IMPROVING SEARCH RESULTS USING FACETS MINING FROM QUERIES

Mr. Sanket D. Bankar
M.E student,
Computer Science & Engineering Dept.
HVPM's COET, Amravati (M.S)

Prof. P L Ramteke
Head of Department, Information Technology
HVPM's COET, Amravati (M.S)

Abstract:

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.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. Search results based on used method will simply improve the efficiency of users' ability to find information easily. 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.

***Index Terms*— Query Facet, Faceted Search, User Behavior.**

1. Introduction

Query facets provide interesting and useful knowledge about a query and thus can be used to improve search experiences in many ways 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.. 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. 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 also implement a faceted search [1], [2] based on the mined query facets 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.

2. Problem Analysis

As the first trial of mining query facets, we propose automatically mining query facets from the top retrieved documents. Query facet extraction is the problem of finding query facets for a given query q from available resources, such as web search results. We will implement a system called QDMiner which discovers query facets by aggregating frequent lists within the top results. This is because important information is usually organized in list formats by websites and they may repeatedly occur in a sentence that is separated by commas, or be placed side by side in a well-formatted structure. This is caused by the conventions of webpage design. Also other reason is 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.

3. Module

1. List and context extraction

List context will be used for calculating the degree of duplication between lists .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..

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 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 .An individual list usually contains a small number of items of a facet and thus it is far from complete. 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.

4. Facet and item ranking

After the candidate query facets are generated, we evaluate the importance of facets and items, and rank them based on their importance. 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. Based on our motivation that a good facet should frequently appear in the top results. 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. The importance of an item depends on how many lists contain the item and its ranks in the lists.

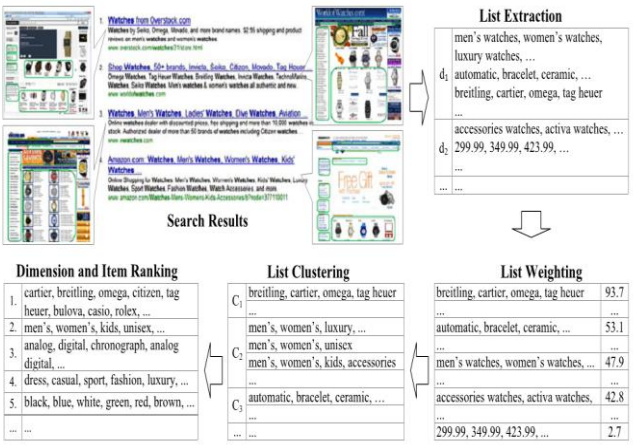

Fig.1 System overview of QDMiner

4. Methods for Evaluation

1. Data

Here we will consider two services for evaluating query facets. The first service implies for finding facets, and invites 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". First we 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. We then aggregate the qualified items in the facets returned by all algorithms we want to evaluate, and ask the subject to assign unlabelled items into the created facets. A facet named "misc" is automatically created for each query to
help subjects to distinguish between bad and items that are not judjed. During evaluation, "misc" facets are discarded before mapping generated facets to manually labelled facets. The rating that is most chosen by subjects is regarded as the final rating of the facet. The higher one is used if two ratings get the same number of votes. The ratings for "misc" facets are automatically set to "Bad" by default and subjects are unable change them.

2. Evaluation Metrics

We have to measure the quality of query facets in two aspects:-

2.1 Quality of clustering- Firstly each facet should only contain items reflecting the same facet of the query, and the items referring to the same information should not be separated into multiple facets. Here we will use some existing metrics [3] such as Purity, NMI (Normalized Mutual Information), RI (Random Index), and F measure, to evaluate the quality of clusters. Clusters will contain the different elements of a similar item. Because of this
We will get the appropriate cluster from its quality.

2.2 Ranking effectiveness of facets- This method is to check, how a facet ranking will influence the readers mind to know he actual status of product item. We aim
to rank good facets before bad facets when multiple
facets are found. We use the nDCG measure (Normalized Discounted Cumulative Gain), which is widely used in information retrieval, to evaluate the ranking of query facets.

5. Models Used

1. Unique Website Model

This model mainly used to avoid the duplication of data. In the Unique Website Model, we used websites a simple signal for creating groups. Because a same website usually delivers similar information, multiple lists from a same website within a facet are usually duplicated. 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

In this section, we want to further explore better ways for modeling the duplication among lists for weighting facets. 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. we do find the dependence between some websites and the lists from these websites are sometimes duplicated, including but not limited to the cases such as **Mirror websites** contains different domain names but contain same contents. Secondly **content republishing** where contents of the a website which is original in nature, are copied or republished by other website by making small changes or as it is. See Fig Below

Fig: - An example of copied pages

Also Different websites may publish content using the **same software.**

6. Conclusion

This paper results in evaluation of applied models in order to make search results faster and reliable to the user. In this paper, we studied the problem of extracting query facets from search results. We studied evaluation metric for this task to combine recall and precision of facet terms with grouping quality 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. Experimental results showed that the supervised method significantly out-performs other unsupervised methods, suggesting that query facet extraction can be effectively learned.

7. 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] C. D. Manning, P. Raghavan, and H. Schtze, Introduction to Information Retrieval. Cambridge University Press, 2008.

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

[5] 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.\

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

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

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

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

[10] 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..

[11] 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

[12] 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.

[13] 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.

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