IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Extracting Facets For Queries From Search Results

[1]Minu Augustine

[1]Assistant Professor

[1]Department of Computer Science & Engineering,

[1]Nehru College of Engineering & Research Centre

***Abstract:*** A query facet is a set of items that describes an important aspect of an inquiry. We address the matter of finding inquiry facets and link to the specified web page. In order to solve this problem, we introduce an efficient solution refer to as QDMiner, which discovers query items by aggregating frequent lists restrain in top search results. We further examine the issue of list duplication and discover the better query aspects can be mined by displaying fine-grained similarities among records and penalizing the copied records. We introduce a website wrapper called Anchor based data extraction to increase the quality of extracted list. From the generated facets users can go to the desired high ranking pages by selecting the item in the facets. Search based on this method will improve the efficiency of user's ability to search the information easily.

***Index Terms*** **– Facet, Faceted search, Summarization, Website Wrapper**

## I. INTRODUCTION

A query may have variety of facets that encapsulate the significant data concerning about the query from alternate view inquiries. The process of finding query aspects which are in the form of multiple groups of words or phrases are address as a problem to explain and summarize the content covered by an item. Query facets provide fascinating and useful knowledge about a inquiry and thus can be used to accustomed upgrade search experiences. Users will understand some prime aspects of an inquiry while not browsing tens of pages. Query aspects might give direct information or instant answers that users are seeking. The items are generated supported the users style. The user can also go to the specified web page by choosing the inquires from the facets. So, it saves the user's time wasted on sort out the information in tens or thousands of pages. We implemented a faceted search [1], [2] related to the query. User can clarify the selecting facet items and the search results could be restricted to the relevant documents. The figure 1 shows that sample items for some inquires. The query "watches" has a query aspect about the main information to be specified.

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: **lost**
1. season 1, season 6, season 2, season 3, season 4, season 5
2. matthew fox, naveen andrews, evangeline lilly, josh holloway, ...
3. jack, kate, locke, sawyer, claire, sayid, hurley, desmond, boone, ...
4. what they died for, across the sea, what kate does, the candidate, ...

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,...
3. matthew fox, naveen andrews, evangeline lilly, jorge garcia, ...
4. season 1, season 3, season 2, season 6, season 4

Query facets also contain ordered information covered by the query, and thus they can be utilized in other fields besides conventional web search, such as semantic search or entity search. We observe that significant pieces of data about an inquiry are usually presented in list styles and repeated many times among top retrieved documents. In the QDMiner method, to automatically mine query aspects by extracting and aggregating frequent lists within the highest search results.

More precisely, QDMiner extracts lists from free text, HTML tags, and frequent patterns contained in the highest search outcome, groups them into clusters supported the objects they contain, then ranks the clusters and objects supported however the lists and objects seems within the high results. Here we use two models, the Distinctive website Model and therefore the Context Similarity Model, to rank query facets. In the Distinctive Website Model, we presume that lists from the similar website might contain same information, while different websites are self-governing and each can give a divided vote for weighting aspects. However, we discover that occasionally two lists can be same data, still if they are from unlike websites. For example, mirror websites are using variety of 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 supported and the software may generate duplicated lists in different websites.

## II. RESEARCH METHODOLOGY

Mining query aspects is identified for the several existing methodologies. In this area, we briefly survey them and discuss the advantages and importance for our approach.

### *2.1 Query Facet Mining and Faceted Search*

Faceted search is a system for permitting users to digest, explore and analyze through multidimensional information. It is generally applied in e-commerce and computerized libraries. A robust view of faceted pursuit is past the area of importance for this paper. Most existing faceted search [4], [5] and features era frameworks are built on a specific domain (like item look) or predefined aspect categories [12]. For example, Dakka and Ipeirotis presented an unattended procedure for automatic extraction of aspects that are valuable for browsing content databases. Facet hierarchies are manipulated for a rather assortment, instead for a given inquiry. Li et al. proposed Facetedpedia, a faceted retrieval framework for data discovery and investigation in Wikipedia. Facetedpedia concentrates and aggregates the main semantic details from the specific knowledge database Wikipedia. In this work, here discover the naturally inquiry query-dependent facets for open-domain inquiries based on a general Web Internet searcher. Aspects of an inquiry are naturally mined from the top list search results of the query with no extra space domain information required. As query facets are great outlines of an inquiry and are potentially valuable information for users to understand the query and help them investigate information, they are conceivable data sources that enable a general open-area faceted exploratory search. Like us, Kong and Allan [6] as of late built up a supervised approach related on a graphical model to mine query aspects. The graphical model figures out how likely a candidate term is to be an aspect item and how likely two terms are to be gathered composite in a facet [7].

### *2.2. Query Reformulation and Recommendation*

Query reformulation and query recommendation (or query suggestion) are two famous approaches to help clients better analyze their data need. Query reformulation is the path toward of changing an inquiry that can better match a user's data need [8], and query suggestions techniques produces alternative inquiries semantically similar like the same query [9], [13]. The main goal of mining features is not quite the same as from query recommendation. The previous is to outline the knowledge and data accumulated in the query, while the last is to find a list of related or expanded inquiries. However, query facets incorporate semantically related expressions or terms that can be utilized as inquiry reformulations or inquiry suggestions now and again. Different from transitional query suggestions, we can utilize query facets to generate structured query suggestions, i.e., various groups of linguistics related query suggestions. This potentially provides richer information than traditional query suggestions and might help users find a better query more effectively.

### *2.3. Web Data Extraction*

The World Wide Web data contains a large content of unstructured and semi-structured data that is exponentially increasing with the coming of the Web 2.0 [9]. HTML was really designed to display information to a human user, so applications HTML wrappers that can make the content of HTML pages directly available to them. The purpose of the World-Wide Web Wrapper is the rapid design, generation, and integration in applications of such wrappers. Specifically, the key features are: fully declarative specifications, light- weight components, rapid development, robustness, direct integration into Java programs and re-usability. Some people have already argued that there is no more need for HTML wrappers because data sources will soon serve XML documents [10]. In fact, there already are countless HTML pages on the Web and the information that many of them contain will have to be displayed in XML in a relatively near future. More generally, we believe light-weight HTML wrappers are currently indispensable for Web interoperation and Web information integration. In particular, such wrappers turn out to be also an excellent tested for the construction of smarter customized applications for e-commerce, digital libraries, etc.

### *2.4 VIPS: A Vision-based Page Segmentation*

Recently the Web has become the largest information source for people. Mostly the information retrieval framework on the Web mainly web pages as the smallest and undividable units, but a web document as a whole may not be suitable to define a single semantic. A web page usually contains different contents such as navigation, interaction and contact data, which are not combined to the choice of the web-page. Furthermore, a web page usually contains different type of topics that are not necessarily related to each other. Therefore, detecting the semantic items in a structure of a web page could potentially increase the performance of web data retrieval. In this work, here describes an approach because the VIPS (Vision-based Page Segmentation) algorithmic program [11] to extract the linguistics structure for online page content. Such linguistics field structure is a hierarchical structure with in which each node will correlated to a block. Every node frame are outlined a worth (Degree of Coherence) to denoted that however coherent of the content within the block related on manual perception. The VIPS algorithmic program makes full type of page layout frame: it first emerged all the related blocks from the HTML DOM tree, and so it tries to search out the separators between these contained parts. Here, separators defined the horizontal or vertical lines in an online page that semantically cross with no parts. Finally, supported these separators, the linguistics structure field for the online page is constructed. VIPS algorithm emerged a top-down approach, which is very effective and maintainable. The algorithm is evaluated manually on a big data set, and also used for selecting good expansion terms in a pseudo-relevance feedback process in web data retrieval, both of which achieve very performance.

## III. MINING QUERY FACETS

In the QDMiner method, When an inquiry is given then from the search results the lists are obtained using the list content algorithm, all obtained items are given weights, unused or noisy items that sometimes happens in a page are assigned by small weights. Same categories of items are sorted into clusters using the weighted Threshold Algorithm. Facets are then evaluated and ranked. High rank is given for lists extracted from similar context and lists having higher weight. We tend to address the matter of finding inquiry facets and link to the specified web page. In the list content method extracts low quality lists. So, here we propose a website wrappers are introduced to extract the high quality lists.

## IV. SYSTEM ARCHITECTURE

In QDMiner, given a query *q* we retrieve the top *k* search results from the search engine. Then collect all documents to form a combination of input. Query items can be mined by using 4 steps.
The overall structure for our system is demonstrated in Figure 2.

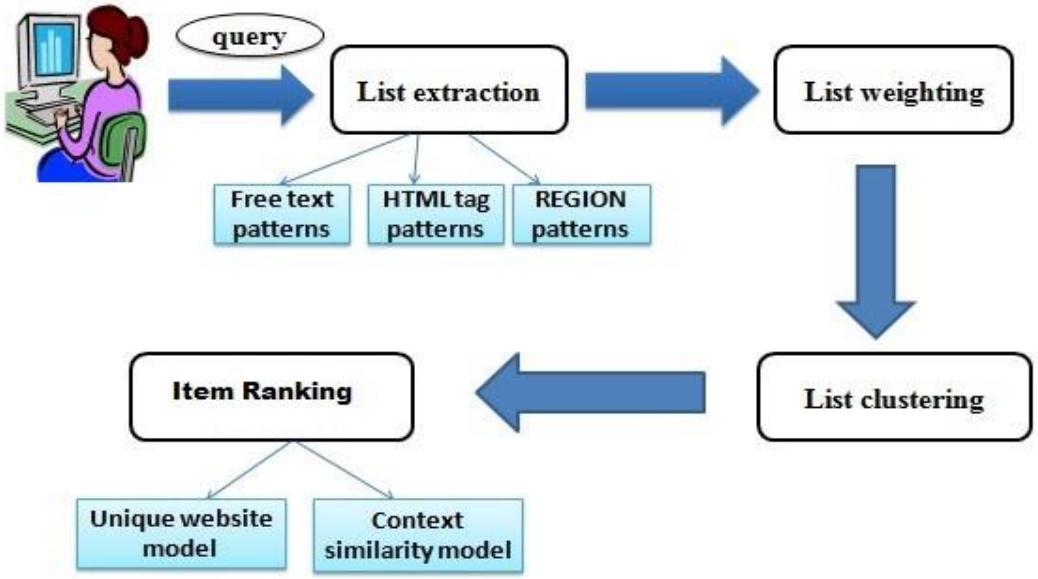


Fig.2 Process Flow

### *4.1. List and context Extraction*

From each document in the search result set to extract a combination of lists from the HTML content of based on three different types of patterns, namely text patterns, HTML tag patterns, and same 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 extract all items in italic font and also extract lists from continuous lines of two parts separated by colon or dash. In HTML tag patterns, we extract items from several list-style tags including SELECT, UL, OL, TABLE. The Region-based pattern in a webpage extracts all leaf HTML nodes within every block and group them by their styles. List content will be used for determining the duplication between items in each list To increase the quality of extracted list here we used the website wrapper method. A wrapper is software that turns an online source into a place that can be inquired as a database. Website wrappers are used to accustom acquire the structured data from a knowledge regions, objects and relations. Here we tend to use an online wrapper called Anchor related information extraction. A website wrapper can tolerate any change in the internal structure of the web page items like the changes made by developers, page link. Website wrapper can be adapted to similar paper to gather and extract sources from web pages. An online wrapper is created by identifying anchors on the typical web page. Then it is saved as a file for extracting data from similar pages on the website. An anchor is a textual field that marks the start or destination of a data region or as a keyword in a data region that distinguishes it from the other of the pages, like the title, highlighted words, constants, keywords. For making anchors the sample page is loaded into the required web browser. The element of the page gets highlighted and the anchor is created related on the elements features.

We further process each extracted items by avoiding the useless symbols, remove the stop words and then converting uppercase to lowercase.

### *4.2. List Weighting*

Some of the extracted items are not useful or even noisy. Some of them are content errors. The lists may be navigational links which are enveloped to help the clients navigate between web pages. They are not useful to the query. Several types of information are mixed together. Thus, to penalize these lists and rely more on better lists to generate good aspects. We discover that a better list is usually supported by many websites and appear in many documents, partially or exactly. Evaluate the unique items in each document by using the

document matching weight. The document matching weight can be calculated by using the tf-idf method. Finally, we sort all items by final weights for the given query.

*4.3. List Clustering*

Group the similar lists together to form a cluster. To compute the distance between two clusters of lists, when every two lists of them are similar enough. We use a weighted quality threshold algorithm (WQT) that groups data into high quality clusters. This method prevents dissimilar data under the same dimension and ensures high cluster quality. In WQT, the number of clusters is not required to be specified.

1) Initialize the edge distance allowed for clusters and therefore the minimum cluster size.

2) Build a candidate cluster for every data points by together with the highest point, the consequent related point and then on, until the space of the cluster surpasses the edge.

3) Save the candidate cluster with the foremost points because the first true cluster and take away all points within the cluster from further considerations.

4) Repeat with the every set of points till no additional cluster may be fashioned having the minimum cluster size.

The weight of a cluster is often calculated supported the quantity of websites from which its items are extracted. After the clustering process, same items will be sorted into a candidate query facet.

*4.4. Item Ranking*

After the candidate query facets are generated, to evaluate the main content of aspects and items, and rank they supported their importance. The better items should sequentially seem within the main results, a aspect is additional necessary if the lists are extracted from additional distinctive content of search results. Here we emphasize "unique" content; as a result of generally there are duplicated content and items among the top search results. We estimate the degree of duplication between two lists supported on the similarity of their contexts but not the entire pages.

The two models, the distinctive website model and the context similarity model, to rank the query facets

*1)* Distinctive Website Model: A same website usually deliver similar information, multiple lists from a same website within an aspect are usually duplicated. A simple method for dividing the lists into different groups is checking the websites they belong to. And to assume that different websites are independent and each distinct website has one and only one separated vote for weighting the facet.

*2)* Context Similarity Model: To further explore better ways for modeling the duplication among lists for weighting inquires. Here the similarity is mostly about the duplication between two lists, in terms of whether two lists are representing dependent sources, while the original similarity used for clustering lists into facets are mainly about whether two lists are about same type of information, and whether they contain be in a same facet. For example, mirror websites are using variety of domain names but are publishing duplicated content.

List Duplication Estimation: The text obtained in the context to analyze the list similarity. There are several ways to measure the similarity between two pieces of text. Here we use the SimHash [14] method to first encode each lists into a 64-bit fingerprint. To extract all lists and their items contained in all documents, and building their fingerprints into index with low space value in search engines. Similarity between two lists $l_1$ and $l_2$ is then estimated based on Hamming Distance *dist* ($l_1$ , $l_2$) between the fingerprints of their items.

$$Dup_L(l_1, l_2) = 1 - \frac{dist(l_1, l_2)}{LS} \qquad (1)$$

Where *LS* is the length of fingerprint used and *LS* = 64.

In a facet, the importance of an item depends on how many lists contain the item and position of the facets in the lists. The weight contributed by a group lists and the average rank of item within all lists extracted from group. To sort all items within a facet by their weights and to define an item is a qualified item of aspect.

## V. RESULTS AND ANALYSIS

The evaluation of the system can be measured in terms of accuracy and precision. In the existing approach, the list content method extracts the low quality lists. If there is any change in the internal structure of the webpage like the page link or the addition of items to the website may produce low quality lists. So, here we introduce the website wrapper method to extract high quality lists from reliable websites. Adding these lists may increase both accuracy and precision of query facets. The experiment is performed real time in Bing using the Bing search engine. When the inquiry is given then from the search results to be mined the facets. From the generated items users can go to the related high ranking pages by choosing the lists in the facets.

*Evaluation based on Accuracy of results*

The proposed QDMiner method is evaluated on the basis of accuracy that is the results obtained for same set of queries are evaluated. In the existing system evaluated and the search result obtained from the same query with our query approach. For the purpose of analysis the top most facets and their links are analyzed. The result obtained is shown in figure 3. All the analysis results in the aspects of accuracy, precision and execution time shows that our system works with much more precision.

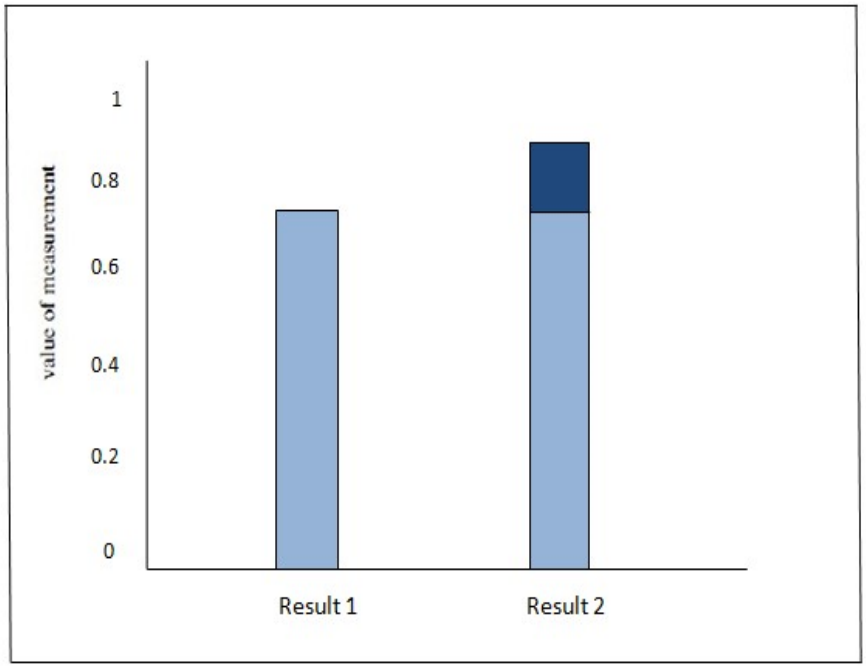


Fig.3 Comparison graph of search result quantity

## VI. CONCLUSIONS AND FUTURE WORKS

This work presents a methodology of generating meaningful items from the user query search results. The facets here are generated using four steps; List content extraction, list weighting, list clustering and list ranking. In the list extraction step, the specific website wrapper method is also used for extract high-quality items from reliable websites. Adding these lists may develop both accuracy and recall of query facets. Then from these generated patterns can be weighted and similar lists to be clustered. Finally, the item ranking can be done. Search based on this method can extract high quality lists from the top k query search results. Hence these high quality items can be used to generate meaningful lists. These inquires are generate related on the user's interest. User can navigate to the related page by choosing the facet on the item list to get detailed information. The future work possible in this method includes the Part-of-speech data content can be used to check the homogeneity of lists and improve the quality of query facets. Semi-supervised bootstrapping list extraction algorithms can be also used to iteratively acquire more lists from the top results.

## VII. REFERENCES

[1] O. Ben-Yitzhak, N. Golbandi, N.Har'El, A. Neumaan, S.Yogev, D. Sheinwald, "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] W. Kong and J. Allan, "Extracting query facets from search results," in Proceedings of SIGIR '13, ser. SIGIR '13. New York, NY, USA: ACM, 2013, pp. 93–102.

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

[8] 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.

[9] E.Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, "Web data extraction, applications and techniques: A survey," Knowledge –Based Systems, Vol. 70, pp.301-323, 2014.

[10] D. Cai, S.Yu, J.R Wen and W.Y Ma, "Vips: A Vision-based Segmentation algorithm," Microsoft technical report, MSR-TR-2003-79, 2003

[11] 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.

[12] S. Basu Roy, H. Wang, G. Das, U. Nambiar, and M. Mohania, "Minimum-effort driven dynamic faceted search in structured databases," in Proceedings of CIKM '08, 2008, pp. 13–22.

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

[14] G. S. Manku, A. Jain, and A. Das Sarma, "Detecting near Duplicates for web crawling," in Proceedings of WWW '07. New York, ACM, 2007, pp. 141-150.