ISSN XXXX XXXX © 2018 IJESC

Research Article Volume 8 Issue No.1

Review Paper on Implementation of Mining Facets Automation for the Searched Queries

Shraddha Dinanath Londhe[1], Pallavi Dhande[2]
ME Student[1], Assistant Professor[2]
Department of Computer Science and Engineering
Dr. Rajendra Gode Institute of Technology and Research, Amravati, India

Abstract:
Data mining is the process of sorting through large data sets to identify patterns and establish relationships to solve problems through data analysis. Data mining tools allow enterprises to predict future trends. 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. 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 phrase. A query may have multiple facets that summarize the information about the query from different perspectives. 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. Query facets provide interesting and useful knowledge about a query and thus can be used to improve search experiences in many ways. To assist information for finding faceted queries a technique is explore that represent 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 user's ability to find information easily.

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. 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, style and colours.

Example query facets mined by QDMiner Query: Watches

1. cartier, breitling, omega, citizen, tag heuer, bulova, casio, rolex, . . .
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 facets provide interesting and useful knowledge about a query and thus can be used to improve search experiences in many ways. 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 in particular useful for vague or ambiguous queries, such as "apple". We could show the products of Apple Company. In one facet and different types of the fruit apple in another. 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. 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.

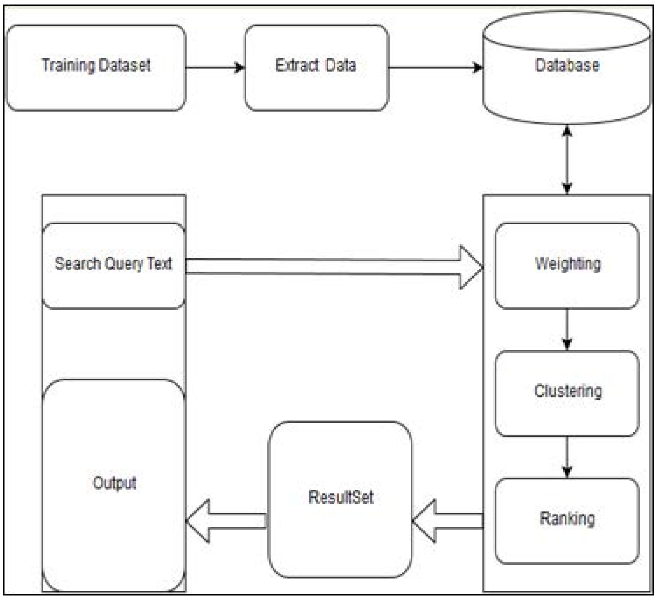
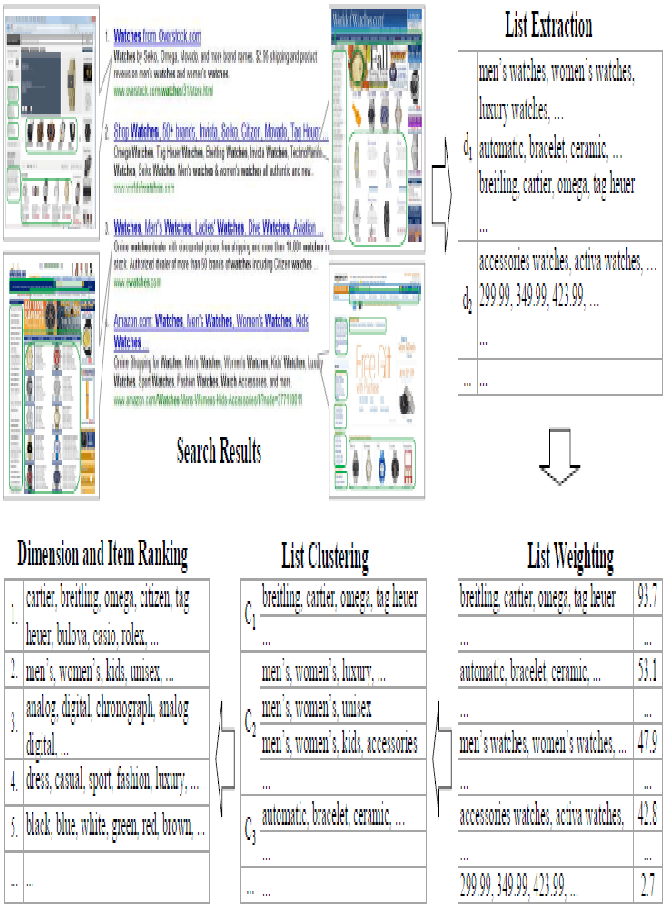


Figure.1. Mining Query Facets



List Extraction
d1
men's watches, women's watches, luxury watches, ...
automatic, bracelet, ceramic, ...
breitling, cartier, omega, tag heuer
...
d2
accessories watches, activa watches, ...
299.99, 349.99, 423.99, ...
...
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
C1 breitling, cartier, omega, tag heuer
C2 men's, women's, luxury, ...
men's, women's, unisex
men's, women's, kids, accessories
C3 automatic, bracelet, ceramic, ...
List Weighting
breitling, cartier, omega, tag heuer 93.7
automatic, bracelet, ceramic, ... 53.1
men's watches, women's watches, ... 47.9
accessories watches, activa watches, 42.8
299.99, 349.99, 423.99, ... 2.7

Figure.2. system overview of QDMiner

QDMiner in Figure 2. In QDMiner, given a query q, we retrieve the top K results from a search engine and fetch all documents to form a set R as input.

Query facets are mined by the following four steps:

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.

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. Table 3 shows some sample lists for the query "watches". The first three lists are navigational links which are designed to help user navigate between WebPages. They are not informative to the query.

TABLE 3
Less informative list examples

	Items (separated by commas)
1	we recommend, my account, help
2	home, customer service, my account, tracking, faq's
3	read, edit, view history
4	movado 605635 luno two tone... 547.50 717.00 1 rating 1 review, movado museum strap 0690299... 225.00 395.00 1 rating, citizen calibre 2100 av0031... 299.00 350.99 11 ratings

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.(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.

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 colours based on how frequent the facets occur and how relevant the supporting documents are. Within the query facet on gender categories, "men's" and "women's" are ranked higher than "unisex" and "kids" based on how frequent the items appear, and their order in the original lists. 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 is more important if: (1) The lists in facet are extracted from more unique content of search results and (2) the lists in facet 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 So by using these four steps, we extract the data or list from website1, website2 and after that all extracted list is weighted and then similar list are group together to compose a facet by using clustering after that we evaluate and rank that facet and their items. In this paper we also implement the impression and click technique. Impression is a measurement of responses from a Web server to a page request from the user browser, which is filtered from robotic activity and error codes, and is recorded at a point as close as possible to opportunity to see the page by the user. An impression (in the context of online advertising) is when an ad is fetched from its source, and is countable. Whether the ad is clicked is not taken into account. Each time an ad is fetched, it is counted as one impression. For example Person begins to type and then clicks anywhere on the page like a search result, ad, or related search Person types a search and then clicks the "Search" button, presses Enter, or selects a predicted query from the drop-down menu Person stops typing, and the results are displayed for a minimum of three seconds You'll sometimes see the abbreviation "Impr" in your account showing the number of impressions for your ad. In this topic by using impression we show most visited items by user in our search engine. Click-through rate (CTR) is the ratio of users who click on a specific link to the number of total users who view a page, email, or advertisement. It is commonly used to measure the success of an online advertising campaign for website as well as the effectiveness of email campaigns.

II. LITERATURE REVIEW:

Mining query facets is hot subject to many existing research topics. In this section, we briefly evaluate and discuss difference from our approach.

Query reformulation and Recommendation

Mainly standard information fetch models use a single source of information for query formulation task such as query weighting etc. The process of query formulation modify the keyword submit by the user to the search engine. The formulated query is used as the input to search engine ranking algorithm. Therefore the main goal of the query reformulation is to get better the overall superiority of the ranking process. For example, for the query "what are the fastest animals in the world". we generate a dimension "cheetah, pronghorn antelope, lion, Thomson's gazelle, wildebeest,..." which includes animal names that are direct answers rather than query reformulations to the query animal names that are straight answers rather than query reformulations to the query. Different from transitional query suggestions, we can make use of query facets to create structured query suggestions, i.e., several groups of semantically related query suggestions. Potentially provides richer information than traditional query suggestions.

Query-based Summarization

Building a summary document satisfying Request for information expressed by a query. The summary is the sequences of sentences which can be extracted from several documents or from single documents. There are number of sources for a summary that are single document summary and multiple document summaries.

There are two types of summary construction methods as follows

- Abstractive
- Extractive

Steps in query based summarization

- Identification of the correct sections
- Summary generation

And might help users find a better query more easily. We will examine the problem of generating query suggestions based on query facets in future work.

Entity Search

The Entity search engines intend at providing the user with entities and relationships between these entities, instead of providing the user with links to web pages.[5],[6],[7] Data is everywhere. The traditional search approach is to verify every document in any order. An entity refers to any object or thing that can be distinctively identified in the world. Each entity is match with other entity.

Benefits of entity search

- It is considered into taxonomy
- The first task is to create a decision
- More ordered than compared to others
- More explicable by human
- amplify precision
- Less time consuming

Query Facets Mining and Faceted Search

Faceted search is a technique for allowing users to digest, analyze, and navigate through multidimensional data. It is widely applied in e-commerce and digital libraries. A robust review of faceted search is beyond the scope of this paper. Most existing faceted search and facets generation systems [1], [2], [3], [8], [9], are built on a specific domain (such as product search) or predefined facet categories. For example, Dakka and Ipeirotis [9] introduced an unsupervised technique for automatic extraction of facets that are useful for browsing text databases. Facet hierarchies are generated for a whole collection, instead of for a given query. Li et al. proposed faceted pedia [8], a faceted retrieval system for information discovery and exploration in Wikipedia. Faceted pedia extracts and aggregates the rich semantic information from the specific knowledge database Wikipedia. In this paper, we explore to automatically find query-dependent facets for open-domain queries based on a general Web search engine. Facets of a query are automatically mined from the top web search results of the query without any additional domain knowledge required. As query facets are good summaries of a query and are potentially useful for users to understand the query and help them explore information, they are possible data sources that enable a general open-domain faceted exploratory search

III. CONCLUSION:

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. 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. 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.

IV. 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 contention 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]. W. Kong and J. Allan, "Extending faceted search to the general web," in Proceedings of CIKM '14, ser. CIKM '14. New York, NY, USA: ACM, 2014, pp. 839–848.

[5]. T. Cheng, X. Yan, and K. C.-C. Chang, "Supporting entity search: a large-scale prototype search engine," in Proceedings Of SIGMOD '07, 2007, pp. 1144–1146.

[6]. K. Balog, E. Meij, and M. de Rijke, "Entity search: building bridges between two worlds," in Proceedings of SEMSEARCH'10, 2010, pp. 9:1–9:5.

[7]. M. Bron, K. Balog, and M. de Rijke, "Ranking related entities: components and analyses," in Proceedings of CIKM '10, 2010, pp. 1079–1088.

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

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

[10]. A. Herdagdelen, M. Ciaramita, D. Mahler, M. Holmqvist, K. Hall, S. Riezler, and E. Alfonseca, "Generalized syntactic And semantic model of query reformulation," in Proceeding of SIGIR'10, 2010.

[11]. M. Mitra, A. Singhal, and C. Buckley, “Improving automatic query expansion,” in Proceedings of SIGIR ’98.

[12]. P. Anick, “Using terminological feedback for web search refinement: a log-based study,” in Proceedings of SIGIR ’03.

[13]. S. Riezler, Y. Liu, and A. Vasserman, “Translating queries into snippets for improved query expansion,” in Proceedings of COLING ’98, 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.

[15]. 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.

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

[17]. 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.

[18]. Z. Zhang and O. Nasraoui, “Mining search engine query logs for query recommendation” in Proceedings of WWW ’06, 2006.

[19]. L. Li, L. Zhong, Z. Yang, and M. Kitsuregawa, “Qubic: An adaptive approach to query-based recommendation,” J. Intell. Inf. Syst., vol. 40, no. 3, pp. 555–587, Jun. 2013.

[20]. I. Szpektor, A. Gionis, and Y. Maarek, “Improving recommendation for long-tail queries via templates,” in Proceedings of WWW. New York, NY, USA: ACM, 2011, pp. 47–56.