Extracting Facets for Queries from Search Results

Minu Augustine[1*], Vishnu Priya R[2], Chitra S Nair[3]

[1]PG Scholar, Computer Science and Engineering, NSS College of Engineering, Kerala, India
[2]PG Scholar, Computer Science and Engineering, NSS College of Engineering, Kerala, India
[3]Assistant Professor, Computer Science and Engineering, NSS College of Engineering, Kerala, India

Abstract: ***A query facet is a set of items that explains an underlying aspect of a query. To address the issue of finding query facets which are various groups of words or phrases that clarify and summarize the content covered by a query. In order to solve this problem, we propose a systematic solution refer to as QDMiner, which discovers query facets by aggregating frequent lists contained in top search results. We further analyse 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. And also a method for creating web wrappers to extract data records from web pages. In our approach, we depend mainly on the page related content to be independent of the HTML structure. Therefore the generated wrappers are sturdy and easier to maintain and can be adapted to similar websites to collect and integrate information.***

Keywords: Query Facet, Faceted Search, Web Wrapper Generation, Web Information Extraction.

1. Introduction

To address the issue of discovering query facets which are numerous assembling of words and expressions. A query may have multiple facets that describe the significant information about the query from alternate view inquiries. The table 1 shows that sample facets for some queries.

Facets for the query "Watches" spread the information about watches in five different views, including brands, gender categories, supporting components, styles and colours. The query "lost" has a query aspect about the main events to be specified. Query facets provide the useful knowledge about a query aspect and thus can be improve in many perspectives. First, the original result of a query can be displayed in a suitable way. Thus the users can understand the importance of a query without browsing many pages. For example the user could learn search for the different brands and supporting components of watches. 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. Second, the query facets may provide the related information that the users are request. For example the query "lost season 5" all episode titles are viewed in one facet and main actors are shown in another. In this instance, displaying query facets could save their time. Third, the query facets are used to improve the variety of links. We can re-rank the list items to abstain from viewing the pages that are close copied in query facets at the top.

Existing query facet mining algorithms mainly rely on the top search results from search engines. In all these existing solutions, facet items are extracted from the top search results from a search engine (e.g., top 100 search results from Bing.com). More specifically, facet items are extracted from the lists contained in the results. Kong et al. proposed two supervised methods, namely QF-I and QF-J, to mine query facets from search results.

Table 1: 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: **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

The problem is that the coverage of facets mined using this kind of methods might be limited, because some useful words or expressions might not appear in the list within the search results used and them have no chance to be mined. We analysed that the important concepts about query are usually viewed in list styles repeated many times among top retrieved documents. Thus we propose the QDMiner, a system can automatically mine query facets by aggregating the frequent lists by extracting and grouping within the top search results. More specifically the lists are extracted by HTML tags (like <select> and <table>), text patterns and repeat content blocks contained in web pages. Here two models are also being proposed for to rank the inquiry features, the Unique Website Model and the Context Similarity Model. In the Unique Website Model, we expect that rundowns from the same site may contain copied data, while distinctive sites are free and each can contribute an isolated vote in favour of weighting aspects. Be that as it may, we find that occasionally two records can be copied, regardless of the fact that they are from various sites. For example, mirror sites are utilizing diverse area names yet they are distributed copied content and contain the same records. Some substance initially made by a site may be re-distributed by different sites; henceforth

the same records contained in the substance may show up multiple times in various sites. Moreover, distinctive sites may distribute content utilizing the same programming and the product may create copied records in various sites.

Contrasted with past takes a shot at building feature hierarchies [2], [3] our methodology is extraordinary in two perspectives: (1) Open area. We don't confine questions in a particular space, similar to items, individuals, and so forth. Our proposed methodology is bland and does not depend on particular area learning. Along these lines it can manage open-space questions. (2) Query subordinate. Rather than a settled outline for all inquiries, we remove aspects from the top recovered records for every inquiry. Therefore, diverse inquiries may have distinctive aspects.

2. Related Works

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

2.1 Query Facet Mining and Faceted Search

Faceted search is a system for permitting users to digest, explore and analyse through multidimensional information. It is generally applied in e-commerce and computerized libraries. A robust view of faceted pursuit is past the scope of this paper. Most existing faceted search [4], [5] and features era frameworks are built on a specific domain (such as item look) or predefined aspect categories [12]. For example, Dakka and Ipeirotis presented an unsupervised procedure for automatic extraction of aspects that are valuable for browsing content databases. Facet hierarchies are generated for a rather collection, instead for a given query. Li et al. proposed Facetedpedia, a faceted retrieval framework for information discovery and investigation in Wikipedia. Facetedpedia concentrates and aggregates the rich semantic details from the specific knowledge database Wikipedia. In this paper, here discover the naturally inquiry query-dependent facets for open-domain inquiries based on a general Web Internet searcher. Aspects of a query 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 a query 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 based 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 together in a facet. Different from our approach, they used the supervised methods. They encouraged built up a facet search framework based on the mined features [7].

2.2 Query Reformulation and Recommendation

Query reformulation and query recommendation (or query suggestion) are two famous approaches to help users better analyse their data need. Query reformulation is the way 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 fundamental goal of mining features is not quite the same as from query recommendation. The previous is to outline the knowledge and information contained 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 query suggestions now and again. Different from transitional query suggestions, we can utilize query facets to generate structured query suggestions, i.e., various groups of semantically related query suggestions. This potentially provides richer information than traditional query suggestions and might help users find a better query more effectively. We will research the issue of generating query suggestions based on query aspects in future work.

2.3 Web Data Extraction

The World Wide Web contains a large amount of unstructured and semi-structured data that is exponentially increasing with the coming of the Web 2.0. We intend to briefly survey the fields of application, in particular enterprise and social applications, and the methods are used this approach and solve the problem of the extraction of information from Web sources: during last year's many approaches were developed [10], some inherited from past studies on Information Extraction (IE) systems, many others Studied ad hoc to solve the related problems. We can generically define a Web data extraction system as a sequence of procedures that extracts information from Web sources. We can infer the two important aspects of the problem defined as the Interaction with web pages and Generation of a Wrapper. There are some future directions and challenges that can be foreseen. Some of them comprise how to address enormous scaling issues of the extraction problem, the robustness of the process and the design and implementation of auto-adaptive wrappers.

2.4 VIPS: a Vision-based Page Segmentation Algorithm

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 as the VIPS (Vision-based Page Segmentation) algorithm [11] to extract the semantic structure for web page content. Such semantic field structure is a hierarchical structure in which every node will correlated to a block. Each node frame will be defined a value (Degree of Coherence) to denoted that how coherent of the content in the block related on visual perception. The VIPS algorithm makes full use of page layout frame: it first emerged all the related blocks from the HTML

DOM tree, and then it tries to find the separators between these contained blocks. Here, separators defined the horizontal or vertical lines in a web page that semantically cross with no blocks. Finally, based on these separators, the semantic structure field for the web 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.

3. Mining Query Facets

We propose the method QDMiner that discovers frequent lists within the top search results to mine query aspects. More specifically, extracts lists from free text patterns, HTML frames, and repeat regions contained in the top search results, joining them into clusters related on the items they contain, and then ranks the clusters and items based on how the lists and items appear in the top results. We propose two models, for rank the query facets such as the Unique Website Model and the Context Similarity Model. In the Unique Website Model, we assume that lists from the similar website might contain copied information, whereas different websites are independent and each can contribute a separated vote for weighting facets. However, we discover that sometimes two lists can be duplicated, even if they are from different websites. For example, mirror websites are using various domain names but they are publishing copied content and contain the similar lists. So the content originally created by a website might be re-published by other websites. And also a method for creating web wrappers to extract data records from web pages. In our approach, we depend mainly on the page related content to be independent of the HTML structure. Therefore the generated wrappers are sturdy and easier to maintain and can be adapted to similar websites to collect and integrate information.

3.1 System Overview

Methodologies are the process of analysing the principles or procedure for behavioural characterizing of discovering query aspect.

1. List and Context Extraction
2. List Weighting
3. List Clustering
4. Facet and Item Ranking

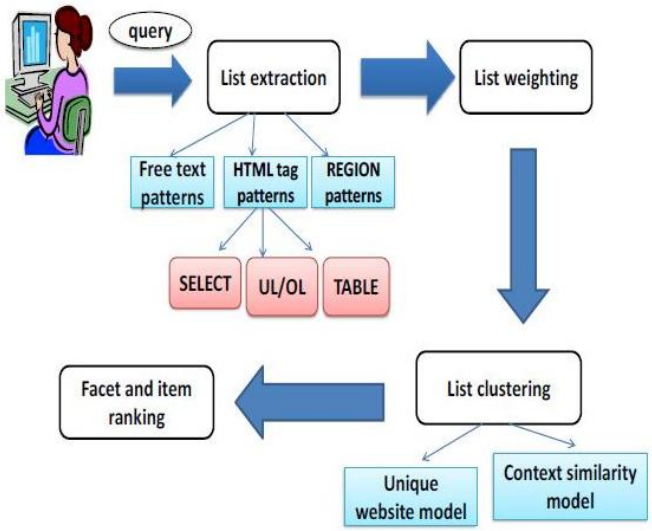

Figure -1: Process Flow

3.1.1 List and Context Extraction

From each document in the search result set to extract a set of lists from the HTML content of 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. Lists and their connection are removed from every record in R. "men's watches, women's watches, extravagance watches" is an illustration list removed.

3.1.2 List Weighting

Some of the extracted lists are not informative or even noisy. Some of them are extraction errors. The lists may be navigational links which are designed to help users navigate between web pages. They are not informative 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 find that a good list is usually supported by many websites and appear in many documents, partially or exactly. A good list contains items that are informative to the query. All extricated records are weighted, and in this manner some insignificant or boisterous records, for example, the value list "299.99, 349.99, 423.99 . . .” that infrequently happens in a page, can be allotted by low weights.

Document matching weight: Items of a good list should frequently occur in highly ranked results. And the document matching weight is the supporting score by the percentage of items contained and measures the importance of document.

Average invert document frequency: A list comprised of common items in a quantity is not informative to the query. Finally, we sort all lists by final weights for the given query. The assigned low weights as they have low average invert document frequencies. Its most items just appear in one document in top results hence it has a low document

matching weight.

3.1.3 List Clustering

To group similar lists together to compose aspects. Two lists can be grouped together if they share enough items. To use the complete linkage distance to compute the distance between two clusters of lists. This means that two groups of lists can only be merged together when every two lists of them are similar enough. List Clustering Similar records are assembled together to com-represent an aspect. Thus, use a modified QT (Quality Threshold) clustering algorithm to group similar lists. QT is a clustering algorithm that groups data into high quality clusters.

A modified QT (quality threshold) algorithm:

1. Merge lists that contain similar items.
2. Find large clusters whose diameters do not exceed a user-defined diameter threshold.
3. Lists with higher weights will be merged first.

3.1.4 Facet and Item Ranking

After the candidate query facets are generated, to evaluate the importance of aspects 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 the lists are extracted from more unique content of search results. Here we emphasize “unique” content, because sometimes there are duplicated content and lists among the top search results. 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. Facets and item ranking facets are evaluated and positioned. For instance, the aspect on brands is positioned higher than the feature on hues in light of how incessant the features happen and how pertinent the supporting records are. Inside the question aspect on sex classes, "men's" and "women's" are positioned higher than "unisex" and "children" in view of how regular the things show up, and their request in the first records.

Unique 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.

Context Similarity Model: To further explore better ways for modelling the duplication among lists for weighting facets. 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 should be in a same facet.

In a facet, the importance of an item depends on how many lists contain the item and its ranks in the lists. As a better item is usually ranked higher by its creator than a worse item in the original list, and to calculate the weight of an item within an aspect. 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.

3.2 Web Wrapper Generation

In our method, we try to pretend the way people mainly look for information on a web page. It mainly rely on the visual hint on the page like colours, text, fonts as well as the keyword constants, highlighted words etc. we defines the textual element with the term anchor. Here the anchor as the textual signal that marks the start or end of a data stream or a pattern within a data record that defines the web page. This approach is mainly build by the web wrapper is used to define the anchor on a typical page by the client; then the wrapper is stored as a file to be used for obtaining information from relative pages on the similar website or different websites. In the following, we define the steps required for web wrapper generation.

3.2.1 Creating Anchors

The client loads a web page into an embedded web browser to define the anchors. To facilitate this piece of work, the web page items get highlighted as the client linger the mouse pointer over them. After selecting the item, it allows for creating an anchor related on the item’s features. The main content may specify an anchor using the terms of some fixed and semantic field attributes such as id, class name and text pattern.

3.2.2 Creating Region Patterns

After creating the anchors, the client must define how the data regions are selected using these anchors. The client originates some region patterns related on one or a group of anchors. Each pattern can be defined either for data obtaining or to avoid the occurrence of the different patterns. The method that relies on the page visible content, same anchors and patterns can be finding in related websites providing the same service. Therefore, the client often needs to follow the similar procedure to create wrappers for same website and it makes creating the wrappers easier.

So, here propose a systematic solution as QDMiner, which discovers query facets by aggregating frequent lists contained in top search results. We further analyse 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. And also a method for creating web wrappers to extract data records from web pages. In our approach, we depend mainly on the page related content to be independent of the HTML structure. Therefore the generated wrappers are sturdy and easier to maintain and can be adapted to similar websites to collect and integrate information.

4. Conclusions

In this paper, we analyse the issue of discovering finding query facets. We propose a precise solution, which we refer to as QDMiner, to consequently mine inquiry facets by amassing successive lists from free text, HTML labels, and repeat regions inside top search results. Exploratory results show that valuable query facets are mined by the methodology. We further analyse the issue of duplicated lists, and find that aspects can be enhanced by demonstrating fine-grained similitudes between records inside a feature by comparing their likenesses. The approach for creating data records from web pages based on the textual anchors within the page content. Anchors contain the key elements within the region patterns and each region pattern contains sub patterns for the regions within it. As proposed method describes mainly on the page content more than the HTML structure, the generated wrappers are strong and easy to maintainable.

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] 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.