ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 6, Issue 1, January 2017

Mining Facets For The Searched Queries: A Review

Mr. Sanket Dnyaneshwar Bankar
Computer Science and Engineering
H.V.P.M.'s College of Engineering & Technology, Amravati.

Prof. P. L. Ramteke
Information Technology
H.V.P.M.'s College of Engineering &Technology, Amravati.

Abstract:-

The process of finding query facets which are in the form of multiple groups of words Or phrases will be address as a problem to explain and summarize the content covered by a query. It is assumed 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. To assist information for finding faceted queries, a technique is explore that represents interesting facets of a query using groups of semantically related terms extracted from search results. Web search queries are often multi-faceted, which makes a simple ranked list of results inadequate. So, a method is used, 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. Search results based on used method will simply improve the efficiency of users' ability to find information easily.

Index Terms—Query Facet, Faceted Search, Summarization

I. INTRODUCTION

A query facet is a set of items which describe and summarize 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. For example facets for the query "watches" cover the knowledge about watches in five unique aspects, including brands, gender categories, supporting features, styles, and colors. Query facets provide interesting and useful knowledge about a query and thus can be used to improve search experiences in many ways. In this work, we attempt to extract query facets from web search results to assist information finding for these queries. We define a query facet as a set of coordinate terms { i.e., terms that share a semantic relationship by being grouped under a more general a "relationship". 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. For example, a user could learn different brands and categories of watches. We can also implement a faceted search [1], [2], [3] based on the mined query facets. 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.

II. PROPOSED WORK

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. Furthermore, different websites may publish content using the same software and the software may generate duplicated lists in different websites.

III. MODULE

1. List and context extraction

Lists and their context are extracted from each document in R. "men's watches, women's watches, luxury watches," is an example list extracted. From each document d in the search result set R, we extract a set of lists Ld from the HTML content of d based on three different types of patterns, namely free text patterns, HTML tag patterns, and repeat region patterns. For each extract list, we extract its container node together with the previous and next sibling of the container node as its context. We define that a container node of a list is the lowest common ancestor of the nodes containing the items in the list. List context will be used for calculating the degree of duplication between lists.

2. List weighting

All extracted lists are weighted, and thus some unimportant or noisy lists, such as the price list "299.99, 349.99, 423.99 . . ." that occasionally occurs in a page, can be assigned by low weights. Some of the extracted lists are not informative or even useless. Some of them are extraction errors

3. List clustering

Similar lists are grouped together to compose a facet. For example, different lists about watch gender types are grouped because they share the same items "men's" and women's". We do not use individual weighted lists as query facets because:

(1) An individual list may inevitably include noise. For example, the first item of the first list in Table 2, i.e., "watch brands", is noise. It is difficult to identify it without other information provided;

(2) An individual list usually contains a small number of items of a facet and thus it is far from complete;

(3) Many lists contain duplicated information. They are not exactly same, but share overlapped items. To conquer the above issues, we group similar lists together to compose facets.

After the clustering process, similar lists will be grouped into a candidate query facet.

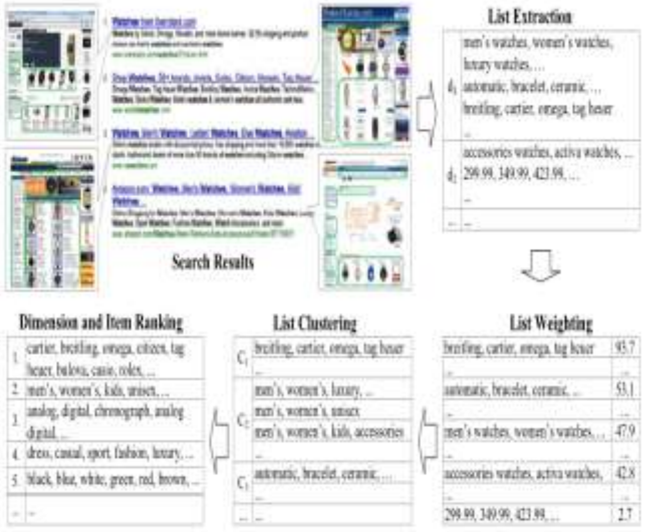
List Extraction
d_1
men's watches, women's watches, luxury watches, ...
automatic, bracelet, ceramic, ...
breitling, cartier, omega, tag heuer
d_2
accessories watches, activa watches, ...
299.99, 349.99, 423.98, ...
Search Results
Dimension and Item Ranking
1 cartier, breitling, omega, citizen, tag heuer, bulova, casio, rolex, ...
2 men's, women's, kids, unisex, ...
3 analog, digital, chronograph, analog digital, ...
4 dress, casual, sport, fashion, luxury, ...
5 black, blue, white, green, red, brown, ...
List Clustering
C_1 breitling, cartier, omega, tag heuer
C_2 men's, women's, luxury, ...
men's, women's, unisex
men's, women's, kids, accessories
C_3 automatic, bracelet, ceramic, ...
List Weighting
breitling, cartier, omega, tag heuer 95.7
automatic, bracelet, ceramic, ... 53.1
men's watches, women's watches, ... 47.9
accessories watches, activa watches, 42.8
299.99, 349.99, 423.98, ... 2.7

Fig.1 System overview of QDMiner

4. Facet and item ranking

Facets and their items are evaluated and ranked. For example, the facet on brands is ranked higher than the facet on colors based on how frequent the facets occur and how relevant the supporting documents are. After the candidate query facets are generated, we evaluate the importance of facets and items, and rank them based on their importance. Based on our motivation that a good facet should frequently appear in the top results, a facet c is more important if:

(1) The lists in c are extracted from more unique content of search results; and

(2) The lists in c are more important, i.e., they have higher weights. Here we emphasize "unique" content, because sometimes there are duplicated content and lists among the top search results. we estimate the degree of duplication between two lists based on the similarity of their contexts but not the entire pages.

IV. MODELS USED

1. Unique Website Model

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 [4], [5]. 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 republished by other websites, hence the same lists contained in the content might appear multiple times in different websites. Furthermore, different websites may publish content using the same software and the software may generate duplicated lists in different websites. Ranking facets solely based on unique websites their lists appear in is not convincing in these cases.

2. Context Similarity Model

Hence we propose the Context Similarity Model, in which we model the fine-grained similarity between each pair of lists. More specifically, we estimate the degree of duplication between two lists based on their contexts and penalize facets containing lists with high duplication

V. QD MINER

It is observe that important pieces of information about a query are usually presented in list styles and repeated many times among top retrieved documents. So we propose aggregating frequent lists within the top search results to mine query facets and implement a system called QDMiner**.** It discovers query facets by aggregating frequent lists within the top results.

We propose this method because:

(1) Important information is usually organized in list formats by websites. They may repeatedly occur in a sentence that is separated by commas, or be placed side by side in a well-formatted structure (e.g., a table). This is caused by the conventions of webpage design. Listing is a graceful way to show parallel knowledge or items and is thus frequently used by webmasters.

(2) Important lists are commonly supported by relevant websites and they repeat in the top search results, whereas unimportant lists just infrequently appear in results. This makes it possible to distinguish good lists from bad ones, and to further rank facets in terms of importance. It automatically mine query facets by aggregating frequent lists from free text, HTML tags, and repeat regions within top search results.

VI. CONCLUSION

In this paper, we studied the problem of extracting query facets from search results. We developed a supervised method based on a graphical model to recognize query facets from the noisy facet candidate lists extracted from the top ranked search results. We proposed two algorithms for approximate inference on the graphical model. We designed a new evaluation metric for this task to combine recall and precision of facet terms with grouping quality. Experimental results showed that the supervised method significantly out-performs other unsupervised methods, suggesting that query facet extraction can be effectively learned.

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 Proceedings of WSDM '08, 2008.

[2] M. Diao, S. Mukherjea, N. Rajput, and K. Srivastava, "Faceted search and browsing of audio content on spoken web," in Proceedings of CIKM '10, 2010.

[3] D. Dash, J. Rao, N. Megiddo, A. Ailamaki, and G. Lohman, "Dynamic faceted search for discovery-driven analysis," in CIKM '08, 2008**.**

[4] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, "Facetedpedia: dynamic generation of query-dependent faceted interfaces for wikipedia," in Proceedings of WWW '10. ACM, 2010.

[5] W. Dakka and P. G. Ipeirotis, "Automatic extraction of useful facet hierarchies from text databases," in Proceedings of ICDE '08, 2008, pp. 466–475.

[6] M. Mitra, A. Singhal, and C. Buckley, "Improving automatic query expansion," in Proceedings of SIGIR '98.

[7] P. Anick, "Using terminological feedback for web search refinement: a log-based study," in Proceedings of SIGIR '03.

[8] S. Riezler, Y. Liu, and A. Vasserman, "Translating queries into snippets for improved query expansion," in Proceedings of COLING '98, 2008, pp. 737–744.

[9] 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.

[10] 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, Feb. 2015.\

[11] J. Huang and E. N. Efthimiadis, "Analyzing and evaluating query reformulation strategies in web search logs," in Proceedings of CIKM. New York, NY, USA: ACM, 2009, pp. 77–86.

[12] R. Baeza-Yates, C. Hurtado, and M. Mendoza, "Query recommendation using query logs in search engines," in Proceedings of EDBT'04, 2004, pp. 588–596.

[13] Z. Zhang and O. Nasraoui, "Mining search engine query logs for query recommendation," in Proceedings of WWW '06, 2006