IPHV8I3019X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

AN EFFECTIVE AND DYNAMIC FACET SELECTION ALGORITHMS FOR WEB PRODUCT SEARCH

#1G.VANI, M.Tech Student,
#2B.UMARANI, Assistant Professor,
#3Dr.M.SUJATHA, Associate Professor,
Department Of CSE,
JYOTHISHMATHI INSTITUTE OF TECHNOLOGICAL SCIENCES, KARIMNAGAR T.S.INDIA.

ABSTRACT: In the past few years, a growing amount of e-commerce information has been published online either as Linked Open Data or embedded as Micro data or RDF a mark up inside HTML pages. Unfortunately, the usage of such data for product search and comparison is hampered by the products and services being themselves specific and heterogeneous with regard to their relevant characteristics, and by the search process that involves learning about the option space. In this paper, we present an adaptive faceted search interface over product offers in RDF. Our search interface is directly based on the popularity of schema elements in the data and does not rely on a rigid conceptual schema with hardwired product features, thereby being suitable for arbitrary product domains and product evolution. Further it supports learning during the search process. As a proof of concept of our work, we provide two use cases, namely one with product offers from an automobile database, and a second one with real product data collected from the Web.

Keywords: *E-Commerce, Product Comparison, Faceted Search, Usability, SUS, HCI, Linked Data, Semantic Web, RDF, SPARQL.*

I. INTRODUCTION

Online product search has nowadays become more important than ever, as consumers purchase more often on the Web [3]. One explanation for this is that the Web facilitates the user in finding products that better match their needs and that it offers more variation than traditional selling channels. Not only do the users have access to more information (e.g., user reviews, exact product information), they also find it easier to shop from their homes. On the other hand, because of the many options, users are often overwhelmed and find it difficult to browse through the available products.

Multifaceted search, also sometimes referred to as 'guided navigation', is a popular interaction paradigm that allows users to navigate through multidimensional data [1]. One of the main uses of multifaceted search is in the domain of ecommerce, i.e., Web shops. It is being employed to solve the parametric product search problem for Web shops that have collected local offerings and product information. For example, in a Web shop the user might enter a query like 'samsung, gps' in order to search for a Samsung phone that has built-in GPS capabilities. After showing the initial result set, most Web shopping interfaces display the facets of the products in the result set, which can be used to further drill down into the results set. The facets in this case are product attribute/value combinations. Some examples of such product facets are 'connectivity:hspda' and 'screensize:3-4 inch'. An important problem of multifaceted search is the selection of facets that should be displayed for each query. Because products have so many attributes that could be displayed as facets, Web shops usually have some static business logic to display certain facets for each result set. Although this works for local Web shops that do not have many product categories, the creation of this business logic is a time consuming process and is not appropriate for Webwide product search. One solution to this problem is to employ an optimized facet selection process. The goal of such an optimization process is to show facets that effectively partition the product search space so that the user can easily drill down and find its desired product. In literature, this is referred to as the facet selection problem, which can be expressed as the optimization of a hyperactive media link generation process [2].

In this paper, we propose new algorithms for the facet selection problem in product search. We evaluate several approaches and compare our proposed algorithms against several state-of-the-art facet selection algorithms from the literature. Our proposed algorithms aim to partition the space in the most effective manner and thus allow the user to drill down in the least amount of time. We perform the evaluation on a large data set and analyze the results, differently from previous works, across three different measures.

II.RELATED WORK

The main goal of this paper is to analyze the performance of several facet selection algorithms in a product search environment. In the literature, facet optimization is a field that has gained popularity in the last few years [4, 8, 11, 13]. The author of [5] shares her experiences from experiments on (faceted) user interface design, as part of UC Berkeley's Flamenco Search Interface project. Most of the studies investigate the optimization of the facet selection process. However, there is not much research that explores this topic in an e-commerce environment. The authors of [7] aim to make facet optimization personalized, a solution that should be able to deal with the large variety in user preferences towards facet drill down approaches. From this work we adopt several aspects of the used user modeling. The main idea is to have a collaborative filtering and personalization of the search interface, with respect to the user's behavior. The authors propose a utility-based framework, which we also adopt in this paper. The authors of [9] investigate how facet optimization can be performed for general-purpose Web documents. Three different strategies for simulating a faceted search session are employed in the evaluation. In this paper, we benchmark our proposed methods against those presented in [9], using the same simulation strategies. The reason for this is that the strategies cover a wide range of type of users and that it allows for a more fair comparison than we would have come up with new simulation strategies. Similar to [7] and [9], which are directly related to our work, there are other studies that have investigated the use of facets in information retrieval. The authors of [14] focus on column-wise faceted search. This approach allows the user to drill down in a hierarchical fashion, going from left to right. The authors claim that this has an advantage as users can make use of both directional and non-directional faceted search. We do not include this approach in our comparisons as we do not focus on directed faceted search. The authors of [6] investigate how the use of Semantic Web technology can aid in multi-faceted product search. The authors propose an information search and retrieval framework based on a semantically annotated multi-faceted product family ontology. The key idea in their approach is to use a document profile model that suggests semantic tags for the annotation process. Because the approach presented by these authors requires a product family ontology, we did not include it in our comparisons (our approach is more flexible because we do not require product family onto logies). The study presented in [10] focuses on an approach where both the querying and navigation of Web documents are considered as equally important activities. In this paradigm, querying is seen as a way to identify the starting points for navigation, and navigation is guided based on information obtained from the query. The authors present a formal model for this paradigm and report empirical results obtained from experiments over a large Web corpus. The results show that in the case of ambiguous queries, the proposed retrieval model identifies good starting points for post query navigation. However, for queries that are less ambiguous, the output tends to match that of a conventional retrieval system. Because this approach deals with Web documents and hyper-links, we exclude it from the comparisons, as our approach does not involve hyper-links but instead just uses directly product/facet associations.

III. REQUIREMENTS FOR FACETED SEARCH

This section defines important requirements for product search that can to a large extent be readily met by faceted search interfaces over RDF data.

Regard multi-dimensionality of products: The complexity and dynamics of products and services necessitate multi-parametric searches based on distinguishing properties and attributes of product entities, which, on the Web of Data, can be realized by considering the structure of the available data. – Support learning about the option space: Search is an iterative, incremental learning process (e.g. [12, p. 9]) rather than a static, one-shot query. For example, users grasp new information about the option space in every search turn [5], possibly leading to changes in price expectation. Thus, users need a way to relax or refine their constraints and preferences based on how those modify the size of the option space. – Facilitate incremental, user-driven schema alignment: For product search with incremental learning, it is not only vital to assist in navigating and pruning the option space, but also to actively engage the user in the search process. Since users are likely to learn about correspondences in the underlying product features during the user interaction, the approximate alignment of conceptual elements should be integrated in the iterative search process, and be fed back to the graph. E.g., a user interface could ask the user for approval of a possible match between two product features. In an RDF environment, corresponding axioms can be easily added to the existing data as named RDF graphs – potentially managed on a per-user basis. – Take into account the popularity of conceptual elements in the instance data: A user interface that is solely based on the schema elements defined in the underlying ontologies is inefficient, because the user lacks information about the availability of matching data (e.g. whether a property is used at all) and the relevance of a constraint on the option space (e.g. whether products differ in that property). Due to a sparsely populated graph of product information on the Web, efficient user interfaces should thus adapt to the actual usage of schema elements in the data rather than be based on schema definitions. – Utilize metrics for the efficiency of the search process: An efficient search interface presents choices to the user that help to quickly

IPHV8I3019X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

narrow down the option space, e.g. by proposing discerning features that partition the option space in the best possible way, or by suggesting properties that promise the highest utility to a given user need. The user dialog in faceted search is fundamentally a decision tree problem, where the user interaction steps are branches of the tree. Because the facets are orthogonal to each other, the decision tree can be constructed in any order [13]. However, if we want to optimize the search efficiency for the user, we have to create and, if necessary, update the resulting tree based on a "best split" strategy known from decision tree research in data mining [18, p. 158]. Popular algorithms from literature, e.g. ID3 [14], iteratively choose attributes maximizing the information gain. In this context, [10] mention some popular facet-pair suggestion strategies, namely relying on frequency, probability, and the information gain. The authors in [22] further give an overview over different metrics appropriate for product search to help decide which facets to present to the user.

IV.EVALUATION

In this section, we first discuss the used data set and the experimental design, before presenting the evaluation results. We use a data set that is gathered from Kieskeurig.nl, the largest price comparison site in the Netherlands. This service does not only provide price comparisons, but also has very detailed information on products. For this evaluation, we focused on consumer electronics and chose mobile phones to be the category of products that we use in the experiments. The data set contains 980 products for which we have key/value pairs, i.e., product attributes. All product information is in Dutch, but should be understandable also for non-Dutch speaking people because of the frequently used English terminology in the product attributes. Using the product attributes, we created the facets using the following rules. A facet is a combination of a product attribute and a value (or range of values). Binary product attributes, such as 'GPS', were mapped to one facet that was in the form of 'GPS:yes'. For product properties that represent multivalued qualitative values, such as 'Supported Video formats', we created a binary facet for each value. Similarly, for single-valued qualitative product attributes, we created a single facet for each value. For all the quantitative properties, we manually defined the ranges that would represent the different facets. As a result of this facet creating process, we obtained 487 facets for the 980 products. The size and variety of this data set allows for a thorough evaluation of the facet selection algorithms that we propose in this paper.

Experiments. For the evaluation, we simulate a user that is in a faceted search session. There are two aspects that are important in this type of simulation. First, we need a way to generate queries that are sufficiently realistic for the experiments. Second, we need one or more simulation strategies of users in order to simulate the clicking on a facet. Before we go into the details of these two aspects, let us first explain on a high level how we have designed the simulation. Given a query, we submit it to the product search engine, after which, for every product that we consider as a possible target product, we simulate a faceted search session. In this paper, the set of possible target products consists of the first 100 products after the top-m products. We set m = 10, which results in performing the simulation with each product ranked in the range [11, 111] as a target product. The reason for this is that we want to measure how the algorithms perform for many different target products. Next, the ranked search results are obtained and a faceted search session is simulated, where a user is aware of the target product, but is only able to recognize it when it appears in the top-10 results. The user keeps clicking on a facet (described shortly) until either the target product appears in the top-10, or the target product disappears from the result set. The latter can happen if the user simulation strategy does not assume that the user has perfect knowledge of the target product and therefore sometimes clicks on a facet that does not belong to the target product. In each simulation of a faceted search session, we record how many products the target product has been promoted over for each click. We repeat this process for $l \in \{3, 5, 7\}$, i.e., the number of facets to be displayed. For the estimation of the target document identity probability $p(d = d_q)$, we employ a Zipfian distribution, as the authors of [9] have done. For $j \in \{11, \ldots, 111\}$, we have $p(d_j = d_q) = \gamma \cdot j$ -2 , where dj is the product returned by the search engine at rank j for query q, and γ is a normalization constant such that P d_j $p(d_j = d_q) = 1$. Queries. For the queries, we used the product attributes to manually create a list of terms of all sensible qualitative values, such as 'iOS', 'autofocus', 'flash', and 'led'. Using this terms list, we created 1000 queries that each consisted of three terms. We ensured that no queries were generated that had an empty result set. In order to avoid trivial drill down, we chose to filter out queries that have a result set with a size of less than 20. In our experiments, we use disjunctive semantics for the queries in which at least one of the terms needs to be present in a product for it to appear in the search results. Products that match on more terms are ranked higher.

User Simulation Strategy. For the user simulation strategies, we adopt the three approaches used in [9]. The first simulation strategy that we consider is the Conjunctive User Strategy. This strategy assumes that the user has perfect knowledge about the target product and that it selects all facets in one drill down. In a user interface, this is usually achieved by allowing the user to select multiple facets using a check box. The Best Facet User Strategy is the second simulation strategy that we consider. It also assumes that the

IPHV8I3019X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

user has perfect knowledge, however, instead of choosing all facets that correspond to the target product, it selects the single facet that promotes the target product the most. The third simulation strategy that we consider is the Probabilistic User Strategy. This strategy differs from the other two because it does not assume that the user has perfect knowledge. It involves a probabilistic process where the user scans the facets in the order that they are presented and the choice whether to click or not is modeled using two Bernoulli distributions, depending on whether the facet is actually associated to the target product or not. For facets that are associated to the target product we set p = 0.9 and for facets that are not associated to the target product we set p = 0.1. This means that a user might select a facet that does not belong to the target product with 10% probability.

Implementation. We provide a faceted search engine implementation (http://faccy.net) that supports the same query semantics we used in the experiments, as well as all the algorithms that we have evaluated in this paper. The user can enter a query, separated by comma's to perform a disjunctive query, after which the facets will be shown. By default, the method 'All Facets' is used, which simply shows all possible facets. The user can click at any point in time on a different facet selection algorithm, shown at the top of the page. Both the experiments and the Web application have been implemented in Python, using MongoDB as the database.

Experiments This paper investigates the appropriateness of faceted search interfaces for the Web of Data. To test for two fundamental aspects of search interfaces, namely search efficiency and usability, we first measure the impact of specificity in product search on the size of the result set using a simulation of random walks. Then, we conduct a usability study where we contrast a data-driven, adaptive faceted search interface with a second alternative with hard-wired product features.

4.1 Impact of Search Specificity on the Size of the Result Set

We simulated a number of product searches to find out how dispersed the search space for products is and how well a faceted search approach on average performs regarding partitioning the option space.

Method. We took a random sample of 875 automobile offers3 from the mobile.de car listing Web site. We extracted the product features from the respective Web pages and populated an RDF graph via mapping product features to properties from the VSO ontology4 . For the sake of simplicity, we did not take into account quantitative values for our simulation, but only qualitative and data type properties. The variety of qualitative and data type properties over the whole dataset is shown in Table 1. These numbers give a total of 113 possible property-value pairs. From this range of possible property-value combinations, we drew one item at random and started from their 100 random walks with each simulating ten consecutive selection steps. After every selection step, we randomly picked a property-value pair from the reduced option space, which we obtained by issuing a proper SPARQL query.

Property	Variety of Values
http://purl.org/vso/ns#bodyStyle	6
http://purl.org/vso/ns#color	24
http://purl.org/vso/ns#condition	5
http://purl.org/vso/ns#feature	60
http://purl.org/vso/ns#fuelType	10
http://purl.org/vso/ns#meetsEmissionStandard	5
http://purl.org/vso/ns#transmission	3

Results. Figure 1 outlines the results of our simulation. At the beginning (step 0), the option space always entails the full range of 875 car offers. In search step 1, the median of the 100 iterations already goes down to circa 150 results, i.e. in 50% of the cases the first filtering step sorts out an average of more than 700 out of 875 automobiles. After having selected three product features, the median of the option space decreases to only three items. As a possible constraint, our random walk does not include UNION clauses, i.e. the disjunctive selection of multiple facet values which would expand the option space (e.g. select a car that offers either manual or automatic transmission). However, we argue that this expansion operation does anyway occur rarely in practice when users seek interesting product offers.

Discussion. We can see clearly from the analysis that the space of possibly matching products decreases logarithmically with the number of features specified in a query. This confirms our assumption that learning about the option space, i.e. how relaxing and refining requirements and preferences based on the set of remaining choices, is a critical part of product search interaction. It also highlights that in specific branches of product search and thus sparsely populated decision trees, a search interface can benefit from being dynamically generated directly from the data about products and their characteristics. Of course, the findings presented are currently based on a single sample data set of 875 cars, albeit those have been selected randomly from a very significant real dataset from a car sales portal. The effect of the number of features might be less significant if we took into account the correlation of features (e.g. that a stronger engine is likely to be found in combination with more seating capacity), which we deliberately abstracted from by selecting the features randomly. We would counter, however, that exactly these correlations between product features are unknown ex ante to a person exploring a product space and thus stress the importance of the learning effect of iterative product search.

ISSN Online: 2319 – 9253
Print: 2319 – 9245

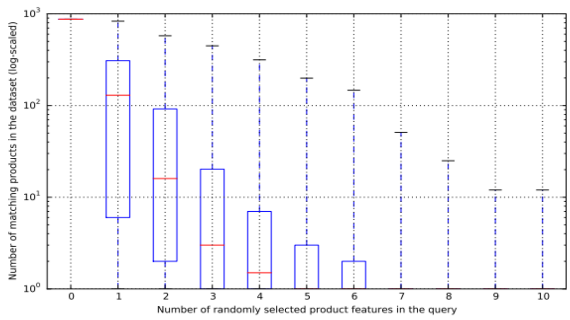

Fig. 1. Random walk simulation over a decision tree for 875 automobile offers

4.2 Usability Studies of Faceted Search

Interfaces for Products Faceted search interfaces have recently attracted significant research interest. Various demonstrators, user studies, and evaluations repeatedly attest them superior usability in contrast with other search paradigms. In a survey in [23], the authors systematically compare faceted search with other popular search paradigms.

In here, we conduct a user study in order to find out whether an instance driven search interface has a negative impact on usability, because hard-wired, consolidated user interfaces found in today's commercial faceted search applications have the advantage that the displayed facets can be based on popular mental models of human users. Instance-driven, adaptive faceted search interfaces bear the risk of being confusing to users, as the facets and facet names presented to users may change dynamically depending on the available data.

Method. In order to evaluate a potentially negative effect, we first developed an instance-driven, adaptive faceted search interface5 for product comparison on the Web of Data that addresses the requirements outlined in Sect. 2. Then, we prepared an identical search interface except for relying on hard-wired product

Table 2. Results of SUS experiments

	Students		Crowdsourcing	
	A	B	A	B
No. participants	39	29	50	50
No. incorrect answers	5	3	13	9
No. answers considered	39	29	37	41
Avg. SUS score	66.54	72.59	65.00	68.75

features6. As the data to present in our two search interfaces, we used a random subset of 25 car offers out of the random sample of 875 car offers from mobile.de. We set up a usability study according to the System Usability Scale (SUS) [7] score. The questionnaire encompasses ten brief questions where each response is represented by a five-point Likert scale ranging from strongly agree to strongly disagree. SUS questions are designed to alternate between positive and negative statements. In addition, we included a gold question to filter out unreliable candidates based on an incorrect response. We placed the gold question at the end of the questionnaire. Otherwise, we feared that participants would possibly give up too early, because it required a bit of effort to look at the information displayed in the search interface. Finally, we asked for optional feedback, which we used in a later analysis for interpreting the results. We put the questionnaire online so that users could test the search interface and answer to questions remotely.

We conducted two separate usability studies. The first one we ran with undergraduate students from our University, who specialize in business management or related fields. They were asked to assess the usability of the original, dynamic search interface A and, later, to repeat the same task with the amended search interface B. Our second experiment was harnessing crowd workforce from the Crowd Flower platform. As compared to the students experiment, we ran the usability test for both search interfaces A and B in parallel with two distinct groups of participants.

Results. In the following, we report on the empirical results obtained from the two usability studies, as summarized in Table 2.

Usability Experiment with Students. The task completion rate (cf. [17]) for students was 34/39 = 87% for search interface A, and 26/29 = 90% for search interface B. For students' ratings, we decided against eliminating incorrect answers to the gold question, because a closer investigation of individual responses revealed that students were not fooled by the alternating pattern of SUS questions rotating between positive and negative statements. Search interface A achieved an average SUS score of 66.54, which is slightly below the average of 687 , which was the mean SUS score among 500 system usability studies. Taking on the qualitative, "adjective" rating introduced in [4], the search interface is considered good (SUS score close to 71.4). By comparison, search interface B obtained an average SUS score of 72.59. We stated the following null hypothesis to test the difference in the usability scores for significance:

Null Hypothesis. There is no difference among SUS scores for search interfaces A and B obtained by two student samples from the same population.

A Shapiro-Wilk test revealed that we cannot assume that both SUS score samples are normally distributed (p-values of 0.03 and 0.06), thus we compared the two samples using a non-parametric statistical test, the Wilcoxon rank-sum test. The average usability scores assigned by our students to search interface A (median = 70.00) did not differ significantly from usability scores assigned to search interface B (median = 75.00), $W = -1.45$, $p = 0.15$, $r = -0.18$.

Usability Experiment with Crowdsourcing. Unlike in the previous experiment, we did only accept contributions by crowd workers who correctly answered the gold question.

The task completion rate for crowd workers was 37/50 = 74% for search interface A, and 41/50 = 82% for search interface B. Search interface A achieved an average SUS score of 65.00, which is below 68, but still good according to [4]. Search interface B obtained an average SUS score of 68.75. The null hypothesis below was used to test whether the usability scores significantly differ:

Null Hypothesis. There is no difference among SUS scores for search interfaces A and B obtained through two different samples of crowd workers. A Shapiro-Wilk test revealed that we cannot assume that both SUS score samples are normally distributed (p-values of 0.13 and 0.01), thus we compared the two samples using a non-parametric statistical test, the Wilcoxon rank-sum test. The average usability scores assigned by the first group of crowd workers to search interface A (median = 65.00) did not differ significantly from usability scores assigned to search interface B by the second group of crowd workers (median = 73.75), $W = -1.30$, $p = 0.19$, $r = -0.15$.

Discussion. This analysis shows that, in principle, a fully dynamic search interface directly based on product features found in the data, is not systematically less intuitive for users than one based on established, hard-wired product features used in existing car portals. However, we see a small negative effect in usability, which we expected, because the static, hard-wired set of search dimensions allows a higher degree of users' familiarity with the terminology and conceptual model of a search interface. We conclude from that small negative effect that a data-driven search interface for products comes at a cost, which must be compensated for by additional gains in precision, recall, and eventually the utility of the finally selected product.

We would also like to stress that a usability-based evaluation of novel search interfaces has a systematic weakness, because it only analyzes how well a user can handle the interface, but not the quality of the choices eventually made (e.g. how well the finally selected product meets the user's needs). As we have shown in the first part of this section, the sparsity and heterogeneity of the product space indicates that a more precise navigation in the option space can return much better product matches.

V. CONCLUSION

In this paper, we focused on automatic facet selection in the domain of e-commerce, for the purpose of minimizing the number of steps required by the user in order to find its desired product. We proposed several facet selection algorithms, which we evaluated against the state-of-the-art algorithms from literature. Furthermore, we implemented all considered facet selection algorithms in a freely available Web application called faccy.net.

The small-scale usability study in this paper also indicates that users apparently have gotten used to search interfaces that expose rigid navigation structures optimized for individual application domains. While viable in smaller and controlled settings, it is not feasible for e-commerce over Linked Open Data, where diverse and dynamic product domains need to be consolidated. A large-scale evaluation with real e-commerce data from the Web is planned for future work.

REFERENCES

1. Abel, F., Celik, I., Houben, G.J., Siehndel, P.: Leveraging the Semantics of Tweets for Adaptive Faceted Search on Twitter. In: Proceedings of the 10th International Semantic Web Conference (ISWC 2011). pp. 1–17. Springer Berlin Heidelberg, Bonn, Germany (2011)

2. Arenas, M., Cuenca Grau, B., Kharlamov, E., Marciuška, S., Zheleznyakov, D.: Faceted Search over Ontology-Enhanced RDF Data. In: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM 2014). pp. 939–948. ACM, Shanghai, China (2014)

3. Arenas, M., Cuenca Grau, B., Kharlamov, E., Marciuška, S., Zheleznyakov, D.: Towards Semantic Faceted Search. In: Poster Proceedings of the 23rd International World Wide Web Conference (WWW 2014 Companion). pp. 219–220. ACM, Seoul, Korea (2014)

4. Bangor, A., Kortum, P., Miller, J.: Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. Journal of Usability Studies 4(3), 114–123 (2009)

5. Bates, M.J.: The Design of Browsing and Berrypicking Techniques for the Online Search Interface. Online Information Review 13(5), 407–424 (1989)

[5] M. Hearst. Design Recommendations for Hierarchical Faceted Search Interfaces. In ACM SIGIR Workshop on Faceted Search, pages 1–5, 2006.

6. S. Johnson Lim, Y. Liu, and W. Lee. Multi-facet product information search and retrieval using semantically annotated product family ontology. Information Processing & Management, 46(4):479–493, 2010.

7. J. Koren, Y. Zhang, and X. Liu. Personalized interactive faceted search. In Proceedings of the 17th international conference on World Wide Web (WWW 2008), pages 477–486. ACM, 2008.

8. B. Kules, R. Capra, M. Banta, and T. Sierra. What do exploratory searchers look at in a faceted search interface? In Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2009), pages 313–322. ACM, 2009.

9. S. Liberman and R. Lempel. Approximately optimal facet selection. In Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC 2012), pages 702–708. ACM, 2012.

10. S. Pandit and C. Olston. Navigation-aided retrieval. In Proceedings of the 16th International Conference on World Wide Web (WWW 2007), pages 391–400. ACM, 2007.
11. G. Sacco and Y. Tzitzikas. Dynamic Taxonomies and Faceted Search, volume 25. Springer, 2009. [
12. C. Shannon, W. Weaver, R. Blahut, and B. Hajek. The Mathematical Theory of Communication, volume 117. University of Illinois press Urbana, 1949.
13. D. Tunkelang. Faceted search. Synthesis Lectures on Information Concepts, Retrieval, and Services, 1(1):1–80, 2009.
14. M. Wilson, P. Andr´e, et al. Backward highlighting: Enhancing faceted search. In Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST 2008), pages 235–238. ACM, 2008.
15. Wei, B., Liu, J., Zheng, Q., Zhang, W., Fu, X., Feng, B.: A Survey of Faceted Search. Journal of Web Engineering 12(1), 41–64 (2013)