ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 6, April 2017

Facets mining from search results using BatchSTS algorithms

Anju G R [1], Karthik M[2]
[1]Student, Mohandas College of Engineering and Technology, Anad
Thiruvanthapuram, Kerala, India
anjugireesan1@gmail.com
[2]Asst Professor, Mohandas College of Engineering and Technology, Anad
Thiruvanthapuram, Kerala, India

***Abstract*—Faceted search help users to navigate through multidimensional information space. It is achieved by combining keyword search with the drill down options in each facet. In this paper mainly address the problem of finding the query facets from the search results. A query facet is a set of items that summaries an important aspect of a query. Faceted search can also be termed as a navigational search. Important documents are arranged as list style in every websites. To overcome the problems in the existing system introducing a new method BatchSTS algorithm for clustering.**

***Index Terms*—Query facet, faceted search, summarization, user intent**

I. INTRODUCTION

Information search should be a pleased experience. Search engines currently become an indispensable tool for every web users to locate information, but still needs a skilled searcher to perform an effective search. Although widely used search engines still suffer from many issues such as information overload. To overcome this can use faceted search.

Faceted search also known as navigational search is widely used today in many ecommerce sites such as “e bay, Amazon “etc. However this idea is not well explored in the general web search because of the heterogeneous nature of the web. In this paper mainly address the problem of finding query facets from the search results. After getting the facets can refine the search results using particular facets.

A query facet is a set of items that describe or summaries an important aspect of a query. Facet items may be typically a word or a phrase. So a query facets focus on the important of a particular query. A query may have multiple facets

Query facets provide interesting and useful knowledge about a query and thus can be used to improve search experiences in many ways. First, 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 study other brands and categories of watches. User can clarify their specific intent by selecting facet items. Then search results could be restricted to that 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. [1]These multiple groups of query facets are in particular useful for vague or ambiguous queries, such as “apple”. Show the products of Apple Inc. in one facet and fruit apple in another. Second, query facets may serve straight information that users are seeking.

In the figure 1.1 faceted search most popular, content, price are the facets and for the most popular facets the facet items are top rated ,very good. For the content facets the facets items are cook book, romance, travel etc.

In the figure 1.2 when search for computer the facets will be retrieved from the corresponding documents that means from the original search results

Compared to previous works on creating facet hierarchies [1], [2], [3], [8], [9], this approach is unique in two aspects: (1) **Open domain** and do not limit queries in a specific domain, like products, people, etc. New method is wide and does not depend on any specific domain knowledge. So the main advantage is it can deal with all queries. (2) **Query dependent**, instead of a permanent schema for all queries, extract facets from the top retrieved documents for each query. Different queries have different facets. E.g., query “computer” and “apple “have totally different query facets.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 6, April 2017

Fig 1.1: Faceted Search

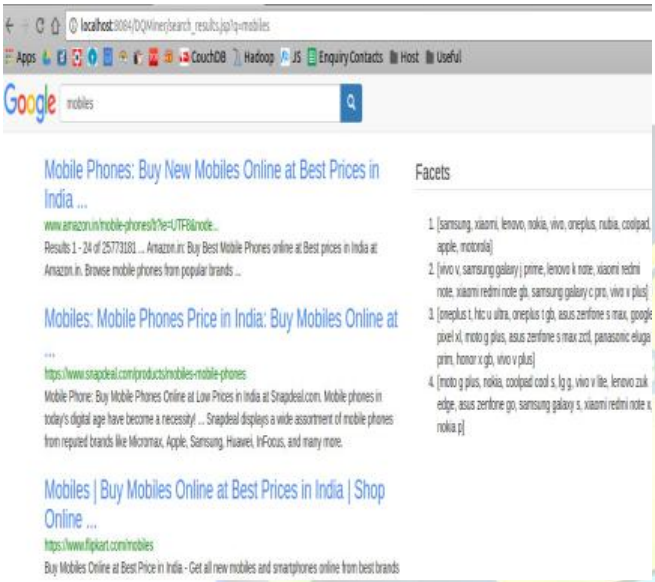

Fig 1.2: Facets

TABLE 1
Example of Query Facets

Example query facets
Query: Watches 1. cartier,omega,citizen,casio,rolex,movado 2. men's watch, women's watch , kids watch 3. analog ,digital, quartz, mechanical 4. black, blue ,white, green
Query: what are the fastest animals in the world? 1. cheetah, pronghorn antelope, lion, Thomson's gazelle, Wildebeest, cape hunting dog, elk, coyote, quarter horse, 2. birds, fish, mammals, animals, reptiles 3. science, technology, entertainment, nature, sports, lifestyle, travel, gaming, world business

Table 1 [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.

In this, automatically discover query facets for open-domain queries based on a common Web search engine. Facets of a query are automatically mined from the top web search results without any supplementary domain knowledge [1].

II. RELATED WORK

There are number of techniques developed to achieve faceted search. In this section, discuss query subtopic mining, search result diversification, search result clustering and query suggestion [2]

To address multi-faceted queries, much previous work studied mining query subtopics. A query subtopic is often defined as distinct information need relevant to the original query.) For example, {"news", "cnn", "latest news", "mars curiosity news"} is a query subtopic for the query mars landing, which describes the search intent of Mars landing news. {"Photos", "NASA", "new photos", "and curiosity rover photos"} is another query subtopic for the query, which describes the search intent about Mars landing photos. Subtopics and facets are different in that the terms in a query subtopic are not restricted to have any specific semantic relations or structures [4]

Search result diversification has been studied as a method of tackling ambiguous or multi-faceted queries, while a ranked list of documents remains the primary output feature of Web search engine today. The purpose is to diversify the ranked list to account for different search intents or query subtopics.

Query suggestion (or query recommendation) is a common technique used by search engines to assist users in reformulating queries. In the suggestion process, a user begins with issuing an initial query that may be useless. Then, the system provides a set of alternative queries that may better address the user's information need as suggestions. After that, the user can select one of the query suggestions to search again. [5] discussed about Automatic Number Plate Recognition (ANPR), Automatic Number Plate Recognition (ANPR) is a real time embedded system which automatically recognizes the license number of vehicles. In this paper, the task of recognizing number plate for Indian conditions is considered, where number plate standards are rarely followed.

Semantic class extraction [6, 7] is to automatically mine semantic classes interpreted as their class instances from certain data corpora. For example, it may extract USA, UK, and China as class instances of semantic class country. Due to the similar semantic relationships between terms inside a facet and a semantic class, semantic class extraction can be used for facet generation

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 6, April 2017

III. SYSTEM OVERVIEW

Mine Query facets by aggregating frequent list with in the top search results. Here uses a query dependent miner to mine facets from the top search results. Query dependent means instead of a fixed schema for all queries, it extract facets from the top documents with in the search results. That means it refresh the search results using the corresponding facets.

This approach use the list format because important information is organized as list formats in the top search results and also important list will be supported by many websites [1]. Unimportant list will not be supported by the top search results. So from this it's clearly understood good list and bad list.

Initially retrieve top documents from a search engine say A. Then A will be the set of documents retrieved from the top results A= $\{d_1,d_2,d_3,\ldots,d_n\}$. Then the query facets are mined using the following steps.

- List Extraction
- List Weighting
- List Clustering
- Facet Ranking

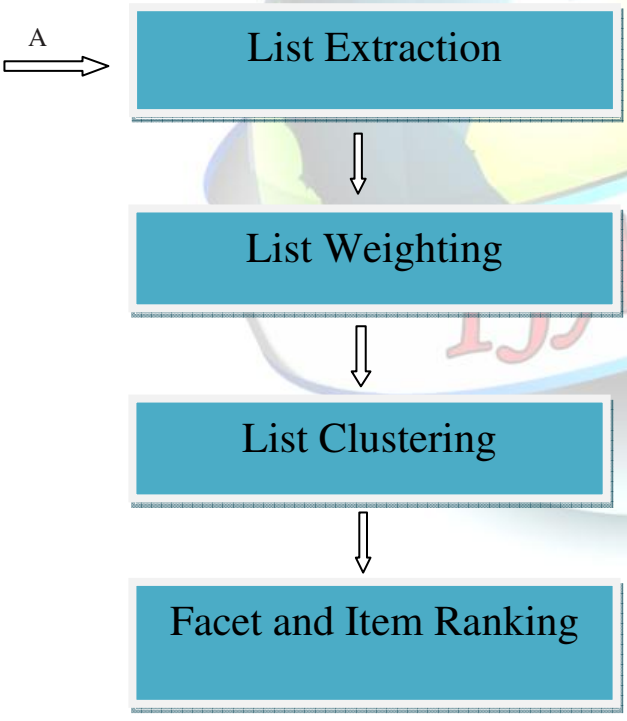

Fig 2: Work Flow

IV. IMPLEMENTATION DETAILS

4.1 *List Extraction*

From the set A should get the list from each of the documents. That means the list will be extracted from the document's contained in the set A. The list will be extracted using three methods

- Free Text Patterns
- HTML TAG Patterns
- Repeated Region Patterns

4.1.1 Free Text Pattern

The free text patterns mean the sentences and paragraphs which can be found in the document set A. From the sentences and paragraphs the list will be extracted using regular expressions. In this employ the pattern,

Item {, item}*(and \or) {other} item

Also use another pattern to get list from the semi structured paragraphs. Semi structured paragraph means there are some paragraphs which are separated by: or -. For such type paragraphs use another pattern as shown below,

{^item (: | -. +$}+

For example in the sentence "We shop for the mobiles *Samsung, Apple, and Nokia* ".When applying the regular expression get the list as { Samsung, Apple, Nokia }. Name this type of pattern as ***Textp.***

Apply the pattern on the textual content [1] (ignoring HTML tags and formatting in the webpage) of the search results, and extract matched items as a candidate list. To give an example, for the sentence, "The airlines servicing this airport are AA, Delta, and JetBlue", can extract the candidate list {"AA", "Delta", "JetBlue"} using the lexical pattern

4.1.2 HTML Tag Pattern

Extract candidate lists based on several HTML patterns that target list structures in HTML webpages, including drop-down lists, ordered lists, unordered lists and tables.

The HTML tag pattern, the list is extracted from the HTML tags such as SELECT, UL, and TABLE etc. From the tags only the items is selected. Table 2 shows the HTML tags. SELECT For the SELECT tag; simply extract all text from their child tags (OPTION) to create a list. Moreover, remove the first item if it starts with some predefined text, such as "select" or "choose". UL/OL For these two tags, also simply extract text within their child tags (LI). Describe the extraction in details below

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 6, April 2017

- **SELECT**: For the SELECT tag, extract textual content in the OPTION tags as a candidate lists. For the example in Table 2, extract candidate list {"watch brands", "Acutrim", "bulova"…."caravels"}.

- **OL**: For the OL tag, extract textual content in the LI tags as a candidate lists.

- **UL**: Similarly, for the UL tag, extract textual content in the LI tags as a candidate lists. For the example in Table 2, extract candidate list{"Titanium", "Automatic", "Quartz", "Gold"}.Note that in this case, when extracting textual content in the LI tags, ignore other HTML tags/formatting (i.e., "<a>"

4.1.3 Repeated Region Pattern

In some websites can see that important information is organized as well structured visual blocks. Figure 3 shows a repeated region comprised of 2 blocks in a repeated style. The block contains a picture and descriptions. The picture is ignored and the contents are extracted.

Table 2
Example HTML Sources that Contain Lists

Example HTML Sources that Contain Lists
SELECT: <select name="ProductFinder2" id="ProductFinder2" > <option value="WatchBrands.htm" >Watch Brands</option> <option value="Brands-Accutron.htm">Acutrim</option> <option value="Brands-Bulova.htm">Bulova</option> <option value="Brands-Caravelle.htm">Caravels</option> <option value="Brands-Seiko.htm">Seiko</option></select> UL: Dive   Titanium   Automatic   Quartz   Gold

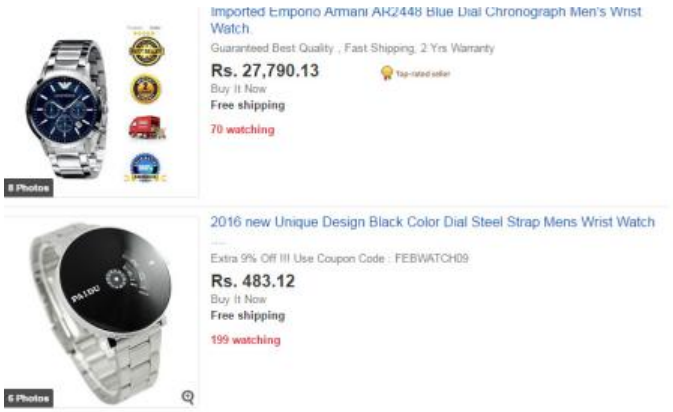

Fig 3: Repeated Region

4.2 List Weighting

After the list extraction, the output is the set of lists from each document set A. But some of the list is not informative to the particular query so it should be eradicate from the list. For that the list is weighted such that low weighted list can be crossed out easily. From each document, document matching weight is calculated.

$$S_{DOCUMENT} = \sum_{d \in A} (S_d^m \cdot S_d^r) \quad (1)$$

S_d^m Is the percentage of items contained in the document and S_d^r is the importance of the document.

- $S_d^m = \frac{N_{i,d}}{|l|}$ where $N_{i,d}$ is the item both contained in list and the document. |l| is the total no of items in the list.
- $S_d^r = \frac{1}{\sqrt{rank_d}}$ where $rank_d$ is the rank of document d. The higher d is ranked, the larger its score S_d^r is.

V. PROPOSED SYSTEM

5.1 List Clustering

Individual weighted list cannot be used as query facets since it may contain noises such as price tags etc. so it should be avoided to get the good facet. After the list weighting, the output is the weighted list. The clustering is done to group the related items. For the clustering process, BatchSTS algorithm is used. After the clustering process, twin lists will be grouped into a query facet.

Initially in the BatchSTS algorithm the entire list will be grouped under one say S. So the S array contains the entire weighted list. From S each list is checked whether it belongs to any cluster. The process is explained in the below algorithm.

Consider two list represented in the term vector model, $v_a = (t_{1,a}, t_{2,a}, \ldots, t_{N,A})$ and $v_b = (t_{1,b}, t_{2,b}, \ldots, t_{N,b})$. Each dimension corresponds to a separate term, and N is the number

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 6, April 2017

of dimensions. Since define that the weights of terms are equal, if the term t_i occurs in the listv_a, ti, a will be set to 1. Otherwise, ti, a will be set to 0. Note that the vectors are not normalized to unit length; the cosine similarity of two lists is as follows

$$sim(va, vb) = \begin{cases} \frac{va \cdot vb}{D} & \text{if} \quad va \cdot vb \leq D \\ 1 & if \quad va \cdot vb > D \end{cases} \quad (2)$$

Where $va \cdot vb$the inner product of two vectors and D is is a positive integer constant. The design of Equation 2 indicates that as long as there exists more than or equal to D common terms in two lists, these two lists can be viewed as with the highest similarity.

Definition 1 (list-list distance). The distance between two lists va and vb is defined as:

$$dis(va, vb) = \frac{1}{sim(va,vb)} - 1 \quad \text{if } sim(va, vb) \neq 0$$

$$\infty \quad \text{If } sim(va, vb) = 0 \quad (3)$$

With such definition, when va and vb have more than or equal to D common terms (i.e.,$sim(va, vb) = 1$), the distance between them is zero.

Definition 2 (center of cluster). Let $\{v1, v2, \ldots, vi, \ldots, vn\}$ be the set of lists belonging to cluster Ce , where $vi = (t1, i, t2, i, \ldots, tj, i, \ldots, tN, i)$. The center vc_e of cluster Ce is defined as:

$$vc_e = (tc1, e, tc2, e, \ldots, tcj, e, \ldots, tcN, e) \quad (4)$$

$$tc_{j,e} = \sum_{p=1}^{n} t_{j,p} \quad (5)$$

Definition 3 (list-cluster distance). The similarity between list vi and cluster Ce (whose center isvc_e) is defined as:

$$dis(vi, Ce) = \begin{cases} \frac{1}{sim(v_{i,Ce})} - 1 & \text{if } sim(vi, Ce) \neq 0 \\ \infty & \text{if } sim(vi, Ce) = 0 \end{cases} \quad (6)$$

Algorithm BatchSTS

Input: S: the list set
θ r: the radius threshold
Output: top-k clusters which have top-k most lists

1. Initialize C = Ø;
2. For each element v i of S
3. If there exists any cluster C j where dis (vi, Cj) is not infinite
4. Add v i into anyone of these clusters;
5. Else
6. Form a new cluster C new with the list $v\,i$;
7. C = C ∪ C new;
8. For each non-single-point element C i of C
9. While the radius of C i is larger than or equal to θ r
10. For each list v j in C i
11. If dis $(v\,j, Ci) \geqq$ θ r
12. Exclude v j from C i;
13. Check whether v j can be merged with other excluded lists;
14. Output top-k clusters in C which has top-k most lists;

End

Algorithm 1, is to find all connected items of the list set S. The items belonging to the same component will be merged as a cluster. It can be imagined that there will be a link between two lists as their distance is not infinite.

In lines 2-7, for each list vi of S, examine whether there is any existing cluster Cj where $dis(vi\,, Cj\,)$ is not infinite. If there is, this list will be added into anyone of these clusters. Otherwise, a new single-point cluster is formed with the list vi .

In lines 8-11, calculate the distances between two centers of any pair of non-single-point clusters, and if the distance between two clusters is not infinite, they will be merged together. Subsequently, in lines 12-17, the objective of the second main step is to guarantee the radius of each cluster is smaller than the threshold θr.

To meet this requirement, for each non-single-point clusterCi , exclude the list vj where $dis(vj\,, Ci)$ is larger than or equal to θr. Meanwhile, in line 17, it will be checked whether vj can be merged with other excluded lists. After this step, all clusters will satisfy the radius restriction, and finally, BatchSTS outputs the $topk$ clusters with top-k most lists.

4.2 Facet Ranking

After the clustering process query facets are generated. Each facet may contain many list of same type. Merge this entire list to get one list as a query facet.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 6, April 2017

Facets are evaluated based on how frequently it appears in top search results. For example for gender in the watches men's and women's will be ranked higher than others so that the search results will be displayed according to the importance. The importance of facet c is as follows:

$$S_c = \sum_{G \in \mathbb{C}(c)} S_G = \sum_{G \in \mathbb{C}(c)} \max_{l \in G} S_l. \quad (7)$$

Here C© is ideally the set of independent groups of lists Contained in query facet c. S_G is the weight of a group of lists G, and S_l is the weight of a list l within the group G.

4.3 Item Ranking

In the item ranking actual output is displayed. For example within a brand cluster for watch, it contains Cartier, omega, titan etc. In the item ranking decide which should be displayed first according to frequency count. Calculate $Se|c$, the weight of an item e within a facet c, by:

$$S_{e|c} = \sum_{s \in \mathbb{C}(c)} w(c, e, \mathbb{C}) = \sum_{G \in \mathbb{C}(c)} \frac{1}{\sqrt{AvgRank_{c,e,G}}}, \quad (8)$$

Where $w(c, e, G)$ is the weight contributed by a group of lists G, and $AvgRank(c; e; G)$ is the average rank of item e within all lists extracted from group G. Suppose $L(c, e, G)$ is the set of all lists in c and G (G € c) that contain item e, then

$$AvgRank_{c,e,G} = \frac{1}{|L(c, e, G)|} \sum_{l \in L(c,e,G)} rank_{e|l}. \quad (9)$$

VI. EXPERIMENTAL RESULTS

The following figures show the results of the proposed methods. In the figure 4.1 when a user type a keyword computer, initially all the documents is retrieved from a search engine. From each document from the search engine facets are mined using list extraction, list weighting, list clustering and facet and item ranking. In the figure 4.2 the mined facets for keyword computer is displayed.

Fig 4.1: Type Keyword in Search Engine

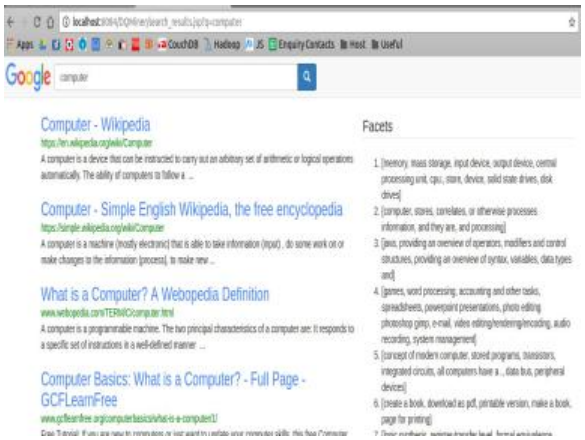

Fig 4.2: Facets

VII. PERFORMANCE ANALYSIS

The performance can be analyzed using the total no of clusters formed. For example in the keyword computer QT nearly 45 clusters and BatchSTS has nearly 30 clusters. By using BatchSTS algorithm the number of clusters is reduced and the similar items are mostly grouped in a similar cluster. QT clustering is performed based on one candidate cluster. According to the first candidate the clusters are grouped but in BatchSTS there is no candidate cluster. Outliers are also detected in the BatchSTS algorithm.

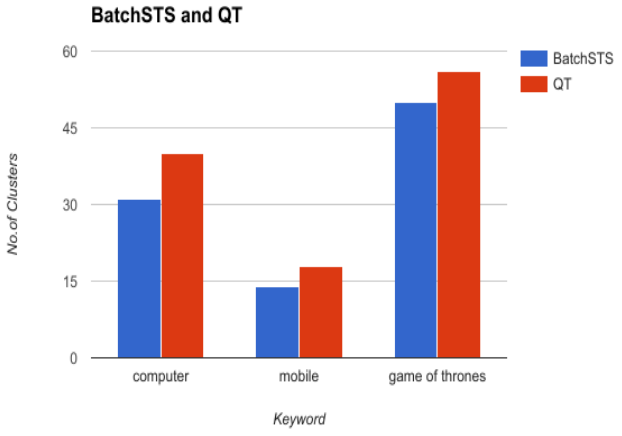

Fig 5: Comparison with QD miner

In the existing method QD Miner [1] it uses the QT algorithm to cluster the weighted list. In this use BatchSTS algorithm to cluster weighted list and tested 3 queries and in all cases new method is better than QT.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 6, April 2017

VIII. CONCLUSION AND FUTURE WORK

The main thing is to find the query facets to improve the search results. A query facet is a set of items that summaries an important aspect of a query. After getting the query facets can re rank the search result's using the particular facets such that the searching can become easier for non-technical users also. This approach can be improved in many aspects. For example, some 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. Part-of-speech information can be used to further check the homogeneity of lists and improve the quality of query facets. Explore these topics to refine facets in the future. 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. Automatic generation of description for the facets is an important future scope.

REFERENCES

[1] Zhicheng Dou, Member, IEEE, Zhengbao Jiang, Sha Hu, Ji-Rong Wen, and Ruihua Song"Automatically Mining Facets for Queriesfrom Their SearchResults"ieeetransactions on knowledge anddata engineering, vol. 28, no. 2, february 2016
.

[2] Weize Kong and James Allan Center for Intelligent Information Retrieval School of Computer Science
University of Massachusetts Amherst . CIKM'14, November 3–7, 2014, Shanghai, Chin

[3] M. Bron, K. Balog, and M. de Rijke, "Ranking related entities:Components and analyses," in Proc. ACM Int. Conf. Inf. Knowl.Manage., 2010, pp. 1079–1088.

[4] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, "Facetedpedia:Dynamic generation of query-dependent faceted interfaces forwikipedia," in Proc. 19th Int. Conf. World Wide Web, 2010,pp. 651–660.

[5] Christo Ananth, K.Kalaiselvi, S.Selvakani, P.Sorimuthu Iyan, "ANPR for Developing Countries", International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE), Volume 2, Special Issue 13, March 2016, pp: 21-34

[6] A. Herdagdelen, M. Ciaramita, D. Mahler, M. Holmqvist, K. Hall,S. Riezler, and E. Alfonseca, "Generalized syntactic and semanticmodels of query reformulation," in Proc. 33rd Int. ACM SIGIRConf. Res. Develop. Inf. retrieval, 2010, pp. 283–290.

[7] M. Mitra, A. Singhal, and C. Buckley, "Improving automaticquery expansion," in Proc. 21st Annu.Int. ACM SIGIR Conf. Res.Develop. Inf. Retrieval, 1998, pp. 206–214.

[8] P. Anick, "Using terminological feedback for web search refinement:A log-based study," in Proc. 26th Annu. Int. ACM SIGIRConf. Res. Develop. Inf. Retrieval, 2003, pp. 88–95.

[9] S. Riezler, Y. Liu, and A. Vasserman, "Translating queries intosnippets for improved query expansion," in Proc. 22nd Int. Conf.Comput. Ling., 2008, pp. 737–744