Interactive Search Group Mining from Annotated Web Pages

P. Sijin and H. N. Champa

Abstract Machine learning facet is a query-dependent upper layer service. Each facet is a set of items which describe and summarizes one important aspect of a query. Faceted search allows the users to search on a facet list of a query to pick out the desired one without browsing for a long time. The proposed facet mining framework is called facet mining from Annotated Documents (FMAD) integrates keyword search by category browsing and produces an interface which has several conceptual dimensions. The FMAD is designed to achieve an interactive data summarization process by liberalizing topic identification and interpretation on user side. It is an ontology-based query-dependent facet framework which follows a series of phases from initial facet weighting to facet item ranking. The FMAD produced high-quality clusters by using a context similarity approach which provides high cluster discrimination. FMAD facets automatically extracted from document description, annotated documents, and metadata records.

Keywords Facet · Metadata · Item ranking · Datapoint

1 Introduction

Facet mining is the process of exploration and discovery within an information collection of selected choices, and it is used to summarize the knowledge and information contained in the knowledge base with high-quality structured data, large databases of text-annotated objects, semantically related feature set such as WordNet for a query [1–3]. Query reformulation and query recommendation are the methods widely used by the users to explore their intention. Direct and instant answer groups are available with faceted search, e.g., cell phone groups with display size, color, OS used. In the

P. Sijin (✉)
University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India
e-mail: psijin@gmail.com

H. N. Champa
University Visvesvaraya College of Engineering, Bengaluru, India

M. S. Kaiser et al. (eds.), *Information and Communication Technology for Competitive Strategies (ICTCS 2021)*, Lecture Notes in Networks and Systems 401,
https://doi.org/10.1007/978-981-19-0098-3_53

Table 1 Query facets for query: "Watches"

Facet name	Group items
Watch category	Men, women, kids, etc.
Watch brand	Alpina, Citizen, Favre Leuba, Movado, Victrinox, etc.
Watch color	Black, brown, pink, red, etc.
Watch type	Analog, digital, quartz, etc.

case of vague, ambiguous queries which are short in nature and noisy, facets are useful by provided with a clarified list of choice of user interest. Similar to normal and approximate query results, facets are also abundant in nature. In order to avoid the facet boom, it is possible to rank the facets and could display the top-k diversified links of choice [4–7]. Table 1 shows the facet lists to item "watch" in various categories. Facet lists to watch are crawled for four categories of watch, namely watch category, watch brand, watch color, and watch type. The user can display the query facets along with top search results, since the facets reveal the different aspects of a query, the user will get an additional set of results which may relate to the given query and could be used for approximate search process [8–11].

The quality of a facet can be measured in terms of specificity and dispersion. Specificity is the quality of belonging or relating uniquely to a particular subject. Dispersion can be defined as the action of distributing something or appearance of a facet on multiple lists. The proposed dynamic facet mining process for Table 1 data generates faceted items with high ranks on an online database by a systematic ranking process based on metadata and is given in Fig. 1.

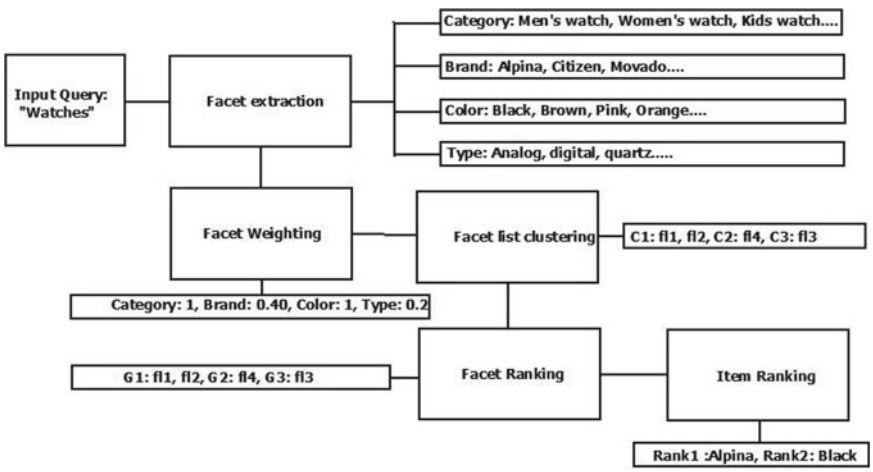

Fig. 1 Dynamic facet mining process

2 Literature Review

Takahashi et al. [12] proposed a new, user behavioral approach such as link anomaly-based detection, to detect the emergence of topics in a social network stream. The basic idea of this approach is to focus on the social aspect of the posts reflected in the mentioning behavior of users instead of the textual contents. The works by Yan and Wan [13] used a heterogeneous ranking algorithm called SRRank. A heterogeneous graph is constructed for sentences, semantic roles, and words as nodes. Using SRRank algorithm, the nodes are ranked and the nodes with the highest ranks are taken for generating summaries.

Transitional query suggestion approaches such as annotations, query logs, online summarization methods are used to design database schemas and to generate metadata attributes [8, 14–16].

Dou et al. [5] proposed a framework to list out the facets from millions of tweets called QDMiner. QDminer extracts lists from the top search results, groups them into clusters based on the features and ranks them according to how they appear in the search results.

Vandic et al. [4] proposed a framework for dynamic facet mining based on specificity and the dispersion of facet values. The proposed facet optimization algorithm ranks properties based on their importance and sorts the values within each property.

3 Methodology

The proposed facet mining process has various phases such as facet extraction, facet weighing, facet list clustering, facet ranking, item ranking. Facets are used to prepare multiple groups of semantically related queries [3].

3.1 Problem Statement

For a document Doc_i annotated with metadata M_k and a query Q_i produce facets for each Doc_i with weight Doc_{fi}, these facets are clustered and grouped to sort out top-k search items with ranks WRank_{ei} where fi is the weight of each facet associated with that document and ei is the rank of an item e.

3.2 Facet Extraction

The FMAD performs a facet extraction of metadata of documents [6, 17, 18]. It can also able to extract the required information from web forms by applying tag

Table 2 Examples for data extraction from web page

Extraction type	Facet group	Faceted data (all data not included)
Metadata	Watch brand	Shop watches from 25+ brands like Anne Klein
Tag-based	Price range	Price = 1000–10000, 1k–10k

level extraction. The metadata of the document contains lots of information about the contents and properties related to that document. The meta properties like title, type, url, image, description, site name, sign-in client-id, sign-in cookie policy, sign-in scope are useful for achieving document categorization and data security [8], [19]. Table 2 lists out the results of metadata extraction of the query "watches" over ClueWeb dataset. After list extraction, the obtained results are listed as facets such as category, brand, color, and type.

3.3 Facet Weighing

The weight of a facet in a document is calculated based on the facet count in the document and the number of shared keywords in facet keyword lists and document. Let M_k be the metadata keyword lists for a facet. If a document accounts for the presence of more number of a particular facet in it, then that document shows the high-term frequency for that particular facet; hence, facet count is valuable for facet weight (Facet_w) calculation. In order to balance the effect of common items in a document inverted document frequency (IDFE) of that document, represented as idf_e can be considered during weight calculation and the entire equation is given in (1).

$$\text{Facet}_w = \text{Doc}_w * \text{Facet}_{\text{count}} * \frac{\text{idf}_e}{|l|} \tag{1}$$

where

$$\text{Doc}_w = \frac{N_{\text{ld}}}{|l|} \tag{2}$$

where $N_{\text{ld}} \leq M_k$ is the number of shared keywords in the facet keyword list and the document used and l is the total number of lists identified as given in (2). idf_e is calculated as in (3).

$$\text{idf}_e = \log \frac{N - N_e + 0.5}{N_e + 0.5} \tag{3}$$

where N is the total number of words in the given document and N_e is the number of terms which has a direct or semantic relation to the given item e. After facet weighing normalized value list obtained is given as (Category: 1, Brand: 0.40, Color: 1.0, Type: 0.2).

3.4 Facet List Clustering

During facet clustering list with similar intentions are grouped together. If two facet lists are related proportionally distance between them can be calculated as in (4). Quality threshold with weighted data point and web page count (WQTWC) is used to reduce the overlap of clusters.

$$d(f_{l_1}, f_{l_2}) = 1 - \frac{|f_{l_1} \cap f_{l_2}|}{\min(|f_{l_1}|, |f_{l_2}|)} \tag{4}$$

The pairwise distance obtained between the facet lists is $d(f_{l_1}, f_{l_2}) = 0.2$, $d(f_{l_1}, f_{l_3}) = 1$, $d(f_{l_1}, f_{l_4}) = 0.94$, $d(f_{l_2}, f_{l_3}) = 1$, $d(f_{l_2}, f_{l_4}) = 0.91$, $d(f_{l_3}, f_{l_4}) = 1$. WQTWC algorithm sets a cluster diameter of user choice and here it is 0.96. All the points with threshold value up to 0.96 are locked in the initial cluster as given in Algorithm 1. Algorithm 1 chooses minimum weighted list $W_{\min}$ from faceted list. Then it choses a threshold diameter $\text{Diameter}_{\max}$ and randomly selected a target facet list with weight $\text{Target}_{\text{weight}}$. The pairwise distances among lists are calculated and made a core cluster for initializing the clustering. Then set $\text{Target}_{\text{weight}}$ as locus of the core cluster. After this added all faceted lists with diameter less than $\text{Diameter}_{\max}$ to core cluster. The remaining faceted lists are also clustered similarly. A lookup table table is maintained for faceted list for getting website count. Based on website count performed a re-clustering operation to produce final clusters. The algorithm produced three clusters, namely C_1, C_2, C_3, are shown in Fig. 2. The faceted lists $d(f_{l_1}, f_{l_2}, f_{l_4})$ are grouped in cluster C_1. Faceted lists $d(f_{l_3})$ are grouped in cluster C_2, and faceted lists $d(f_{l_4})$ are grouped in cluster C_3. The typical WQT algorithm works on weighted list and grouped f_{l_4} in cluster C_1. The normalized website count for the given faceted lists is calculated as (Category: 1, Brand: 1, Color: 0.6, Type: 0.01). The website count for f_{l_4} is comparatively less and that increased its threshold distance above 0.96, and it is grouped in C_3.

3.5 Facet Ranking

Facet ranking is the process of grouping lists based on their hamming distance in each cluster. The weight of a group is the number of lists in that group. Here the group is defined as the lists in a cluster which have close similarity. A list grouping approach is used for this purpose. This approach set minimum weight $W_{\min} = 1$ and

Algorithm 1: Quality threshold with weighted data point and web page count

Data: The facet keyword lists of various items.
Result: The pairwise distances among faceted list and facet clusters for the query.
initialization;
1. Choose a minimum weight $W_{\min}$ for facets.
2. Choose a threshold diameter $\text{Diameter}_{\max}$ for facets.
3. Choose a target facet with weight $\text{Target}_{\text{weight}} \geq W_{\min}$
4. Calculate the pair-wise distance among all pairs of facet lists according to (4).
5. Make a core cluster with $\text{Target}_{\text{weight}}$ as locus.
6. Add all faceted lists with weight $\leq \text{Diameter}_{\max}$ to the core cluster.
7. Go to step 2 and Cluster the remaining faceted list according to their weight.
8. Calculate website count $\text{Web}_{\text{count}}$ for each facet list.
9. Calculate the new weight by multiply (2) with $\text{Web}_{\text{count}}$.
10. Go to step 2 and restructure the clusters for new facets by feedbacking current cluster details.

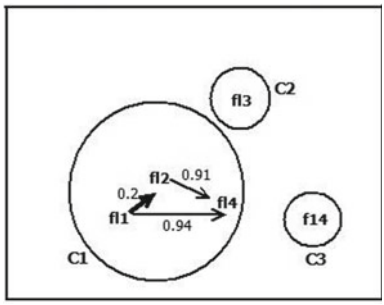

Fig. 2 Facet clustering with WQTWC algorithm

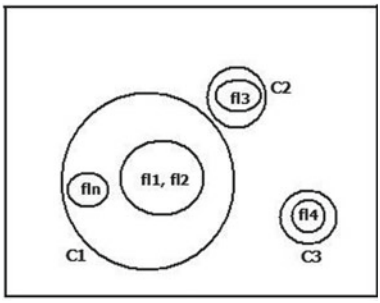

Fig. 3 During facet ranking related groups are clustered

a threshold value FLdup are identified as the similarity hamming distance, in order to form a sub-group in selected clusters. The cluster with the highest weighted lists is identified, and the highest weighted list is made as the locus of subgroups within the group. The other lists with weight less than FLdup added to the new subgroup. The same process has done to other n top-most clusters. The hamming distance for watch category $d(f_{l_1})$ and watch brand $d(f_{l_2})$ is calculated as in (5), and the subgroups formed in cluster C_1 are shown in Fig. 3.

$$\mathrm{HD}_{f_{l_1}, f_{l_2}} = 1 - \frac{\mathrm{hd}(d(f_{l_1}, f_{l_2}))}{\mathrm{LFP}} \tag{5}$$

where LFP is the length of fingerprint used by default it is 64, and $\mathrm{HD}_{f_{l_1}, f_{l_2}}$ is the hamming distance between two faceted lists (f_{l_1}, f_{l_2}).

3.6 Item Ranking

The importance of an item e in a list depends on its rank in the list and the number of lists accessing it. The rank of faceted item in a group depends on the item count in the lists of that facet. The weight of an item in a faceted list f_{l_i} is calculated as in (6).

$$\mathrm{WRank}_{\mathrm{ek}} = \sum_{f_{l_i} \in C} \frac{1}{\sqrt{\mathrm{AvgRank}}} \tag{6}$$

where average facet rank AvgRank is calculated as

$$\mathrm{AvgRank} = \frac{1}{|\mathrm{SFL}|} \sum_{l \in \mathrm{SFL}} \mathrm{Rank}_e \tag{7}$$

where SFL is the set of all list in the facet and the faceted group which contain the item e.

4 Performance Evaluation

4.1 Effect of List Weighing Method

A list which contains a term frequently in it would show a high document weight for it, but at the same time some valid facets may degrade because of less frequent appearance. If including idf_e in calculation this problem can solve. Document weight as obtained from (1) is the measure of facets which are on various lists, and it improved the facet count in lists. Figure 4 shows the effect of list weighting in facet mining.

4.2 Effect of Search Result Quantity

When the number of search results is increased, then the facet groups are enriched with more lists; these have more facets, and hence, the overall quality of facets is

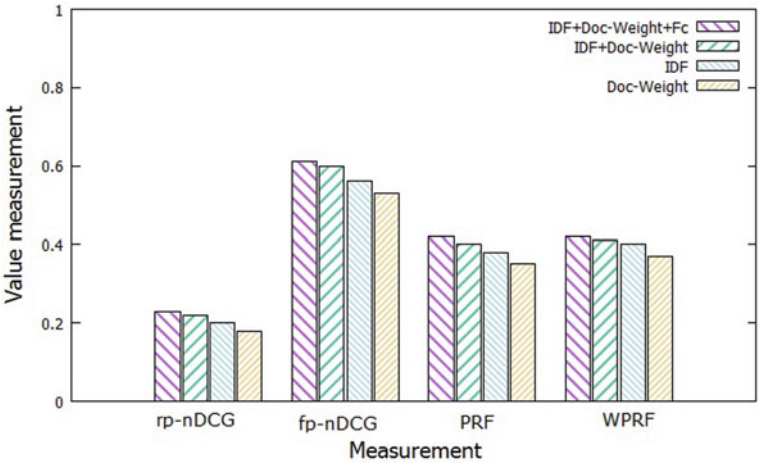

Fig. 4 Facet weight with various measurement methods

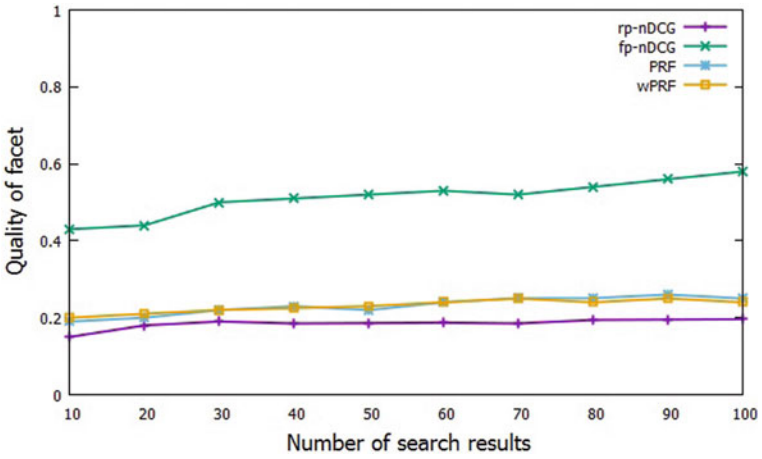

Fig. 5 Effect of search result quantity during facet mining

improved, and Fig. 5 shows this. From Sect. 3.3, it is clear that the facet count has an important role in the facet list weighing process. So if the results are more there is a possibility of their facet counts also to be more. Specific properties whose facets match many products have high impurity.

5 Conclusion

The proposed facet mining approach followed a series of phases such as facet clustering and grouping in order to organize the items, for which user is able to drill down with an intention over a navigational search platform. Because of the abundance of facet results, the FMAD followed a facet weighing and clustering scenario. Quality clusters are derived by considering disjoint facet count. Hamming distance is measured for cluster grouping by considering each facet list in a cluster. Average facet rank is calculated to list out top-k items which are specifically for query and dispersed on query dimension.

References

1. Latha KVK (2010) Facet generation framework for document retrieval. In: IEEE International conference on electro/information technology (EIT). In: 2010 International conference on advances in computer engineering, pp 602–607
2. Jiang Z, Dou Z, Wen J-R (2016) Generating query facets using knowledge bases. IEEE Trans Knowl Data Eng 29(2):315–329
3. Herdagdelen A, Ciaramita M, Alfonseca E (201) Generalized syntactic and semantic models of query reformulation. In: Proceedings of the 33rd International ACM SIGIR conference on research and development in information retrieval, pp 283–290
4. Vandic D, Aanen S, Frasincar F, Kaymak U (2017) Dynamic facet ordering for faceted product search engines. IEEE Trans Knowl Data Eng 29(5):1004–1016
5. Dou Z, Jiang Z, Hu S, Wen JR, Song R (2016) Automatically mining facets for queries from their search results. IEEE Trans Knowl Data Eng 28(2):385–397
6. Liu Q, Chen E, Xiong H, Ding CH, Chen J (2011) Enhancing collaborative filtering by user interest expansion via personalized ranking. IEEE Trans Syst Man Cybern Part B (Cybern) 42(1):218–233
7. Kim HJ, Zhu Y, Kim W, Sun T (2014) Dynamic faceted navigation in decision making using semantic web technology. Decis Support Syst 61:59–68
8. Ruiz EJ, Hristidis V, Ipeirotis PG (2014) Facilitating document annotation using content and querying value. IEEE Trans Knowl Data Eng 26(2):336–349
9. Liu J, Yan D (2016) Answering approximate queries over XML data. IEEE Trans Fuzzy Syst 24(2):288–305
10. Hahsler M, Bolaños M (2016) Clustering data streams based on shared density between micro-clusters. IEEE Trans Knowl Data Eng 28(6):1449–1461
11. Nivetha K, Ram GR, Ajitha P (2016) Opinion mining from social media using fuzzy inference system. In: International conference on communication and signal processing, pp 2171–2175
12. Takahashi T, Tomioka R, Yamanishi K (2014) Discovering emerging topics in social streams via link-anomaly detection. IEEE Trans Knowl Data Eng 26(1):120–130
13. Yan S, Wan X (2014) SRRank: leveraging semantic roles for extractive multi-document summarization. IEEE/ACM Trans Audio Speech Lang Process 22(12):2048–2058
14. Wang Z, Shou L, Chen K, Chen G, Mehrotra S (2015) On summarization and timeline generation for evolutionary Tweet streams. IEEE Trans Knowl Data Eng 27(5):1301–1315
15. Zo H, Ramamurthy K (2009) Consumer selection of e-commerce websites in a B2C environment: a discrete decision choice model. IEEE Trans Syst Man Cybernet Part A Syst Hum 39(4):819–839
16. Kules B, Capra R, Banta M, Sierra T (2009) What do exploratory searchers look at in a faceted search interface. In: Proceedings of the 9th ACM/IEEE-CS joint conference on digital libraries, pp 313–322

17. Liberman S, Lempel R (2014) Approximately optimal facet value selection. Sci Comput Prog 94:18–31
18. Vandic D, Frasincar F, Kaymak U (2013) Facet selection algorithms for web product search. In: Proceedings of the 22nd ACM international conference on information and knowledge management, pp 2327–2332
19. Kong W, Allan J (2014) Extending faceted search to the general web. In: Proceedings of the 23rd ACM international conference on information and knowledge management, pp 839–848