e-ISSN: 2456-3463
International Journal of Innovations in Engineering and Science, Vol. 2, No.2, 2017
www.ijies.net

A Survey on Automatically Mining Facets for Queries from Their Search Results

Atul Kumar Ojha[1], Amit Singh Bhadoria[1], Nikhil SharadAhire[1],Ganesh Baburao Mane[1]
Prof.T.Arivanantham[2]

[1,2]Dr.D.Y.Patil Institute of Engineering and Technology, Pune, India,411018

Abstract *–We deal with the problem of discovering query facets which are several groups of words or phrases that make clear and review the content enclosed by a query. We believe that the significant aspects of a query are usually presented and recurred in the query's peak retrieved documents in the style of lists, and query facets can be mined out by aggregating these important lists. We propose an organized answer, which we refer to as QDMiner, to automatically supply query facets by extracting and grouping recurrent lists from free text, HTML tags, and duplicate regions within top search results. Experimental outcome show that a big number of lists are present and valuable query facets can be mined by QDMiner. We further analyze the problem of list duplication, and find superior query facets can be mined by modeling fine-grained similarities between lists and punishing the duplicated lists.*

Keywords-Query facet, faceted search, summarize

INTRODUCTION

WE tackle the problem of finding query facets. A query facet is a set of items which explain and summarize one significant aspect of a query. Here a facet item is typically a word or a phrase. A query may have various facets that summarize the information about the query from different perspectives. Table 1 shows sample facets for some queries. Facets for the query "watches" cover the information about watches in five distinctive aspects, as well as brands, gender categories, sustaining features, styles, and colors. The query "visit Beijing" has a query facet about trendy resorts in Beijing (Tiananmen square, forbidden city, summer palace, . . .) and a facet on tour related topics (attractions, shopping, dining, . . .). Query facets provide remarkable and useful information about a query and thus can be used to get better search experiences in many ways. First, we can present query facets together with the original search results in a suitable way. Thus, users can understand some significant aspects of a query without browsing tens of pages. For example, a user could study different brands and categories of watches. We can also apply a faceted search based on the mined query facets. User can make clear their specific intent by selecting facet items. Then search outcome could be restricted to the documents that are associated to the items. A user could drill down to women's watches if he is looking for a gift for his wife. These various groups of query facets are in particular useful for vague or uncertain 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 offer direct information or instantaneous answers that users are looking for. For example, for the query "the flash", all event titles are shown in one facet and main actors are shown in another. In this case, showing query facets could save browsing time. Third, query facets may also be used to get enhanced diversity of the ten blue links. We can re-rank investigated outcome to avoid showing the pages that are near-duplicated in query facets at the top. Query facets also contain ordered information covered by the query, and thus they can be used in other fields besides conventional web search, such as semantic search or entity search. We observe that significant 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 outcome to mine query facets and implement a system called QD Miner. More specifically, QD Miner extracts lists from free text, HTML tags, and repeat regions contained in the top search outcome, groups them into clusters based on the objects they contain, then ranks the clusters and objects based on how the lists and objects appear in the top results. We illustrate QD Miner in Fig. 1.We recommend two models, the Unique Website Model and the Context Similarity Model, to rank query facets. In the Unique Website Model, we presume that lists from the same website might contain duplicated information, while different websits are self-governing and each can give a divided vote for weighting facets. However, we find that occasionally two lists can be duplicated, still if they are from unlike websites. For example, mirror

Table 1
Example Query Facets Mined by QD Miner

Query: watches
1. Cartier, breitling, omega, citizen, tag heuer, bulova, casio,rolex, audemarspiguet, seiko, accutron, movado,
2. men's, women's, kids, unisex
3. analog, digital, chronograph, analog digital, quartz, mechanical, . . .
4. dress, casual, sport, fashion, luxury, bling, pocket, . . .
5. black, blue, white, green, red, brown, pink, orange, yellow, . . .

Query: lost
1. season 1, season 6, season 2, season 3, season 4, season 5
2. matthew fox, naveenandrews, evangelinelilly, josh holloway,jorgegarcia, danieldaekim, michaelemerson
3. jack, kate, locke, sawyer, claire, sayid, hurley, desmond,boone, charlie, ben, juliet, sun, jin, . . .
4. what they died for, across the sea, what kate does, the candidate, the last recruit, everybody loves hugo, the end, . . .

Query: lost season 5
1. because you left, the lie, follow the leader, jughead, 316, . . .
2. jack, kate, hurley, sawyer, sayid, ben, juliet, locke, miles,desmond, charlotte, various, sun, none, richard, daniel, . . .
3. matthew fox, naveenandrews, evangelinelilly, jorgegarcia,henryiancusick, josh holloway, michaelemerson, . . .
4. season 1, season 3, season 2, season 6, season 4

Query: what is the fastest animal in the world
1. cheetah, pronghorn antelope, lion, thomson's gazelle, wildebeest, cape hunting dog, elk, coyote, quarter horse, . . .
2. birds, fish, mammals, animals, reptiles
3. science, technology, entertainment, nature, sports, lifestyle,travel, gaming, world business

Query: visit beijing
1. tiananmen square, forbidden city, summer palace, great wall, temple of heaven, beihai park, hutong, . . .
2. attractions, shopping, dining, nightlife, tours, tip, . . .

websites are using different domain names but they are publishing duplicated content and contain the same lists. Some content initially produced by a website might be re-published by other websites, hence the same lists contained in the content might appear various times in different websites. Furthermore, different websites may publish content using the similar software and the software may generate duplicated lists in different websites.

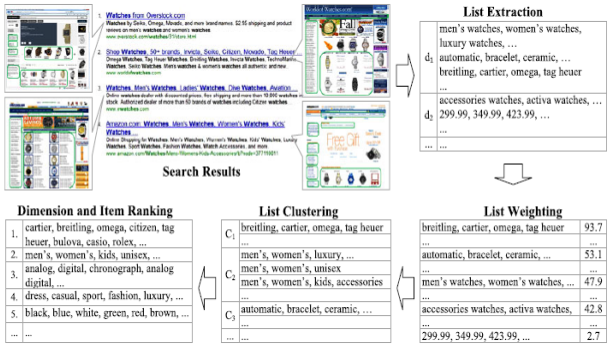

Fig. 1- System overview of QDMiner

Review of Existing Work

This section reviews the main existing work found in the scientific literature that applies on Automatically Mining Facets for Queries from Their Search Results.

[1] This paper extends established faceted search to support more affluent information discovery tasks over more difficult data models. Our first extension adds elastic, active business intelligence aggregations to the faceted application, enabling users to gain insight into their data that is far richer than just knowing the quantities of documents belonging to each facet. We see this potential as a step toward bringing OLAP capabilities, traditionally supported by databases over relational data, to the domain of free-text queries over metadata-rich content. Our second addition shows how one can proficiently extend a faceted search engine to support interrelated facets - a more intricate information model in which the values associated with a document across multiple facets are not independent. We show that by reducing the difficulty to a recently solved tree-indexing scenario, facts with correlated facets can be efficiently indexed and retrieved.

[2]Spoken Web is a network of Voice Sites that can be accessed by a phone. The substance in a Voice Site is audio. Therefore Spoken Web provides an alternate to the World Wide Web (www) in rising regions where low Internet access and low literacy are barriers to accessing the conservative www. Searching of audio content in Spoken Web through an audio query-result interface presents two key challenges: indexing of audio content is not precise, and the arrangement of results in audio is sequential, and therefore cumbersome. In this paper, we apply the concepts of faceted search and browsing to the Spoken Web search problem. We use the concepts of facets to index the meta-data associated with the audio content. We provide a means to rank the facets based on the search results. We develop an interactive query

interface that enables effortless browsing of search results through the top ranked facets. To our understanding, this is the first system to use the concepts of facets in audio search, and the first result that provides an audio search for the rural population. We present quantitative results to illustrate the accuracy and usefulness of the faceted search and qualitative results to highlight the usability of the interactive browsing system. The experiments have been conducted on more than 4000audio documents composed from a live Spoken Web Voice Site and evaluations were carried out with 40 farmers who are the intended users of the VoiceSite.

[3]We recommend a dynamic faceted search structure for discovery- driven analysis on data with both textual content and structured attributes. From a keyword query, we want to dynamically choose a little set of "appealing" attributes and present aggregates on them to a user. Similar to work in OLAP discovery, we define "interestingness" as how astonishing an aggregated value is, based on a given expectation. We make two new contributions by proposing a novel" navigational" expectation that's chiefly helpful in the background of faceted search, and a novel interestingness measure through sensible application of p-values. Through a user survey, we find the new expectation and interestingness metric quite valuable. We develop an efficient dynamic faceted search system by improving a accepted open source engine, Solr. Our system exploits compressed bitmaps for caching the posting lists in an inverted index, and a novel directory structure called a bitset tree for fast bitset intersection. We conduct a broad experimental study on huge real data sets and show that our engine performs 2 to 3 times quicker than Solr.

[4] Faceted search helps users by presenting drill-down options as a complement to the keyword input box, and it has been used fruitfully for many vertical applications, including e- commerce and digital libraries. However, this scheme is not well explored for general web search, even though it holds great potential for supporting multi-faceted queries and exploratory search. In this paper, we discover this potential by extending faceted search into the open-domain web setting, which we call Faceted Web Search. To tackle the diverse nature of the web, we propose to use query-dependent automatic facet generation, which generates facets for a query instead of the entire corpus. To incorporate user feedback on these query facets into document ranking, we examine both Boolean filtering and soft ranking models. We assess Faceted Web Search systems by their utility in assisting users to clarify search intent and locate subtopic information. We illustrate how to construct reusable test collections for such tasks, and propose an evaluation method that considers both gain and cost for users. Our experiments testify to the potential of Faceted Web Search, and show Boolean filtering feedback models, which are commonly used in conventional faceted search, are less efficient than soft ranking models.

[5] As the Web has evolved into a data-rich repository, with the typical "page view," current search engines are more and more inadequate. While we regularly search for a variety of data units, nowadays engines only get us in a roundabout way to pages. Hence, we propose the representation of *entity search*, a significant departure from conventional document retrieval. Towards our goal of supporting entity search, in the *WISDM*1 project at UIUC we build and assess our prototype search engine over a 2TB Web corpus. Our demonstration shows the viability and assurance of a large-scale system architecture to sustain entity search.

[6] We reflect on the task of entity search and inspect to which degree state-of-art information retrieval (IR) and semanticweb (SW) technologies are skilled of answering information needs that focus on entities. We also investigate the potential of combining IR with SW technologies to develop the end-to- end performance on a specific entity search task. We arrive at and encourage a proposal to unite text-based entity models with semantic information from the Linked Open Data cloud.

[7]Associated entity finding is the task of returning a ranked list of homepages of significant entities of a specified type that need to engage in a given association with a given source entity. We propose a framework for addressing this task and execute a detailed scrutiny of four core components; co-occurrence models, type filtering, context modeling and homepage finding. Our initial spotlight is on recall. We examine the performance of a model that only uses co-occurrence statistics. While it identifies a set of related entities, it fails to rank them successfully. Two types of fault emerge: (1) entities of the incorrect type spoil the ranking and (2) while somehow linked to the source entity, some retrieved entities do not engage in the right relation with it. To address (1), we add type filtering based on category information obtainable in Wikipedia. To correct for (2), we add related information, represented as language models derived from documents

in which source and target entities co-occur. To complete the pipeline, we find homepages of top ranked entities by combining a language modeling approach with heuristics based on Wikipedia's outer links. Our method achieves very high recall scores on the end-to-end task, providing a concrete starting point for expanding our focus to improve precision; supplementary heuristics lead to state-of-the-art performance.

[8]This paper proposes Facetedpedia, a faceted recovery system for information innovation and investigation in Wikipedia. Given the set of Wikipedia articles resulting from a keyword query, Facetedpedia generates a faceted interface for navigating the product articles. Compared with other faceted retrieval systems, Facetedpedia is completely automatic and dynamic in both facet production and hierarchy construction, and the facets are based on the rich semantic information from Wikipedia. The heart of our approach is to build upon the mutual vocabulary in Wikipedia, more specifically the exhaustive internal structures (hyperlinks) and folksonomy (category system). Given the sheer size and complexity of this corpus, the space of probable choices of faceted interfaces is prohibitively big. We propose metrics for ranking individual facet hierarchies by user's navigational cost, and metrics for ranking interfaces (each with kfacets) by together their average pairwise similarities and average navigational expenses. We thus build up faceted interface discovery algorithms that optimize the ranking metrics. Our experimental assessment and user study verify the usefulness of the system.

[9]Databases of text and text-annotated data comprise a major fraction of the information available in electronic form. Searching and browsing are the characteristic ways that users locate items of interest in such databases. Faceted interfaces represent a new dominant paradigm that proved to be a successful complement to keyword searching. Thus far, the recognition of the facets was either a manual procedure, or relied on apriori information of the facets that can potentially appear in the underlying collection. In this paper, we present an unsupervised technique for automatic extraction of facets valuable for browsing text databases. In particular, we observe, through a pilot study, that facet terms hardly ever appear in text documents, showing that we need external resources to make out useful facet terms. For this, we first classify important phrases in each document. Then, we develop each phrase with "context" phrases using external resources, such as WordNet and Wikipedia, causing facet terms to emerge in the expanded database. Finally, we compare the term distributions in the original database and the expanded databaseto identify the conditions that can be used to construct browsing facets. Our widespread user studies, using the Amazon Mechanical Turk service, show that our techniques produce facets with high precision and recall that are better to existing approaches and help users locate interesting items quicker.

[10]We present a narrative approach to query reformulation which combines syntactic and semantic information by way of comprehensive Levenshtein distance algorithms where the substitution process costs are based on probabilistic term rewrite functions. We look into unsupervised, solid and capable models, and provide empirical evidence of their usefulness. We further discover a generative model of query reformulation and supervised combination methods providing enhanced performance at variable computational costs. Among other preferred properties, our likeness measures integrate information-theoretic interpretations of taxonomic relations such as specification and generalization.

[11]Most informal users of IR systems type **short** queries. Latest research has shown that adding up new words to these queries via odhoc***feedback*** improves the retrievaleffectiveness of such queries. We look into ways to improve this query expansion process by refining the set of documents used in feedback. We start by using physically formulated Boolean filters along with proximity constraints. Our advance is similar to the one proposed by Hearst[l2]. Next, we investigate a completely automatic method that makes use of expression co-occurrence information to estimate word correlation. Experimental outcome show that refining the set of documents used in query extension regularly prevents the query drift caused by blind expansion and yields substantial improvements in retrieval effectiveness, both in terms of average precision and precision in the peak twenty documents. More importantly, the fully automatic approach developed in this study performs competitively with the most excellent manual approach and requires small computational overhead.

[12]Even though interactive query reformulation has been keenly studied in the laboratory, little is known about the real behavior of web searchers who are

presented terminological feedback along with their search results. We scrutinize log sessions for two groups of users interacting with variants of the AltaVista search engine –a baseline group given no terminological opinion and a opinion group to whom twelve refinement terms are offered along with the search results. We examine uptake, refinement usefulness, situation of use, and refinement type preferences. Although our measure of overall session "success" shows no variation between outcomes for the two groups, we find proof that a subset of those users presented with terminological feedback does make valuable use of it on a continuing basis.

[13]User logs of search engines have lately been applied effectively to improve various aspects of web search quality. In this paper, we will apply pairs of user queries and snippets of clicked results to instruct a machine translation model to link the "lexical gap" involving query and document space. We show that the blend of a query-to-snippet translation model with a big n-gram language model trained on queries achieves enhanced contextual query expansion compared to a system based on term correlations.

[14] In this paper we propose a technique that, given a query submitted to a search engine, suggests a record of related queries. The related queries are based in formerly issued queries, and can be issued by the user to the search engine to adjust or readdress the search process. The method planned is based on a query clustering process in which groups of semantically similar queries are recognized. The clustering procedure uses the content of historical preferences of users registered in the query log of the search engine. The method not only discovers the related queries, but also ranks them according to a significance criterion. Finally, we show with experiments over the query log of a search engine the helpfulness of the method.

[15]Users frequently modify a previous search query in hope of retrieving better results. These modifications are called query reformulations or query refinements. Existing research has studied how web search engines can propose reformulations, but has given less attention to how people perform query reformulations. In this paper, we aim to better understand how web searchers refine queries and form a theoretical foundation for query reformulation. We study users' reformulation strategies in the context of the AOL query logs. We create a taxonomy of query refinement strategies and build a high precision rule-based classifier to detect each type of reformulation. Effectiveness of reformulations is measured using user click behavior. Most reformulation strategies result in some benefit to the user. Certain strategies like add/remove words, word substitution, acronym expansion, and spelling correction are more likely to cause clicks, especially on higher ranked results. In contrast, users often click the same result as their previous query or select no results when forming acronyms and reordering words. Perhaps the most surprising finding is that some reformulations are better suited to helping users when the current results are already fruitful, while other reformulations are more effective when the results are lacking. Our findings inform the design of applications that can assist searchers; examples are described in this paper.

[16]A significant way to improve users' contentment in Web search is to assist them by issuing more valuable queries. One such approach is query reformulation, which generates new queries according to the existing query issued by users. A common method for conducting reformulation is to produce some candidate queries first, then a scoring method is engaged to assess these candidates. At present, most of the offered methods are context based. They depend heavily on the context relation of terms in the history queries and cannot notice and preserve the semantic uniformity of queries. In this article, we propose a graphical model to achieve queries. The proposed model exploits a latent topic space, which is repeatedly derived from the query log, to detect semantic dependency of terms in a query and reliance among topics. Meanwhile, the graphical model also captures the term context in the the past query by skip-bigram and n-gram language models. In addition, our representation can be easily extended to consider users' history search interests when we carry out query reformulation for different users. In the task of candidate query generation, we examine a social tagging data resource—Delicious bookmark—to generate adding up and replacement patterns that are employed as supplements to the patterns generated from query log data.

CONCLUSIONAND FUTURE WORK

In this paper, we learn the problem of finding query facets. We propose a methodical key, which we refer to as QDMiner, to involuntarily mine query facets by aggregating recurrent lists from free text, HTML tags, and repeat regions inside top search results. We generate

two human annotated data sets and pertain existing metrics and two new joint metrics to evaluate the superiority of query facets. Experimental results show that helpful query facets are mined by the approach. We further scrutinize the problem of duplicated lists, and find that facets can be enhanced by modeling fine-grained similarities among lists within a facet by comparing their similarities.

As the first approach of finding query facets, QDMiner can be bettered in many aspects. For example, some semi supervised bootstrapping list extraction algorithms can be used to repeatedly extract more lists from the top results. Specific website wrappers can also be engaged to extract high-quality lists from reliable websites. Adding these lists may develop both accuracy and recall of query facets. Part-of-speech information can be used to further ensure the homogeneity of lists and improve the quality of query facets. We will discover these topics to purify facets in the future. We will also inspect some other associated topics to finding query facets. Superior descriptions of query facets maybe helpful for users to improved understand the facets. Automatically produce meaningful descriptions is an fascinating research topic.

ACKNOWLEDGMENT

This job was supported by the National Key Basic Research Program (973 Program) of China under grant No.2014CB340403, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China No. 15XNLF03, and the National Natural Science Foundation of China (Grant No. 61502501).

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, 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 SIGIRConf. Res. Develop. Inf. Retrieval, 2003, pp. 88–95.

[13] S. Riezler, Y. Liu, and A. Vasserman, "Translating queries into Comput. Ling., 2008, pp. 737–744.

[14] L. Bing, W. Lam, T.-L. Wong, and S. Jameel, "Web query Reformulation via joint modeling of latent topic dependency and term context," ACM Trans. Inf. Syst., vol. 33, no. 2, pp. 6:1–6:38, eb. 2015.

[15] R.Baeza-Yates, C.Hurtado, and M. Mendoza, "Query Recommendation using query logs in search engines," in Proc. Int. Conf. Current Trends Database Technol., 2004, pp. 588–596.

[16] Z. Zhang and O. Nasraoui, "Mining search engine query logs for 2006, pp. 1039–1040.