Automatically Mining Facets for Queries from Their Search Results

G. ANITHA REDDY1, K. HIMAJA2, SHAIA ASMA RIFAT3, D. JAYANARAYANA REDDY4, DR. S. PREM KUMAR5

1Research Scholar, Dept of CSE, G. Pulliaah College of Engineering & Technology, Kurnool, AP, India, E-mail: honeyanithareddyg@gmail.com.
2Research Scholar, Dept of CSE, G. Pulliaah College of Engineering & Technology, Kurnool, AP, India, E-mail: himahimaja6@gmail.com.
3Research Scholar, Dept of CSE, G. Pulliaah College of Engineering & Technology, Kurnool, AP, India, E-mail: asmarifath@gmail.com.
4Assistant Professor, Dept of CSE, G. Pulliaah College of Engineering & Technology, Kurnool, AP, India, E-mail: djnreddy@gmail.com.
5Professor, Dept of CSE, G. Pulliaah College of Engineering & Technology, Kurnool, AP, India, E-mail: csehod@gpcet.ac.in.

Abstract: We address the problem of finding query facets which are multiple groups of words or phrases that explain and summarize the content covered by a query. 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 QD Miner, to automatically mine query facets by extracting and grouping frequent lists from free text, HTML tags, and repeat regions within top search results. Experimental results show that a large number of lists do exist and useful query facets can be mined by QD Miner. We further analyze the problem of list duplication, and find better query facets can be mined by modeling fine-grained similarities between lists and penalizing the duplicated lists.

Keywords: HTML, QD Miner.

I. INTRODUCTION

We address the problem of finding query facets. 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. Table 1 shows sample facets for some queries. Facets 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. For example, a user could learn different brands and categories of watches. We can also implement a faceted search [1], [2], [3], [4] 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 2. attractions, shopping, dining, nightlife, tours, tip, transport, . . . 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 [5], [6], [7]. 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 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 rank 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 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. 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. Compared to previous works on building facet hierarchies [1], [2], [3], [8], [9], our approach is unique in two aspects:

- **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.
- **Query Dependent**: instead of a fixed schema for all queries, we extract facets from the top retrieved documents for each query. As a result, different queries may have different facets. E.g. query “watches” and query “lost” have totally different query facets, as shown in Table 1.

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 results ranked 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. 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.

Disadvantages: The problem of finding query facets.

III. 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 as shown in
Automatically Mining Facets for Queries from Their Search Results

Fig.1. 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 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.

Advantages:
- We propose the Context Similarity Model, in which we model the fine-grained similarity between each pair of lists.
- Query dependent is instead of a fixed schema for all queries; we extract facets from the top retrieved documents for each query.

IV. MODULES

- List and context extraction
- List weighting
- List clustering
- Facet and item ranking

A. Module Description

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.

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.

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”.

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.

V. CONCLUSION

In this paper, we study the problem of finding query facets. We propose a systematic solution, which we refer to as QDMiner, 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 lists within a facet by comparing their similarities. We have provided query facets as candidate subtopics in the NTCIR-11 IMine Task [39]. As the first approach of finding query facets, QDMiner can be improved in many aspects. For example, some semi-supervised bootstrapping list extraction algorithms can be used to iteratively extract more lists from the top results. Specific website wrappers can also be employed to extract high-quality lists from authoritative websites. Adding these lists may improve both accuracy and recall of query facets.

Partof-speech information can be used to further check the homogeneity of lists and improve the quality of query facets. We will explore these topics to refine facets in the future. We will also investigate some other related topics to finding query facets. Good descriptions of query facets may be helpful for users to better understand the facets. Automatically generate meaningful descriptions is an interesting research topic.

V. REFERENCES