IADIS International Journal on WWW/Internet
Vol. 12, No. 2, pp. 115-130
ISSN: 1645-7641

MOTIVE-BASED SEARCH USING A RECOMMENDATION-DRIVEN VISUAL DIVIDE AND CONQUER APPROACH

Andreas Both. *Unister GmbH, Barfußgäßchen 11, 04109 Leipzig.*

Viet Nguyen. *Unister GmbH, Barfußgäßchen 11, 04109 Leipzig.*

Mandy Keck. *Technische Universität Dresden, Chair of Media Design, Nöthnitzer Str. 46, 01187 Dresden.*

Martin Herrmann. *Technische Universität Dresden, Chair of Media Design, Nöthnitzer Str. 46, 01187 Dresden.*

Dietrich Kammer. *Technische Universität Dresden, Chair of Media Design, Nöthnitzer Str. 46, 01187 Dresden.*

Rainer Groh. *Technische Universität Dresden, Chair of Media Design, Nöthnitzer Str. 46, 01187 Dresden.*

Dana Henkens. *queo GmbH, Tharandter Str. 13, 01159 Dresden.*

ABSTRACT

A novel generation of (e.g. touch-driven) applications leads to a new universe of interaction paradigms and a growing need for simple, inspiring and smart interfaces while the size of searchable data sets is increasing permanently (big data). A system intended for non-experts should only present information the user needs to solve his task, instead of confronting him with the large and complex underlying data structure. In this paper, we focus on users wanting to perform a product search driven by a vague information need. We call this kind of search motive-based search, which is often initiated by unconscious motives and expectations that are difficult to transform into a specific search query at the beginning of the search process. Hence, the user needs guidance to fulfill his search task. A search approach will be developed in this paper, which allows a step-by-step reduction of the result set by selecting (visualized) concepts such as "beach", "relaxing" and "culture". Concepts are often organized as multiple faceted hierarchies (polyhierarchies) to represent different views on things. Hierarchies can also be used as navigation paradigm, named faceted search. We will present the significant flaws of this approach concerning larger knowledge bases. Alternatively, we propose a selection-based recommendation-driven search, based on the principle of divide and conquer. An experiment compares both approaches proving that the proposed approach allows to solve the given search tasks in shorter time and with less effort.

KEYWORDS

Information Retrieval in Semantic Web; Search Interfaces; Emotional Interfaces; Interfaces for e-Commerce; Interfaces for Recommender Systems; Interfaces for Big Data.

1. INTRODUCTION

During the last decade, several innovations pushed the usage of data-driven applications. One main reason for this trend is the permanently increasing amount of data. According to a long term observation (Hilbert & Lopez 2011), the size of saved data is growing by at least 23% every year. A great progress was made by creating (semi-)structured data sets (i.e. knowledge bases). For example, crowd-based approaches (e.g. Wikipedia.org) or a growing number of public structured databases in the Web of Data using Semantic Web (Hitzler et al. 2011) technologies (in particular the Linked Open Data Cloud, see Bizer et al. 2009) lead to data sets that can be processed with greater expressiveness than a set of unstructured texts, e.g. DBpedia (a knowledge base extracted from Wikipedia, see Lehmann et al. 2014). This evolution of data stores can also be observed considering enterprise data (Wood 2010).

Hence, there is a great demand in improving and building intuitive search-driven applications that lead to good results in short time. However, the challenge of handling huge amounts of data is not solved, as still little effort is invested in taking advantage of the available data structures. In particular, users will not accept having to invest more time for the same search task because of the increased amount and size of data over the last year. Search-driven applications have to take advantage of the structures (i.e. semantics) the creator of the knowledge base has already integrated during the knowledge engineering process (Studer et al. 1998).

In this paper, we consider the well-known hierarchical search and will uncover the disadvantages of this search approach. Based on these drawbacks, the contribution of the paper is to derive a novel selection-based search approach following the principles of divide and conquer. This novel approach is embedded in our main research agenda, that focuses on product search and big data, where the user has only a vague idea of his search goal and is not able to form a precise search query right away (e.g. looking for a suitable gift, planning a relaxing holiday). This can also be described as motive-based search (Keck et al. 2013), where a motive defines the cause of a search as well as particular conditions such as the price limit for a product. In this paper, we address motives that are strongly influenced by emotions and interests and therefore cannot be easily transformed into a textual search query.

2. RELATED WORK

There are various search activities that can be distinguished in the context of web search. They can be classified into Lookup and Exploratory Search (Marchionini 2006). Lookup is the most basic kind of search task, returning well-structured results to analytical queries such as "when", "where" or "who". In contrast, Exploratory Search is a more complex process that can start with a vague information need, and therefore requires multiple iterations of learning, investigation and reformulating the search query (Marchionini 2006). Motive-based search scenarios refine Explorative Search by specifying the motive and the aim of the process to find a suitable product. Thus, in the context of product search interfaces are required that support the user in expressing his information need based on these conditions.

Therefore, two search paradigms are well established in the world of web search: Direct Search and Navigational Search (Tunkelang 2006). Most conventional product search interfaces use the Direct Search paradigm in the form of text boxes and forms. These interfaces are still very lookup-oriented and require the user to transform a possibly vague information need into a specific query (Dörk et al. 2012). For motive-based search, this paradigm alone is neither adequate nor effective. In contrast, Navigational Search systems use taxonomies to provide the user an overview of the offered product categories and an access point for his search. Hierarchical taxonomies demand to browse the information space in a predetermined order, whereas tags or Faceted Classifications allow multiple access points (Stefaner & Müller 2007). Hence, Faceted Browsers (e.g. amazon.com) and Tag Clouds (e.g. flickr.com/photos/tags/) are established interfaces that are offered to the user to examine the structured web content (Russel-Rose & Tate 2013). Navigational Search systems are suitable for motive-based search-scenarios because the user does not need to know exactly for what he is looking for in contrast to the Direct Search scenarios. However, they can result in very complex structures. Hence, also tag-based systems or Faceted Classifications use categories or hierarchies to structure the taxonomy. If one concept (or tag) can be placed in more than one category, even polyhierarchical structures arise.

Even though Faceted Browsers and Tag Clouds have been under intensive study (Stefaner & Müller 2007, Polowinski 2009, Waldner et al. 2013), there are still many unanswered questions like automatic classification of complex or unstructured datasets or the visualization of excessive numbers of tags. Furthermore, there are still only a few adequate metrics to evaluate Faceted Search applications. In addition to subjective metrics such as "easy to use" or "flexibility" and relevance metrics, cost-based metrics play an important role in evaluating the efficiency of a search interface (Wei et al. 2013). Therefore, the time to finish a retrieval task is still one of the most common measurements to compare facet-based search interfaces.

2.1 Textual Approaches

To support the user to transform a possibly vague information need into a specific search query, the interface has to offer search functionality that goes beyond the basic search box or the standard search experience. Textual approaches like autocomplete and autosuggest offer support for query creation and reformulation. Additionally, autosuggest can encourage exploratory search by providing related concepts (Morville & Rosenfeld 2006). Apart from providing inspiration and new ideas for extending and refining a query, related searches can be used to help clarify an ambiguous information need (Russell-Rose & Tate 2013). For example, Bing and Google emphasize extensions to a query by highlighting these elements in the search box.

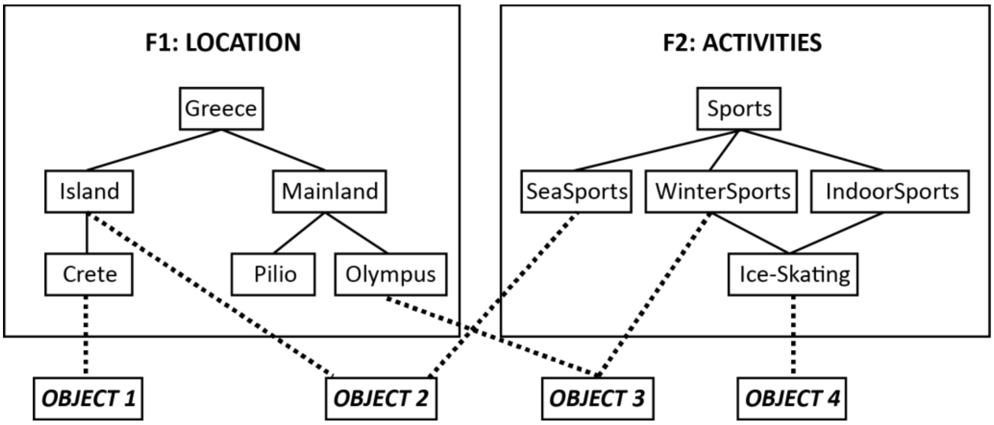

Figure 1. Example data set

Besides these textual approaches in which an information need has still to be articulated by a query, faceted search provides more effective information-seeking support to users with vague information need and can be combined with keyword search approaches (Hearst 2006; Russell-Rose & Tate 2013). It allows multiple access points for the search, the iterative refinement of the result set by selecting and combining different concepts relevant to a domain, and encourage the construction of complex search queries. Therefore, a Faceted Classification is required, which can be considered as a set of taxonomies, each one describing the domain of interest from a different point of view (Ranganathan 1965). Figure 1 shows an example consisting of two facets, namely "location" and "activities", and four indexed hotel web pages (Sacco & Tzitzikas 2009). Each facet contains different facet values that can be organized as flat lists or a hierarchy (Morville & Rosenfeld 2006).

Although faceted search interfaces are powerful tools for exploratory search and discovery-oriented problem solving, they face some challenges handling complex faceted classifications. Some approaches use scrollable or extensible containers to show the values on demand. However, even with these extensions it is assumed that the lists of facets and facet values are a manageable size (Russell-Rose & Tate 2013). Dealing with large information systems, can moreover result in very complex structures. As seen in Figure 1, one object could not be placed in only one category (e.g. Object 3) and a category can have multiple parent categories (e.g. “Ice-Skating”). Hence, a polyhierarchical structure arises. Instead of confronting the user with these complex data structures, the interface should only present the information he needs to solve his search task.

2.2 Non-Textual Approaches

In this section, we will point to related approaches that do not force the user to enter a well formulated textual query. Within the context of inspirational holiday planning interfaces addressing emotions, former experience or imagination is better suited than a simple text based form. The desired holiday can be considered as a summary of fuzzy concepts like adventure, pleasure or relaxation. Since these concepts have very different meanings to every individual, it is hard to evaluate them through a textual query. To overcome this problem, several travel providers use pictures associated with these concepts. Based on the pictures the user likes or dislikes, the provider can create a profile to recommend personalized travel offers.

"Picture your holiday"[1] from British Airways provides an example of a simple non-textual search. The interface consists of a large wall of tiled pictures showing places, objects, food or people. After picking five inspiring pictures, the system presents places related to the selection. This is a very simple way of communicating the personal idea of a perfect holiday. But since it is only possible to choose 5 pictures, which may in addition represent nearly the same concept, the results may not be very accurate.

"Inspire me"[2] is another example for a simple concept based approach. By selecting images for different categories like "eat", "see", "listen" or "do", the user can specify his interest in a simple way, while the system can match the selections with the underlying result set.

3. PROBLEM DESCRIPTION

This section describes the challenges users have to deal with while searching within a knowledge base. We will derive the requirements for the search approaches and the key performance indicators for the following experiment (cf. Section 5). Without loss of generality, we assume that a knowledge base is given where the data structure definitions are closely related to the definitions used for representing entities such as Linked Data (Bizer et al. 2009) using the RDF standard of the W3C (Lassila & Swick 1999). It should define categories (often concepts) and searchable entities, whereas categories are organized hierarchically (e.g. “Sport” with subcategory “WinterSports"). A subcategory can have multiple parent elements, allowing complex polyhierarchies. Searchable entities (e.g. London) are linked to at least one of these categories.

Definition 1. A searchable entity $e \in EE$ *within a knownledge base is defined by a unique ID. Additional properties are allowed.*

Hence, it is possible that a knowledge base contains a city defined by a unique number as ID and having a label like “London” and a depiction.

Definition 2. A category $c \in CC$ *within a knownledge base is defined by a unique ID. Additional properties are allowed.*

Example: An element of the type category might be labelled with “Sports”.

Definition 3. The relation subtype *is defined as* subtype $\in CC \times CC$.

Now, it is possible to represent sub-categories (e.g., “WinterSports” is a subtype of “Sports”).

The relation subtype is transitive, the materialization of the transitive relation is called subtype+.

[1] http://pictureyourholiday.ba.com/, last access: 2014/12/15
[2] http://www.iberia.com/gb/inspire-me/, last access: 2014/12/15

Definition 4. The relation subtype+ *is defined as* subtype+ $\in CC \times CC$*, where the recursive definition holds:*
subtype+(c_1, c_2) $:\Leftrightarrow$ subtype(c_1, c_2) $\vee$ subtype(c_1, c_3) $\wedge$ subtype+(c_3, c_2), where $c_1, c_2, c_3 \in CC$

This enables to express that "Ice-Skating" is also a subtype of "Sports" (cf. Figure 1).

Definition 5. The relation relatedTo *is defined as* relatedTo $\in EE \times CC$.

Hence, it is possible to have relations between a searchable entity $e \in EE$ and several categories $c \in CC$. With no loss of generality, it is assumed that for each e the relation relatedTo is not empty (i.e., a searchable element is always related to at least one category). In particular, it is possible that e is related to categories from two or more categories that are located within different hierarchies (compare example in Figure 1).

Definition 6. A data set contains polyhierarchical data structures if and only if there exist entities e_1, e_2 that are related to at least two categories which are located in different hierarchies. This is expressed by different root categories c_3, c_4 of each hierarchy which are not subtype of another category:
$\exists\, e_1, e_2 \in EE : c_1, c_2, c_3, c_4 \in CC$ $\wedge$ relatedTo(e_1, c_1) $\wedge$ relatedTo(e_2, c_2) $\wedge$ subtype+(c_1, c_3) $\wedge$ subtype+(c_2, c_4) $\wedge$ $c_3 \neq c_4$, where $\nexists c_5 \in CC$: subtype(c_3, c_5) and $\nexists c_6 \in CC$: subtype(c_4, c_6).

Using the given definitions it is possible to represent complex taxonomies, hierarchies and polyhierarchies within a knowledge base. In particular, several category hierarchies can be implemented separately and (later) applied to the same data items. Please note that we distinguish between the categories and searchable items, only the later might be contained within the search result set.

As described earlier, a common approach for searches within knowledge bases is based on categories and the (poly-)hierarchies they constitute. While browsing the (poly-)hierarchy, the user selects appropriate concepts to refine his search query and reduce the number of results. While searching hierarchically, the user has to select all categories while traversing top-down from the root element(s) of the hierarchy. Considering this approach may lead to several problems:

(1) The number of search results within the choice of categories might differ significantly.
(2) The number of selectable categories (concepts) at a refinement step might differ significantly. If the number is very high, the user might be overwhelmed by the given refinement options. If it is very low, the user might be annoyed because of an interaction perceived as unnecessary.
(3) A user has no criteria to decide which of the refinement options will optimize (i.e. reduce) the number of refinement steps until the searched item is found.
(4) If none of the presented sub-categories (refinement options) match the user's intention, then the last chosen option has to be canceled. The user has to start over on the previous level of the hierarchy.

4. SELECTION-BASED RECOMMENDATION APPROACH

We claim that the disadvantages of the hierarchical search approach have a significant negative impact on the user experience. Therefore, we specify the following requirements to provide a better user-experience:

- Requirement 1: Considering the user is not acquainted with the structure of the underlying knowledge base, he has to be provided with recommendations to start and refine his search.
- Requirement 2: After each refinement step, the result set should update immediately to provide the opportunity to evaluate the previous selection.
- Requirement 3: The presented concepts should lead to a desired result with the fewest number of interaction steps possible.

Hence, a search-driven application is confronted with the challenge to provide information to the user allowing to reduce the searchable elements within a knowledge base as fast as possible, even if the user is not familiar with the knowledge base. This process is also called step-by-step refinement of the search query.

Another requirement is to provide feedback about the remaining items within the knowledge base. Considering textual search, Google claims that a user can reduce the time needed for search tasks by 2–5 seconds (Google 2014) if the search results for the current search query are provided instantly (Google instant search). The user experience during the search process depends on the time (Wei et al. 2013) and consequently of the number of interactions needed during the search process. Hence, we derive in accordance with (Wei et al. 2013) that a search process is better than another if the time needed for finishing the search task is lower in comparison.

In conclusion, while searching within a knowledge base, the user should take advantage of the structures provided by the designer of the knowledge base, for example from the searchable entities classified using categories. Here, we focus on categories that structure the items contained in the knowledge base hierarchically.

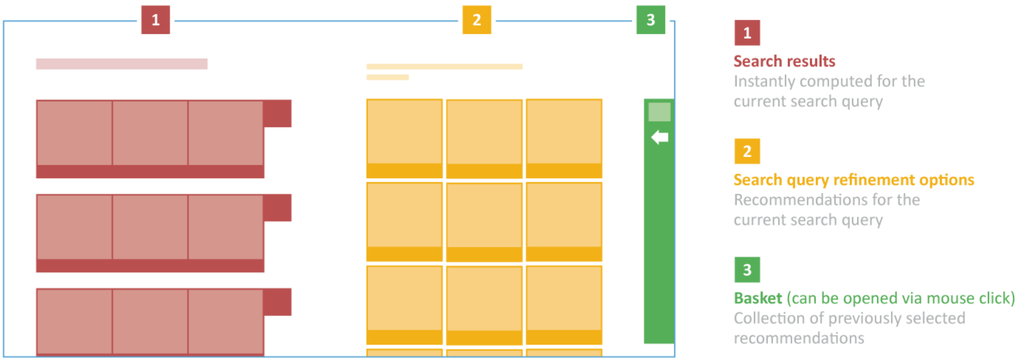

Figure 2. Wireframe of the experimental application

4.1 Concept

The search application was designed with focus on getting inspired regarding the content of a knowledge base. In particular, the user should be supported during the process of intuitively exploring a large number of unknown data items organized by hierarchical categories. Therefore, we implemented the following light-weight workflow consisting of two steps:

(1) Provide a well selected set of refinement recommendations (see Requirement 1).
(2) Provide immediate feedback of the search results (see Requirement 2).

A wireframe of the designed user interface is shown in Figure 2. The concept includes a region for showing search results [1], another region that displays recommendations [2] and a basket [3] for storing selected search results.

4.2 Algorithm

To select the refinement options, an algorithm (loosely based on the principle of binary search) was designed. Binary search is also called half-interval search algorithm (Cormen et al. 2001). The main concept here is to find the best discrimination rate for the search result sets. I.e., while searching for an item within an ordered list, at each refinement step, the number of left search results is $\lceil n/2 \rceil$, where n is the number of elements before the refinement. Hence, binary search is a divide and conquer approach executing in O(log(n)) time.

Given a real-world example, one can think about a set of n persons, where n=32, and the given search task to find a specific person p (like a famous person, e.g., Alan Turing). There are several strategies to find p. First of all, one can ask every one of the 32 people sequentially if it is the searched person. As this is only guessing, the needed time is within O(n), more precisely $\lceil n/2 \rceil$= 16 questions (refinement options) are needed to find p in the average case. However, if one is capable of asking always a next (intelligent) question, so that half of the people can answer with "yes", then the reduction rate is optimal, i.e. 32, 16, 8, 4, 2, 1 people are within the result set after each refinement step. Hence, the number of refinement steps until finding the person is $\log(n)$ = 5. The difference of the number of refinement steps on actually large knowledge bases is of course way higher.

```
in: sq          // input:  a search query
out: ro         // output: ordered list of refinement options

 // initialization
sr <- all search results for sq;
nn <- the number of search results for sq;
cc <- the categories of the knowledge base;

// compute the reduction rate for all available categories
foreach c in cc do
        n <- the number of items in sr classed with category c;
        if n/nn <= 0.5 then
                r <- n/nn;
        else
                r <- 1 - n/nn;
        fi
        ro.append( { c, r } ); // append a tuple of category and reduction rate
done

// sort the refinement options by rate r of each category c
foreach i in [0..cc-1] do
        foreach j in [i+1..cc] do
                { c1, r1 } <- ro[i];
                { c2, r2 } <- ro[j];
                if r1 < r2 then
                        ro[i] <- { c2, r2 };
                        ro[j] <- { c1, r1 };
                fi
        done
done
```

Listing 1. Computing refinement options

Given a knowledge base as defined in Section 3, the refinement options `ro` are equal to the properties of the entities. Note: The considered knowledge base does not require a total order of all searchable items.

In Table 1 the example of Figure 1 is represented using an explicit notation form of the categories. In addition, a new hierarchy is added called “Proximity” describing the character of the nearby close proximity of the object which has the subclasses “Culture” and “Shopping”. It holds: related(Object 1, Culture), related(Object 2, Shopping), related(Object 3, Culture), related(Object 4, Shopping). At Table 1 the four categories are marked by the blue dots that might be picked intuitively by a human to describe the data in a compact way, because they have the best coverage (reduction rate is 50%) of the data in the given knowledge base and describe it significantly. Note: It is possible that an entity is not related to all hierarchies.

Table 1. Example data (w.r.t. Figure 1) with intuitively chosen refinement options

Location			**Activities**			**Proximity**		**Entities**
Greece	Island •	Crete				Character	Culture •	Object 1
Greece	Island •		Sports	SeaSports		Character	Shopping •	Object 2
Greece	Mainland	Olympus	Sports	WinterSports •		Character	Culture •	Object 3
			Sports	WinterSports •, IndoorSports	Ice-Skating	Character	Shopping •	Object 4

Hence, an algorithm is needed that is providing refinement options `ro` for the current search query `sq` considering the current search result set. The half-interval search algorithm is needed to reduce the number of search results close to 50% (optimal reduction rate) using the algorithm in Listing 1. Therefore, it is required that the algorithm for computing refinement options is returning an ordered list of refinement options where the first element provides the best reduction rate.

The algorithm computes the normalized rate search results (Line 10–18) for each category if the category might be picked. The reduction rate `r` computed in Line 13 and 15 is finally always lower than or equal 0.5.

Note, the closer `r` to 0.5 the better (with regard to the reduction rate). Afterwards, the list of categories will be sorted (Line 21–30), so that the category closest to the optimal reduction rate (i.e. 50%) is at the first position of `ro`. The following properties of the algorithm hold: (1) Any category can have the best reduction rate (not only sub-categories of the category selected at the latest); (2) Categories leading to a reduction of 40% or 60% are counted as equally optimal refinement options (cf. Line 12–16). Hence, the best k refinement options for the current search query `sq` are at index l to k. Therefore, Requirement 3 is fulfilled.

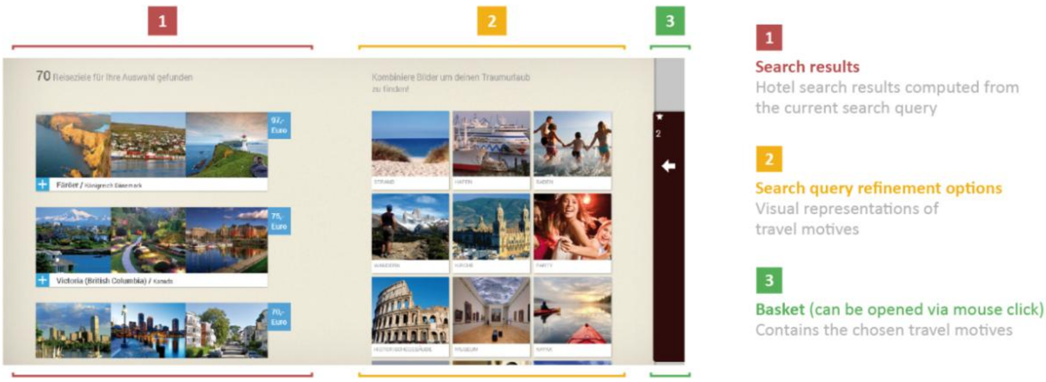

Figure 3. Experimental application with visual concepts for travel searches

5. EVALUATION

In our user study, we compared the selection-based recommendation approach with a common hierarchical approach. Both approaches rely on the same set of concepts, organized within the same hierarchy and the same knowledge base (i.e., the same possible search results).

5.1 Test Setting

The designed user interface used in our test setting is shown in Figure 3. The interface consists of three graphical regions (cp. Section 4.1).

(1) Results: The list is instantly computed for the current search query and can be scrolled. The region is empty at the start of the application.

(2) Recommendations for the current search query: The best k search suggestions provided to refine the search query. By default, a maximum of 9 search suggestions

are shown (loosely following Miller's Law (Miller 1994)). As these search suggestions should consist of neither minor fractions (e.g. 1% coverage) nor options leading to just a marginal reduction of the search results (e.g. 99% coverage), the user gets a better insight of the significant categories of the knowledge base independently from their location within the concept (poly-)hierarchy.

(3) Basket: The basket contains the previously selected search suggestions (i.e. the tokes of the search query) and therefore enables the user to check the parameters of the current search query if needed.

The graphical representation reflects the two steps as described in Section 4.1. After selecting a refinement option of `ro` [2], it is stored in the basket [3] and the search results [1] are updated (cf. Figure 4). Hence, the user receives an immediate feedback about the implications of his recent search (cf. Requirement 2). The selection-based recommendation approach (SB) starts with a scrollable choice of concepts on the first page. After picking one of these categories, the system offers a new collection computed based on the previous selection. In contrast to these automatic concept suggestions, the hierarchical approach (H) allows browsing the tree structure of the categories. At the first position within [2], a button to return to a higher hierarchy level is presented (cf. Figure 5, left).

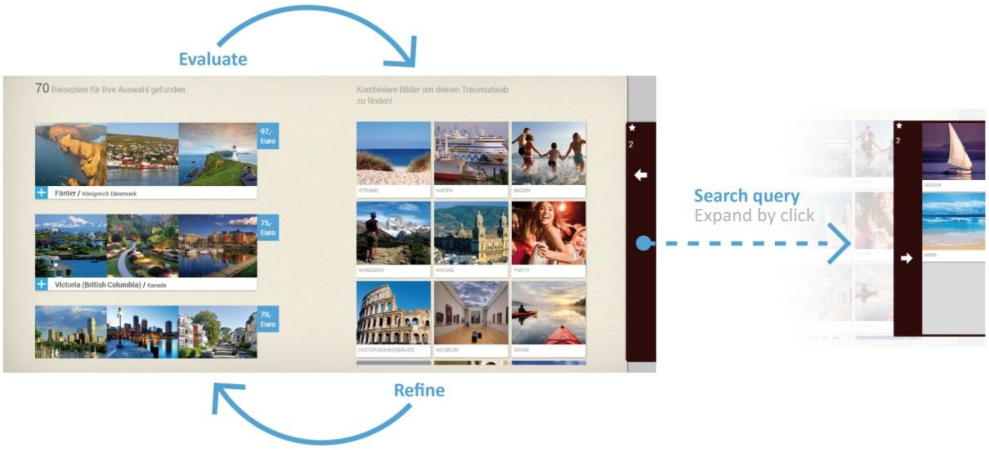

Figure 4. Lifecycle of the experimental application

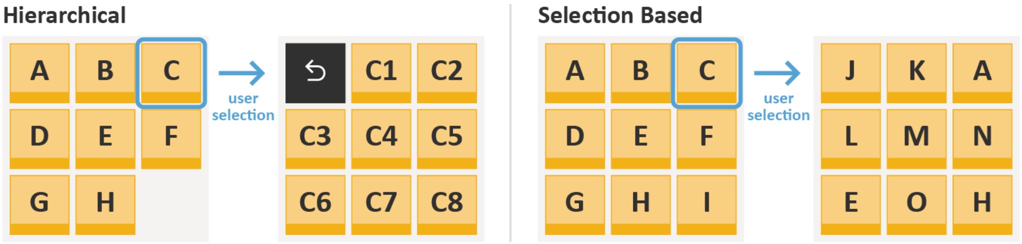

Figure 5. Differences between hierarchical and selection-based approach

For the experiment, a domain should be used well-known to most people (targeting non-expert users). In this way, we expect to balance the expert level of participants. We finally choose the travel domain, since most people already have knowledge of this domain, it is easy to understand and it addresses motive-based search as many people have an emotional connection to their holidays that also are considered as motives. Within the context of inspirational holiday search interfaces former experiences, emotions and imagination will strongly influence the user’s choices. The desired result can be considered as a summary of fuzzy concepts like adventure, pleasure or relaxation.

Especially at transactional and informational search applications, it is popular to use images as a "compelling source of entertainment and inspiration" that help to "create connections to other people and remote places," and "help us reminisce about our personal past" and "provide a way of navigating both the Web and the physical world" (Chew et al. 2010).

Since travel concepts are hard to describe in a short textual query and therefore may lead to ambiguities, several travel providers use pictures associated with these concepts. Based on the pictures the user likes (or dislikes), the provider can create a profile to recommend personalized travel offers.

Moreover, using pictures for travel searches is already a well-known approach (cp. “Picture your holiday” and “Inspire Me” in Section 2.2). This is a very simple way of communicating the personal idea of a perfect holiday. For these reasons, we used pictures in addition to every textual concept (such as "beach", "relaxing" and "culture") in the knowledge base (cf. Figure 3, [2]). The pictures were selected by a group of three domain experts to achieve the best quality and consistency with the concepts (e.g. a concept “beach” is represented by a typical beach picture).

5.2 Methodology

This study included 29 participants (15 females) in the age range of 20 – 64 years (mean value AVG = 33.1, standard degression SD = 13.1). They had to solve 12 different tasks, divided into 4 blocks (3 tasks per block): 2 blocks with tasks for a travel search with vague information need (e.g. "looking for a place to relax"), and 2 blocks with concrete information need (e.g. "looking for a holiday destination in the mountains where you can practice winter sports and join parties"). The participants started randomly with one approach and changed the approach after every task block. They had to select pictures that match best to their given task. After every selection the offered pictures and the result set were updated with regard to the respective algorithm. A task was solved when the participant reduced the result set to a limited number of results (1 – 10 results) and could not find any more concepts that matched his search task. He or she was free to decide when this was the case. Before the participants started with the experiment, they were asked to familiarize themselves with both interfaces. During the experiment, we measured the time (start and end of each task were indicated by the user) and the number of clicks needed to solve each task. Afterwards, they had to complete two questionnaires to evaluate each system regarding learnability, comprehensibility and efficiency. A Likert scale (0 = Strongly Disagree, 1 = Disagree, 2 = Neutral, 3 = Agree, 4 = Strongly Agree) was used to point out features of each approach they liked or disliked.

5.3 Results

Time and click data were subjected to 2 (system: selection-based (SB), hierarchical (H)) x 2 (task: vague, concrete) repeated measures ANOVA. Subjective ratings were compared between both systems with t-tests. For solution time, there was a significant main effect for the system, $F(1,28) = 9.510$, $p < 0.005$. The tasks could be solved with the selection-based approach in less time (SBAVG = 44 sec, HAVG = 52 sec). There was no significant effect of task, $F(1,28) = 1.65$, $p = 0.209$, and no interaction $F(1,28) = 2.34$, $p = 0.137$ (cf. Figure 6, middle). Although the interaction was not significant, planned contrasts indicated that SB was faster than H for concrete tasks, $p < 0.001$, but there was no difference between both systems for vague tasks, $p = 0.363$.

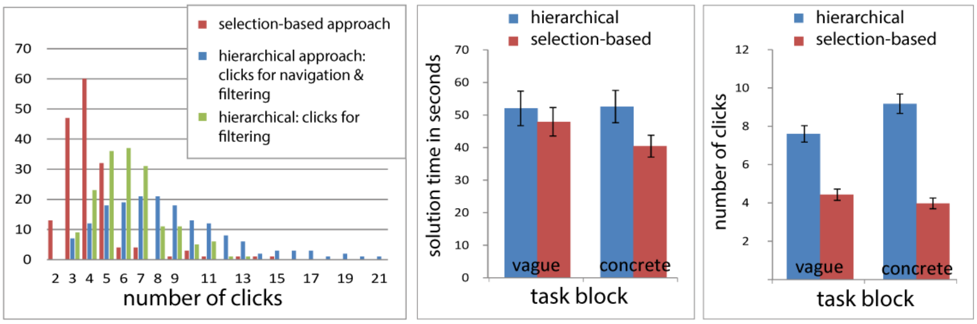

Figure 6. Histogram of clicks (left), solution time for both task blocks (middle), number of clicks for both task blocks (right)

We analyzed clicks to evaluate the effort to solve the given tasks with each approach. Due to one failure of logging the click data, only 27 measured values are considered. There was a significant main effect of system, $F(1, 27) = 114.62$, $p < 0.001$. Users made twice as many clicks in H than SB (8.4 vs. 4.2). There was no main effect of task, $F(1, 27) = 1.72$, $p = 0.201$, but an interaction of both factors, $F(1, 27) = 7.87$, $p = 0.009$. Although the difference between both systems was highly significant in both tasks, both $p < 0.001$, a higher task specificity (vague vs. concrete) only increased click numbers for H, $p = 0.036$ but made no difference for SB, $p = 0.198$ (cf. Figure 6, right). As the hierarchical system requires users to click the "Back" button to return to a higher level in the hierarchy, this might artificially increase the click numbers for this approach. Therefore, we repeated the analysis only including clicks made to select an image, excluding clicks for navigation. However, this did not change the pattern of results (cf. Figure 6, left).

The evaluation of the questionnaire (shown in Table 2) results in no significant differences between both approaches referring to perceived learnability (Q2: SBAVG = 3.556, HAVG = 3.276), efficiency (Q4: SBAVG = 3.222, HAVG = 2.897), and satisfaction (Q1: SBAVG = 3.481, HAVG = 3.138), all $|t| < 1.5$, all $p > 0.1$. In both systems, the participants were equally satisfied with the provided result set (Q6: mean value is almost identical), but the selection-based approach SB allows generating it in less time and with fewer clicks. In both parts of the questionnaire, the ranking of the aspects (Q3: AVG = 3.198) and the description of the advantages of the interface, the participants pointed out that the inspiration by pictures is well suited in the context of travel search. Furthermore, they felt well supported by the interface with regard to the inspiration for a holiday (Q12: AVG = 3.144), the exploration of unknown, interesting destinations (Q7: AVG = 3.126) and the capability to express their information needs by the provided pictures (Q11: AVG = 2.963). As disadvantage, the risk in misinterpreting the provided pictures was mentioned, because they do not possess equal expressiveness.

Table 2. Results of questionnaire

Question	H_{AVG}	H_{SD}	SB_{AVG}	SB_{SD}
Q_1: I was able to solve the majority of the tasks successfully.	3.138	0.915	3.481	0.849
Q_2: The handling of the tool was easy to learn.	3.276	0.797	3.556	0.577
Q_3: While searching for a holiday location I found pictures more inspiring than text.	3.138	1.026	3.259	0.903
Q_4: I had the feeling of finding a suitable result in a fast way.	2.897	1.012	3.222	0.892
Q_5: I could imagine to use the tool more often.	3.000	1.102	3.074	0.997
Q_6: To a large extend the results met my expectations.	3.034	0.778	3.074	0.730
Q_7: In many cases the proposed travel destinations did not meet my expectations, yet were still interesting as possible result.	3.103	0.673	3.148	0.718
Q_8: While using the tool I learned about travel destinations which I will investigate further.	2.828	1.104	2.963	0.980
Q_9: I often needed to delete images from the selection, since the results did not meet my expectations.	1.966	1.149	1.593	1.083
Q_{10}: In many cases I was given images (concepts) not corresponding to my wishes.	1.724	1.099	1.407	1.010
Q_{11}: The offered Images helped me to express my ideas.	3.000	0.886	2.926	1.072
Q_{12}: The tool helped me to get inspired for my next holiday.	3.103	1.081	3.185	0.962

5.4 Discussion

Our experimental results show a major impact on the users. They are in average more satisfied in comparison to the traditional hierarchical search approach. In particular, this is represented by question Q1 and Q2 of the questionnaire (cf. Table 2). The clear results of the time benchmark (cf. Figure 6, middle) are also obviously underpinned by the subjective judgment of the users (Q4). Hence, although the questions have not provided statistical significant results, the average ratings are clearly showing the major benefits for the users. The selection-based approach SB leads to faster solutions and drawbacks of the hierarchical search H are reduced (cf. Q9 and Q10). Considering the standard degression HSD and SBSD, we assume that the higher values can be explained by the fact that the users' justifications are highly influenced by missing concepts (e.g. Q8 and Q9). As the experimental application has no industry standard knowledge base there might be concepts not modeled that the user expected as recommendations. Finally, the results show that the underlying visual concept of the experimental application is functional (cf. Q5 and Q6). In addition, Q7, Q8, Q11 and Q12 show that the visual representation of the knowledge base concepts leads to new insights on the current search results. Although the users have not expected this, they were – using both approaches – pleased about this application behavior.

6. CONCLUSION AND FUTURE WORK

In this article, we have discussed the problems of the well-known hierarchical search approaches based on category systems. Our main contribution is an approach for searching in knowledge bases that are organized using (hierarchical) category systems (e.g. used for ontology modelling at the Semantic Web). The underlying algorithm computes recommendations for the search query refinement focusing on the optimization of the user's decision path. Even more, it inspires the user regarding the search refinement options. As category systems are well established and often used, our approach can take advantage of the structure within existing knowledge bases. In addition, it is sound even if polyhierachies are used for structuring (modern) knowledge bases at it is the case for example at the latest release of DBpedia which is a crystallization point for the Web of Data. Moreover, the approach is robust if entities are present within the knowledge base that are not related to at least one category of all given hierarchies.

We conducted an extensive user study based on a search-driven application considering motive-based search scenarios, which are driven by a search goal that cannot be formulated precisely at the beginning of the search task. A comprehensive case study was conducted in the field of e-commerce (e-travel). Our overall results show that solving the search tasks can be improved significantly. In particular, the users' time for solving the tasks was reduced by 15.58% (average improvement) and the number of clicks dropped by 31.85%, while the other key performance indicators are not deteriorating. The experimental results show an obvious trend of increased user experience as well as inspiration while the search result quality has not dropped (rated by the users).

We see this work as one step in a larger research agenda. Based on our current results, we aim to develop a new paradigm for realizing search-driven applications, which employs knowledge bases for support in solving search tasks in a world of an increasing amount of data. Future research has to aim truly on exploiting the semantics (e.g. structures and meanings) of the knowledge bases. Also the understanding of the visual content conveyed by the used images should be noted. A picture usually contains multiple objects. If the user selects an image, his intent may not be the same as the intent of the designer. Offering multiple images for one concept or allowing the user to specify regions of interest inside the image (Sang et al. 2013) could reduce these ambiguities. Another option is the computation of prototypical images for each category. We expect that research done in this field will provide knowledge base engineers as well as usability engineers with valuable foundations on which they can develop future intuitive and inspiring search-driven applications.

ACKNOWLEDGEMENT

This work has been supported by the European Union and the Free State Saxony through the European Regional Development Fund (ERDF). The research presented in this article has been conducted in cooperation of the Chair of Media Design – Technische Universität in Dresden, Unister GmbH from Leipzig and queo GmbH from Dresden, Germany. Thanks are due to Severin Taranko, Fred Funke, Steffi Schattenberg, Marcus Kirsch and Romy Müller for their invaluable feedback and support in this research.

REFERENCES

Bizer, C. et al., 2009. Linked data - the story so far. *Int. J. Semantic Web Inf.* Syst. Vol. 5, No. 3, pp. 1–22.

Chew, B. et al., 2010. Understanding the everyday use of images on the web. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. NordiCHI '10, ACM, New York, pp. 102–111.

Cormen, T. H. et al., 2001. Introduction to Algorithms, *2nd ed. McGraw-Hill Higher Education*.

Dörk, M. et al., 2012. Navigating tomorrow's web: From searching and browsing to visual exploration. *ACM Trans. Web*. Vol. 6, No. 3, pp. 13:1–13:28.

Inc., G. Google Instant – Inside Search. *Website*, 2010. Available online at http://www.google.com/insidesearch/instant-about.html; visited on 2014/12/15.

Hearst, M. A., 2006. Clustering versus faceted categories for information exploration. *Commun. ACM*, Vol. 49, No. 4, pp. 59–61.

Hilbert, M. and Lopez, P., 2011. The world's technological capacity to store, communicate, and compute information. *Science*, Vol. 332, No. 6025, pp. 60–65.

Hitzler, P. et al., 2011. Foundations of semantic web technologies. *Chapman and Hall/CRC.*

Keck, M. et al., 2013. Improving motive-based search: Utilization of vague feelings and ideas in the process of information seeking. In *Distributed, Ambient, and Pervasive Interactions*, N. Streitz and C. Stephanidis, Eds., Vol. 8028 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 439–448.

Lassila, O. and Swick, R. R., 1999. Resource Description Framework (RDF) Model and Syntax Specification. *W3C recommendation*

Lehmann, J. et al., 2014. DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. In *Semantic Web*, IOS Press, 2014. ISSN 2210-4968.

Marchionini, G., 2006. Exploratory search: From finding to understanding. *Communications of the ACM*. Vo. 49, No. 4, pp. 41–46.

Miller, G. A., 1994. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, Vol. 101, No. 2, pp. 343.

Morville, P., and Callender, J., 2010. *Search Patterns - Design for Discovery*. O'Reilly.

Morville, P., and Rosenfeld, L., 2006. *Information Architecture for the World Wide Web*, 3nd ed. O'Reilly & Associates, Inc., Sebastopol, CA, USA.

Polowinski, J., 2009. Widgets for faceted browsing. *Human Interface and the Management of Information*. Designing Information Environments, pp. 601-610, Springer Berlin Heidelberg

Ranganathan, S. R., 1965. *The Colon Classification*. Rutgers Series on Systems for the Intellectual Organization of Information; 4. New Brunswick : Graduate School of Library Science.

Russell-Rose, T. and Tate, T., 2013. *Designing the Search Experience - The Information Architecture of Discovery*, Morgan Kaufmann, Waltham, USA.

Sacco, G. M., and Tzitzikas, Y., 2009. Dynamic Taxonomies and Faceted Search Theory, *Practice, and Experience*. Springer, New York.

Sang, J. et al., 2013. Interaction design for mobile visual search. *IEEE Transactions on Multimedia*, Vol. 15, No. 7, pp. 1665–1676.

Stefaner, M. and Müller, B., 2007. Elastic lists for facet browsers. *DEXA Workshops*, pp. 217-221, IEEE Computer Society. ISBN: 0-7695-2932-1.

Studer, R. et al., 1998. Knowledge engineering: principles and methods. *Data & knowledge engineering*, Vol. 25, No. 1, pp. 161–197.

Tunkelang, D., 2006. Dynamic category sets: An approach for faceted search. *ACM SIGIR*, Vol. 6.

Waldner, M. et al., 2013. FacetClouds: exploring tag clouds for multi-dimensional data. *Proceedings of Graphics Interface 2013 (GI '13)*. Canadian Information Processing Society, Toronto, Ont., Canada, pp. 17-24.

Wei, B. et al., 2013. A survey of faceted search. *Journal of Web Engineering*, Vol. 12, No. 1-2, pp. 41–64.

Wood, D., 2010. *Linking enterprise data*. Springer ISBN 978-1-4419-7665-9.