queries are often ambiguous or multi-faceted, which makes a simple ranked list of results inadequate. To assist information finding for such faceted queries, we explore a technique that explicitly represents interesting facets of a query using groups of semantically related terms extracted from search results. As an example, for the query baggage allowance", these groups might be different air-lines, different flight types (domestic, international), or different travel classes (rst, business, economy). We name these groups query facets and the terms in these groups' facet terms. We develop a supervised approach based on a graphical model to recognize query facets from the noisy candidates found. The graphical model learns how likely a candidate term is to be a facet term as well as how likely two terms are to be grouped together in a query facet, and captures the dependencies between the two factors. We propose two algorithms for approximate inference on the graphical model since exact inference is intractable. Our evaluation combines recall and precision of the facet terms with the grouping quality. Experimental results on a sample of web queries show that the supervised method significantly outperforms existing approaches, which are mostly unsupervised, suggesting that query facet extraction can be e effectively learned.

KEYWORDS: Query facet, faceted search, summarization, user intent.

I. INTRODUCTION

Current information retrieval systems, including Web search engines, have a standard interface consisting of a single input box that accepts keywords. The keywords submitted by the user are matched against the collection index to find the documents that contain those keywords, which are then sorted by various methods. When a user query contains multiple topic-specific keywords that accurately describe his information need, the system is likely to return good matches; however, given that user queries are usually short and that the natural language is inherently ambiguous, this simple retrieval model is in general prone to errors and omissions.

Query facets provide interesting and useful knowledge about a query and thus can be used to improve search experiences in many ways. First, we can display query facets together with the original search results in an appropriate way. Thus, users can understand some important aspects ofa query without browsing tens of pages. For example, a user could learn different brands and categories of watches. We can also implement a faceted search based on the mined query facets. User can clarify their specific intent by selecting facet items. Then search results could be restricted to the documents that are relevant to the items. A user could drill down to women's watches if he is looking for a gift for his wife. These multiple groups of query facets are in particular useful for vague or ambiguous queries, such as "apple". We could show the products of Apple Inc. in one facet and different types of the fruit apple in another. Second, query facets may provide direct information or instant answers that users are seeking. For example, for the query "lost season 5", all episode titles are shown in one facet and main actors are shown in another. In this case, displaying query facets could save browsing time. Third, query facets may also be used to improve the diversity of the ten blue links. We can re-rank search results to avoid showing the pages that are near-duplicated in query facets at the top. Query facets also contain structured knowledge covered by the query, and thus they can be used in other fields besides traditional web search, such as semantic search or entity search.

We observe that important pieces of information about a query are usually presented in list styles and repeated many times among top retrieved documents. Thus we propose aggregating frequent lists within the top search results to mine query dimensions and implement a system called QDMiner. More specifically, QDMiner extracts lists from free text,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

HTML tags, and repeat regions contained in top search results, and groups them into clusters based on the items they contain. Compared to previous works on building facet hierarchies, our approach is unique in two aspects:
(1) Open domain: we do not restrict queries in a specific domain, like products, people etc. Our proposed approach is generic and does not rely on any specific domain knowledge. Thus it can deal with open-domain queries.
(2) Query dependent : instead of a same pre-defined schema for all queries, we extract dimensions from the top retrieved documents for each query. As a result, different queries may have different dimensions. For example, although "lost" and "lost season 5" are both TV program related queries, their mined dimensions are different.As the problem of finding query dimension is new, we cannot find publicly available evaluation datasets. There fore, we create two datasets, namely UserQ, containing 89 queries that are submitted by QDMiner users, and RandQ, containing 105 randomly sampled queries from logs of a commercial search engine, to evaluate mined dimensions. We use some existing metrics, such as purity and normalized mutual information (NMI), to evaluate clustering quality, and use NDCG to evaluate ranking effectiveness of dimensions. We further propose two metrics to evaluate the integrated effectiveness of clustering and ranking.Experimental results show that quality of query facets mined by QDMiner is good. We find that quality of query facets is affected by the quality and the quantity of search results. Using more results can generate better facets at the beginning, whereas the improvement of using more resultsranked lower than 50 becomes subtle. We find that the Context Similarity Model outperforms the Unique Website Model, which means that we could further improve quality of query facets by considering context similarity of the lists during ranking the facets and items.

II. LITERATURE SURVEY

1. Zhicheng Dou, Member, IEEE, Zhengbao Jiang, Sha Hu,Ji-RongWen, and Ruihua Song, February 2016, "Automatically Mining Facets for Queries from Their Search Results".

In this paper, we study the problem of finding query facets. We propose a systematic solution, which we refer to as QD Miner, to automatically mine query facets by aggregating frequent lists from free text, HTML tags, and repeat regions within top search results. We create two human annotated data sets and apply existing metrics and two new combined metrics to evaluate the quality of query facets. Experimental results show that useful query facets are mined by the approach. We further analyze the problem of duplicated lists, and find that facets can be improved by modeling fine-grained similarities between listswithin a facet by comparing their similarities. We have provided query facets as candidate subtopics in the NTCIR-11 IMine Task.

2. Feng Zhao, Jingyu Zhou, Chang Nie,Heqing Huang, Hai Jin,year2015, "Smart Crawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces".

In this paper, we propose an effective harvesting framework for deep-web interfaces, namely Smart- Crawler. We have shown that our approach achieves both wide coverage for deep web interfaces and maintains highly efficient crawling. Smart Crawler is a focused crawler consisting of two stages: efficient site locating and balanced in-site exploring. Smart Crawler performs site-based locating by reversely searching the known deep web sites for center pages, which can effectively and many data sources for sparse domains. By ranking collected sites and by focusing the crawling on a topic, Smart Crawler achieves more accurate results. The in-site exploring stage uses adaptive link-ranking to search within a site; and we design a link tree for eliminating bias toward certain directories of a website for wider coverage of web directories. Our experimental results on a representative set of domains show the effectiveness of the proposed two stage crawler, which achieves higher harvest rates than other crawlers. In future work, we plan to combine pre-query and post-query approaches for classifying deep web forms to further improve the accuracy of the form classifier.

3.LidanShou, He Bai, Ke Chen, and Gang Chen, year2014,"Supporting Privacy Protection in Personalized Web Search".

This paper presented a client-side privacy protection framework called UPS for personalized web search. UPS could potentially be adopted by any PWS that captures user pro_les in a hierarchical taxonomy. The framework allowed users to specify customized privacy requirements via the hierarchical profiles. In addition, UPS also performed online generalization on user profiles to protect the personal privacy without compromising the search quality. We proposed two greedy algorithms, namely Greedy DP and Greedy IL, for the online generalization. Our experimental results

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

revealed that UPS could achieve quality search results while preserving users customized privacy requirements. The results also confirmed the effectiveness and efficiency of our solution. For future work, we will try to resist adversaries with broader background knowledge, such as richer relationship among topics (e.g., exclusiveness, sequentiality, and so on), or capability to capture a series of queries from the victim. We will also seek more sophisticated method to build the user profile, and better metrics to predict the performance (especially the utility) of UPS.

4.Anju G R1 and KarthikM2 ,Year2016. "**Minning Queries From Search Results : A Survey".**
As the primary methodology of discovering question features, can be enhanced in numerous angles. For instance, some semi administered bootstrapping list extraction calculations can be utilized to iteratively extricate more records from the top results. Particular site wrappers can likewise be utilized to concentratetop notch records from legitimate sites. Including these rundowns may enhance both precision and review of inquiry features. Grammatical feature data can be utilized to further check the homogeneity of records and enhance the nature of inquiry aspects. We will investigate these points to refine aspects later on. We will likewise research some other related themes to discovering inquiry aspects. Great portrayals of question aspects might be useful for clients to better comprehend the features. Automatically create significant depictions is an intriguing examination subject.

III. METHODOLOGY

We do not find any publicly available dataset for evaluating query facets. Therefore, we build two datasets from scratch. First, we build a service for finding facets, and invite human subjects to issue queries on topics they know well. We collect 89 queries issued by the subjects, and name them as "UserQ". As this approach might induce a bias towards topics in which lists are more useful than general web queries, we further randomly sample another set of 105 English queries from a query log of a commercial search engine, and name this set of queries as "RandQ". For each query, we first ask a subject to manually create facets and add items that are covered by the query, based on his/her knowledge after a deep survey on any related resources (such as Wikipedia, Freebase, or official web sites related to the query). We then aggregate the qualified items in the facets returned by all algorithms we want to evaluate, and ask the subject to assign unlabeled items into the created facets. New facets will be created for the items that are not covered by the existing facets. A facet named "misc" is automatically created for each query by the labeling system. Subjects can add the noisy or irrelevant items into this facet. The main purpose of creating this "misc" facet is to help subjects to distinguish between bad and unjudged items. During evaluation, "misc" facets are discarded before mapping generated facets to manually labelled facets.

IV. EXISTING SYSTEM

We address the problem of finding query facets. A query facet is a set of items which describe and summarize query one important aspect of a query. Here a facet item is typically a word or a phrase. A query may have multiple facets that summarize the information about the query from different perspectives. In sample facets are for some queries.
Facets are for the query "watches" cover the knowledge about watches in five unique aspects, including brands, gender categories, supporting features, styles, and colors. The query "visit Beijing" has a query facet about popular resorts in Beijing (Tiananmen square, forbidden city, summer palace, . . .) and a facet on travel related topics(attractions, shopping, dining, . . .). Query facets provide interesting and useful knowledge about a query and thus can be used to improve search experiences in many ways. First, we can display query facets together with the original search results in an appropriate way. Thus, users can understand some important aspects of a query without browsing tens of pages. Second, query facets may provide direct information or instant answers that users are seeking. For example, for the query "lost season", all episode titles are shown in one facet and main actors are shown in another. In this case, displaying query facets could save browsing time. Third query facets may also be used to improve the diversity of the ten blue links. We can re-rank search results to avoid showing the pages that are near-duplicated in query facets at the top. Query facets also contain structured knowledge covered by the query, and thus they can be used in other fields besides traditional web search, such as semantic search or entity search.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

V. PROPOSED SYSTEM

We propose aggregating frequent lists within the top search results to mine query facets and implement a system called QDMiner. More specifically, QDMiner extracts lists from free text, HTML tags, and repeat regions contained in the top search results, groups them into clusters based on the items they contain, then ranks the clusters and items based on how the lists and items appear in the top results. We propose two models, the Unique Website Model and the Context Similarity Model, to rank query facets. In the Unique Website Model, we assume that lists from the same website might contain duplicated information, whereas different websites are independent and each can contribute a separated vote for weighting facets. However, we find that sometimes two lists can be duplicated, even if they are from different websites. For example, mirror websites are using different domain names but they are publishing duplicated content and contain the same lists. Some content originally created by a website might be re-published by other websites; hence the same lists contained in the content might appear multiple times in different websites.

VI. CONCLUSION AND FUTURE WORK

We assume that the important aspects of a query are usually presented and repeated in the query's top retrieved documents in the style of lists, and query facets can be mined out by aggregating these significant lists. We propose a systematic solution, which we refer to as QDMiner, to automatically mine query facets by extracting and grouping frequent lists from free text, HTML tags, and repeat regions within top search results.

REFERENCES

1. O. Ben-Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. Shekita, B. Sznajder, and S. Yogev, "Beyond basic faceted search," in Proc. Int. Conf. Web Search Data Mining, 2008, pp. 33–44.
2. M. Diao, S. Mukherjea, N. Rajput, and K. Srivastava,, "Faceted search and browsing of audio content on spoken web," in Proc. 19th ACM Int. Conf. Inf. Knowl. Manage., 2010, pp. 1029–1038.
3. D. Dash, J. Rao, N. Megiddo, A. Ailamaki, and G. Lohman, "Dynamic faceted search for discovery-driven analysis," in ACM Int. Conf. Inf. Knowl. Manage., pp. 3–12, 2008.
4. W. Kong and J. Allan, "Extending faceted search to the general web," in Proc.ACMInt. Conf. Inf. Knowl. Manage., 2014, pp. 839–848.
5. T. Cheng, X. Yan, and K. C.-C. Chang, "Supporting entity search: A large-scale prototype search engine," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2007, pp. 1144–1146.
6. K. Balog, E. Meij, and M. de Rijke, "Entity search: Building bridges between two worlds," in Proc. 3rd Int. Semantic Search Workshop, 2010, pp. 9:1–9:5.
7. M. Bron, K. Balog, and M. de Rijke, "Ranking related entities: Components and analyses," in Proc. ACM Int. Conf. Inf. Knowl. Manage., 2010, pp. 1079–1088.
8. C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, "Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia," in Proc. 19th Int. Conf. World Wide Web, 2010, pp. 651–660.
9. W. Dakka and P. G. Ipeirotis, "Automatic extraction of useful facet hierarchies from text databases," in Proc. IEEE 24th Int. Conf. Data Eng., 2008, pp. 466–475.
10. A. Herdagdelen, M. Ciaramita, D. Mahler, M. Holmqvist, K. Hall, S. Riezler, and E. Alfonseca, "Generalized syntactic and semantic models of query reformulation," in Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. retrieval, 2010, pp. 283–290.
11. M. Mitra, A. Singhal, and C. Buckley, "Improving automatic query expansion," in Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 1998, pp. 206–214.
12. P. Anick, "Using terminological feedback for web search refinement: A log-based study," in Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2003, pp. 88–95.
13. S. Riezler, Y. Liu, and A. Vasserman, "Translating queries into snippets for improved query expansion," in Proc. 22nd Int. Conf. Comput. Ling., 2008, pp. 737–744.
14. X. Xue and W. B. Croft, "Modeling reformulation using query distributions," ACM Trans. Inf. Syst., vol. 31, no. 2, pp. 6:1–6:34, May 2013.