Evaluation of a Search Interface for Preference-Based Ranking - Measuring User Satisfaction and System Performance

Dagmar Kern
GESIS - Leibniz-Institute for the Social Science
Cologne, Germany
dagmar.kern@gesis.org

Wilko van Hoek
Bonn, Germany
wilko.vanhoek@gmail.com

Daniel Hienert
GESIS - Leibniz-Institute for the Social Science
Cologne, Germany
daniel.hienert@gesis.org

ABSTRACT
Finding a product online can be a challenging task for users. Faceted search interfaces, often in combination with recommenders, can support users in finding a product that fits their preferences. However, those preferences are not always equally weighted: some might be more important to a user than others (e.g. red is the favorite color, but blue is also fine) and sometimes preferences are even contradictory (e.g. the lowest price vs. the highest performance). Often, there is even no product that meets all preferences. In those cases, faceted search interfaces reach their limits. In our research, we investigate the potential of a search interface, which allows a preference-based ranking based on weighted search and facet terms. We performed a user study with 24 participants and measured user satisfaction and system performance. The results show that with the preference-based search interface the users were given more alternatives that best meet their preferences and that they are more satisfied with the selected product than with a search interface using standard facets. Furthermore, in this work we study the relationship between user satisfaction and search precision within the whole search session and found first indications that there might be a relation between them.

Author Keywords
Search interface, Information filtering, Preference-based ranking, Weighted facets, Evaluation of whole sessions

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces – Evaluation/methodology, Graphical user interfaces (GUI); H.3.3. Information Storage and Retrieval : Information Search and Retrieval – Query formulation

INTRODUCTION

The number of consumers who browse or buy products online grows further [10], and they are all facing the challenge to choose the best option out of a huge set of alternatives. Web shops and service providers (e.g. hotel booking services or apartment finders) try to support users in finding the right product. One well-established method for that is to provide search facets in the user interface. [8]. Facets allow users to filter products by predefined categories or features. In this way, users can exclude products they are not interested in and obtain a smaller and more manageable number of alternatives. An overview about faceted search in general is given by [21, 23]. Studies have shown that search interfaces providing facets are considered as intuitive and easy to use, see e.g. [8]. Furthermore, they provide a high level of control and transparency [24]. However, user preferences can not always be encoded into a boolean logic. Sometimes, some features are not mandatory but nice to have and some of them are considered more important than others. But, using facets means that all selected facet terms are mandatory and equally weighted. Selecting less important facet terms may remove potentially interesting products unintentionally, while specifying only a few criteria may lead to a too large result set where interesting alternatives are not obvious [22].

To counterbalance disadvantages of facet search, recommender systems [17] are often additionally applied to suggest possible alternatives that might fit users' preferences. In current systems, the recommendations rely on user profiles and the use of automated recommendation techniques [17]. In general, common commercial recommender systems offer no or little options for users to explicitly influence the recommendations, leaving them no opportunity to express their preferences. However, in the research literature, one can find some evidence that, regarding the user satisfaction, it is beneficial to put users in control of their recommendations (e.g. [1, 7]) and of the ranking process of search results in general (e.g. [14]). One possibility to give users control over recommendation and ranking is to let them weight terms according to their preferences (e.g. [11, 22]). From a user perspective, these systems have been evaluated as very helpful, and they are able to increase user satisfaction (e.g. [4, 7]). However, little attention has been given so far considering both sides to evaluate a search system – user feedback as well as system performance and their relationship to one another. We want to close this gap by providing new insights about a search interface that uses preference-based ranking in the form of

weighted facet terms. We want to know if the user's perceived satisfaction and system support can be backed up by analyzing system performance measures. For that purpose, we tracked the changes in recall and precision over the course of whole search sessions.

RELATED WORK

Most of the existing commercial search interfaces using facets still offer few opportunities for the user to adapt the search query explicitly to her preferences. However, in research, several attempts have been made to include user preferences in *product search*. For example, Stolze [20] proposed a soft navigation approach for finding products in an electronic product catalog. He distinguished between hard and soft constraints and allows weighting the importance of product features. The proposed system requires from the user to learn a rule based syntax. Therefore, it is stated to be more suitable for frequent users. However, in the research field of *search interfaces* using facets, there are surprisingly few approaches focusing on allowing the user to express their preferences through weighting facet terms. Han et al. [6] focus in their people search system on using slider-based facets to specify the importance of three predefined categories. Their results show that the users consistently interact with the sliders to fine-tune the result ranking to achieve a better ranking. An approach very similar to ours is presented by Voigt et al. [22]. They distinguish in their VisBoard application between must-have and optional facet terms that can be weighted by the user with drag-and-drop in a configuration area. They utilized this approach for the specific task of selecting a visualization component based on its characteristics. Unfortunately, they only performed a preliminary user study with five participants showing the general potential of this approach to support the user expressing her preferences.

In the field of *recommender systems*, a lot of research approaches show the benefit of including user's preference to increase recommendation accuracy, user satisfaction and user experience. Critiquing-based recommender systems (e.g. see [4, 2]) allow users to interactively criticize recommendations and thus put the users in control of finding a product that fits their preferences. With the possibility to weight critiques (e.g. "less expensive" or "compromise distance"), a user can perform trade-offs while searching for products, a concept which can still be very rarely found in commercial systems. Studies have shown that the critiquing-based approach leads to a higher decision accuracy compared to non critiquing-based systems such as a ranked list with one ranking criteria at a time [15, 16]. In TasteWeight [1] users can adjust their taste interactively during a recommendation session via slider-weights components. Thus, they weight suggested recommendation terms and influence directly the recommendations. Results of a user study showed a positive effect on user satisfaction. Harper et al. [7] also put the user in control of her recommendations by providing means for tuning system generated recommendations according to her preferences (e.g., "show more popular items"). Results of their user study show that if users have control over a recommender system, they evaluate suggested recommendations more positively than automatically generated recommendations. A study with SetFusion [13], an interactive system that allows weighting the influence of three different recommender algorithms, also shows that due to the interaction and visualization, the users have a greater sense of perceived control and transparency. The approaches above focus primarily on ranking or criticizing a generated set of recommendations and recommended terms. Loepp et al. [11] in contrast, integrate different recommender algorithms with several interactive filter techniques in one hybrid recommender system. This system allows the selection of hard and soft filter criteria from different facets by the user. The selected facet values can be weighted by the user and serve as input for collaborative and content-based recommender techniques. Results of a user study showed that users feel more in control with the hybrid recommender system than with a standard faceted filtering system. They find the interaction to be more appropriate for generating recommendations.

In the field of *information retrieval* different concepts have been proposed to involve the user's preferences more interactively in the ranking of search results. A core concept is relevance feedback [18], in which the user implicitly or explicitly influences the ranking by marking some result items more relevant than others. Another concept that is strongly related to relevance feedback, is query expansion [3]. Here, user search terms are expanded with additional terms originating from knowledge sources, search results, the document corpus or the user's history. These expanded terms are often assigned a different weight in the overall query, to balance the effect of query expansion. On the user interface side, Frei and Qui [14] allow the user to weight query terms in the context of document retrieval. They showed that weighted queries perform significantly better than Boolean retrieval regarding usefulness and precision. In the context of a digital library, an approach for conditional weighting like preference A is more important than B was formally introduced by [19].

In this paper, we build up on different concepts of the works above: allowing users to enter query terms, a recommender for facets, the possibility to weight free and facet terms, a certain fuzziness for specific facets and a highly interactive search interface. We especially concentrate on a thorough evaluation measuring user feedback and system performance based on the whole search session.

HOTEL SEARCH – AN EXAMPLE APPLICATION FOR A PREFERENCE-BASED RANKING SYSTEM

Searching for a hotel includes a wide variety of facets (hotel features) on the system side as well as some different preferences on the user side. We chose this as a use case for studying the potential of a preference-based ranking system. Our search interface allows optional search terms to enhance the result list with alternatives which do not necessarily fulfill all of the user's preferences. The user can specify which of her preferences have to be matched exactly ("must-have terms") and which preferences are nice to have ("optional terms"). Furthermore, the user can explicitly exclude hotels containing features which are not wanted ("must-not have"). "Must-have" and "must-not have" terms cause a trimming of the result list, while the weighting of "optional terms" influences the ranking so that the best-matched hotels are at the top.

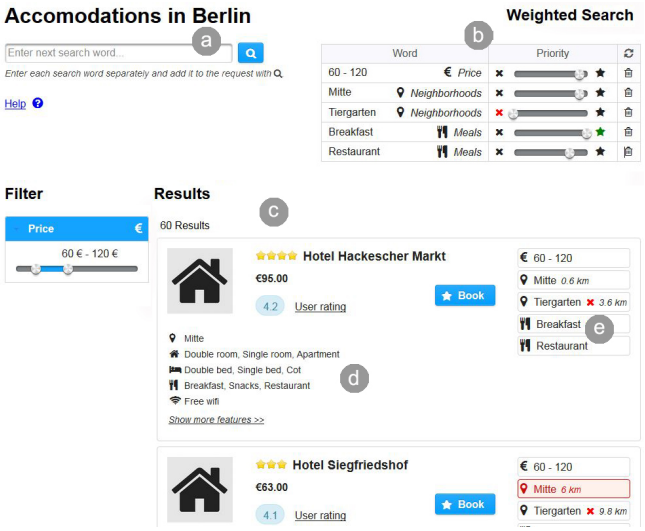

Figure 1. The preference-based hotel search interface consists of a) input field, b) weighting area, c) result list, d) list of product features, e) visual feedback on matched or mismatched search terms.

User Interface

The search interface consists of three main components: input field, weighting area, and search result list. Figure 1 shows the implemented interface. In the input field, the user can enter her preferences one after another (Figure 1a). While typing, a recommender suggests a list of facet terms or facet categories that match the current character string. If the search term does not match a facet term, it is marked as a free text term. In the weighting area (Figure 1b) all search terms which formulate the search query are listed with the possibility to weight each of them. Initially, all search terms have the same almost highest weight. This means that all products are still in the result list. With the provided slider the user can weight the impact of each term. By setting the slider to the minimum of the scale, the search term is considered as a "must-not have" and by setting the slider to the maximum of the scale the search term is considered as a "must-have". With criteria marked as "must-haves" and "must-not haves" the number of results can be decreased. With a click on the recycle bin icon, the search term can be easily removed from the search query. Any interaction in the weighting area leads directly to a new calculation of the search results (Figure 1c) and provides immediate feedback to the user. On the left side of each result item, information on all important hotel features are shown (Figure 1d). On the right side matched and mismatched search terms are visualized (Figure 1e). This gives the user the opportunity to judge the agreement of search results with her preferences and provides transparency over the ranking process.

Determining the Result Set

The weights for each search criterion are mapped to floating point numbers between zero and one. This weighting factor is used to determine which hotel will be included in the result set. If the weight is set to one, a hotel will be retrieved if it satisfies the required criterion. In contrary, if the weight is set to 0, the weighting acts as a NOT operator, in which hotels that satisfy

		Criterion Classes	
Class	*Operator / Function*	*Description*	*relevance score (rs)*
Text	FULL_TEXT	Default full text or phrase search.	tf-idf value
Nominal Facet	EQUAL	Filter criteria like "Breakfast included", "Double room", "Single room", etc.	1
Numerical Facet	(HALF-) GAUSS, (TRI-) LINEAR	Fuzzy criteria, like geo distance, price, etc.	Result of applied function

Table 1. Criterion classes and relevance scores.

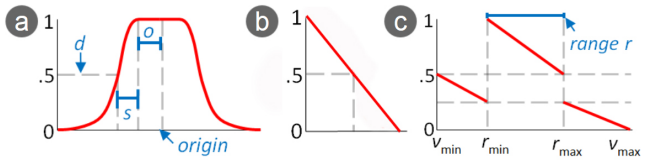

Figure 2. a) Customizable Gaussian function for relevance scoring, b) linear function applied on directed Likert-scales, c) Tri-linear descendant directed scoring function.

the criterion will be eliminated from the result list. This allows users to exclude unwanted criteria. In both cases, hotels will not be included if they lack the given criterion. These cases can be considered as filtering the result set.

Ranking Model

Each hotel in the result lists gets a summed relevance score (*srs*) according to that *srs* the result list is ordered. The hotel with the highest *srs* is on top of the list followed by the others in descending order. The summed relevance score for each hotel is computed as $srs = \sum_{i=1}^{n} w_{Ci} * rs_{Ci}$, whereas (C1-n) are the hotel's criteria the search terms are mapped to. w is the user defined weight for each criterion and *rs* a relevance score that depends on the criterion class the search term is mapped to (see Table 1). Search terms that cannot be mapped to the given facets are assigned to the class *Text*, for which a tf-idf scoring is applied to calculate a relevance score taking all textual description of the hotel into consideration. A hotel criterion that fulfills a nominal facet (like "breakfast" or "single room") gets a relevance score of 1. For criteria that fulfill numerical facet terms, we suggest using a fuzzy relevance scoring approach whereby hotels that fully match the stated criterion get the highest *rs* and hotels whose feature fall in a range around the stated value get a lower *rs* according to the applied function. We propose the usage of a "Gaussian", "linear" or a "tri-linear" scoring function. Figure 2a shows a customizable Gaussian function that can be applied to a criterion for which a single value is given. An optional offset parameter can widen the range where the score will be the maximum. A linear function can be applied where a maximum or minimum is stated (shown in Figure 2b), for example, for user ratings . In case a scoring is applied to a criterion given by a range r (e.g. "price from $100-$120") a tri-linear function can be used which primarily favors the selected range and secondly the border ranges in order of direction (Figure 2c). Here, each range is valued differently, so that one range border gets a higher value than the other. In our price example, this means that the lowest price of the selected range gets the

Search term	*Category*	*Hotel's Criterion*	*User Weight (w)*	*Relevance Function*	*Relevance score (rs)*
60-120€	Price	95 €	0.9	Tri-linear	0.71
Mitte	Neighbor-hood	Mitte (0.6km)	0.9	Gauss	0.69
Tiergarten	Neighbor-hood	✓	-	-	-
Breakfast	Meals	✓	1	EQUAL	1
Restaurant	Meals	✓	0.8	EQUAL	1

srs (Hotel Hackerscher Markt) = 0.9 * f_{Tri}(95) + 0.9 * f_{Gau}(0.6) + 1 + 1 = 3.26

Table 2. Example mapping from the search terms to the hotel criteria of the first hotel in the results of Figure 1 including provided user weights, applied functions and calculated relevance scores.

highest relevance score (*rs*). The *rs* decreases linearly until the highest price of the selected range is reached. Prices that do not fell into the price range get lower *rs* whereby lower prices get higher *rs* than higher prices, in each case the *rs* is linearly descending from the lowest to the highest price in each range border. For the first hotel in Figure 1, Table 2 shows the mapping of the search terms to the hotel criteria, the weights provided by the user, the relevance functions that are used and the calculated relevance score for each criterion. The hotel gets, therefore, a summed relevance score of 3.26.

Implementation

For implementing our search interface we utilized Elastic Search[1] as a search engine. The software architecture was designed to support various types of data and data sources. The prototype is implemented in Java-EE and provides a generic library and the actual hotel search application. The user interface is a generic JSF web fragment which provides a custom component and java controller (bean)[2].

The hotel search application contains specific facets with different classes and functions. For the facet "neighborhood", a Gauss scoring (Figure 2a) is applied to consider the distance of a hotel to a specified neighborhood. That means hotels that are nearer to the neighborhood indicated in the search query are weighted higher than those farther away. For the "customer rating values" and "hotel stars" linear functions (Figure 2b) are used. For the price range, a tri-linear function (shown in Figure 2c) is used with a range extended by 20% to each side of the specified price maximum and minimum. For all nominal facets, an equal value comparison is applied.

EVALUATION

We performed a lab study to evaluate the presented hotel search interface against a standard search interface using facets (see Figure 3). Additionally to the explicit user feedback and user task performance, we are interested in the system performance during a whole search session and how this relates to the user's feeling about system support.

[1]https://www.elastic.co/de/products/elasticsearch

[2]The prototype's source code is available here: https://git.gesis.org/iir/preferenced-based-search.

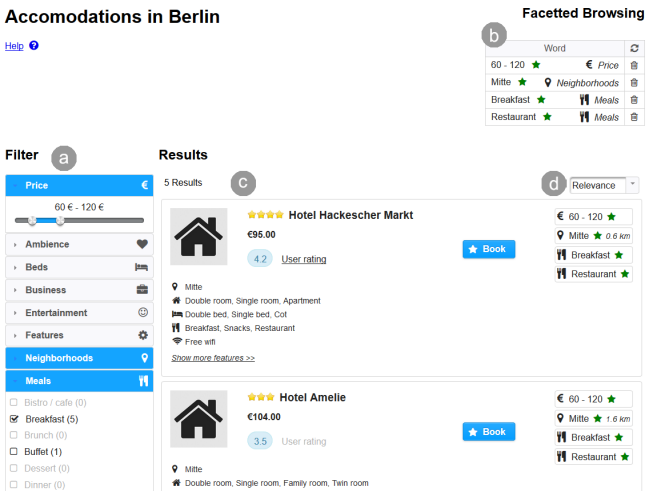

Figure 3. Comparative prototype used in the user study as a representative of a faceted search interface.

Apparatus

Prototypes

For the user study, we used the hotel search prototype introduced above – in the following called weighted prototype. For the comparison with a conventional search interface using facets we developed a second prototype as a representative of a state-of-the-art hotel search interface (see Figure 3) – in the following called facet prototype. This prototype provides filter functionalities in the form of facets, which are separated into different categories (see Figure 3a). For each facet term, the number of remaining results in the result list after choosing this facet is shown after the term in brackets. Selected facets are represented on the right-top side of the interface where the user can also delete selected facets by clicking on the recycle bin icon (see Figure 3b). The results shown in the result list (see Figure 3c) match all selected facet terms. The representation of the result list is the same as in the weighted prototype. However, an additional sorting function is available to sort the results by relevance, price, hotel stars and customer ratings (see Figure 3d). Both prototypes include a logging component to record user actions as well as the status of each presented result list.

Scenario

In order to compare the two prototypes and to analyze whole search sessions under the same conditions, we created a use case all participants had to perform with each prototype. The scenario includes "must-haves", "must-not haves" and "optional" preferences for a hotel in Berlin. The participants were asked to book a hotel for Paul who wanted to attend a conference in Berlin. His intended price range is between €60 and €120. Participants were provided with Paul's preferences: "Paul would prefer to stay in district 'Mitte' (neighborhood) or in another district with good access to public transport (transport). By no means he wants to stay in the district 'Tiergarten' (neighborhood). In any case, he would like to have breakfast (meal). Furthermore, he would appreciate if the hotel has a restaurant (meal) or a bar (entertainment). Additionally, he would welcome, if the hotel has a fitness center (sport) and

that he can pay on invoice (payment type). He could do without these two last items, whereas paying on invoice would be more important for him than the fitness center." The categories of the conditions were explicitly mentioned in the scenario because we are not interested in comparing how fast the participants could find the desired category compared to how fast they could type in the search terms. There is no hotel in the data set that matches all criteria. Otherwise participants would find this hotel with both prototypes very quickly – it would be the best hit after selecting all facets. In our scenario, Paul's preferences have to be weighted against each other and compromises have to be made finding a hotel that matches Paul's preferences properly.

Data sets and hotel relevance scores

We created a data set of 150 hotels in Berlin. The initial set of accommodations is based on information gained through the public Yelp-API. We received a list of hotels in Berlin with addresses, user comments, user ratings and categories. Each hotel was enriched by further features assigned to 18 different categories (such as price, neighborhood, type of room, type of catering, customer rating, etc.). This information was taken either from the hotel's website or information provided by Booking.com. To be able to have two comparable sets of hotels that can be used in the evaluation and to avoid a learning effect, we copied the hotel data set and changed only the names of the hotels. All information in the data set were provided in German.

For evaluating how well a selected hotel fits Paul's preferences, we generated a graded relevance score (*grs*) for all hotels in our dataset. The algorithm is illustrated in Figure 4. First, we checked if the hotel match the "must have" / "must-not have" criteria. When these mandatory criteria are fulfilled the hotel gets a first *grs* of one. In our case, that means, the hotel has a price between €60 and €120, breakfast is included and the hotel is in the district "Mitte" and if it is not in "Mitte" it has access to public transport, and it is not in the district "Tiergarten". All other hotels that do not meet these criteria are considered to be not relevant and are not considered further. They get a *grs* of zero. Then, we checked the additional optional criteria and awarded additional relevance scores depending on Paul's preferences. That means, a hotel with a fitness center gets two additional scores, as this was Paul's least preferred criterion. Hotels that provide the opportunity to pay by invoice gets three additional scores, as this was more preferred by Paul than the fitness center and hotels with a restaurant or a bar get four additional scores, as Paul preferred these features most. A hotel that meets all criteria can gain a *grs* of 10 (= 1 (mandatory criteria) +2 (fitness studio) +3 (paying on invoice) +4 (restaurant or bar)). In our data set only 15 hotels meet Paul's must-have preferences. No hotel meets all requirements, but five hotels have a graded relevance score of eight.

Setup

For the user study, we used a laptop with internet access. Through the Firefox browser 46.0.1 both prototypes were accessed on our server. To make sure that all participants see exactly the same part of the user interface a 21″ monitor with

```
FOR EACH hotel in the data set
  IF (60€ <= price <= 120€) AND "breakfast" AND
  ("Mitte OR ((NOT "Mitte) AND "public transport"))
  AND NOT "Tiergarte" THEN grs = 1
          IF "fitness center"  THEN grs=grs+2
          IF "invoice" THEN grs=grs+3
          IF "restaurant" OR "bar" THEN grs=grs+4
```

Figure 4. Algorithm for calculating Hotel's graded relevance score (grs) based on Paul's preferences provided in the scenario.

the same resolution was used in all sessions. Furthermore, an external keyboard and a mouse were provided to the participants as input modalities. The activities on the screen were recorded with the screen capture software Camtasia for further analysis. The used questionnaires were on paper.

Methodology

Data collection took place in a laboratory setting at a university and our institute in single sessions. We used a within-subject design approach with the two prototypes and the two data sets being the independent variables. The four resulting experimental conditions were randomly but equally assigned to the participants. The study took about 45 minutes per participant. As dependent variables, we collected time-on-task, clicks-on-task, graded relevance scores of selected hotels, subjective ratings, free-form text comments. Furthermore, we calculated the normalized discounted cumulative gain (NDCG) [9] of each result list.

Procedure

The study was performed in single sessions and followed a detailed trail protocol with a counter-balanced order of the four experimental conditions. First, the experimenter explained the purpose of the study and that all activities on the screen are recorded. The participant agreed to the procedure by signing a consent form. Before the actual experiment started, the participant filled out a questionnaire in which we asked a few questions regarding demographic information and the experience with hotel search systems. Afterwards, the experimenter showed and explained the first prototype and asked the participant to familiarize herself with the interface for about four minutes. Then, the participant was given the assignment with the scenario description. The task ends with selecting a hotel for Paul. There was no time limit for the task. In a questionnaire, the participant was asked why she selected the hotel, how satisfied she is with her choice, how well the system supported her while searching for a hotel and if the ranking of the result list was comprehensible. The same procedure was followed with the second prototype. At the end of the study, the participant filled out a final questionnaire. In this, we asked for advantages, disadvantages of each prototype as well as for suggestions for improvements with open questions. Each participant received €10 as a compensation for expenses.

Participants

24 native speaking German participants took part in the user study. Half of them were university students from different

	Selected Hotels				
	grs = 8	*grs = 7*	*grs = 0*	*Neighbarhood "Mitte"*	*average price*
weighted prototype	22	0	2	9	78 €
facet prototype	21	2	1	17	95 €

Table 3. Number of hotels selected by the participants for each prototype divided according to the graded relevance score, neighborhood "Mitte" and average price.

fields of study. They were recruited through an announcement on notice boards or mailing lists. The other half were recruited by word-of-mouth recommendation and they all had either a university degree or a completed apprenticeship. 12 participants were female. Participants' age ranged from 18 to 51 years (M=28.04, SD=8.3). Three of them have never used a hotel search system. 16 of the participants are familiar with more than one hotel search system like Booking.com or Trivago.com and 17 used such a system at least once in the last six months. 81% of these participants are satisfied or very satisfied with the functionalities such systems provide and 76% are also satisfied or very satisfied with the search results of such systems.

EVALUATION RESULTS

In the following section, we describe the results of the user study by first focusing on the selected hotels, then on the provided user feedback and finally on system performance. Given the fact that all participants had to perform the task following a given scenario we are able to analyze changes in recall and precision over the whole search sessions and combine them with the provided user feedback on system support. We want to know if there are relations between user feedback and system performance measures. If not stated others, we use a Wilcoxon signed-rank test to compute differences between two paired groups with $\alpha \leq 0.05$.

Analysis of selected hotels

In almost all sessions, hotels with the highest possible graded relevance score were selected with both prototypes (see Table 3). Having a closer look at the selected hotels, it is striking that with the facet prototype 17 participants chose a hotel in the neighborhood "Mitte", while with the weighted prototype only nine participants did the same. This might be an indication of a different level of elaborateness during the task performance. While the neighborhood was an equally weighted condition by Paul we found evidence in the log files that ten of the participants using the facet prototype did not consider that alternative at all. Furthermore, the log files showed that in total 16 participants had not examined all of Paul's preferences in the facet prototype. Beside of the preference "outside Mitte with access to public transport", the preference "fitness center" was often remained unconsidered. In the following, we will also have a closer look at those two user groups - those who examined all of Paul's preferences in the facet prototype (full examiners, n=8) and those who did not (incomplete examiners, n=16).

One interesting finding is the significant difference in price of the selected hotels (see Table 3). With the weighted prototype, the accommodation price is significantly lower than with the facet prototype (facet M=99.63, weighted M=80.25, p=0.011). Whereby, in the group of full examiners there is no significant difference in the hotel price (facet M=85.82, weighted M=74). Nine participants also stated price as an important factor while searching for a hotel.

In the weighted prototype condition, two participants failed in solving the task correctly based on the provided information about the hotel. They chose a hotel outside the neighborhood Mitte without access to public transport. However, one of them answered to our question why she chose this specific hotel with "There is good access to public transport." which indicates real knowledge about the neighborhood. With the facet prototype, also one participant chose a hotel outside Mitte with no access to public transport. We do not know if these two participants have the same reason (knowledge about the neighborhood) without stating it in the questionnaire. 18 participants provided further comments to the reasons why they chose a hotel. 15 stated that they included the user rating in their relevance decision process. For nine participants the price played an important role and comments from four participants allow the conclusion that they had further knowledge about Berlin, especially about the neighborhoods and the distances.

Quantitative User Feedback

We asked the participants on five-point likert scales how satisfied they are with the selected hotel ("very satisfied"=1 to "not at all satisfied"=5), how well they felt supported in their search by the system ("very well"=1 to "not well at all"=5) and if the result sorting was comprehensible ("very comprehensible"=1 to "not at all comprehensible"=5). The results show that there is no significant difference regarding support and comprehension of both presented search systems. However, participants are significantly more satisfied with the selected hotel when they use the weighted prototype (M=1.67) than the facet prototype (M=2.13, Z=-2.082, p<.05). A closer look at our to user groups showed that this is true for the incomplete examiner (weighted M=1.63, facet M=2.19, p=0.039) but there was no significant difference in the group of full examiner. They were similar satisfied with the hotel selected in the weighted prototype (M=1.75) and in the facet prototype (M=1.88).

For analyzing time-on-task and clicks-on-task needed to perform the task, we consulted the recorded screencast videos as well as the log file data. In both cases, we could find significant differences between the two prototypes. Participants are significantly faster and significantly fewer clicks are needed in the facet prototype than in the weighted prototype. In the facet prototype it took 5.3 minutes (σ= 2.9) and 16.5 clicks (σ= 10) on average to select a hotel compared to the weighted prototype with 7.36 minutes (σ= 2.96) and 25.83 clicks (σ= 9.76) on average (time-on-task: Z=-2.714, p<.05, clicks: Z=-3.244, p<.05). These results seems not surprisingly given the fact that the number of incomplete examiner is with 2/3 relatively high. In the group of full examiner, we could not find a significant difference in time-on-task (weighted prototype M=6.6 minutes, facet prototype M=7.52 minutes) and clicks (weighted prototype M=18.13, facet prototype M=27.13). While in the

group of incomplete examiner, the differences in time-on-task (weighted prototype M=7.64 minutes, facet prototype M=4.16 minutes) and clicks (weighted prototype M=24.69, facet prototype M=11.06) are significant (time: p<.0001; clicks: p=.000).

Qualitative User Feedback

In the final questionnaire, we collected qualitative user feedback in open questions on advantages, disadvantages and suggestions for both prototypes. In the following, we only provide feedback comments that were stated by more than one participant.

Weighted prototype's advantages 22 participants answered the question about advantages for the weighted prototype. Altogether we collected 34 different statements. The benefit most often stated, by eleven participants, was the opportunity to weight optional criteria. Five persons liked the "must-not have"-opportunity and also five found the interface well structured. Three participants appreciated the free-text search function and two other liked that they got more potential results that they can compare.

Weighted prototype's disadvantages 18 participants provided 28 disadvantage statements for the weighted prototype. Six persons disliked that there was no category list. Five missed an explicit sort-by function for price or ratings. Four were annoyed by typing in the search criteria.

Suggestions for the weighted prototype We collected 18 suggestions for the weighted prototype, provided by 16 participants. Six persons would like to have an explicit sort function by price or ratings. Showing all available criteria was suggested by four participants.

Facet prototype's advantages 24 participants provided feedback to the question about advantages of the facet prototype, wherewith we collected altogether 30 statements. 15 persons liked to select the facet terms from lists shown on the left side of the interface, as they know it from online shops and travel portals. Four participants appreciated the opportunity to sort the results by a predefined sorting criterion. Two persons mentioned positively that they do not have to type in a search term and another two liked the clear arrangement.

Facet prototype's disadvantages We received 24 statements from 19 participants to the question about disadvantages of the facet prototype. Six persons missed that there was no possibility to weight the criteria. For four participants the interface was too overloaded, and three criticized the cutting of hotels that did not fulfill all criteria. Two participants stated that it was cumbersome to find out which criteria led to a no-result list and two others missed the opportunity to compare alternatives that did not match all criteria.

Suggestions for facet prototype 18 participants provided suggestions for the facet prototype. Altogether 21 comments could be collected of which eight would like to have a weighting function. Three would add a search field and two the possibility to exclude results matching a "must-not have" criterion.

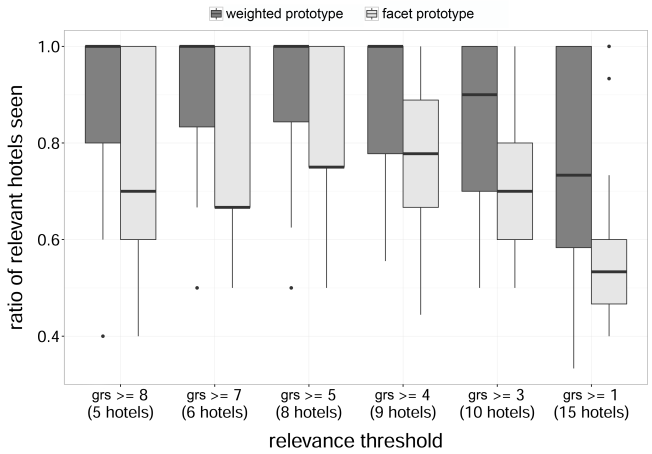

Figure 5. Boxplots of the ratio between relevant hotels that were visible during all participants' sessions and the total number of relevant hotels, grouped by systems.

System Performance Measure

When analyzing the facet and the weighted prototype regarding system performance, we have to keep in mind that their individual functionalities influence recall and precision differently. In a faceted search system, recall is massively influenced by the filtering functionality of the facets. The weighting of facets, in contrast, mainly influences the precision. Concerning individual result lists, the two systems are hardly comparable. Therefore, we will evaluate them on their own on the level of the complete search session and combine results with user feedback.

Relevant hotels seen

In our context, recall would be defined as the ratio of the number of relevant hotels that the system retrieved for a user query and the total number of relevant hotels in the data set. However, in our study, the data set contained only 150 hotels. Stating queries with all relevant hotels within the result set is not a complex task. Instead of looking at the complete result list that a system retrieved, we will only consider the visible part of it, namely, all the results that have been displayed on the result pages the users browsed through. Also, we will not only focus on single queries, but we will incorporate all queries within a session.

Let us consider a participant's session in which multiple searches are conducted to find a suitable hotel. These queries can be triggered by entering search terms, filtering or reordering the result list, changing the weight of a criterion, and including or excluding criteria. Each query action will lead to a new result list of which the first 15 hotels are displayed on the first result page. These hotels form the visible part of the result list. If the participant proceeds to the next page of a result list, the number of visible results is increased. If we collect all visible hotels from each participant's search session, we can assess *how many of the relevant hotels have been visible to the participants* using a specific system.

Figure 5 shows boxplots of the ratio between relevant hotels that were visible during all participants' sessions and the total

number of relevant hotels, grouped by systems. We calculated the ratio with a decreasing relevance threshold and grouped the results. The first two boxplots on the left show the ratio of hotels with a graded relevance score (*grs*) higher or equal 8 for the facet and the weighted prototype. The third and fourth boxplots show the ratio of visible hotels with a *grs* higher or equal 7 and so forth. It can be observed that when using the weighted system until a relevance threshold of 4 a high mean of 1.0 and a first quartile of around 0.8 was achieved whereas the mean and first quartile using the facet prototype was lower. Overall, a non-parametric Mann-Withney test ($\alpha \leq 0.05$) found the number of relevant hotels seen with the weighted system to be significantly higher. The results of the test are as follows: grs$\geq$8 ($u = 2.668, p = .008$), grs$\geq$7 ($u = 2.645, p = .008$), grs$\geq$5 ($u = 2.544, p = .011$), grs$\geq$4 ($u = 2.900, p = .004$), grs$\geq$3 ($u = 3.099, p = .002$), and grs$\geq$1 ($u = 3.478, p = .001$). Having a closer look at the two groups, it is not surprisingly that the group of incomplete examiner is again responsible for that result. They have seen significantly more hotels in the weighted prototype than in the facet prototype (for example grs$\geq$8 facet prototype M=3.50 vs. weighted prototype M=4.50, p=.002; and for grs$\geq$1 facet prototype M=7.94 vs. weighted prototype M=12.13, p<.0001). In the group of full examiner there are no significant differences.

Result list precision

For analyzing the quality of the result lists, we used the normalized discounted cumulative gain (NDCG) [9]. Plotting the NDCG values of all searches conducted during a search session against time gives an overview of the development of the participant's session from the perspective of precision. As both systems' modes of operation have different impacts on the NDCG, the NDCG plots of the two systems should not be compared. Therefore, we will not compare the two systems concerning precision values, but we will investigate the relationship between precision and the perceived satisfaction with the systems' which was asked by the question "How well did you feel supported by the system?" ("very well"=1 to "not well at all"=5).

Figure 6 shows the NDCG plots for the weighted prototype and Figure 7 for the facet prototype. Each point represents the NDCG value of a search at a specific point of time during the session. In both figures, the plots are grouped along the participants' answers to the question how well they felt supported by the system during their search. As the answers "not well" and "not well at all" were only given by one or none participant we only show the plots for "very well supported", "well supported" and "neither well nor not well supported". For example, in the upper plot of Figure 6 ("very well"), each point represents the NDCG value of a search result list of a search session, where the participant felt very well (n=10) supported by the system. In addition to the data points, we generated LOESS smoothing curves[3] which helped to identify common characteristics. The dashed blue line is a LOESS curve created for the complete data set regardless of the participant's answer to the question of system support satisfaction. The solid red

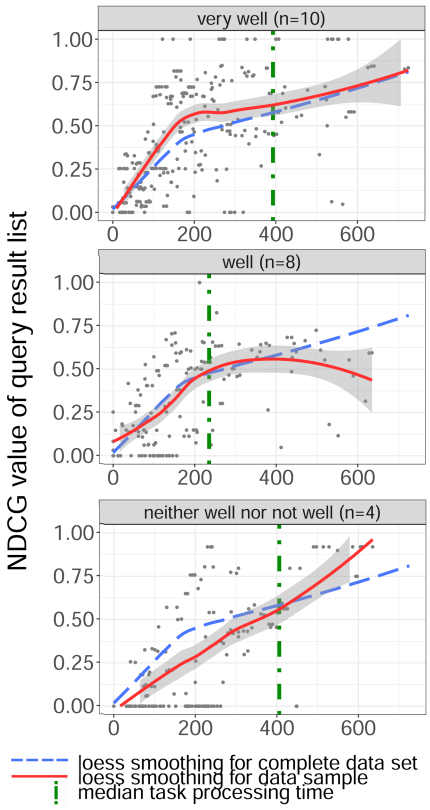

Figure 6. NDCG plots for the weighted prototype, grouped along the answer to the question 'How well did you feel supported by the system?'.

lines are a LOESS curve generated for the specific group.[4] In addition to the LOESS curves, for each plot, there exists a vertical dot dashed green line, which indicates the point of time, where the median of the group's participants has finished their task.

The LOESS curves generated for the whole data set in Figure 6 and Figure 7 show one common characteristic. The search process seems to be divided into two phases. First a take-off, where the precision of search requests increases until roughly 180 seconds into the session, where a break of slope indicates the beginning of the second phase. During the take-off phase, the steepness of the curve indicates a fast improvement of search precision. For the weighted prototype (Figure 6), this first phase ends at an NDCG value of around 0.4. After this, the NDCG still increases, but with a lower gradient. For the facet prototype (Figure 7), the break of slope builds the maximum of an NDCG value of around 0.7. After the maximum, the curve slowly declines until a minimum of around 0.4 after roughly 600 seconds. Notably, the break of slope lies before the median task processing time. This means that at least half of the participants (in most cases even 75%) were still working on the task at the time point of the break of slope. This supports the assumption that the search sessions are indeed divided into two phases, as it means that the break of slope

[3]LOESS was first introduced by [5]. In this paper we use the implementation which is part of the R-Project: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/loess.html

[4]All LOESS curves are generated with a degree of 2 and a span of 0.75.

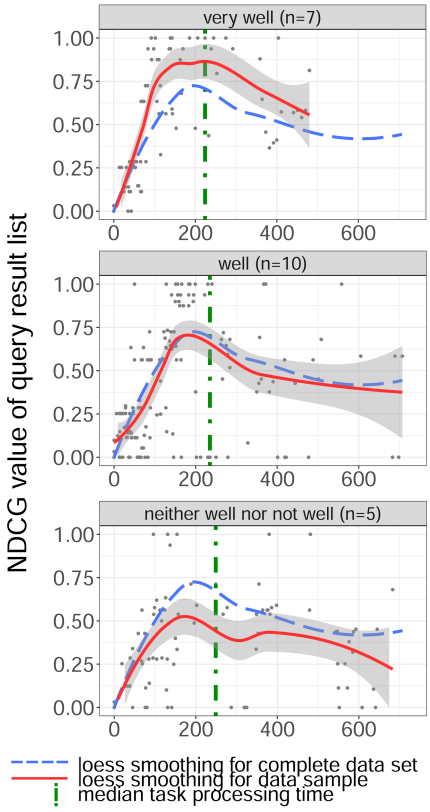

Figure 7. NDCG plots for the facet prototype, grouped along the answer to the question 'how well did you feel supported by the system?'.

does not coincide with the completion of the task.

Comparing the LOESS curve of the complete data sets and the curves of each individual subgroup one can observe a relationship between the satisfaction with the system's support and the precision of the search requests of the participants. In Figure 6, the LOESS curve of the group of participants that felt very well supported by the system primarily lies above the curve of the complete data set. Even if it is also divided into two phases, the take-off phase is steeper, and the break of slope lies on a higher level. Participants in this group were faster and better in formulating more precise search requests than the users in the other groups and felt better supported by the system. Similar results can be observed in the facet prototype condition.

When comparing the individual groups' LOESS curves with the overall curves, we can identify similar relationships between those two for both systems. The lower the perceived level of support is, the lower the curve. The curve for the group "very well" is the highest, the curve for "well" is close to the overall curve, and the curve for "neither well nor not well" lies mostly underneath the overall curve. Overall, we can conclude that there seems to be a connection between perceived system support and the development of the search precision.

DISCUSSION

In this paper, we compare two different concepts for searching products: (1) a pure facet concept, well established in all kinds of (product) search engines and (2) a preference-based approach using weighted facets which allows users to express preferences of certain product, in our case hotel features. We took user feedback, system performance and a combination of both into consideration to evaluate both approaches.

Our evaluation results show that the participants are significantly more satisfied with the selected hotels found in the weighted prototype than in the facet prototype. One possible explanation for that is given by the recall analysis which showed that there were significantly more relevant hotels visible during the whole search session. Therefore, participants were able to compare more hotels, even those that only partly fulfill their requirements, and might get a better feeling for their buying decision. This is inline with McSherry's findings that a decision maker wants to be informed of all items that are likely to be of interest [12]. The high number of relevant hotels shown is explainable by the different interaction techniques to compare alternative hotels. In the weighted prototype, based on the task, users themselves push more relevant hotels higher in the list with the interactive sliders, and therefore these hotels are better visible. In the facet prototype, the effort to see more relevant hotels is higher as the user has to select and deselect each facet and facet combinations explicitly to see its influence on the results. It seems that the incomplete examiner were not willing to take the extra effort. Apparently they are willing to select a hotel more quickly that matches at least parts of the preferences without knowing other, probably better, alternatives. Therefore, it is not surprising that these participants were faster and needed fewer clicks in the facet prototype to select a hotel. The time the full examiner took to find a hotel in the weighted and in the facet prototype did not differ significantly. Half of them were even faster in the weighted prototype than in the facet prototype. Furthermore, no participant complained in the questionnaire that finding a hotel with the weighted prototype, in general, took too much time or effort. The fact that with the weighted prototype the hotel price of the selected hotel is significantly lower than with the facet prototype could provide an incentive for some users to spend more effort for the search process.

Besides our comparison of the two prototypes regarding user satisfaction and recall, we were able to find similar characteristics within the search process of our participants by analyzing the search precision. Overall, the search sessions are divided into two phases. During the first phase, the take-off phase, the requirements defined in the scenario are transferred into the system, which leads to an increase in precision. During the second phase, the precision decreases and increases alternatively leading to a change of the precision curves slope. The reason here might be, that there is no perfect hotel for the task given and the participants had to change their queries to find alternatives that are close to the criteria given.

Regarding result list quality we observed a relationship between the perceived system support and the precision of the participants. When plotting the NDCG values grouped by the

participants answer to the question how well they felt supported by the system, one could observe, that a certain groups' precision curve lies above the average if the user felt very well supported and below if she felt neither well nor not well supported. This indicates that there might be a relation between the perceived system support and the precision of the participants' searches. However, this method is new, and we have not yet understood enough to draw solid conclusions, but we believe that the analysis of the search precision can aid in the task of measuring user satisfaction. In our case, the two prototypes do not allow for a comparison of the result list precision, as they generate those list differently. When comparing two similar systems, this method could produce comparable results, which would allow to evaluating two different versions of a system.

One important lesson learned from our study have to be mentioned. Knowledge about Berlin, the city we chose as an example in our study, might have influenced the results, as participants selected hotels knowing that the neighborhood has a good connection to public transport. In further user studies, we will use a fictitious city. Furthermore, there are some other limitations in our approach which should not go unmentioned. The number of hotels in our data set with 150 hotels is rather low. We could not find an existing hotel data set with a sufficient number of facets for each hotel. So, we created our own by manually enhancing the dataset with a lot of different facets. However, we cannot preclude that the dataset size might have an effect on our evaluation results. In future user studies, we will use a data set with a higher number of hotels. In order to perform a combined analysis of user feedback with system performance it was necessary that all users performed the same task in both conditions. Also in this case, we cannot preclude that with a different task the results might be different. This is also an aspect we have to address in future research to examine further the relation between system performance and user satisfaction measures.

CONCLUSIONS

In this paper, we present an evaluation of a search interface using a preference-based ranking approach. Users can select, exclude and weight (optional) search criteria by their preferences and thus influence the ranking of the result list. In a user study, we compared this search interface to an interface using standards facets. 24 participants had the task to find a hotel according to predefined preferences with both search interfaces. We evaluated the interfaces from a user and a system performance perspective and found out that:

- Users are significantly more satisfied with the selected hotel found with the weighted prototype.
- Users were given more relevant hotels in the result lists with the weighted prototype.
- There is no significant difference regarding time-on-task and clicks when users examine all preferences in both prototypes.
- Users, who do not consider all preferences in their search queries in the facet prototype were significantly faster and needed fewer clicks to select a hotel than with the weighted prototype.
- Users chose a significantly cheaper hotel with the weighted prototype.
- Both user interfaces show characteristic differences in the evolvement of precision during a search session.
- Users that were able to generate result lists with a higher precision seem to felt better supported by a system.

The last two results based on observations on the analysis of the relation between the participants' answers to the question how well they felt supported by the system and the precisions (measured by NDCG) of the result lists. In future work, we want to research the potential of analyzing the evolvement of search precision over whole search sessions as an indication for user satisfaction in more detail.

ACKNOWLEDGEMENT

We would like to thank Marco Janc for implementing the prototype. Additionally, we thank him and Maria Lusky for supporting us with the execution of the user study. We are also very grateful to those who participated in our study.

REFERENCES

1. Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. 2012. TasteWeights: a visual interactive hybrid recommender system. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 35–42.
2. R. D. Burke, K. J. Hammond, and B. C. Yound. 1997. The FindMe approach to assisted browsing. *IEEE Expert* 12, 4 (Jul 1997), 32–40. DOI: http://dx.doi.org/10.1109/64.608186
3. Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. *ACM Comput. Surv.* 44, 1 (Jan. 2012), 1:1–1:50. DOI:http://dx.doi.org/10.1145/2071389.2071390
4. Li Chen and Pearl Pu. 2012. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction* 22, 1 (2012), 125–150. DOI: http://dx.doi.org/10.1007/s11257-011-9108-6
5. William S Cleveland. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association* 74, 368 (1979), 829–836.
6. Shuguang Han, Danchen Zhang, Daqing He, and Qikai Cheng. 2016. User exploration of slider facets in interactive people search system. *IConference 2016 Proceedings* (2016).
7. F. Maxwell Harper, Funing Xu, Harmanpreet Kaur, Kyle Condiff, Shuo Chang, and Loren Terveen. 2015. Putting Users in Control of Their Recommendations. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15)*. ACM, New York, NY, USA, 3–10. DOI: http://dx.doi.org/10.1145/2792838.2800179
8. Marti Hearst, Ame Elliott, Jennifer English, Rashmi Sinha, Kirsten Swearingen, and Ka-Ping Yee. 2002. Finding the Flow in Web Site Search. *Commun. ACM* 45,

9 (Sept. 2002), 42–49. DOI: http://dx.doi.org/10.1145/567498.567525

9. Kalervo Järvelin and Jaana Kekäläinen. 2000. IR Evaluation Methods for Retrieving Highly Relevant Documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '00)*. ACM, New York, NY, USA, 41–48. DOI: http://dx.doi.org/10.1145/345508.345545

10. Matt Lindner. 2016. "Online sales will reach $523 billion by 2020 in the U.S.". Online. (29 January 2016). Retrieved September 20, 2016 from https://www.internetretrailer.com/2016/02/29/online-sales-will-reach-523-billion-2020-us.

11. Benedikt Loepp, Katja Herrmanny, and Jürgen Ziegler. 2015. Blended Recommending: Integrating Interactive Information Filtering and Algorithmic Recommender Techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 975–984. DOI: http://dx.doi.org/10.1145/2702123.2702496

12. David McSherry. 2003. *Similarity and Compromise*. Springer Berlin Heidelberg, Berlin, Heidelberg, 291–305. DOI:http://dx.doi.org/10.1007/3-540-45006-8_24

13. Denis Parra, Peter Brusilovsky, and Christoph Trattner. 2014. See What You Want to See: Visual User-driven Approach for Hybrid Recommendation. In *Proceedings of the 19th International Conference on Intelligent User Interfaces (IUI '14)*. ACM, New York, NY, USA, 235–240. DOI: http://dx.doi.org/10.1145/2557500.2557542

14. Hans peter Frei and Yonggang Qiu. 1993. Effectiveness of Weighted Searching in an Operational IR Environment. (1993).

15. Pearl Pu and Li Chen. 2005. Integrating Tradeoff Support in Product Search Tools for e-Commerce Sites. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC '05)*. ACM, New York, NY, USA, 269–278. DOI: http://dx.doi.org/10.1145/1064009.1064038

16. Pearl Huan Z. Pu and Pratyush Kumar. 2004. Evaluating Example-based Search Tools. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC '04)*. ACM, New York, NY, USA, 208–217. DOI: http://dx.doi.org/10.1145/988772.988804

17. Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. *Introduction to Recommender Systems Handbook*. Springer US, Boston, MA, 1–35. DOI: http://dx.doi.org/10.1007/978-0-387-85820-3_1

18. J. J. Rocchio. 1971. Relevance feedback in information retrieval. In *The Smart retrieval system - experiments in automatic document processing*, G. Salton (Ed.). Englewood Cliffs, NJ: Prentice-Hall, 313–323.

19. Nicolas Spyratos and Vassilis Christophides. 2006. Querying with Preferences in a Digital Library. In *Proceedings of the 2005 International Conference on Federation over the Web*. Springer-Verlag, Berlin, Heidelberg, 130–142. DOI: http://dx.doi.org/10.1007/11605126_8

20. Markus Stolze. 2000. Soft navigation in electronic product catalogs. *International Journal on Digital Libraries* 3, 1 (2000), 60–66. DOI: http://dx.doi.org/10.1007/PL00021475

21. Daniel Tunkelang. 2009. Faceted Search. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1, 1 (2009), 1–80. DOI: http://dx.doi.org/10.2200/S00190ED1V01Y200904ICR005

22. Martin Voigt, Artur Werstler, Jan Polowinski, and Klaus Meissner. 2012. Weighted Faceted Browsing for Characteristics-based Visualization Selection Through End Users. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '12)*. ACM, New York, NY, USA, 151–156. DOI: http://dx.doi.org/10.1145/2305484.2305509

23. Bifan Wei, Jun Liu, Qinghua Zheng, Wei Zhang, Xiaoyu Fu, and Boqin Feng. 2013. A Survey of Faceted Search. *J. Web Eng.* 12, 1-2 (Feb. 2013), 41–64. http://dl.acm.org/citation.cfm?id=2481562.2481564

24. Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. 2003. Faceted Metadata for Image Search and Browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 401–408.