International Journal on Emerging Technologies **11**(2): 342-350(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Automatic Extraction of Facets for user Query in Text Mining [AEFTM]

Ramya R.S.[1], Naveen Raju[2], Pushpa C.N.[3], Venugopal K.R.[4], S.S. Iyengar[5] and L.M. Patnaik[6]
[1]Research Scholar, Department of Computer Science and Engineering,
University Visvesvaraya College of Engineering, Bangalore, Karnataka, India.
[2]UG student, Department of Computer Science and Engineering,
University Visvesvaraya College of Engineering, Bangalore, Karnataka, India.
[3]Associate Professor, Department of Computer Science and Engineering,
University Visvesvaraya College of Engineering, Bangalore,Karnataka, India.
[4]Vice Chancellor, Bangalore University, Bangalore, Karnataka, India.
[5]Professor, Department of Computer Science Florida International University, USA.
[6]INSA, National Institute of Advanced Studies, Indian Institute of Science Campus, India.

(Corresponding author: Ramya R.S.)
(Received 16 December 2019, Revised 10 February 2020, Accepted 12 February 2020)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Query facet is a group of items that describes the content covered by a user query. Every word in a facet item that assign significance to the facet. One single word is more appropriate than a big sentence if it is capable of giving the complete meaning of the sentence. However, identifying a single word efficiently is the challenging task. Normally, the important information of a query exists in the top retrieved document that are in the form of lists. Extracting query facet within the top search results is also a challenging task in Text mining. In this work, we propose a framework Automatic Extraction of Facets in Text Mining [AEFTM] for User Queries that extract the query facets automatically by grouping the list based on three categories namely HTML tags, free text patterns and repeat regions. Grouping G of the list is based on domain sites present in the list. We observe that some of the lists contains noise and irrelevant information for extracting the facets. In order to prune these lists, the importance of each item present in the lists from the group G is evaluated and Cosine Similarity (CS) between two items is calculated. Further, to extract more facets High Quality Clustering (HQC) algorithm is proposed to cluster the items that has the most number of point in each iteration. Finally, the top most items from each cluster are selected and provided as the best facets for the user query. Experiments are conducted on User Q and Random Q dataset. It is observed that the proposed method AEFTM out performs QD Miner method by removing duplicate items and provide a large number of useful and high relevant query facets for user submitted user queries.

Keywords: Faceted Search, Query Facet, Query Processing, Search Satisfaction, Text Description.

I. INTRODUCTION

The collection of query facets are identified for the user submitted queries. Facet is a set of items that describes the essential aspect of a query. A facet item means a word or a phrase. Due to the increasing population in the web application field both in terms of usability and data collection. There is a large collection of data that makes difficult for the user to search and identify the exact information for their submitted queries. The efficiency of an application [2-5] is important factor to be considered in this competitive environment, one factor is that how it helps in mining the large amount of data efficiently. One single word is more appropriate than a big sentence if it is capable of giving the complete meaning of the sentence. However, identifying a single word efficiently is the challenging task. Any query can be explained using a word or collection of words that are termed as facets. Every word in a facet item that assign significance to the facet. Each query might have distinct view that results in the generation of more number of facets. In the previous work author focus on the extraction of query facets by clustering the similar lists for the top search results [1]. However, these lists may contain duplicate items, irrelevant information and noise. Some of the lists may contain only few items. For example, a set of facets for the user query *Dresses* includes information about dresses in distinct views, like *Gender, Color, designer Brand, categories, type of design* etc. These facets helps the users to narrow down their search area as facets and provide them with sufficient information about the query. For example, a user wants to buy clothes and he types a sentence *clothes for men* in the search box. The proposed method aids the search engine to display the query together with the facets that guides the user to gain the knowledge about all the available categories namely colors, designs and brands of clothes so that the user can easily choose the cloth based on his choice. Thus, the user can easily learn some important aspect of a query without searching several web pages. Another way of faceted search can help the user to select from particular facets that leads to few search results that are accurate avoiding vagueness. Sometimes, the facets may lead to ambiguous information, for example, an user may search for *apple*, and facet items may be related to both *fruit apple* and a *mobile phone gadget apple*. However, in such situation the user knows the intent of search, hence, he preferably chooses the required facets. The proposed method utilizes High Quality Clustering (HQC) algorithm in order to "Extract more number of facets to the user".

Most of the times there is a chance that two or more websites might lead to redundant information. Previous works focused to avoid the redundant information and choose only those websites which have unique information. Suppose there are ten websites which contain almost similar information. Then our methodology ranks those websites based on the amount of unique information they contain as well as the coverage of information which helps user o choose from best ranked websites. This is another way extracting exact information with minimal page spanning and time consumption.

Applications of faceted search is the counting of result documents across several facets [11]. However, aggregations of these results support better decision making and shows how to efficiently index and search documents with correlated facet values by reducing the problem to a recently solved instance of indexing shared content in an ordered tree of documents. Faceted search is also used in ecommerce application. It works by extracting facets for user query to select and adjust the facet ranking according to the user satisfaction. There are two components in faceted search namely facet generation and facet feedback. In facet generation, the facets are extracted by first retrieving the candidates from the search results based on predefined patterns and then refine the candidates by clustering methods. On the other hand in facet feedback the terms are selected by the user to adjust the search results.

Motivation: In the QDminer method [1] the author focus on the ex-traction of query facets by clustering the similar lists for the top search results. However, these lists may contain duplicate items, irrelevant information and noise. Some of the lists may contain only few items. Therefore, the number of facets generated to the user query in QDminer method is very less.

Contributions: The main contributions of this work are as follows:

– The items in the lists are grouped based on the similar domains present in the dataset.

– Item similarity is computed using Cosine Similarity (CS) measure that are present in the group.

– To extract more number of facets to the user by High Quality Clustering (HQC) algorithm.

Organization: The rest of the paper is organized as follows: Section II introduces a detailed overview of related works. In Section III proposed AEFTM Framework is presented. Section IV discusses the performance evaluation. Finally, Section VI contains the conclusions.

II. RELATED WORKS

Due to increase in the usage of internet, the user can fetch a lot of information through the net. Therefore, can be considered as an efficient way to generate huge amount of data in the electronic format. Facet based searching is a powerful paradigm in this scenario. In the previous studies, identification of facets was manually generated or based on some prior information. Dakka *et al.,* proposed an unsupervised technique to produce facets that utilizes the external references like Wikipedia to find the context for every phrase in a document. The obtained facets are compared with the external and original databases and then the decision is taken on the facet usage. However, navigating through large database that consists of more number of tuples to identify any document can consume more search effort and time [6].

Different feature extraction methods for extracting topics from the text documents were described [7, 8]. Ramya *et al*., proposed a hierarchial clustering method for extracting spatial objects for the user submitted queries with improved response time when compared to other state of arts techniques [9]. Diao *et al.,* proposed a methodology for the Spoken web to generate facets by indexing the metadata of the audio content with the help of facets and further, these search results are ranked. The proposed approach is very helpful for less educated population especially for farmers in addressing their queries [10]. Bhat *et al.,* introduced an approach to classify the email by keyword stemming so that the facets are discovered quickly [16]. Latha *et al.,* proposed a dynamic facet ordering mechanism for e-commerce. A fully automatic algorithm for ranking the facets where top ranked ones help in better drill down during fetching the documents is developed. However, the method is inefficient due to under coverage of facets [13]. Dou *et al.,* (2015) proposed a mining technique where facets are mined based on the entities present in the freebase corresponding to search queries [1]. This enhances coverage of facets along with efficiency. However, the technique does not address for huge data. Facet based searching has been useful paradigm way to drill down the search results in terms of sites like e-commerce. It is not much expanded in general web due to its heterogeneous nature. Researchers proposed dynamic association rule mining using genetic algorithms and also a data analysis for forensic applications [17,19].

Li *et al.,* (2010) proposed facet based retrieval system, Facetopedia to generate facets dynamically for wikipedia. The facet generation is dynamic and has builds strong hierarchy based on semantics of Wikipedia [3].

User can use this facets in the interface provided and can easily browse through articles in Wikipedia.

Searching through the web is a very time consuming problem. Many researchers have tried to address this problem by various ways.

Le *et al.,* extend the concept of facets to semantic search using social networks i.e. on linked data. Authors devise an algorithm to graphically visualize the facets to represent the nodes of ontological relations and for filtering the search results [15]. Roy *et al.,* [4] proposed an automated facet generation technique, which is independent of domains. The algorithm concentrates on users previous move to suggest the next facet sets.

When data is multidimensional, it needs multi-facets to analyze them rigorously. Liberman and Lempel (2012) described an approximation algorithm to generate multi-facets which approximately selects optimal facet list according to the user query [5].

Due to uncertain amount of data generation, heterogeneous facets have been an integral part of any data set. Zhao *et al.,* proposed *pivotslice*, an interactive visualization technique for faceted browsing to easily identify the relationships between the data items [18].

In most of the search engines where faceted search is employed, we often find a fixed set of facets working actively. The demerits of having fixed set of facets is, they may become out of use if all the products belong to that facet [20]. Also it takes significant amount of time and effort to build a fixed list.

Faceted search is very powerful when it comes to structured data. But incorporating facets have problems like meeting exact user intent and dealing with huge data. Jiang *et al.,* (2016) proposed a method of combining structured data with logs from web query [12]. They employ various disambiguation techniques to avoid ambiguity while fetching the keywords from web query logs.

In every computing system multithreading and multiprocessing has been an integral part. Facets related to memory access might manifest in multithreaded environment and lower their efficiency. To overcome this, Viglas [21] proposed a design that is scalable across different platforms considering the underlying hardware.

Zhao *et al.,* [22] address the present inconvenient way of retrieving ranked list of videos when there are multiple facets. Hierarchical method is proposed to help the users to visualize and understand the meaning of any facet quickly. So that they can choose their facets accordingly to fetch only required list of videos.

Madhav (2017) proposed an approach of facet mining and ranking to reduce the number of pages to be navigated to reach desired resources by aggregating many search result lists and rank them by assigning weights [23]. In order to prioritize the facets the model utilizes utility mining [33, 34].

User satisfaction has been most focused concern in today's situation. Capturing user satisfaction is difficult due to many reasons. However, Chen *et al.,* proposed a technique to check the prediction of user satisfaction by the mouse movement and mouse click data during browsing.

Table 1: Comparison of Different Indexing Techniques.

S. No.	Authors	Model	Concept	Advantage	Disadvantage
1.	Vidhya & Saravanan (2018) [24]	Content and Auxilary Model	Removes irrelevant features based on selected features	Efficient on high dimensional data.	
2.	Diao *et al.,* (2010) [10]	K-means algorithm	Proposal of query facet engine which fetches, weighs and prioritizes the facets to reduce the number of pages to be navigated.	It is a semantic solution which avoids hassle during web surfing.	Usage of utility mining may lead to the common problem of frequent pattern mining.
3.	Jiang *et al.,* (2016) [25]	QDMKB	Query mining technique which makes use of freebase to generate facets corresponding to query.	Proposed methods significantly enhances facet coverage.	Not highly effective when it comes to general queries.
4.	Bing *et al.*, (2015) [26]	Graphical Model	Automatically exploits a latent topic.	Mine substitution patterns.	Works on one word level
5.	Li *et al.,* (2013) [27]	QUBIC	Produces a history of user URL bipartite collection.	Extract connected components with high precision.	User click information is not investigated.
6.	Zhao *et al.*, (2013) [18]	-	Pivotslice helps user to dynamically search through facets data and to have best visualization of relationships between data items.	It is easy to learn and use also efficient in deriving relationship between data sets.	Pivotslice is not extended to all kinds of datasets.
7.	Li *et al.,* (2010) [3]	CASTANET	Generating facet hierarchies from available facets using lexical dataset	It achieves higher quality results compared to other automatic techniques.	Less item coverage. No mechanism to identify morphological variation and spelling mismatch.
8.	Basu *et al.,* (2008) [28]	WQT for clustering facet term used in QF-I QF-J	Supervised way of Creating of query facet groups that are semantically related to generate effective facets.	Supervised learning technique and graphical models outperform any unsupervised techniques.	Graphical model uses approximate inference technique as exact determination of terms to belong to any group is intractable.
9.	Roy *et al.,* (2010) [4]	A greedy set-cover algorithm	Framework developed by author is a Inter-domain, unsupervised, semantically related technique of facet generation.	Efficiency in generating multi-facets on random domain.	There is no way to identify multiple occurrence of a word and word coverage is not complete.
10.	Grineva *et al.,* (2011) [29]	Optimization-based methods and a learning-to-rankbased method	Identifying facets through automatic unsupervised technique for text databases	Comparing with original and extended database helps discover efficient facets.	Specific for only text database and makes use of external database to identify facets.

Zhang *et al.*, (2019) designed nested facet system based on ontologies to help human data interaction in order to explore data in biomedical domain and enhance human interaction with the system [35]. Table 1 and Table 2 shows the comparison of different facet search systems. Guo *et al.,* proposed weakly supervised method to extract topic specific facets using Label Propagation algorithm (*LPA*). The methods extract complete collection of facets better than other state of arts facet mining approach [36]. Fukuda *et al.*, [37] [38] designed a two dimensional clustering facet cube in order to co-relate the set of facets from one document to more number of documents and further group into at least one cluster. Cluster centre are calculated to find out facets thar are located near the cluster centre.

Table 2: Summary of the existing faceted search Systems.

S. No.	Authors	Model	Data sources	Facet Term Extraction	Evaluation Metrics	Search Paradigm	Facet Ranking
1.	Vandic *et al.,* (2017) [20]	Facet Optimization	Web database	Extracting attributes based on Clicks	Specificity and Dispersion	Keyword search	Drill Down
2.	Jiang *et al.,* (2016) [25]	QDM_{KB}	Freebase	Automatically extracting entities from freebase	rp-nDCG, F1- nDCG	Keyword search	None
3.	Dou *et al.,* (2015) [1]	QDMiner	Wikipedia	Automatically grouping frequent list	nDCG, rpnDCG	Keyword based search	Context Similarity
4.	Zhao *et al.,* (2013) [18]	PivotSlice	Unstructured documents	Automatically extracts relation between two attributes from database	None	Form-based search	None
5.	Zhao *et al.,* (2011) [22]	TEXplorer	Text database	Automatically extracting attributes from database	INDG score	Form-based search	Significance measure
6.	Grineva *et al.,* (2011) [29]	Blognoon	Web pages	Automatically extracting attributes from database	None	Keyword search	Relevance to a search query
7.	MacAvaney *et al.,* (2019) [30]	CAR	Wikipedia	Manually extracts low utility facets	nDCG, MAP	Text based search	None
8.	Van Zwol *et al.,* (2010) [31]	MediaFaces	Semi-structured data	Automatically selecting from annotation information	None	Keyword search	Using recent query logs
9.	Girgensohn *et al.,* (2010) [14]	DocuBrowse	Unstructured enterprise documents	Manual extracted from database	None	Form-based search	None
10.	Roy *et al.,* (2009) [32]	TEXplorer	Relational database	Automatically extracting attributes from database	None	Form-based search	Navigation cost

III. FRAMEWORK: AUTOMATIC EXTRACTION OF FACETS FOR USER QUERIES IN TEXT MINING [AEFTM]

A. Problem Definition

On a website, for a given user input query q_u, the problem is to extract a set of facets of a user query, satisfying the user preference without browsing more number of web pages. The objectives are the following:

– To group the items based on the similar domains.

– To provide more number of facets to the user query.

B. Assumptions

It is assumed that the user is online and preprocessing is offline.

C. Facet Extraction Framework

The facet extraction framework retrieves collection of query facets by combining the top frequent lists within top search results for a user given text query as shown in Fig. 1. The framework has five different modules namely: (1) Pre-Processing Phase (PP) (2) List Extraction Phase (LE) (3) Items Grouping Phase (IG) (4) Ranking Phase (RP) (5) Similarity Computation Phase (SC) and (6) Clustering Phase (CP). The modules are explained in the following sections.

(a) Pre-Processing Phase (PP): In this work, two dataset namely Random Q and User Q dataset is used to retrieve the text data. Dou *et al.,* [1] built these dataset by randomly collecting queries from a commercial search engine. The dataset is in XML format and in total 105 queries and 90 queries are collected for Random Q and User Q dataset. Each query consists of 100 documents respectively. Each document consists of *Document id, Title, URL, description, rank of the document, document text, repeat region and converted HTML list.* From the above contents for each document, we extract only *query, query id, document text* and further extract a collection of list $L_{doc}l_i$ from HTML content of the document based on three patterns namely text patterns, HTML patterns and repeat region patterns.

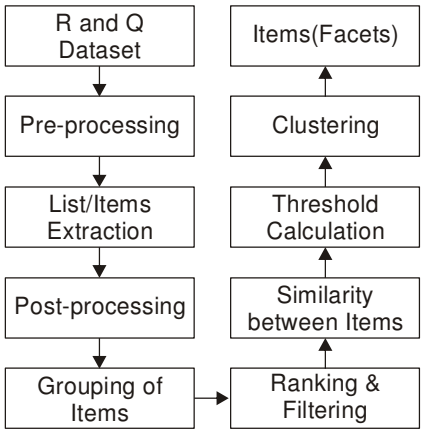

Fig. 1. Automatic Extraction of Facets in Text Mining [AEFTM].

Text Patterns: Text paragraphs are extracted within each document *doc* and are splitted into separate sentences. Further, pattern word {word*(and|or)} is employed similar to [1] to mine the words from each sentences. These sentences based patterns are named as $TEXT_{SP}$.

Text based patterns are extracted by using the pattern {word(:, |, –, and)} from a semi structured text paragraphs. Text based pattern $TEXT_{PT}$ extracts the lists from a continuous lines that consists of two different parts separated by a colon or a dash. Initially, part of these lines are extracted as a lists.

HTML Patterns $HTML_P$: In HTML patterns, the lists are extracted from various style list namely UL, SELECT, TABLE and OL and these patterns are named as $HTML_P$. For example:

UL:

```
<ul><li><a href=/rst.asp?q=digital>Digital</a></li><li><a
href=/rst.asp?q=smartwatch>SmartWatch</a></li>
<li><a href=/rst.asp?q=analog>Anolog</a></li></ul>
```

SELECT:

```
<select name=ProductFinder1id=ProductFinder1>
<optionvalue=WatchBrand.htm>WatchBrand</option>
<option value=Brands-Rolex.htm>Rolex</option>
<option value=Brands-Titan.htm>Titan</option>
<option value=Brands-Omega.htm>Omega</option></select>
```

TABLE:

```
<table width=100%>
<tr><td width=10%></td><td>Purple</td></tr>
<tr><td></td><td height=20>Blue</td></tr>
<tr><td></td><td height=20>Lavender</td></tr>
<tr><td></td><td height=20>Black</td></tr>
<tr><td height=4 colspan=2></td></tr></table>
```

Repeat Regions Patterns: In webpages, some of the data is arranged in a structured blocks (visual form). Each of these blocks consists of four attributes namely: *image, name of the place, description of the place and the user rating.* Three attributes are extracted by ignoring the image attribute since we are focusing only on textual data. These attributes are extracted based on the DOM trees. The entire text is extracted as a list and is named repeat region text as $REPEAT_T$.

(b) List Extraction Phase (LE): In this module, text-based pattern $TEXT_{PT}$, HTML patterns $HTML_P$ and repeat regions patterns $REPEAT_T$ that are extracted from pre-processing stage are considered as a main lists. Further, some of lists are extracted from every document when a text paragraph ends with a full stop. Comma separated items are considered as a list and are extracted from each line in the document. Same procedure is applied to extract the list for all the ten thousand documents. Later, Post Processing (POS) is applied on these list to prune symbol characters that are not relevant for further processing. Special characters like ({, }, (,))are pruned and all the uppercase text are converted to lowercase text. Long items that contain more than ten terms are removed respectively.

(c) Items Grouping Phase (IG): Extracted lists obtained from the list extraction phase consists of duplicate information. These similar list that contain collection of items are grouped together to compose the facets. Grouping is done by combining the lists that belong to the same domain sites. Table 1 illustrates items representation for each document along with the domain site it belongs to. Table 2 shows the grouping of lists based on the same domain and the items are merged for these lists.

(d) Ranking Phase (RP): The importance of an item depends on how many items are present in the group *G* with their ranks. To extract the better facets, items of one group are compared with the items of the remaining group. Therefore, frequency of all the items are computed. Further, the importance of each group G is calculated by the Document Score represented by S_{DS} in Eqn. (1).

$$S_{DS} = \sum d \in R(S_{dc}^{m} \cdot S_{dc}^{T}) \quad (1)$$

S_{dc}^{m} represents the number of items present in the document as shown in Eqn. (2). A group *G* is supported by a document *dc*, if it contains few or all items of G. If there are more items present in *dc*, the more it supports *G*.

$$S_{dc}^{m} = \frac{N_{G,\,dc}}{|\,G\,|} \quad (2)$$

$N_{G,dc}$ is the number of items present in Group *G*. | *G* | is the total number of items present in Group *G Sr dc* is to measure the significance of the document dc as shown in Eqn. (3)

$$S_{dc}^{T} = \frac{1}{\sqrt{dc_{\text{rank}}}} \quad (3)$$

where dc_{rank} is the document rank. Higher the document is ranked, the larger the score Sr dc respectively.

(e) Similarity Computation Phase (SC): In Similarity Computation Phase, the similarity between two items are computed using Cosine Similarity CS measure. In order, to provide more number of the resultant facets and to expand the search space, the items needs to be expanded. More reliable and relevant facets are obtained with the use of these synonyms. The Cosine Similarity measure is used to calculate similarity between two items by Eqn. (4).

$$CS(It_i, It_j) = \frac{TF(i) \times IDF(It_i) + TF(j) \times IDF(It_j)}{TF(i) \times IDF(It_i) + TF(j) \times IDF(It_j)} \quad (4)$$

Where *i, j* = 1, 2,... n where *n* is the number of items in group G.

(f) Clustering Phase (CP): Clusters are constructed for all the group that belong to the same domain as shown in Algorithm 1. The items in each group that are ranked in the Ranking Phase (RP) are arranged in the descending order. The algorithm takes all the sorted items (descending order) and a predefined threshold as an input. Initially, the first item *z* in the list is considered as a centroid of a cluster.

Next, the second item in the list is compared with previous item (centroid item). If the Cosine Similarity between the two items are greater than the predefined threshold, then the particular item is added to the cluster otherwise the item is checked in the later iterations. The average of the cluster is computed based on the number of items containing in each cluster.

Algorithm 1 Automatic Extraction of Facets in Text Mining (AEFTM)

Input : User Input Query q, Cron String C_s, trigger periods t_p, article A.

Output: Ranked Documents as per Relevance Rel_{docs}

begin

Phase I: Offline:

1. Pre Processing of Data (PP).
2. List Extraction (LE).
3. Post Processing (POS).
4. Cosine Similarity (CS)

Use computed Cosine Similarity Score from the repository

Phase II: Online:

For user input query q_u

for *each item* $a \in I_t$ **do**
- $a.centroid \leftarrow a$
- $I_t.remove.(a)$
- **for** *each* $b \in I_t$ **do**
  - **if** $CS(a,b) > Thr$ **then**
    - $c_i.add(b)$
    - $c_i avg \leftarrow c_i.avg + b.r$
    - $I.remove(b)$
  - **else**
    - Continue;
- Compute $c_i.avg = \frac{c_i.avg + a.r}{n(c_i)}$
- $Avg = c_i.avg + Avg$
- $i = i + 1$

$Avg = \frac{Avg}{n(c)}$

for *each* $c_i \in C$ **do**

$Remove(c_i)$

Table 3: Items representation for each document.

Documents	Lists	Items present in the list	Domains
Document 1	$List_1$	$item_1, item_8, item_6, item_{12}, item_{14}, \cdots item_n$	Twitter
	$List_2$	$item_6, item_1, item_{16}, item_{12}, \cdots item_n$	Facebook
	$List_3$	$item_1, item_3, item_6, item_8, \cdots item_n$	Facebook
	$List_4$	$item_1, item_5, item_{16}, item_7, \cdots item_n$	Wikipedia
	$List_5$	$item_5, item_9, item_{14}, item_{15}, \cdots item_n$	Facebook
	$List_6$	$item_1, item_2, item_{18}, item_{10}, \cdots item_n$	Wikipedia
	$List_7$	$item_8, item_2, item_{10}, item_{12}, \cdots item_n$	Twitter
Document 2	$List_1$	$item_{13}, item_{23}, item_{23}, item_{16}, item_8 \cdots item_n$	Facebook
	$List_2$	$item_7, item_4, item_{15}, item_{21}, \cdots item_n$	Wikipedia
	$List_3$	$item_{10}, item_2, item_3, item_{43}, \cdots item_n$	Facebook
	$List_4$	$item_{33}, item_{16}, item_{13}, item_{25}, \cdots item_n$	Wikipedia
	$List_5$	$item_{18}, item_{34}, item_{21}, item_2, \cdots item_n$	Twitter
	$List_6$	$item_1, item_2, item_{18}, item_{10}, \cdots item_n$	Wikipedia
	$List_7$	$item_{13}, item_{17}, item_{19}, item_{16}, \cdots item_n$	Twitter

Table 4: Grouping of items based on domains.

Documents	Lists	Items present in the list	Domains
Document 1	$(List_1, List_7)$	$item_1, item_2, item_6, item_8, item_{10}, item_{12}, item_{14}$	Twitter
	$(List_2, List_3, List_5)$	$item_1, item_3, item_5, item_6, item_8, item_9, item_{12}, item_{14}, item_{15}, item_{16}$	Facebook
	$(List_4, List_6)$	$item_1, item_2, item_5, item_7, item_{10}, item_{16}, item_{18}$	Wikipedia
Document 2	$(List_1, List_5, List_6)$	$item_{13}, item_{23}, item_{23}, item_{16}, item_8, item_{18}, item_{34}, item_{21}, item_2, item_4, item_{37}, item_5, item_{21}$	Facebook
	$(List_2, List_4)$	$item_7, item_4, item_{15}, item_{21}, item_{33}, item_{16}, item_{13}, item_{25}$	Wikipedia
	$(List_3)$	$item_{10}, item_2, item_3, item_{43}, \cdots\cdots item_n$	Twitter

Finally, the top *K* cluster are retrieved based on the number of items present in each cluster. Remaining clusters that consists of few items are discarded. Further, all the items are sorted by their weights and provide the top most item from each cluster to the user.

IV. PERFORMANCE ANALYSIS

The Quality of the clusters can be measured using several metrics namely Purity, Normalized Mutual Information (NMI), Random Index (RI) and F measure.
NMI is a good metric for identifying the quality of cluster and is given by Eqn. (5)

$$NMI\,(CL, C) = \frac{2 * I(CL, C)}{(H(CL) + H(C))} \quad (5)$$

CL and *C*. Random Index is used to measure the similarity between two classes

$$RI = \frac{TP + TP}{TP + FP + FN + TN} \quad (6)$$

where *TP* is number of true positives, *TN* is number of true negatives, *FP* is the number false positive and *FN* is the number of false negatives as shown in Eqn. (6). *F* is the measure for a cluster *C* with respect to certain Class Label *CL* that represents how quality of the cluster describes the Class labels by computing the mean of precision and recall by Eqn. (7).

$$F(C_j, C_j) = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (7)$$

The overall *F* measure is a weighted sum of maximum *F* measure for a cluster in *C* as shown in Eqn. (8)

$$F\,(CL, C) = F(C) = \frac{1}{N} \sum_{i=1}^{K} \max\,\{F(C_j, C_j)\} \quad (8)$$

while experimenting, multiple facets are found. To obtain good facets from multiple facets normalized Discounted Cumulative Gain (nDCG) is used to rank the query facets can be given as Eqn. (9)

$$nDCG_k = \frac{DCG_k}{IDCG_k} \quad (9)$$

where DCGk is the Cumulative gain by correct ordering. DGi is a discounted Gain of ith facet. There are two types of nDCG measures namely first purity nDCG (fp–nDCG) and recallpurity nDCG (rp – nDCG) respectively. The fp-nDCG is based on the initial appearance of each cluster. The purity of each facet fi is considered by multiplying DGi by the percentage of perfectly assigned item.

$$w_i = \frac{|C| \cap f_i}{f_i} \quad (10)$$

rp-nDCG is based on overall resultant facets. Each facet is weighted by

$$w_i = \frac{|C \cap f_i|}{|f_i|} \frac{|C \cap f_i|}{|C|} \quad (11)$$

The Table 3 shows the comparison results of QDminer and AEFTM framework for the dataset User Q and Random Q. It is noticed that the quality of cluster on the user Q is better with high purity score 0.920 when compared to QDminer purity score 0.911 with reasonable scores for NMI, RI, F1, F5. Whereas, in Random Q dataset the NMI score is less i.e., 0.750 than User Q i.e., 0.861. These scores indicate that a small number of facets are generated more in Random Q dataset, than in User Q dataset from the same ground truth classes. This takes place only when the quality of search result is not good to group the list that belong to the similar domains. Table 4 shows the comparison results of QDminer and AEFTM framework on User Q and Random Q. It is noticed that the values of nDCG and fp-nDCG are effective for ranking the query facets on both User Q and Random Q dataset. However, rp-nDCG values are low because only small
percentage of human labelled items are returned as facets. This is because only qualified items are cluster to eliminate the useless items to provide the best items as facets.
In QDminer, the lists are considered from the initial stage to the last level i.e., generation of facets and the keyword match is done only for user query to generate facets. However, in the proposed AEFTM framework, clustering is based on cosine similarity score obtained from CS module to generate more high relevant number of facets compared to QDminer model.
In our experiment, 100 search results are taken ranging from 10 to 100 inorder to investigate whether the query facet quantity affects the quality of the facets respectively. Fig. 2 shows that as the number of facets increases, the quality of facets also increases. Query facets is better when more search results are used because more search results consists of large number of lists and therefore, the items that can generate more facets.

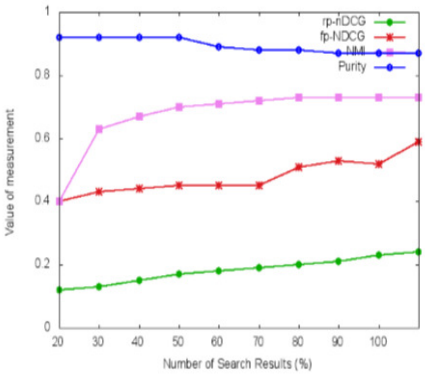

Fig. 2. Experiment search results with search facet quantity.

Table 5: Example of query facets mined by the AEFTM framework.

S. No.	Query	Query Facets
1.	Dresses	(a) Western Dresses, Ethnic Dresses, Party Wear, Festival (b) Womens, Mens, Kids Wear (c) Skirts, Sarees, Gowns, T-shirts, Shirts, Kurthas, Kurtis (d) Red, Blue, Pink, Purple, Yellow, White, Black, Orange
2.	Earphones	(a) BoAt, JBL, Skullcandy, Mi, Sony, Samsung, Sennheiser,Bose (b) Wired, Bluetooth, with Mic, Without Mic (c) Gray, Green, Jazzy Blue, Black, White, Ranging Red
3..	Ice Age	(a) Season1, Season2, Season3, Season4, Season5 (b) The Meltdown, Dawn of the Dinosaurs, Continental Drift, Collision Course (c) Sid, Buck, Scrat, Shira, Eddie, Peaches (d) Gone Nutty, No Time for Nuts, Surviving Sid, Scrat: Spaced Out
4.	Indias World Heritage Sites	(a) Hampi, Khajuraho, Ajanta Caves, Humayuns Tomb, Konark, Ellora Caves, TajMahal, Jaipur City, Elephenta Caves, Agra Fort (b) Karnataka, Madhya Pradesh, Maharashtra, Odisha, Rajasthan, Agra (c) Unesco World Heritage centers, Unesco Monuments in India

Table 6: Query facet quality for the QDminer and AEFTM framework.

Model	Dataset	nDCG	fp-nDCG	rp-nDCG	PRF	wPRF
AEFTM	User Q	0.723	0.684	0.201	0.412	0.425
	Random Q	0.711	0.668	0.205	0.407	0.424
QDMiner [1]	User Q	0.683	0.631	0.189	0.371	0.382
	Random Q	0.697	0.640	0.192	0.385	0.409

Table 7: Query facet quality for the QDminer and AEFTM framework

Model	Dataset	Purity	R1	F1	F5	NMI
AEFTM	User Q	0.920	0.920	0.753	0.510	0.861
	Random Q	0.911	0.810	0.768	0.757	0.750
QDMiner [1]	User Q	0.909	0.890	0.723	0.674	0.809
	Random Q	0.878	0.842	0.728	0.608	0.749

It is observed from the figure that the quality of cluster becomes subtle when the results are more than 50. More the search results are takes, more facet items may be generated. However, the impact on the quality of query facets is less. Increase in the number of search results leads to better query facets because more number of lists do exists when the number of search results are more and therefore facets generation. There is an increase in the purity when compared to other measures like rp-NDCG, fp-NDCG, NMI of query facets as shown in the Fig. 2 (blue line) because in the proposed framework, the items in the lists are merged together based on the similar domain.

V. CONCLUSION

In this research work, the problem of extracting user query facets is addressed. Query facets are the group of items that summarizes the significant aspect of a query. An item is basically a word or a phrase. Automatic Extraction of Facets in Text Mining (AEFTM) framework is proposed to automatically extract facets by grouping the items present in the collection of lists based on text in the document, HTML tag and repeat regions that occur within the top search results for user queries. The Cosine Similarity (CS) measure is utilized to calculate the similarity between the items. Based on the CS score generated, the items are clustered using high quality clustering algorithm in order to provide relevant number of quality facets to satisfy the user preference.

VI. FUTURE SCOPE

In future, we would like to provide meaningful and relevant text description of query facets to the user so that the user can analyze and understand the facets clearly and get exact information for their submitted queries.

Conflict of Interest. The authors confirm that there are no known conflicts of interest associated with this publication of this paper.

REFERENCES

[1]. Dou, Z., Jiang, Z., Hu, S., Wen, J. R., & Song, R. (2015). Automatically Mining Facets for Queries fromTheir Search Results, *IEEE Transactions on Knowledge and Data Engineering*, *28*(2), 385–397.

[2]. Venugopal, K. R., Srinivasa, K., & Patnaik, L. M. (2009). Soft Computing for Data Mining Applications, *Springer*.

[3]. Li, C., Yan, N., Roy, S. B., Lisham, L., & Das, G. (2010). Facetedpe-dia: Dynamic Generation of Query-Dependent Faceted Interfaces for Wikipedia, *In Proceedings of the 19th International Conference onWorld Wide Web*, 651–660.

[4]. Roy, S. B., Wang, H., Nambiar, U., Das, G., & Mohania, M. (2009). Dynacet: Building Dynamic Faceted Search Systems over Databases, *In Proceed-ings of the IEEE 25th International Conference on Data Engineering*,1463–1466.

[5]. Liberman, S., & Lempel, R. (2012). Approximately Optimal Facet Selection,(2012).*In Proceedings of the 27th Annual ACM Symposium on Applied Computing*, 702–708.

[6]. Dakka, W., & Ipeirotis, P. G. (2008). Automatic Extraction of useful Facet Hierarchies from Text Databases. *In Proceedings of the IEEE 24th International Conference on Data Engineering*, 466–475.

[7]. Ramya, R. S., Venugopal, K. R., Iyengar, S. S., & Patnaik, L. M. (2017). Feature Extraction and Duplicate Detection for Text Mining: A Survey, *Global Journal of Computer Science and Technology*, *16*(5),1–21.
[8]. Ramya, R. S., Ganeshsingh, T., Sejal, D., Venugopal, K. R., Iyengar, S. S., & Patnaik, L. M. (2018). DRDLC: Discovering Relevant Documents using Latent Dirichlet Allocation and Cosine Similarity, *In Proceedings of the VII ACM ICNCC International Conference on Network, Communication and Computing*, 2–11.
[9]. Ramya, R. S., Darshan, M., Naveen, R., Sejal, D., Venugopal, K. R., Iyengar, S. S., & Patnaik, L. M. (2019). Efficient Batch Top k Spatial Term Search by Feature Redundancy, *In Proceedings of the IEEE Region 10 Symposium TENSYMP*, 723–728.
[10]. Diao, M., Mukherjea, S., Rajput, N., & Srivastava, K. (2010). Faceted Search and Browsing of Audio Content on Spoken Web. *In Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 1029–1038.
[11]. Venugopal, K. R., & Buyya, R. (2013). Mastering C++, *Tata McGraw-Hill Education*.
[12]. Jiang, Y. G., Wang, J., Wang, Q., Liu, W., & Ngo, C. W. (2016). Hierarchical Visualization of Video Search Results for Topic-based Browsing, *IEEE Transactions on Multimedia*, *18*(11), 2161–2170.
[13]. Latha, K., Veni, K. R., & Rajaram, R. (2010). Agf: An Automatic Facet Generation Framework for Document Retrieval. *In Proceedings of the International Conference on Advances in Computer Engineering*, 110–114.
[14]. Girgensohn, A., Shipman, F., Chen, F., & Wilcox, L. (2010). Docubrowse: Faceted Searching, Browsing, and Recommendations in an Enterprise Context, 189–198.
[15]. Le, T., Vo, B., & Duong, T. H. (2012). Personalized facets for semantic search using linked open data with social networks. In *2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications* (pp. 312-317). IEEE.
[16]. Bhat, V. H., Malkani, V. R., Shenoy, P. D., Venugopal, K. R., & Patnaik, L. (2011).Classification of Email using Beaks: Behavior and Keyword Stemming. *In Proceedings of IEEE Region 10 Conference TENCON*, 1139–1143.
[17]. Shenoy, P. D., Srinivasa, K. G., Venugopal, K. R., & Patnaik, L. M. (2005). Dynamic association rule mining using genetic algorithms. *Intelligent Data Analysis*, *9*(5), 439-453.
[18]. Zhao, J., Collins, C., Chevalier, F., & Balakrishnan, R. (2013). Interactive Exploration of Implicit and Explicit Relations in Faceted Datasets, *IEEE Transactions on Visualization and Computer Graphics*, *19*(12), 2080–2089.
[19]. Bhat, V. H., Rao, P. G., Abhilash, R., Shenoy, P. D., Venugopal, K. R., & Patnaik, L. (2010). A Data Mining Approach for Data Generation and Analysis for Digital Forensic Application. *International Journal of Engineering and Technology*, *2*(3), 313–319.
[20]. Vandic, D., Aanen, S., Frasincar, F., & Kaymak, U. (2017). Dynamic Facet Ordering for Faceted Product Search Engines, *IEEE Transactions on Knowledge and Data Engineering*, *29*(5), 1004–1016.
[21]. Viglas, S. D. (2012). A Comparative Study of Implementation Techniques for Query Processing in Multicore Systems, *IEEE Transactions on Knowledge and Data Engineering*, *26*(1), 3–15.
[22]. Zhao, B., Lin, X., Ding, B., & Han, J. (2011). Texplorer: Keyword-based Object Search and Exploration in Multidimensional Text Databases. *In Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM, 1709–1718.
[23]. Madhav, A. R. M. L. (2017). A Survey on Automatically Mining Facets for Queries from Their Search Results. *International Research Journal of Engineering and Technology (IRJET)*, *4*(3), 2656–2660.
[24]. Vidhya, K., & Saravanan, N. (2018). Enhanced Automatically Mining Facets for Queries and Clustering with Side Information Model, *Bonfring International Journal of Software Engineering and Soft Computing*, *8*(2), 01–06.
[25]. Jiang, Z., Dou, Z., & Wen, J. R. (2016). Generating Query Facets using Knowledge Bases, *IEEE Transactions on Knowledge and Data Engineering*, *29*(2), 315–329.
[26]. Bing, L., Lam, W., Wong, T. L., & Jameel, S. (2015). Web query reformulation via joint modeling of latent topic dependency and term context. *ACM Transactions on Information Systems (TOIS)*, *33*(2), 1-38.
[27]. Li, L., Zhong, L., Yang, Z., & Kitsuregawa, M. (2013). Qubic: An Aaptive Approach to Query-based Recommendation. *Journal of Intelligent Information Systems*, *40*(3), 555–587.
[28]. Basu Roy, S., Wang, H., Das, G., Nambiar, U., & Mohania, M. (2008). Minimum-Effort Driven Dynamic Faceted Search in Structured Databases. *In Proceedings of the 17th ACM Conference on Information and Knowledge Management*,13–22.
[29]. Grineva, M., Grinev, M., Lizorkin, D., Boldakov, A., Turdakov, D., Sysoev, A., & Kiyko, A. (2011). Blognoon: exploring a topic in the blogosphere. In *Proceedings of the 20th international conference companion on World wide web* (pp. 213-216).
[30]. MacAvaney, S., Yates, A., Cohan, A., Soldaini, L., Hui, K., Goharian, N., & Frieder, O. (2019). Overcoming Low-Utility Facets for Complex Answer Retrieval. *Information Retrieval Journal*, *22*(3-4), 395–418.
[31]. Van Zwol, R., Sigurbjornsson, B., Adapala, R., Garcia Pueyo, L., Katiyar, A., Kurapati, K., & Ramani, A. (2010). Faceted exploration of image search results. In *Proceedings of the 19th international conference on World wide web*, 961-970.
[32]. Roy, S. B., Wang, H., Nambiar, U., Das, G., & Mohania, M. (2009). Dynacet: Building dynamic faceted search systems over databases. In *2009 IEEE 25th International Conference on Data Engineering* (pp. 1463-1466). IEEE.
[33]. Yang, C., Teng, D., Liu, S., Basu, S., Zhang, J., Shen, J., & Han, J. (2019). Cubenet: Multi-facet hierarchical heterogeneous network construction, analysis, and mining, 1-2.
[34]. Feddoul, L., Schindler, S., & Löffler, F. (2019, September). Automatic Facet Generation and Selection over Knowledge Graphs. In *International Conference on Semantic Systems* (pp. 310-325). Springer, Cham.
[35]. Zhang, G. Q., Tao, S., Zeng, N., & Cui, L. (2019). Ontologies as nested facet systems for human–data interaction. *Semantic Web*, (Preprint), 1-8.
[36]. Guo, Z., Wei, B., Liu, J., & Wu, B. (2019, April). TF-Miner: Topic-Specific Facet Mining by Label Propagation. In *International Conference on Database Systems for Advanced Applications* (pp. 457-460). Springer, Cham.
[37]. Fukuda, T., Kikuchi, H., & Yotsukura, S. (2019). *U.S. Patent Application No. 15/845,023*.
[38]. Kikuchi, H., & Moriya, Y. (2019). *U.S. Patent Application No. 16/018,586*.

How to cite this article: Ramya R. S., Raju, N., Pushpa C. N., Venugopal K. R., Iyengar, S. S. and Patnaik, L. M. (2020). Automatic Extraction of Facets for user Query in Text Mining [AEFTM]. *International Journal on Emerging Technologies*, *11*(2): 342–350.