

QueryCrumbs: A Compact Visualization for Navigating the Search Query History

Christin Seifert, Jörg Schlötterer, Michael Granitzer
University of Passau
Passau, Germany
 {firstname.lastname}@uni-passau.de

Abstract—Models of human information seeking reveal that search, in particular ad-hoc retrieval, is non-linear and iterative. Despite these findings, today's search user interfaces do not support non-linear navigation, like for example backtracking in time. In this work, we propose QueryCrumbs, a compact and easy-to-understand visualization for navigating the search query history supporting iterative query refinement. We apply a multi-layered interface design to support novices and first-time users as well as intermediate users. The formative evaluation with first-time and intermediate users showed that the interactions can be easily performed, and the visual encodings were well understood without instructions. Results indicate that QueryCrumbs can support users when searching for information in an iterative manner.

Keywords—Information Retrieval; Query History Visualization; Information Re-finding; Search History;

I. INTRODUCTION

A common phenomenon in Web search is that users re-access Web resources that have been found in the past. Information re-access differs mainly from information seeking by being more targeted and more directed involving recognition and recall activities [1]. While active strategies for information re-finding (i.e., explicit storage of the information) would directly support information re-finding, passive strategies with no explicit storage are much more common, especially when search tasks are interrupted [2]. Such passive strategies require to recall how or where the information was found previously. Resuming a search from a previous query relying on human memory has been shown to be only accurate in 72% of the time [3].

The demand to include a search history has also been stated in the context of information seeking models. Models of human information seeking describe and structure the way humans search for information in an information source (for an overview see [4]). These models define human information seeking as an iterative process in which query reformulation is a common step (e.g. [5], [6]). Usually, multiple query reformulations are necessary before the information need is fully satisfied, which can be supported by search history visualizations (e.g. [7], [4]).

In this paper, we propose QueryCrumbs, a compact, interactive, simple-to-understand visualization for accessing, altering, and resubmitting previously issued queries. The

concept is similar to bread crumbing interfaces as navigational aid for web sites [8, p. 221f]. Figure 1 shows the conceptual idea of the QueryCrumbs visualization. Each query is represented by a mark, the position of the mark indicates the position of the query in the sequence of queries. We introduce two different measures for query similarity to capture the general relationship between queries. The similarity is measured on different levels of detail, suitable for different user groups and tasks. In order to evaluate the usefulness of this visual representation, we pursue a layered interface design approach [9] introducing different notions of similarity in each layer. We evaluate the visualization and interaction design in a formative user study with novices. Concretely, the contribution of this paper is as follows:

- We introduce a human querying model as conceptual basis for search history visualizations.
- We propose and evaluate QueryCrumbs, a compact, search-engine agnostic, interactive visualization supporting overview and navigation of the query history while taking up minimal screen space (i.e., for mobile environments or with minimal impact on current search result page designs).
- We account for universal usability by applying the multi-layered user interface design method to the design of the visualization.

This paper is organized as follows: After discussing related work, we describe the human querying model (Section III) and from that derive the conceptual idea for the visualization (Section IV). Then, the multi-layered approach to visualization and interaction design is explained in detail in Section V and evaluated in Section VI. We conclude the paper with an outlook on future work.

II. RELATED WORK

We review insights on human querying behavior gained from web logs and human search models to motivate the human querying model as the conceptual basis for QueryCrumbs. Further, an overview of and design guidelines for search history visualizations, and the relationship to information re-finding behavior and related tools are presented.

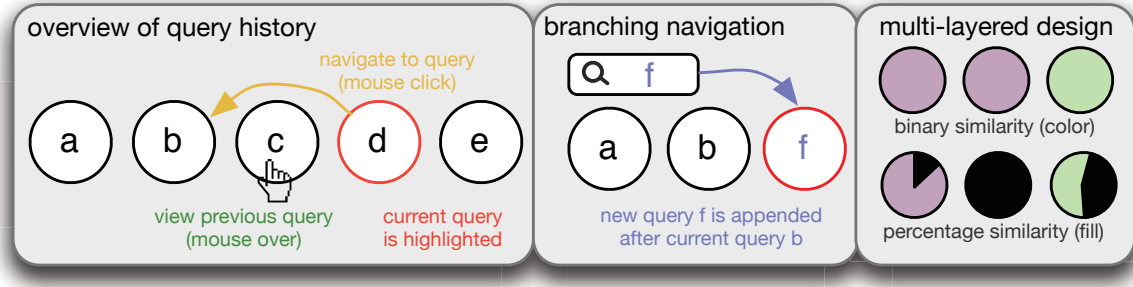


Figure 1. QueryCrumbs visualization concept. Left: A sequence of queries (a,b,c,d,e) is shown, the current query (d) is highlighted, while navigating back to a previous query (b) reissues the query. Hovering over a query (c) shows the query terms and the similarity to all queries. Middle: Issuing a new query from a previous one (b) removes previously subsequent queries (c,d,e) showing only the current path of interest. Right: Query similarity is based on the similarity of the search result lists and can be encoded with different levels of detail.

A. Human Search Models and Querying Behaviour

Human Search Models can be distinguished into models with static information need (e.g., [6]) and models with dynamic information need. Models with static information need assume that the initial information need does not change during the search session. Still, these models describe an iterative process and include the need of query reformulation and potential backtracking. An example model with dynamic information need is the “berry picking” model [5]. Starting with an initial query humans evaluate the results, which leads to new thoughts and to a rephrasing of the query. By repeating this process, the user discovers new resources and thoughts, which is likely accompanied by query modifications.

Web query log analyses provide statistical data about human querying behavior. Broder distinguished transactional, navigational, and informational queries, with 50% of queries being informational queries, ranging from a very broad to a very narrow description of the information need [10]. In an analysis of Altavista query logs, 52% of users modified their queries [11] and 32% of the sessions contained three or more queries. In another study on the same data set 37% of all queries were found to be query modifications of various types [12].

Human search models postulate that human querying is an iterative process in which query refinement is a common step, a finding that is backed up by log analysis results. Insights from the human search models and discussed log analyses are reflected in the human querying model we present as basis for the QueryCrumbs.

B. Search History Visualizations

While the above mentioned models implicitly indicate the requirement for user interfaces supporting search history navigation, this need has been explicitly stated by multiple authors (e.g., [4]). Search history visualizations share many demands with browsing history visualizations [13], [14], which is also reflected in the approaches presented in this

and the next section. A commercial example is Google’s Wonderwheel, a visual tool for interactively finding related queries [15] in which a query is represented as a node in a graph. Similarly, the Footprints [16] tool exploits navigation paths of users to suggest potential paths through the information space. Web pages are represented as nodes in a graph representing most visited paths. Wonderwheel and Footprints visualize the complex information space and focus on exploration of the space generated by other users. QueryCrumbs in contrast focus on exploitation of the user’s own history.

Komlodi et al. present design guidelines and examples for search history visualization based on a study with librarians [17], [18]. This work is similar to ours, while their target user group is different (search experts vs. casual searchers). Their interface follows the information webspace concept [19], and therefore has richer interactions and is much more complex. Padprints [20] visualizes the history of web pages, and is similar to QueryCrumbs in the simplicity of design. Conceptually similar to our work are bread crumbing interfaces [8, p. 221f] introduced as navigational aid for web sites.

C. Information Re-finding

While also relying on history mechanisms, information re-finding differs from information seeking [1]. Information re-finding tasks can be categorized into short-term (retrieving just visited information), mid-term and long-term (re-finding information after months or years) tasks [21]. Re-finding behavior was also observed, when an information seeking task is interrupted [2] and is not well supported by standard Web browsing interfaces [22]. A study by the same authors showed that while being interrupted 58% of users did nothing to explicitly store the retrieved information (passive storage) and relied either on passive (memory, open browser windows) or active retrieval mechanisms (re-querying or browser history) [2].

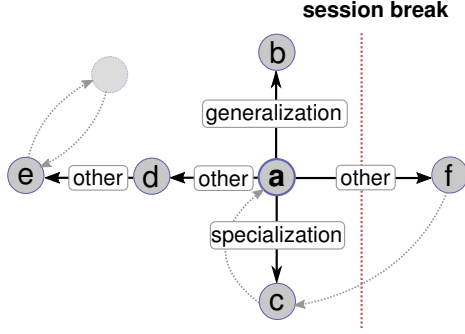


Figure 2. Human querying model. Query modifications with the same search intent include specification and generalization. A session break occurs when the search intent changes.

Tools supporting an active strategy for information re-finding are for example Session Highlights [23] and a plugin for storing web page summaries [24]. While these tools require an explicit user interaction to store the information, the SearchBar [2], SearchPad [25] and YouPivot [26] assume a passive user behavior for information storage. Re-finding tools assuming passive user behavior have also been proposed for other application areas, such as graphical history for visualizations [27], or information re-finding within a Web page [28]. SearchBar, SearchPad and YouPivot are the most similar tools to QueryCrumbs, but require much more screen space and complex information management.

In summary, our approach is search-centric, and covers short-term to long-term re-finding strategies for users that pursue a passive information keeping behavior.

III. HUMAN QUERYING MODEL

Before introducing the concept for the QueryCrumbs visualization, we define the underlying human querying model. Human information seeking models capture the process required to satisfy a user’s information need, they do not model the querying process explicitly. Deriving the information need from a query or a set of queries is ongoing work in the information retrieval community [29]. Multiple queries might reflect the same information need and different information needs might be expressed by the same query. An example for the former are the two queries “buy mobile phone” and “buy phone”, an example for the latter is the query “java” where a user might seek information for the island, the coffee, or the programming language.

As the queries and the retrieved results are the only data that is generally available to a search client, we introduce a human querying model on the basis of this data. This makes the querying model search-engine agnostic, i.e., we do not make any assumption about the type, nature or amount of the back-end search systems. The model has the form of a graph, in which nodes correspond to queries and edges reflect query modifications (see Figure 2). A user starts with an arbitrary query a . When the results for

this query do not satisfy the user’s information need, the user can either generalize the query (if it was too specific) leading to query b , specialize it (if it was too generic) leading to query c , or modify it in other ways leading to query d . Other modifications capturing the same search intent include the use of synonyms or rephrasing. When the search intent changes with the modification of the query (f), a session break occurs. Figure 2 only captures the trellis of the underlying graph, subsequent query modifications could lead to circles (as indicated by the light gray node in the figure). The general graph has an infinite number of nodes (because there is an infinite number of potential queries). Users navigate through this general graph, and the queries a user issues correspond to a (potentially cyclic) subgraph. This human querying model can be seen as a special case of an information seeking model. It does not make any assumption about the underlying information need, but captures the querying and query modification process. The simplification allows to approach an easy-to-understand visualization of the human search process.

IV. QUERYCRUMBS CONCEPT

Conceptually, QueryCrumbs visualizes the most recent path through the general querying graph, i.e., the user’s history of search queries, supporting the four users tasks:

- **Recent Queries:** Get an overview of the recent query history, i.e., the sequence of queries.
- **Navigation:** Navigate back to previous query. Easily access results from previous queries.
- **Simple comparison:** Identify similar searches conducted in the past, and thereby identify search sessions and session breaks.
- **Quantitative comparison:** Compare the quantity of overlapping search results for different queries. Investigate how the result set changed quantitatively.

Figure 1 shows the concept of the visualization and interaction design. We choose to use a simplification of the human querying model introduced in the previous section. We do not show the explicit branching, but rather visualize the query history in a linear fashion, unrolling any cycles. This decision on simplification is supported by a study on web search logs providing details on branching and backtracking behavior [30]. Because queries tend to get more complex at the end of a session, users backtrack to the more general query and start refining it. However, within one session (i.e., one information need) they hardly revisit a path they backtracked from. Also, removing branches after backtracking keeps the visualization small and comprehensible, while at the same time supporting the majority of query refinement steps within one query session. Explicitly displaying all query interaction would result in a rather complex graph. Such graphs are hard to layout in a visually pleasing way and hard to navigate [31] and supporting small screens (e.g., mobile phones) would no longer be possible.

Query marks are arranged from left (older) to right (most recent) to give an overview of **recent searches**. We propose a simple mouse-over interaction for previewing a previous query (i.e., show the query terms for this query), and a mouse click for **navigating** to a query. Navigation to a query means reissuing this query.

A. Measures for Query Similarity

The comparison tasks introduced at the beginning of this section (simple and quantitative comparison) require a notion of similarity between queries. Query similarity can be either calculated on the basis of the query string or on the basis of the results returned. Because the former does not capture semantic similarity, (e.g., the terms “car” and “automobile” are considered as different), we focus on query similarity based on the retrieved results. For example, the two queries “automobile” and “cars” are syntactically different, but could lead to similar results when posed to a search engine. Thus, deriving similarity based on results sets renders the visualization search-engine agnostic.

Typically, search engines return a ranked list of results for a query k . Let this ranked list be denoted by $R_k = [r_k^1, \dots, r_k^i, \dots, r_k^n]$, where r_k^i is the i -th result for query k . Because users of Web search engines only access the top items in the result list [32], [33], the similarity calculation is based on the top τ items, yielding the ranked list R_k^τ .

Two queries can be compared pairwise based on their result list by identifying the overlapping elements. Let $L_k^\tau = \{r_k^1, \dots, r_k^i, \dots, r_k^\tau\}$ be the (unordered) set of results. The similarity sim_r of two queries can then be calculated as the Jaccard coefficient [34] on the two result sets:

$$sim_r = \frac{|L_i^\tau \cap L_j^\tau|}{|L_i^\tau \cup L_j^\tau|} \in [0, 1] \quad (1)$$

sim_r can be expressed as a percentage to which we further refer to as **percentage similarity** (cf. Figure 3, right) and corresponds to the user task **quantitative comparison**.

A binary indicator variable s_r can be obtained by introducing a similarity threshold $\theta \in [0, 1]$, and is calculated as follows:

$$s_r = \begin{cases} 1, & \text{if } sim_r \geq \theta \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

We further refer to s_r as **binary similarity** (cf. Figure 3, left). This similarity corresponds to the user task **simple comparison**.

B. Layered Approach

Intended for the use with general search engines having mostly casual users, novices and first-time users should understand the visualization without instructions. Layer 1 is designed for the tasks “recent queries”, “navigation” and “simple comparison”, and therefore introduces all interactions. Layer 2 adds the more complex notions of similarity is designed for the tasks “quantitative comparison”.

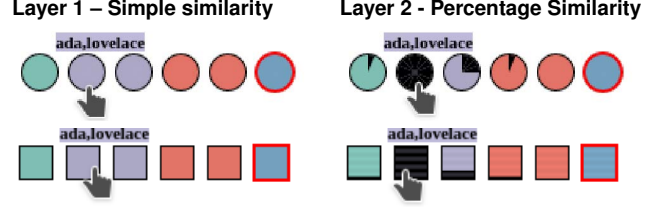


Figure 3. Layers for two different visual marks for queries “ada” “ada lovelace”, “ada byron”, “ada language”, “ada programming”, and “alan turing”. Current query is highlighted in red. [Best viewed in color]

In the design we also considered potential adaptation for mobile devices. Adaptability is ensured by (i) compactness of the visualization and (ii) simple interactions that can be performed with either a mouse, or on a touch display. On the latter, a double-tap is the equivalent to the mouse interactions: the first tap represents the mouse over, while the second tap executes the click.

V. QUERYCRUMBS VISUALIZATION

The visualization concept described in the previous section is implemented in D3.js [35] and released¹. under the MIT license. Next we describe the final design. Design alternatives and choices based on pre-study results are discussed in section V-B.

A. Visualization and Interaction Design

In the basic design each query is represented by a mark. Query similarity is encoded in the mark’s appearance and position is used to show the query sequence. Figure 3 shows an example of the QueryCrumbs visualization.

The mark for a query is either a circle or a square with fixed size. In the user evaluation we addressed the question with which form (circles or squares) the similarity can be more accurately interpreted by users. The currently selected query is outlined with a red border.

In layer 1 (Figure 3, left) the binary similarity s_r from Equation 2 is encoded by color. Similar queries have the same color. We used a color map for qualitative data from ColorBrewer [36]. In a sequence of queries a new query q might be similar to more than one previous query a and b , but a and b might not necessarily be similar to each other. All choices to resolve this coloring ambiguity significantly increase the perceptual complexity of the visualization. We avoid such a complexity by choosing the color of the most recent, similar query instead. This tends to (i) color the new query with the color of the current session if it belongs to it, and (ii) visually shows if the same query or session was issued in the past (with a different session in between).

In layer 2 (Figure 3, right), the percentage similarity from Equation 1 is encoded in the fill-level of the mark. For circles, the angle of the filling and for squares the height of the filling corresponds to the percentage similarity.

¹bit.ly/1PFva5O features the source code and an installable demo

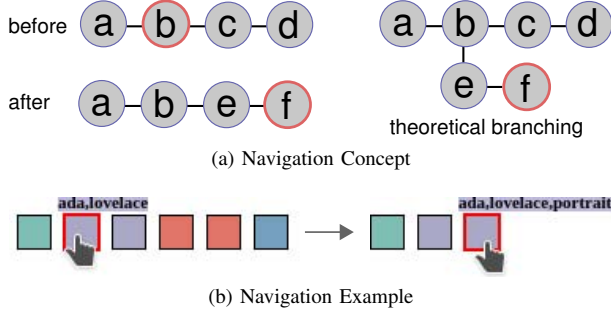


Figure 4. Interactions. (a) Initial history, query “b” selected. Theoretical tree after two new queries “e” and “f” and visualized part of the tree. (b) QueryCrumbs before and after issuing a query from a previous one. [Best viewed in color]

The QueryCrumbs visualization has two simple **interactions**. Mouse over allows to access basic information about a query. In layer 1 this information is the query string, in layer 2 the similarity to other queries is added. An example is shown in Figure 3. A mouse click highlights the selected query mark and reissues the query. If the user issues a new query (being on a previous-to-last query) this would mean a branching of the query history as shown in Figure 4a.

Because this branching could get rather complex as outlined in the introduced querying model, we remove all the query marks on the right of the current query (more recent queries) and append a new query mark. To make the change in the visualization better perceivable, the transition in the layout is animated. Figure 4b shows the step before and after the transition when the query “ada lovelace portrait” is issued from the second query in the history “ada lovelace”.

B. Pre-Study and Design Decisions

In a preliminary study we tested various design alternatives on web-based prototypes with a small user group. This prototype was not fully functional, backtracking and reissuing queries was not implemented. Users were given different designs and were asked whether they find it visually pleasing and understandable.

In this prototype we constantly displayed the query terms for all queries. User found this much text is (i) more hindering than helpful and (ii) poses a layout problem for long queries, which cannot be solved in limited space for arbitrary query lengths. Thus, in the subsequent design we only show the query terms for the current query on mouse-over.

We tested to encode the percentage similarity as edges between subsequent queries, with the width of the edge corresponding to the similarity a design similar to the Footprints interface [16]. Users liked the simpler design more and expected the visualization to be much too complex if all pairwise similarities were encoded via edges.

We also found that the different similarity notions are hard to understand for users, and therefore we introduced

the layered interface design.

The sub-marks of the current query were only colored dark if they reappeared in another result list, different from the hovered one. Users did not understand why results that are currently displayed in the accompanying search result list are not marked in the query mark. Therefore, we decided to compare the hovered query also to itself, which colors the sub-marks for all results in the list dark gray.

VI. FORMATIVE EVALUATION

In the user evaluation we wanted to assess whether the visualization can successfully be used (understanding the visualization and interactions), which benefits users see, and whether they would use it in the future. Therefore, we posed the following hypotheses:

- H1: Layer 1 can be understood and used successfully **without** instruction. This comprises issuing a query, navigating back to a previous query, reissue a query from a previous one and the simple similarity coding.
- H2: The percentage similarity coding (layer 2) is understandable **with** instructions. There is a difference in using the two different marks as query representatives (squares or circles).
- H3: When having experience with the QueryCrumbs users tend to use it in a real-world usage scenario.

A. Design

We used a between subjects design with the independent variable *form* with two levels (square or circle). Dependent variables are *completion time* (in seconds), *task success* (binary) and *understanding* (binary). *Task success* measured the correctness of the performed interaction, i.e., whether the visualization is in the intended state after the interaction and was judged by the evaluator. The variable *understanding* captures whether the user was able to interpret the state of the visualization and was assessed by questions users had to answer after performing a task. The correctness of the answer was judged by the evaluator. In a questionnaire we asked for perceived beauty and helpfulness, expected uptake, and layer preferences (either layer 1 or layer 2). We used a five-point Likert-type scale, with “1” coding the worst value and “5” encoding the best value.

B. Participants and Procedure

20 German-speaking volunteers (undergraduate and post-graduate students) with normal or corrected-to-normal vision participated in the 30 minutes evaluation, 10 males and 10 females. Their age ranged between 20 and 33 years with 50% of the participants being between 22 and 26 years. One participant was a novice computer user, 10 participants rated themselves as intermediate users and 9 as experts.

The evaluation comprised three phases, with predefined tasks sets for each phase (see details on tasks sets below). In the first phase, layer 1 was evaluated using Task Set A.

Table I
TASK OVERVIEW. S(UCCCESS), U(NDERSTANDING), T(IME), NUMBER
AND TYPE OF INTERACTIONS (I). GRAYED VARIABLES WERE
MEASURED, BUT ARE NOT THE FOCUS IN THE EVALUATION. TASKS IN
ITALIC USED TO PREPARE THE VISUALIZATION.

ID	Task Description	Measures			
		S	U	T	I
Task Set A – LAYER 1					
T-A1	issue “passau”, “mauerbau berlin”, “bier”, “dalai lama”, “ebola”	✓		✓	
T-A2	back to previous	✓	✓	✓	
T-A3	issue from previous (“mauerbau berlin 1961”)	✓	✓	✓	
T-A4	issue “luther”, “martin luther”, “luther wittenberg”	✓		✓	
T-A5	estimate binary similarity	✓	✓		
Task Set B – LAYER 2					
T-B1	estimate percentage similarity	✓	✓		
T-B2	issue “lovelace”, “ada lovelace”, “ada countess”, “ada byron”	✓		✓	
T-B3	estimate percentage similarity	✓	✓	✓	
T-B4	estimate percentage similarity	✓	✓	✓	
OPTIONAL USAGE OF QUERY CRUMBS					
T-C	write a blog entry	✓			✓

In the second phase, layer 2 was evaluated using Task Set B. The third phase consisted of task T-C.

At the beginning, participants obtained some general instructions on how to handle the evaluation interface. For each participant the query history was set empty at the beginning, i.e., the QueryCrumbs were not visible. Because we wanted to evaluate whether the visualization in its basic design (layer 1) is understandable without explanations, participants did not receive any explanations about the visualization. Before the second phase, participants received a short introduction (one written paragraph) about the percentage similarity coding (layer 2). For each task in phase 1 and 2 we automatically collected the completion time. Because the results were retrieved on-line from a Web search engine, we controlled for network latency by subtracting the time it took the search engine to respond (which was below 1 second for each query). After the second phase participants had a short break and were told that from now on the completion time was not measured anymore. For the third phase participants received only the task instructions. At the end participants filled out the post-study questionnaire.

Some tasks contained explicit questions and the participants had to speak out loud the answer. For other tasks the correctness of the answer could be judged by observing the state of the user interface. The experimenter noted the correctness for each task.

C. Tasks and Test Material

Table I gives an overview of the tasks and the measured variables task success (S), understanding (U), and completion time (T). *Task Set A* (task T-A1 to T-A5) was performed

with layer 1 of the visualization (see Figure 3, left). With *task set B* layer 2 was evaluated (see Figure 3, right). *Task T-C* is designed to assess potential uptake of the visualization.

The query issuing tasks T-A1, T-A4, and T-B2 required users to type a query in a search field and are used to prepare the visualization for the subsequent tasks. We report them, because we compare the completion time of these tasks to the tasks which required issuing queries using the visualization. Task T-A3 required users to issue a query (with QueryCrumbs) and was used to measure understanding, i.e., whether users can correctly interpret how and why the visualization’s state changed. The instructions for the query issuing tasks (T-A1, T-A3, T-B2, T-A4) were the following (translated from German): *Enter the search terms [...] in the search box*. For T-A3 the task instructions also contained the question *Please explain how the visualization has changed*. The instruction for the similarity estimation tasks were *Which of the previous queries are similar to each other?* and *Please estimate the similarity of the queries X, Y and Z to each other*. Users were asked to mark their estimate (one of 0%, 25%, 50%, 75%, 100%) for all query pairs.

T-C is a creative task, asking users to search for related material on a blog post they are writing. Task T-C was formulated as follows (translated from German, shortened): *You want to write a blog entry about the life of Ada Lovelace. You are looking for images to illustrate your blog entry. Use the browser extension to find relevant images, copy them to a text editor and provide a short description of the image content*. Note that in this task users were not explicitly asked to use the QueryCrumbs visualization, but the extension in general. However, in a previous task (T-B1) the search queries that had to be input were “lovelace”, “ada lovelace”, “ada countess”, and “ada byron”, which would have been a good starting point for a search. The task counted as successfully solved, if users found five images that were relevant for the task. For Task T-C we counted how many users used the QueryCrumbs, and which interactions (I) they performed with the visualization.

For the evaluation we used a browser extension that provides a sidebar alongside each Web page [37]. This extension provides access to the Europeana collection², the European aggregator for digital museum objects, and was modified to collect the evaluation measures.

Figure 5 depicts the sidebar. Users can input a query in the search field ①, and search results are displayed in the result list ② as document surrogates [6]. The QueryCrumbs visualization provides an overview of and access to previous queries ③. Users can start and stop an evaluation task using the controls on the top right ④. When the start button is clicked, an input field for the task id appears ⑤ and disappears after the task id was given. The correctness of the task id is ensured by the evaluator. The measures are

²<http://europeana.eu>

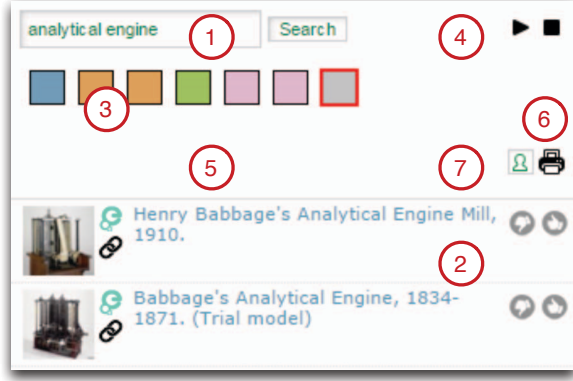


Figure 5. Evaluation User Interface (result list cropped).

stored in the browser’s local storage and can be downloaded at the end of the evaluation ⑥. The layer of the visualization can be set in the user profile ⑦ by the evaluator.

The QueryCrumbs visualization was configured to show eleven previous queries. More queries were not required in the evaluation and the size of the sidebar restricted the size of the displayable queries. The similarity calculations were based on the 16 top-most search results. The query similarity threshold θ was set to 0.1 for the binary similarity which was determined as a good threshold for visually indicating similarity in preliminary experiments.

D. Results

We report the results of the formative user evaluation separately for each layer, the task measuring potential uptake and the questionnaire.

1) *Measured Performance for Layer 1:* Table II shows task success, understanding, and completion time for the tasks performed with layer 1. Values are aggregated over all users independent of whether they used the circles or square condition. We found no influence of the variable *form* (circle or square) and thus omitted the values in the table. In task TA-1 and T-A4, in which users had to issue a query in the search field, we did not measure understanding. Similarly, for task T-A5 measuring completion time was not applicable, because users had to answer a question which required an explanation.

Nearly all, but Task T-A2 (back to previous), were successfully performed by all users. In Task T-A2 only 10 users (50%) successfully navigated back to a specific previous query. This means, the other 10 users either did not choose a previous query at all or did not choose the requested one. Conversely, the understanding rate for this task was high, 13 users (65%) still interpreted the state of the visualization correctly. This means, that although some users did not perform the interaction as intended, they still were able to understand the change in the visualization. Therefore, we

Table II
RESULTS FOR LAYER 1. SHOWING MEAN AND STANDARD DEV FOR COMPLETION TIME AGGREGATED OVER ALL USERS (ONE MISSING VARIABLE IN T-A1 AND T-A2). “N.A.” – MEASURE IS NOT APPLICABLE.

Task	Success [%]	Understanding [%]	Time [sec]
T-A1	100	n.a.	54 ± 17
T-A2	50.0	65.0	23 ± 22
T-A3	100	57.5	24 ± 18
T-A4	100	n.a.	35 ± 13
T-A5	100	100	n.a.

expect an increased success rate on the repeated execution of the task.

Task T-A3 (issue from previous) shows different results. Although all users successfully performed the interaction, only 58% could interpret the result correctly. This means, the navigation concept outlined in Figure 4 was understood by the majority, but not by all users. It is to note, that some users understood part of the result, and we encoded this with a 0.5 value. That is, while they could not interpret the removal of the previous branch correctly, they still understood that the new query was appended at the current position.

If users successfully issued a previous query it took them 10 sec on average (Task T-A2, depending on task success). For a successful reissuing of a previous query users first needed to find the query in the visualization (mouse over), and then click the query mark. Typing a new query of similar length took 24 sec on average (Task T-A3). Interpreting the binary similarity of two queries (Task T-A5) was successfully performed and also correctly understood (query representatives have same color) by all users.

Summing up, we conclude that color coding of the simple similarity was well understood by all participants without instruction. Not all users (50%) performed the interaction for navigating back correctly, but 65% understood the interaction result. Reissuing a previous query is faster with the QueryCrumbs than typing a new query.

2) *Measured Performance for Layer 2:* Table III summarizes the results for layer 2. We do not report task success in this table. All tasks were executed correctly by all users, i.e., task success is 100% for all tasks. Also, all users correctly understood the encoding of the percentage similarity by fill level (understanding is 100%). There was no influence of the variable *form* (circles or squares) on the perception of the similarity coding.

It took users on average between 20 secs and 48 secs to complete a task. There was no significant effect of *form* on completion time for any task ($[F(1, 18) = 0.201, p = 0.660]$ for task T-B2, $[F(1, 16) = 0.945, p = 0.346]$ for task T-B3, $[F(1, 17) = 0.737, p = 0.403]$ for task T-B4).

Summing up, we conclude that query result similarity was correctly interpreted by all participants in all conditions (100% task success), and the form of the mark had no

Table III

RESULTS FOR LAYER 2 AGGREGATED OVER ALL USERS. SHOWING MEAN AND STANDARD DEV FOR COMPLETION TIME (2 MISSING VALUES FOR T-B3, ONE FOR T-B4). “N.A.” – MEASURE IS NOT APPLICABLE .

Task	Understanding [%]		Time [sec]	
	Squares	Circles	Squares	Circles
T-B1	100	100	n.a.	n.a.
T-B2	n.a.	n.a.	35±7	45±27
T-B3	100	100	38±12	48±31
T-B4	100	100	24±6	21±10

Table IV

USAGE STATISTICS FOR TASK T-C

users using QC (any interaction)	12 (60%)
#users reissue with QC	10 (50%)
#users reissue with QC, from previous task	6 (30%)
total #queries	191
average #queries	9.55
#queries with QC	33 (17%)

influence on the completion time.

3) *Usage in Creative Task (T-C)*: In tasks T-C users were free to choose whether or not to use the QueryCrumbs visualization. Table IV shows an overview over the usage of the QueryCrumbs for this task.

12 participants (60%) used QueryCrumbs to find material for their blog post, 6 of them remembered and reissued a query that had been issued in a previous task. The majority of those who used QueryCrumbs reissued a previous query (10 participants), 2 participants only used it for scrolling through the query history (mouse over). In total, 191 queries were issued in this task, 17% of the queries were reissued using the visualization.

In total, 90% of all users successfully completed this task, i.e., found five suitable images to include in the blog post. The task success rate was 91% for participants using QueryCrumbs, and 88% for those not using the visualization. Due to the limited amount of data no conclusions can be drawn for the influence of QueryCrumbs usage on completion time.

Summing up, we conclude that the majority of the participants chose to use QueryCrumbs.

4) *Questionnaire Results*: Table V summarizes the quantitative values from the questionnaire. Generally users rated the QueryCrumbs rather high in all categories, i.e., above the theoretical average of 3 for all variables. The similarity color coding and reissue interaction (both average rating of 4.1) were perceived as especially helpful. Users indicated that they would use both layers in the future (rating of 3.6 for both), but if given a choice, 15 (75%) would prefer the (feature-rich) layer 2. 11 users would prefer circles as marks and 9 squares. Only 5 users deviated in their preference from the condition they had been assigned to (i.e., had been working with circles and would prefer squares). This indicates a bias in favor of familiarity for this question.

Table V

SUMMARY OF QUESTIONNAIRE RESULTS. SHOWING MEAN AND STANDARD DEV (5-POINT LIKERT SCALE, 1 - WORST, 5 - BEST).

Question	Rating
beauty	3.8 ± 0.9
helpfulness of visualizations	3.4 ± 1.1
helpfulness color coding	4.1 ± 0.7
helpfulness fill-level coding	3.5 ± 1.2
helpfulness reissue interaction	4.1 ± 1.3
expected uptake layer 1	3.6 ± 0.9
expected uptake layer 2	3.6 ± 1.1
Choice	Count
prefer layer 1	5 of 20
prefer layer 2	15 of 20
prefer circles	11 of 20
prefer squares	9 of 20

When asked for comments for improvement, 19 participants commented on the overall user interface, and 11 participants commented on the visualization. Comments for the overall user interface included questions like “why is search re-executed and search results are not cached?” and “how do I close the extension?”, and are not further investigated here. Suggestions for improvement of the QueryCrumbs can be categorized into comments on “visual encoding”, “interactions”, and “alternative suggestions”.

For *visual encoding*, one user suggested a different color coding (remove gray as color), usage of gradients to make it more beautiful, or adding additional information to the marks (either the first letter of the query or showing the percentage value instead of the fill-level). Two participants would like to see the marks labeled (with the query terms), and two participants commented that there is no need for improvement (“thumbs up”). In terms of *interactions*, one participant suggested to add the possibility to delete queries from the history. Another participant would prefer to treat the query history as list in which queries are not automatically deleted when reissuing from a previous query. One participant suggested an *alternative* representation as a drop down list (similar to the browser page history).

Summing up, users considered the QueryCrumbs helpful and well-designed. Further we found high indication for potential uptake with preferences for layer 2.

E. Discussion

In the evaluation we distinguished between *task success* (successfully performing the interaction) and *understanding* (correctly interpreting the results). Results show that they are indeed not necessarily related. E.g., in task T-A2, users had only 50% task success on average for navigating back to a previous query, but had understood the result of the changes in the visualization to a larger extend (65% understanding rate). This was due to the fact that although some users did not choose the correct previous query they interpreted the highlighting and change of the current query correctly.

Similarly, for task T-A3 (issue a query from previous) the task success was 100%, but the understanding rate was low (57.5%). This means, some users did not understand the interaction concept (cf. figure 4) when using it the first time. We assume that the low understanding rate due to the fact that users did not expect queries to vanish and also did not observe it. This rate could be improved by decreasing the speed of transition between visualization states.

Thus, hypothesis **H1** can partly be confirmed. The visualization is usable and understandable without instructions (similarity coding, navigate back, issue from previous), but some users had problems navigating back to the requested query and interpreting the visualization state when a new query was issued from a previous one.

The similarity coding was understandable in both layers (hypothesis **H2**), and we found no influence of the form of the mark on accuracy or speed. For **H2** we expected a difference for the two different marks, because reading the percentage similarity from the fill-level in a circle and square requires interpretation of two different visual features which are known to have different acuities [38]. The missing difference might be explained by the fact that users were only required to estimate the correct bin (of size 25%) which was feasible with both angle and area perception.

In the questionnaire users rated the helpfulness and beauty of the QueryCrumbs high and in general stated that they would like to use it in the future (hypothesis **H3**). Most of the users (75%) would prefer to use layer 2 after having gained experience with both layers. The majority of users decided to work with the QueryCrumbs in Task T-C, in which users were free to either do the task with or without the QueryCrumbs. This is also an indication that users expect a benefit in usage and points towards future uptake. There was one participant who requested an improvement towards query history management, in this case the possibility to delete queries from the history. All other users seem to perceive the QueryCrumbs as a search support tool (as intended) and do not think of it as a search history management tool.

While the results are promising, they present an estimate on expected real-world uptake. Providing clear usage statistics in realistic scenarios is subject to future work.

VII. CONCLUSION AND FUTURE WORK

We proposed QueryCrumbs, a simple-to-understand visualization for accessing, altering, and resubmitting previously issued queries. We applied a multi-layered interface approach to the design of the visualization. The formative user study confirmed that both layers were understandable and usable without instructions, and pointed towards directions for improvements, e.g., making the transitions in the visualization slower and thus, better perceptible.

Our preliminary study indicated that static labeling is more hindering than helpful. Because of this, we removed the labels for all (but the current query) for the evaluation.

However, the formal evaluation gives rise to a contrary view, as some users requested to have all labels constantly visible. In the current, horizontal layout of the marks, this would imply a potential overlap of labels for longer queries [39]. Thus, we will investigate the implications of a vertical layout for QueryCrumbs in future work.

The visualization is limited by the space in user interface and the number of base colors of the color scheme (11 colors in the evaluation) influencing the scalability as the number of queries grow. A search session contains 4 queries on average [30], while 67% of the sessions contain 1 or 2 queries, and 33% of the sessions contain 3 or more queries [11]. Even with the limited space for 11 marks (as in our evaluation) QueryCrumbs capture at least 2 search sessions on average. While more marks can be added, we estimate 2 search sessions as lower bound for a useful query navigation support and as a good trade-off between usefulness and support for limited screen-size. QueryCrumbs have been integrated as a visual history tool into a browser plugin for contextualized access to cultural heritage content³ [40].

In the controlled setting of the evaluation, the *layer transition* is done manually. In the prototype, the transition is proposed by the visualization after a number of interactions and needs to be confirmed by the user. This number is currently heuristically chosen to be 50. Improvement of the automatic transition between layers is subject of future work. When a user has successfully interacted with the visualization a specific number of times, we intent to notify her about the existence of the next layer.

REFERENCES

- [1] R. Capra, M. Pinney, and Perez-Quinones, "Refinding is not finding again," Computer Science, Virginia Tech., Tech. Rep. TR-05-10, 2005.
- [2] D. Morris, M. Ringel Morris, and G. Venolia, "Searchbar: A search-centric web history for task resumption and information re-finding," in *Proc. CHI*. ACM, 2008, pp. 1207–1216.
- [3] J. Teevan, "The re:search engine: Simultaneous support for finding and re-finding," in *Proc. UIST*. ACM, 2007, pp. 23–32.
- [4] M. A. Hearst, *Search User Interfaces*, 1st ed. New York, NY, USA: Cambridge University Press, 2009.
- [5] M. J. Bates, "The Design of Browsing and Berrypicking Techniques for the Online Search Interface," *Online Review*, vol. 13, no. 5, pp. 407–424, 1989.
- [6] G. Marchionini and R. White, "Find what you need, understand what you find," *Int. J. Hum. Comput. Interaction*, vol. 23, no. 3, pp. 205–237, 2007.
- [7] B. Shneiderman, D. Byrd, and W. B. Croft, "Clarifying search: a user-interface framework for text searches," *D-lib magazine*, vol. 3, no. 1, pp. 18–20, 1997.

³Parts of this work was developed within the European Union FP7-project EEXCESS under grant agreement number 600601

- [8] M. Levene, *An Introduction to Search Engines and Web Navigation*, 2nd ed. Wiley, 9 2010, no. 978-0-470-52684-2.
- [9] B. Shneiderman, "Promoting universal usability with multi-layer interface design," in *Proc. Conf. Universal Usability*. ACM, 2003, pp. 1–8.
- [10] A. Broder, "A taxonomy of web search," *SIGIR Forum*, vol. 36, no. 2, pp. 3–10, Sep. 2002.
- [11] B. J. Jansen, A. Spink, and J. Pedersen, "A temporal comparison of altavista web searching: Research articles," *J. Am. Soc. Inf. Sci. Technol.*, vol. 56, no. 6, pp. 559–570, Apr. 2005.
- [12] B. J. Jansen, A. Spink, C. Blakely, and S. Koshman, "Defining a session on web search engines: Research articles," *J. Am. Soc. Inf. Sci. Technol.*, vol. 58, no. 6, pp. 862–871, Apr. 2007.
- [13] L. Tauscher and S. Greenberg, "Revisitation patterns in world wide web navigation," in *Proc. CHI*, 1997, pp. 399–406.
- [14] M. Mayer, "Web history tools and revisitation support: A survey of existing approaches and directions," *Found. Trends Hum.-Comput. Interact.*, vol. 2, no. 3, pp. 173–278, 2009.
- [15] D. Sondheim, G. Rockwell, M. Ilovan, M. Radzikowska, and S. Ruecker, "Interfacing the Collection," *Scholarly and Research Communication*, vol. 3, no. 1, 2012.
- [16] A. Wexelblat and P. Maes, "Footprints: History-rich tools for information foraging," in *Proc. CHI*. ACM, 1999, pp. 270–277.
- [17] A. Komlodi, G. Marchionini, and D. Soergel, "Search history support for finding and using information: User interface design recommendations from a user study," *Inf. Process. Manage.*, vol. 43, no. 1, pp. 10–29, Jan. 2007.
- [18] A. Komlodi and D. Soergel, "Search histories for user support in user interfaces," *J. Am. Soc. Inf. Sci.*, vol. 57, pp. 803–807, 2006.
- [19] S. K. Card, G. G. Robertson, and J. D. Mackinlay, "The information visualizer, an information workspace," in *Proc. CHI*. New York, NY, USA: ACM, 1991, pp. 181–186.
- [20] R. R. Hightower, L. T. Ring, J. I. Helfman, B. B. Bederson, and J. D. Hollan, "Graphical multiscale web histories: A study of padprints," in *Proc. HYPERTEXT*. ACM, 1998, pp. 58–65.
- [21] T. Deng and L. Feng, "A survey on information re-finding techniques," *International Journal of Web Information Systems*, vol. 7, pp. 313–332, 2011.
- [22] H. Weinreich, H. Obendorf, E. Herder, and M. Mayer, "Off the beaten tracks: Exploring three aspects of web navigation," in *Proc. WWW*, ser. WWW '06. New York, NY, USA: ACM, 2006, pp. 133–142.
- [23] N. Jhaveri and K.-J. Räihä, "The advantages of a cross-session web workspace," in *Proc. CHI Extended Abstracts*. ACM, 2005, pp. 1949–1952.
- [24] M. Dontcheva, S. M. Drucker, G. Wade, D. Salesin, and M. F. Cohen, "Summarizing personal web browsing sessions," in *Proc. UIST*. ACM, 2006, pp. 115–124.
- [25] K. Bharat, "Searchpad: Explicit capture of search context to support web search," *Comput. Netw.*, vol. 33, no. 1-6, pp. 493–501, Jun. 2000.
- [26] J. Hailpern, N. Jitkoff, A. Warr, K. Karahalios, R. Sesek, and N. Shkrob, "Youpivot: Improving recall with contextual search," in *Proc. CHI*. ACM, 2011, pp. 1521–1530.
- [27] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala, "Graphical histories for visualization: Supporting analysis, communication, and evaluation," *IEEE Trans. Visualization & Comp. Graphics*, vol. 14, no. 6, pp. 1189–1196, Nov 2008.
- [28] T. V. Do and R. A. Ruddle, "The design of a visual history tool to help users refine information within a website," in *Proc. ECIR*. Springer, 2012, pp. 459–462.
- [29] J.-Y. Jiang, Y.-Y. Ke, P.-Y. Chien, and P.-J. Cheng, "Learning user reformulation behavior for query auto-completion," in *Proc. SIGIR*. ACM, 2014, pp. 445–454.
- [30] C. Eickhoff, J. Teevan, R. White, and S. Dumais, "Lessons from the journey: A query log analysis of within-session learning," in *Proc. WSDM*. ACM, 2014, pp. 223–232.
- [31] I. Herman, G. Melancon, and M. S. Marshall, "Graph visualization and navigation in information visualization: A survey," *IEEE Trans. Visualization & Comp. Graphics*, vol. 6, no. 1, pp. 24–43, 2000.
- [32] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *Proc. SIGIR*. ACM, 2005, pp. 154–161.
- [33] P. Qvarfordt, G. Golovchinsky, T. Dunnigan, and E. Agapie, "Looking ahead: Query preview in exploratory search," in *Proc. SIGIR*. ACM, 2013, pp. 243–252.
- [34] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining. (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- [35] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Trans. Visualization & Comp. Graphics*, 2011.
- [36] C. A. Brewer, G. W. Hatchard, and M. A. Harrower, "Colorbrewer in print: A catalog of color schemes for maps," *Cartography and Geographic Information Science*, vol. 30, no. 1, pp. 5–32, 2003.
- [37] J. Schlöterer, C. Seifert, and M. Granitzer, "Web-based just-in-time retrieval for cultural content," in *Proc. Workshop on Personalized Access to Cultural Heritage (PATCH)*, 2 2014.
- [38] C. Ware, *Information Visualization: Perception for Design*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.
- [39] J.-D. Fekete and C. Plaisant, "Excentric labeling: Dynamic neighborhood labeling for data visualization," in *Proc. CHI*. ACM, 1999, pp. 512–519.
- [40] C. Seifert, W. Bailer, T. Orgel, L. Gantner, R. Kern, H. Ziak, A. Petit, J. Schlöterer, S. Zwicklbauer, and M. Granitzer, "Ubiquitous access to digital cultural heritage," *J. Comput. Cult. Herit.*, vol. 10, no. 1, pp. 4:1–4:27, 4 2017.