



QueryCrumbs search query history visualization – Usability, transparency and long-term usage



Jörg Schlötterer^{a,*}, Christin Seifert^b, Christopher Satchell^a, Michael Granitzer^a

^a University of Passau, ITZ/IH, Innstraße 43, Passau 94032, Germany

^b University of Twente, Zilverling 4098, P.O. Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Keywords:

Information retrieval
Queryhistory visualization
Information refinding
Search history
Search experts

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. 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 and expert users. The visualization is evaluated with novice users in a formative user study, with experts in a think aloud test and its usage in a long-term study with software logging. The formative evaluation 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. The evaluation with experts showed that expert users can gain valuable insights into the back-end search engine by identifying specific patterns in the visualization. In a long-term usage study, we observed an uptake of the visualization, indicating that users deem QueryCrumbs beneficial for their search interactions.

1. Introduction

A common phenomenon in Web search is that users re-access Web resources that have been found in the past. Accessing previously found information is different from information seeking, e.g., by being more targeted and more directed involving recognition and recall activities [1]. While active strategies (i.e., explicit storage of the information) would support information refinding, 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. The difficulty of recalling where and how information on the Web was accessed is known as the “Lost in Hyperspace syndrome” [3].

While strictly speaking the “Lost in Hyperspace syndrome” refers to the navigation of hypermedia only, an analysis of human information seeking models shows a similar behavior in the context of Web search. 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 steps have to be taken and multiple query reformulations are necessary before the information need is fully

satisfied. The demand to include a search history for supporting the query reformulation stage has been explicitly stated (e.g., [4,7]). A search history also supports information re-finding for interrupted search sessions, which have been found to occur in 40% of all information seeking tasks in the study of Sellen et al. [8]. Resuming a search from a previous query relying on human memory has been shown to be only accurate in 72% of the time [9]. In the information retrieval community, automatic query refinement is an active research topic (e.g. [10]). This indicates that, in general, queries posed by users are underspecified and need to be refined iteratively.

We propose QueryCrumbs, a simple-to-understand, compact visualization for accessing, altering, and resubmitting previously issued queries. The concept is similar to bread crumbing interfaces as navigational aid for web sites [11]. Fig. 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 and different notions of query similarity are encoded in the mark's visual attributes. We introduce three different measures for query similarity to capture the general relationship between queries and corresponding mappings to the marks' visual attributes. 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

* Corresponding author.

E-mail addresses: joerg.schloetterer@uni-passau.de (J. Schlötterer), c.seifert@utwente.nl (C. Seifert), michael.granitzer@uni-passau.de (M. Granitzer).

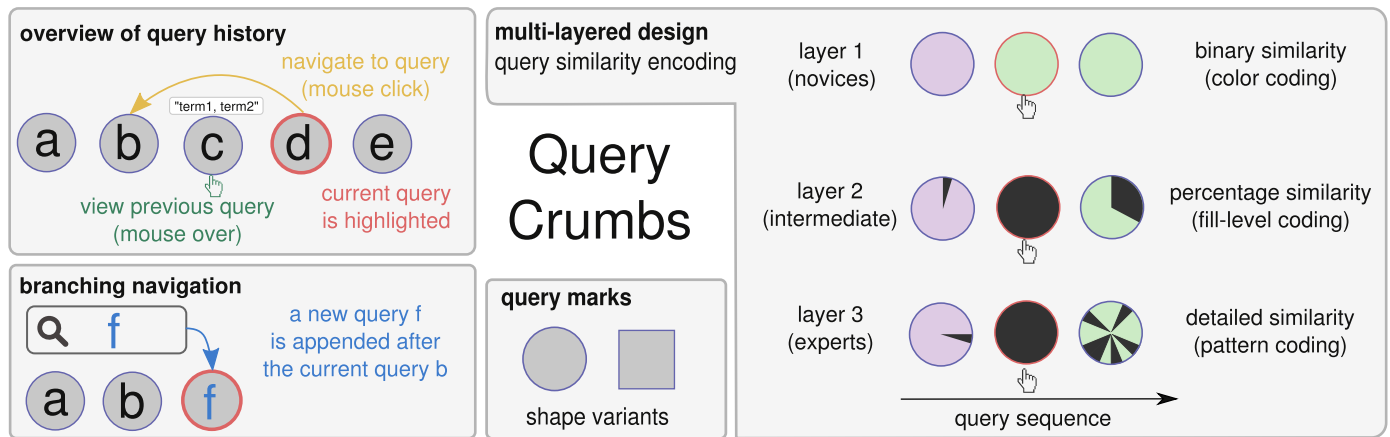


Fig. 1. QueryCrumbs visualization concept. Previous queries are shown. Navigating back to a previous query reissues the query (top left). Issuing a new query from a previous one removes previously subsequent queries showing only the current path of interest (bottom left). Two different shape variants (bottom middle). Query similarity is based on the similarity of the search result lists and can be encoded with different levels of detail (right).

representation, we pursue a layered interface design approach [12] introducing different notions of similarity in each layer. We evaluate the visualization and interaction design in a formative user study with novices. Additionally, we performed a think aloud test with experts in information retrieval to investigate which conclusion can be drawn about the search engine when using the visualization with the advanced similarity encoding. Finally, we evaluate the actual usage of QueryCrumbs outside the lab environment with software logging in a long-term study. Concretely, the contribution of this paper is as follows:

- We introduce a human querying model as the conceptual basis for search history visualizations.
- We propose QueryCrumbs, a search engine agnostic, compact and interactive visualization supporting overview and navigation of the query history
- We account for universal usability by applying the multi-layered user interface design method to the design of the visualization.
- We show, that QueryCrumbs is usable by search lay persons without instructions, search experts can gain valuable insights into search engine internals (transparency) and demonstrate an uptake outside the lab environment.

This paper extends the study in [13]¹ and builds upon the underlying human querying model and basic visualization concepts in previous work [14]². We unify those two papers into a joint representation for self-containedness, extend the discussion of the multi-layered approach and present an evaluation “in the wild”. As other tools, search history visualizations need to be both, useful and usable. In own previous work, a formative user study showed that QueryCrumbs was deemed usable and useful for search lay persons [14]. Further own previous work showed – again in a temporally constrained lab study – that encoding of additional information in QueryCrumbs enabled search experts to gain understanding of the search backend internals [13]. These results encouraged an out-of-lab implementation of QueryCrumbs: Proving usability and usefulness in a lab environment does not tell, whether people are actually using the visualization outside the lab and if so, in which way. To close this gap, we present an

out-of-lab long-term study with a widely used search backend (Google Scholar) in this paper. This crucial extension complements our previous work (proving usability and usefulness) with insights to the actual usage of QueryCrumbs.

The remainder of the paper is organized as follows: After discussing related work, we describe the human querying model in Section 3 and from that derive the conceptual idea for the visualization in Section 4. Then, the multi-layered approach to visualization and interaction design is explained in detail in Section 5. The formative user evaluation described in Section 6 assesses the usability for the two layers designed for first-time and intermediate users. In a study with experts we then assess which conclusions can be drawn about the underlying search engine with the help of the visualization (cf. Section 7). We evaluate the actual usage of the visualization in Section 8 and conclude the paper with a discussion of design decisions and their consequences, limitations and an outlook on future work.

2. 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 refinding behavior and related tools are presented.

2.1. Human querying behavior

Web query log analyses provide statistical data about human querying behavior. Broder [15] derived a basic categorization of web queries from analyzing Altavista logs and survey data distinguishing transactional, navigational, and informational queries. Approximately 50% of queries are informational queries, with which users seek a certain piece of information. The author observed that especially informational queries range from a very broad description to a very narrow description of the information need. A similar observation was made for mobile search [16]. The authors classify queries on a continuum ranging from very specific to very general and 44% of the queries to be imprecise or general. 24% of the queries were classified as difficult because some information to construct a good query was missing, e.g., the name of an actor when searching for movies with this actor.

A detailed analysis on search sessions in AltaVista query logs is provided in Jansen et al. [17]. In this data set 52% of users modified their queries. 48% of search sessions were single-query sessions, 32% of the sessions contained three or more queries. In another study on the

¹ © 2018 IEEE. Reprinted, with permission, from J. Schlötterer, C. Seifert and M. Granitzer: QueryCrumbs for Experts: A Compact Visual Query Support System to Facilitate Insights into Search Engine Internals. 22nd International Conference Information Visualization (IV).

² © 2017 IEEE. Reprinted, with permission, from C. Seifert, J. Schlötterer and M. Granitzer: QueryCrumbs: A Compact Visualization for Navigating the Search Query History. 21st International Conference Information Visualization (IV)

same data set 37% of all queries were found to be query modifications of various types [18]. A taxonomy of query reformulation strategies is given in [19], including stemming and addition of words. The authors built a classifier to automatically detect reformulations and conclude that most of the reformulations can be accurately detected (90% accuracy). They note that some modifications are nearly impossible to detect automatically (e.g., the rephrasing of the query "how to calculate nutritional values" to "weight watchers calculator").

These statistics indicate that human querying is an iterative process in which query refinement is a common step, a result that is reflected in models of information seeking behavior reviewed in the next section. The results from the discussed log analyses are reflected in the human querying model we present as basis for the QueryCrumbs.

2.1.1. Search experts

In terms of query behavior, search expertise has typically been investigated along the following dimensions [20]: query attributes (choice of search terms, query length and syntax) and search strategies and tactics. Aula et al. [21] report the use of widely different definitions of what constitutes an expert, ranging from more than five hours of browsing a week to at least three years of extensive professional experience. For log-based studies, the use of advanced query syntax (such as quoted queries) has been used to identify advanced searchers [22]. Bates has formulated a set of potentially useful search tactics [23], which have later been refined and extended by Smith [24]. We follow the notion of related work in defining experts as being capable to use advanced query operators and applying a variety of different search tactics. In particular, their search tasks are usually more complex, sense-making tasks and therefore, an explorer-type behavior [25] with longer sessions, containing more than the average two queries [26] is expected. This increased complexity is reflected in QueryCrumbs by a multi-layer approach, providing more details to advanced users.

2.2. Human search models and search user interfaces

Multiple models for the human information seeking process have been proposed. The model of Shneiderman et al. consists of the four-stages *formulation* (generating the query), *action* (starting the search), *review* of results, and *reformulation* [7]. Similar models were later proposed by Sutcliffe and Ennis [27], and Marchionini and White [6]. All these models describe an iterative process and include the need of query reformulation and potential backtracking.

In the "berry picking" model of information seeking [5], the information need is not static, but changes with the resources found with each (reformulated) query. 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.

Mulhem & Nigay [28] applied Normans' seven stages of action model [29] to information retrieval tasks. Specifically, they identify two user-system distances spanning the gulf of execution. The first distance is the input semantic distance, which describes the challenge of identifying the semantic description of the information need (e.g., the right concepts for the specific search engine). The second distance is the input articularity distance, which describes the challenge of identifying the physical description of the search query (e.g., the right keywords and search operators). Both distances describe gaps between the conceptual model of the user and the system that cannot be bridged in general. Reasons include the dynamic nature of information sources, concept drift in systems and users [30], and that in some usage scenarios the search engine is a black-box from the view of the search user interface. These two distances point towards the necessity for query adaptation and rephrasing.

The analysis of the models shows that query reformulation, adaptation, and backtracking need to be supported in a search history visualization, which is reflected in the QueryCrumbs interaction design.

2.3. 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., [4,7]). A commercial example is Google's Wonderwheel, a visual tool for interactively finding related queries [31]. A query is represented as a node, clicking expands the node and shows related queries. Wonderwheel allows arbitrary branching from nodes and leads to a complex graph structure. It supports navigation in the query space, users can go back to previous queries and get new queries suggested. Wonderwheel is designed to focus on exploration of the information space (leading to the complex graph structure), while QueryCrumbs focuses on exploitation of the information space.

Segura and Barbosa present tools not specifically tailored towards search history visualizations, but general user interaction logs [32,33]. Several history visualizations focus on identifying entities and concepts and the relations among them in either the search [34] or browsing [35–37] history. In contrast, QueryCrumbs do not account for any semantics of a query, but define query similarity solely by the similarity of retrieved results. Carrasco et al. showed that animated browsing history visualizations can help users to better reflect on their browsing habits [38]. TrackThink [39] aim to track the thought process on Web Search, whereas QueryCrumbs accounts for this thought process only implicitly through the underlying human querying model.

Komlodi et al. present design guidelines and examples for search history visualization based on a study with librarians [40,41]. 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 [42], and therefore has richer interactions and is much more complex. Conceptually similar to our work are bread crumbing interfaces [11] introduced as navigational aid for web sites. In bread crumbing interfaces the navigation history is presented in a compact manner, but similarity of visited sites is not shown.

The notion of flow has been adapted to user interface design [43]. One aspect to support flow is to make the current state in the user interface transparent, alleviating the problem of very limited human short term memory. In the context of this work, this can be translated to presenting the current state of information search in context to previous searches.

2.4. Information refinding

While also relying on history mechanisms, information refinding differs from information seeking [1]. Information refinding tasks can be categorized into short-term (retrieving just visited information), mid-term and long-term (refinding information after months or years) tasks [44]. Refinding behavior was also observed, when an information seeking task is interrupted [2] and is not well supported by standard Web browsing interfaces [45]. 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]. Similar findings have been made in a different study, where 66% of users do nothing to save search results, but re-assess the page via search and 54% of participants reported to use this method at least once a week [46]. A study on query reissuing revealed that 28% of the time queries are misremembered when being asked for it one hour later [9].

Tools supporting an active strategy for information refinding are for example Session Highlights [47] and a Firefox plugin for storing web page summaries [48]. While these tools require an explicit user interaction to store the information, the SearchBar [2] and SearchPad [49] assume a passive user behavior for information storage. Both, SearchBar and SearchPad are centered around search queries, store additional documents and context and require a complex management.

Refinding tools assuming passive user behavior have also been proposed for other application areas, such as history of visualizations [50], or information refinding within a Web page [51]. SearchBar and SearchPad 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 refinding strategies for users that pursue a passive information keeping behavior.

3. 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 [10]. 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 Fig. 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. Fig. 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 capturing all queries and their relations 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.

We illustrate this model by an example. Imagine the user task “prepare something for a children's birthday party tomorrow”. The derived information need might be “find an easy to prepare recipe for cookies”. Query *a* could be “baking cookies”. A generalization of this query would be “cookies” (query *b*), a specialization “baking cookies gluten-free” (query *c*). A modification with the same search intent could

lead to the query “easy cookie recipe” (query *d*) or subsequently to the query “gluten-free baking” (query *e*). We note that the query modifications cannot necessarily be classified exactly by only knowing the query string. The new query “baking muffins” might for instance indicate a session break or a query modification in the session with the same search intent of “preparing something for a children's birthday party”. However we argue that for the purpose of query history visualization an exact classification is not necessary and therefore we kept the human querying model as simple as possible.

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 or the search goal, but captures the querying and query modification process. This simplification allows to approach the visualization of the human search process and is used as a basis for the concept of the QueryCrumbs history visualization.

4. 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 5 users tasks:

- **Overview:** Get an overall overview of the query history, i.e., the sequence of queries.
- **Navigation:** Navigate back to previous queries, thus be able to 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.
- **Qualitative comparison:** Compare the search result ranking across queries. Investigate how the result set changed qualitatively, i.e., identify the location of result set changes.

Fig. 1 shows the concept of the visualization and interaction design. In the previous section we introduced the human querying model as a general graph. For the introduced tasks it is not necessary to show the graph, thus we present only the user's navigation path and unroll any cycles (i.e., display each graph node as often as it has been visited). This choice is further discussed in Section 9. 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. In a first design we constantly displayed the query terms for all queries. A preliminary study showed that this much text is (i) more hindering than helpful for users 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. The navigation interaction corresponds to Shneiderman et al.'s guideline for supporting the query reformulation stage: allow users to review, alter, and resubmit queries [7].

4.1. Layered approach

Intended for the use with general search engines and thus casual users, the visualization should be understandable without instructions by novices and first-time users. To this extend, we apply a multi-layer interface approach [12] to the visualization design. The visualization has three layers depicted in Fig. 3. Layer 1 is designed for the tasks “overview”, “navigation” and “simple comparison”, and therefore introduces all interactions. Layer 2 and 3 add the more complex notions of similarity, and are designed for the tasks “quantitative comparison” and “qualitative comparison”, respectively. Although it might seem that

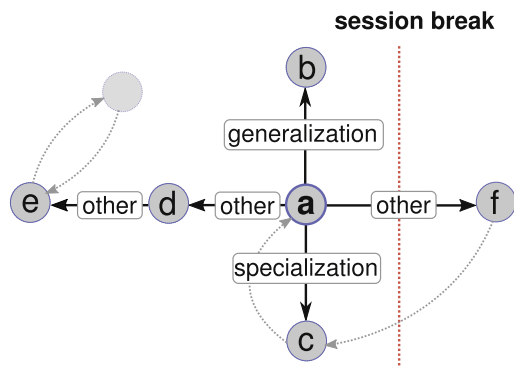


Fig. 2. Human querying model. Query modifications with the same search intent include specification and generalization. A session break occurs when the search intent changes.

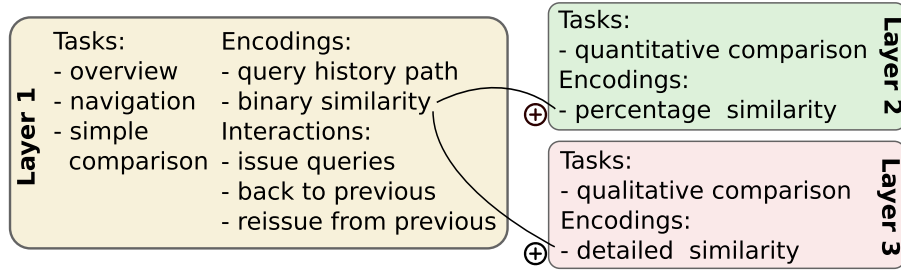


Fig. 3. Multi-layer visualization concept showing tasks, encoded data and interactions for all layers. Layer 2 and 3 extend the similarity notion of layer 1.

layer 2 and 3 only add minor information, we found in preliminary studies that the different similarity notions are hard to understand for users, and therefore we transferred them to additional layers.

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.

4.2. Measures for query similarity

The three comparison tasks introduced at the beginning of this section 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. In the following we introduce three different measures for query similarity capturing different levels of detail.

Typically, search engines return a ranked list of results for a query k . Let this ranked list be denoted by

$$R_k = [r_k^1, \dots, r_k^i, \dots, r_k^n] \quad (1)$$

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 [52,53], the similarity calculation is based on the top τ items, yielding the ranked list R_k^τ . The ranked lists can be directly used to assess the **qualitative comparison**, i.e., comparing which elements in two result lists are similar when viewed side-by-side.

Two queries can be compared pairwise based on their result list and the overlapping elements can be identified. With this similarity, it can be judged if a result that is present in the list R_i is also present at any position in the list R_j . 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 [54] on the two result sets (as opposed to using the ranked lists for the detailed similarity).

$$sim_r = \frac{|L_i^\tau \cap L_j^\tau|}{|L_i^\tau \cup L_j^\tau|} \in [0, 1] \quad (2)$$

sim_r can be expressed as a percentage to which we further refer to as **percentage similarity**. This similarity 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:

$$s_r = \begin{cases} 1, & \text{if } sim_r \geq \theta \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

We further refer to s_r as **binary similarity**. This similarity corresponds to the user task **simple comparison**.

Fig. 3 shows an overview of the presented data, similarity notions

and user interactions available in each layer.

5. QueryCrumbs visualization

The concept of the visualization described in the previous section is implemented in D3.js [55] and released³ under the MIT license.

5.1. Visualization and interaction design

The basic design of the visualization incorporates a mark for each query. Query similarity is encoded in the mark’s visual attributes and position is used to show the query sequence (time). Fig. 4 shows an example of the QueryCrumbs visualization for all three layers with either circles or squares as visual marks for a single query.

The mark for a query is either a circle or a square with fixed size. The currently selected query is outlined with a red border. In layer 1 (see Fig. 4, left column) the simple similarity s_r from Eq. (3) is encoded by color. Similar queries have the same color. We used a color map for qualitative data from ColorBrewer [56]. 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 chose to avoid such a complexity by choosing the color of the most recent, similar query instead. This coloring scheme tends to (i) color the new query with the color of the current session if it belongs to it, and (ii) visually shows if a same query or session was issued in the past (with a different session in between).

In layer 2 (see Fig. 4, center column), the percentage similarity from Eq. (2) is additionally 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. In the user evaluation we also addressed the question with which form (circles or squares) the similarity can be more accurately interpreted by users.

Layer 3 (see Fig. 4, right column) is designed for experts to assess the differences in two search result rankings in more detail. The query mark is divided into τ equally-sized sub-marks, one for each element in the result list, similar to Dense Pixel Displays [57], in which a single data item corresponds to one pixel. The sequence of sub-marks encodes the rank in the search result list and corresponds to the Western reading direction for squared marks and to a clock-wise reading for circles. Queries are compared pairwise, the hovered query (i.e., the query under the mouse pointer) is compared to all other queries (and itself). If a result from the hovered query is present in the result list of another query, the corresponding sub-marks in the hovered and the other query are colored dark gray. In a first version of QueryCrumbs the sub-marks of the current query were only colored dark if they reappeared in another result list, different from the hovered one. However, preliminary user studies showed that this was confusing to interpret for users. Users did not understand why results that are currently displayed in the accompanying search result list are not marked in the query mark.

³ <http://bit.ly/1PFva5O> features the source code and an installable demo

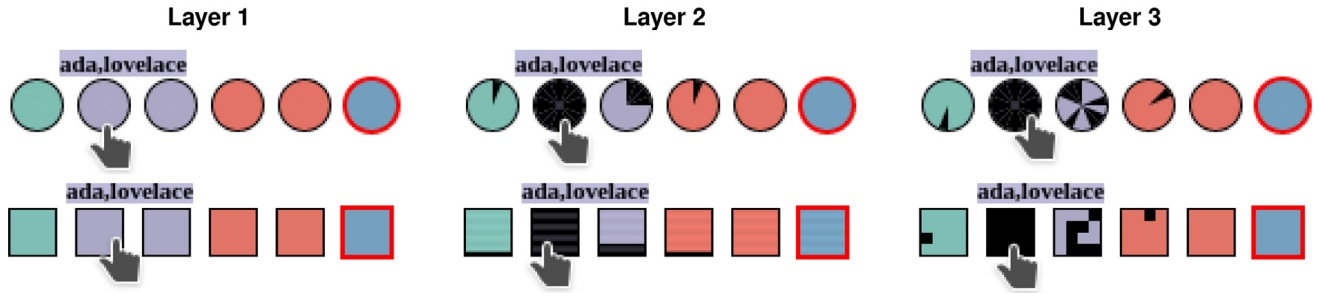


Fig. 4. QueryCrumbs layers for two different visual marks for queries “ada” “ada lovelace”, “ada byron”, “ada language”, “ada programming”, and “alan turing”. From left to right: Layer 1 for novices or first-time users, layer 2 for intermediate users, layer 3 for expert users. Current query is highlighted in red. [Best viewed in color]

Therefore, we decided to compare the hovered query also to itself, which colors the sub-marks for all results in the list dark gray.

5.2. Navigating the history

The QueryCrumbs visualization has two simple interactions for navigating the history. Mouse over allows to access basic information about a previous query. In layer 1 this information is the query string, in layer 2 and 3 additionally, the similarity to other queries is visualized. An example is shown in Fig. 4. 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 Fig. 5a.

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. Fig. 5b 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”.

6. Evaluation with novices

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. We posed the following hypotheses:

H1: Layer 1 can be understood and used successfully **without** instruction. This comprises the following visual encodings and interactions (cmp. Fig. 3):

- “issue query” (perform interaction, understanding of change in visualization)

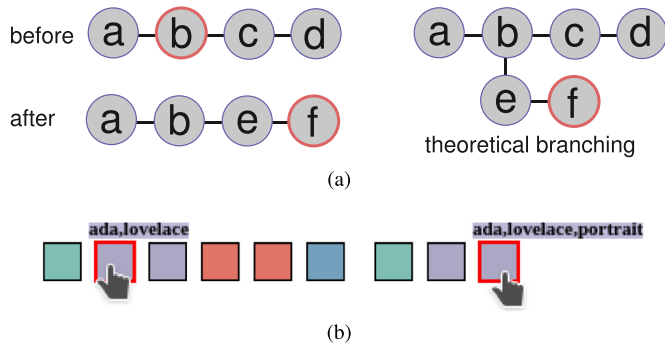


Fig. 5. Interactions. (a) Concept. Initial history, query “b” is selected (top left). Theoretical query tree after two new queries “e” and “f” (top right). Visualized trellis of the tree (bottom left). (b) Example. QueryCrumbs before and after issuing a query from a previous one. [Best viewed in color]

- “back to previous”-navigation (perform interactions, understanding of change in visualization)
- “issue from previous” (understanding of change in visualization: subsequent queries removed, new appended)
- “similarity encoding by color” (understanding)

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.

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 [58]. Interpretation in the human visual system relies on angle perception in the circle, and on area perception in the square.

6.1. Design

We used a between subjects design with the independent variable *form*, i.e., the type of query marks. The independent variable has two levels, it can either be a square or a circle as shown in Fig. 4. Dependent variables are *completion time* (in seconds), *task success* (binary) and *understanding* (binary). *Completion time* was measured automatically in the test user interface accounting for network latencies. *Task success* measured the correctness of the performed interaction (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 then judged by the evaluator.







In the questionnaire we also asked for perceived beauty, perceived helpfulness, expected uptake, and which layer participants would prefer (preferred layer). Preferred layer has either value 1 (for layer 1, simple similarity coding) or value 2 (for layer 2, percentage similarity coding). All other variables were assessed using a five-point Likert scale, with “1” coding the worst value and “5” encoding the best value (e.g., 1 - “not helpful at all”, 5 - “very helpful”).

6.2. Participants

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

Table 1

Task overview (measured variables: S - success, U - understanding, T - time, I - number and type of interactions). Grayed variables were measured, but are not the focus in the evaluation. Tasks in italic font are used to prepare the visualization.

ID	Task Description	Measures			
		S	U	T	I
Trial 1 - LAYER 1					
T-A1	issue queries				
		fx4.eps		fx4.eps	
T-A2	back to previous	✓	✓	✓	
T-A3	issue from previous	✓	✓	✓	
T-A4	issue queries				
		fx4.eps		fx4.eps	
T-A5	estimate binary similarity	✓	✓		
Trial 2 - LAYER 2					
T-B1	estimate percentage similarity	✓	✓		
T-B2	issue queries				
		fx4.eps		fx4.eps	
T-B3	estimate percentage similarity	✓	✓	✓	
T-B4	estimate percentage similarity	✓	✓	✓	
Trial 3 - OPTIONAL USAGE OF QUERY CRUMBS					
T-C	writing a blog entry	✓			✓

6.3. Tasks

Table 1 gives an overview of the evaluation 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 Fig. 4, left). With task set B layer 2 with the fill-level encoding the percentage similarity (see Fig. 4, center) was evaluated. Task T-C is designed to assess potential uptake of the visualization. T-C is a creative task, asking users to search for related material on a blog post they are writing. For this task users were not given explicit instructions on whether to use the QueryCrumbs visualization. They were only told to solve the task with the evaluation user interface, and were free to choose the visualization as an additional tool. We intentionally scheduled this task at the end of the evaluation, where users already had experienced what they can do with the visualization. For Task T-C we counted how many users used the QueryCrumbs, and which interactions (I) they performed with 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 QueryCrumbs. Task T-A3 also 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, selecting one of the values 0%, 25%, 50%, 75%, 100% for all query pairs.

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.

6.4. Test material

For the evaluation we used a browser extension that provides a sidebar alongside each Web page [59]. This extension accesses the Europeana collection⁴, the European aggregator for digital museum objects, and was modified to collect the evaluation measures.

Fig. 6 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 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 11 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.

6.5. Procedure

The evaluation comprised three trials. The first trial concerned layer 1 of the visualization (simple similarity and interactions) and was performed with task set A. In the second trial layer 2 (percentage similarity) with task set B was used. The third trial consisted of task T-C.

Before the first trial, participants obtained some general instructions on how to handle the evaluation interface. This contained the introduction of the start and stop task button and the instruction to start the task in the interface only after having read and understood the task instruction. For each participant the query history was set empty at the beginning, i.e., the QueryCrumbs were not visible at the beginning. 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 trial, participants received a short introduction (one written paragraph) about the percentage similarity coding (layer 2). For each task in trial 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 trial participants had a short break and were told that from now on the completion time was not measured anymore. For the third trial participants received only the task instructions. At the end of the third trial participants filled out the post-study questionnaire.

Some tasks contained explicit questions (e.g., “Which of the search queries you issued are similar to each other?”) 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.

⁴ <http://europeana.eu> - last accessed 2019

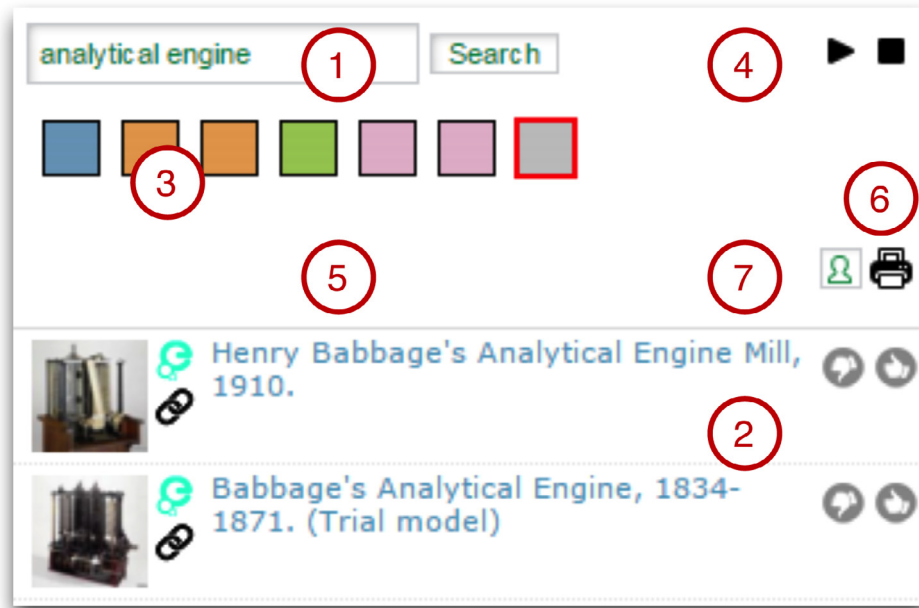


Fig. 6. Evaluation user interface (result list cropped).

6.6. Results

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

6.6.1. Measured performance for Layer 1

Table 2 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.

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

Table 2

Results for layer 1 aggregated over all users. Showing mean and standard deviation for completion time (for task T-A1 and T-A2 there is one missing value). "n.a." indicates measure is not applicable for this task.

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.

Fig. 5 was understood by the majority, but not by all users.

If users successfully issued a previous query it took them 10 sec on average (Task T-A2, depending on task success). For a successful re-issuing 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.

6.6.2. Measured Performance for Layer 2

Table 3 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). This means the group using circles performed all tasks as

Table 3

Results for layer 2 aggregated over all users. Showing mean and standard deviation for completion time (2 missing values for task T-B3, one for task T-B4). "n.a." indicates measure is not applicable for this task.

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 4
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%)

fast as the group using squares, on average.

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 influence on the completion time.

6.6.3. Usage in creative task (T-C)

In tasks T-C users were free to choose whether or not to use the QueryCrumbs visualization. Table 4 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 use QueryCrumbs for 1/3 of their queries.

6.6.4. Questionnaire results

Table 5 summarizes the quantitative values from the questionnaire. Generally users rated the QueryCrumbs rather high in all categories, i.e., above the theoretical average of 2.5 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.

When asked for comments for improvement, 19 participants

Table 5
Summary of questionnaire results. Showing mean and standard deviation (values from 5-point Likert scale, 1 - worst value, 5 - best value).

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

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 no queries are automatically deleted when reissuing from a previous query. One participant suggested an *alternative* representation of the query history as a drop down list (similar to the browser page history).

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

6.7. 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 still had understood the result of the changes in the visualization (65% understanding rate). We would expect users that have understood the results, but not performed the task correctly to become more accurate when performing the same task again (not part of the evaluation). Similarly, for task T-A3 (issue a query from previous) the task success was 100%, but the understanding rate was lower (57.5%). In terms of Norman’s seven stages of action model, for the interaction “back to previous” more users were lost in the gulf of execution, and for the interaction “issue from previous” more users were lost in the gulf of evaluation. This indicates areas for improvement, e.g., the speed of the transition could be decreased when a new query is issued from a previous one.

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. We would have expected an influence of the variable *form* in layer 2 (fill-level coding), but the missing difference might be because we did not ask for the absolute percentage values. 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 support tool while searching (as intended) and do not think of it as a search history management tool.

7. Evaluation with experts

The visualization was designed to show different levels of similarity of search result lists. Layer 3 that conveys the detailed similarity was designed for experts. The assumption is that the QueryCrumbs visualization can give information retrieval or search engine experts deeper insights into the querying process, therefore increasing transparency of the search engine. The goal of the expert user study was to qualify potential insights experts can gain while interacting with the visualization during a search session. We further wanted to understand their reasoning for the insights, and which patterns in the visualization indicate certain findings.

7.1. Tasks

Participants were given the search interface with the QueryCrumbs visualization and were asked to perform the following task first: “Input some queries and investigate the visualization. Please tell us, what you observe, what conclusions you draw and why you draw these conclusions.” In the second task, participants received a set of prepared queries to input. In an optional third task participants could again use their own queries, if they expected more findings when trying to query again. The tasks took approximately 30 minutes to perform.

7.2. Procedure and participants

We recruited eight experts with experience in information retrieval or search engines. Participants were either employed by the University or by a large German library. One participant was a graduate student, one a Post-Doc and six were Phd students. All participants were male, with an average age of 30, ranging from 25 to 37.

Participants were given an explanation of the visualization, with specific focus on layer 3 and had time to get familiar with the interactions. We chose the squared marks because we found no influence of a specific mark on performance in the previous evaluation (cf. Section 6). After that the tasks and the precompiled query sets were introduced and participants were asked to perform the tasks while thinking aloud. A screencast (with audio) was taken during the experiment. The study ended with a questionnaire asking for potential usefulness and application areas of the visualization.

7.3. Test material

We used the same search back-end as in the previous evaluation (cf. Section 6.4) and slightly modified the user interface (similarity calculations based on the top-25 search results, buttons for task recording removed). We prepared 7 sets of queries that we assumed to lead to interesting insights. The sets of queries are listed in Table 6.

In set 1, both queries lead to the same result list, which means the search engine can be assumed to use stemming as a text pre-processing method. The queries in set 2 provide no overlap between the result lists, which indicates that synonyms might not be used by the search engine.

Table 6
Predefined query sets and QueryCrumbs for set 7.

1	car	cars	
2	car	automobile	
3	August Macke	Franz Marc	Paul Klee
4	August Macke	Macke	
5	haystack series	"haystack series"	
6	loom	loom weaving	
7	ada lovelace	"ada lovelace"	ada and lovelace
	ada or lovelace	ada AND lovelace	ada OR lovelace




Table 7

Overview of search engine properties identified by experts using QueryCrumbs. Indicating the number of experts identifying the feature (column 2), and the predefined set or example queries used.

Category	#	Comment
<i>PRE-PROCESSING</i>		
Stemming	8	Set 1
Lemmatization	3	child and children
Stopword Removal	3	to be or not to be and to or not to
Case-sensitivity	1	Insensitivity assumed, not tested
<i>ADVANCED NATURAL LANGUAGE PROCESSING</i>		
Translation	2	car and voiture
Synonyms/Abbreviations	8	Set 2, ww2 and world war 2
Concept matching	1	Set 3
Named entity recognition	1	Set 4
<i>QUERY LANGUAGE</i>		
Phrase query support	8	Set 5, Set 7
Boolean query support	8	Set 7
Boolean operator syntax	7	Set 7
Default Boolean operator	4	Set 7, Set 6
<i>RANKING</i>		
Reproducibility	2	1 user tested a query 8 times
Term Sequence	2	star wars and wars star
Query Term Proximity	1	Set 5

The connection between the three painters (the artist group “Blue Rider”) in set 3 is not made explicit in the search content. In Set 4 the second query leads to results that is a superset of the first query, with the specific property that the common results are at the beginning of the result list. Sets 5 and 6 have one single result in common between the two queries, in set 5 this is because the phrase query only returns 1 result. Set 7 tests the implemented query language of the search engine, e.g., whether a list of terms is implicitly connected via the AND operator (which is true), and whether “and” and “or” in lower-case are interpreted as part of the query language or as query terms (the latter is true).

7.4. Results and discussion

Table 7 provides an overview of the search engine properties that experts have identified during the evaluation. All of the features expected when compiling the predefined result sets were identified, some by all expert users (e.g., stemming, synonyms, boolean query support). Most conclusion could be made about the query language: support for phrase queries, the syntax of the phrase queries and the default operator (AND) were identified. Additional insights were made about the reproducibility, i.e., the randomness of the ranking (2 users) and whether the proximity of terms in the documents has any influence on the final ranking (1 user). Own query sets were used to identify whether the query terms sequence has any influence (2 users) and whether the search engine uses translation of query terms (1 user). Most users (7 of 8) performed the optional third tasks to test for further insights.

In the questionnaire, participants were asked about the usefulness, potential improvements and (further) application areas. Summarizing the comments, participants indicated that the QueryCrumbs interface is well suited for comparing search result lists. However, they remarked that result list comparison is not inherent to the search task and performed only rarely. Accordingly, suggested further application areas comprised applications, where list comparison is a primary task, e.g., comparing friend lists in social networks. Regarding improvements, participants suggested to provide more details when comparing only two elements, e.g., by enlarging the elements or by a different visualization, such as a Venn-chart.

Summing up, the evaluation showed that experts can acquire insights about search engine internals by using QueryCrumbs, making the search process more transparent.

8. Usage evaluation

In the usage evaluation, we wanted to assess whether QueryCrumbs are actually used outside a lab study and if so, which features are commonly used.

8.1. Test material and participants

We implemented a browser extension for the Google Chrome browser to integrate with Google Scholar⁵ (a search engine for scholarly literature). A screenshot of the evaluation prototype is depicted in Fig. 7. Google Scholar was chosen in order to target a larger audience as with the cultural data backend used in the previous prototypes. While the largest audience might have been addressed by integrating QueryCrumbs into Google's general purpose search engine we still opted for Google Scholar, as it has a consistent interface. With Google's general purpose search engine, we faced frequent changes in the interface, both in terms of (CSS-)layout and identifiers. These changes would require frequent updates of our prototype and could invalidate evaluation results as the time between a change in the interface and the corresponding update of the prototype could have a negative influence. Further, with Google's general purpose search engine, several special cases need to be considered, as depending on the query, different verticals and/or info-boxes are displayed. Therefore we opted for Google Scholar, since the interface is simply the query box and a plain result list without frequent changes.

As visible in Fig. 7, the implementation included the first two layers, i.e. color and percentage similarity and used circle shapes for query marks. The similarity calculations were based on the first top-ten results, since that is the default returned by the search engine on the first page. The query similarity threshold θ was set to 0.1. That means, two queries are considered similar in terms of binary similarity (i.e. have same color), if they share a single result. Further, we followed a suggestion from the evaluation with novices and added the option to delete query marks with a double-click. During the evaluation we tracked the following interactions:

- **Draw:** Rendering of the visualization. Occurs every time a new query is issued.
- **Hover:** Shows the query terms and percentage similarity to all queries (comparison).
- **Click:** Navigating back to a previous query and re-issue that query.
- **Double-click:** Removes the query from the history.

Those interactions were logged over a time frame of 2.5 months. The interaction logs were sent to a lightweight node.js⁶ server instance, storing them in a MySQL⁷ database.

21 users volunteered to participate in the long-term evaluation, with the majority of them being computer science students or researchers. One user indicated, to not have used Google Scholar before. A globally unique identifier (GUID) was created for each user and interactions were logged anonymized with the user's corresponding GUID.

8.2. Results and discussion

In this section, we present and discuss the evaluation results, first for the general usage of QueryCrumbs and then for the utilized features individually.

8.2.1. General usage

Out of the 21 users who downloaded and installed the extension, 17

actually used QueryCrumbs at least once. To obtain statistics about the general usage, we calculated the usage duration as the amount of distinct days, the visualization was utilized at least once. This way, we get a more accurate measure for the usage than by simply taking the timespan between the first and last usage (which may contain several days without usage). The results are summarized in Fig. 8.

53% of the participants used the visualization for no longer than one day, 35% for less than a week 12% for more than a week. The high number of participants using the visualization for only a week or less is partially explained by the selection of the search engine. Google Scholar's primary goal is the indexing of full text or metadata of scholarly literature. Even though the majority of participants were students, they don't use Google Scholar on a daily basis and we explicitly asked participants not to deviate from their regular behavior. Students typically use Google Scholar more frequently when writing their thesis or a seminar paper, but less frequently during a regular semester. Therefore, for this part of the participants group, using QueryCrumbs for at most seven distinct days is explained simply by them not using Google Scholar more often. Likely, participants who used the visualization at most for one day, visited Google Scholar only to try out the visualization on that day. They wouldn't have visited otherwise and did not visit it afterwards. We assume, that the two users, who used QueryCrumbs for more than seven distinct days (28 and 35 days) were the researchers in the participant group, who also use Google Scholar on a regular basis.

Summing up, we see an uptake of QueryCrumbs on a scholarly search engine by participants, who use this search engine on a regular basis.

8.2.2. Feature usage

In the second part of the evaluation, we investigated which features of the visualization are used and how often. Table 8 provides an overview of the feature usage.

As can be seen in the table, adding a crumb and comparing queries was performed by all participants, while less participants utilized the navigation and deletion feature. Accordingly, the number of adding and comparing queries is way higher than the number of navigation and deletion actions performed. This behavior is expected, as a new crumb is added automatically every time a query is issued. Also, when performing a comparison, users quite likely take a look at several queries and compare them against each other. Another reason for the number of comparisons being higher than the number of added queries is that a comparison event can be triggered unintentionally. For example when moving the mouse pointer from the result list to the query input box, it may be moved over a query crumb. Hence a comparison event is triggered, while the user did not actually compare queries. On the other hand, when the user compares queries simply by the color coding (binary similarity), we cannot detect this event via software logging and therefore have no information about how many queries were compared in that way.

The usage numbers for the navigation feature are naturally far lower than adding and comparing crumbs, as the navigation is used for backtracking in time. Typically, users do not repeat each and every query several times, but navigate back only to selected queries (if at all). Surprisingly, while the delete feature was used by less users than the navigation feature, they used it more often. Looking closer at this difference, in Fig. 9, we see that the median for the delete feature is even at zero. Whereas the statistics of the other features look as expected, with a few outliers on the feature usage (those are the participants using the scholarly search engine on a regular basis). A potential explanation for not many participants deleting crumbs is that, even though this feature was mentioned in the brief description of the browser extension upon install, participants may not have read that description or forgot about it. From the visualization itself, a double-click is not an obvious interaction to delete a crumb and therefore, participants might not have recognized that possibility. To counter that fact, a tooltip may be added in the future, if the user hovers over a

⁵ <https://scholar.google.com> - last accessed 2019

⁶ <https://nodejs.org> - last accessed 2019

⁷ <https://www.mysql.com> - last accessed 2019

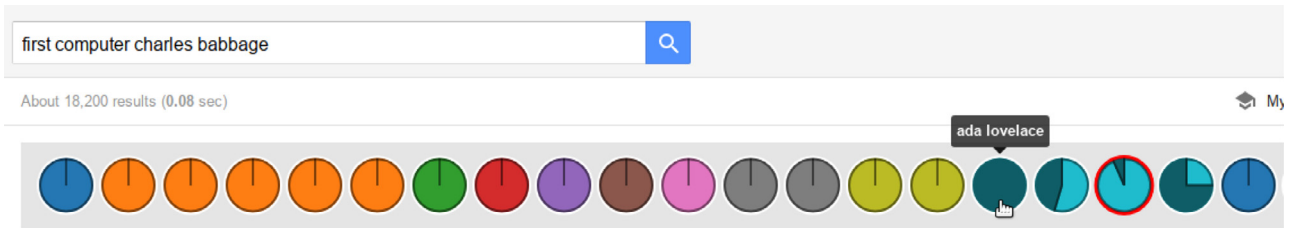


Fig. 7. Screenshot of the evaluation prototype for usage evaluation (cropped). The current query is highlighted by the red circle (backtracked) and query under the mouse pointer compared to all others.

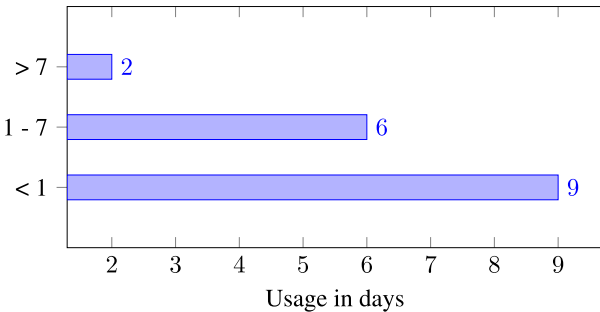


Fig. 8. Results for general usage. Shows how long users that installed the extension used it for

Table 8

Results for feature usage. Shows each feature, how many users used it and how often on average and in total.

Feature	Usage [%]	Average	Total
Add crumb	100	75	2733
Compare crumbs	100	177	3010
Navigate back	58.82	4	66
Delete crumb	35.29	5	88

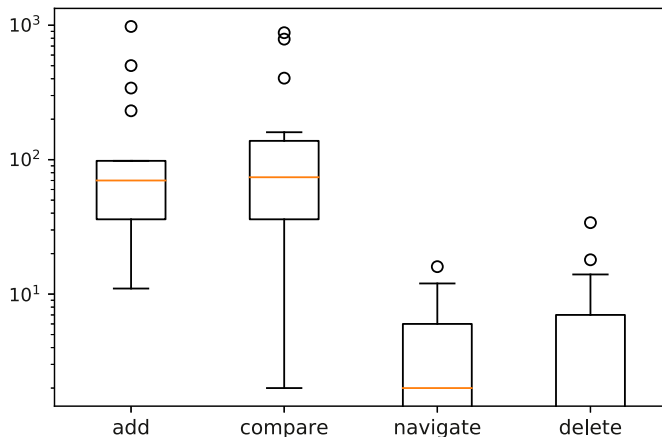


Fig. 9. Boxplots for detailed statistics of the individual features (log-scale).

crumb for a certain time, in order to inform about the delete feature from within the visualization.

Still, it is interesting to see that if participants were aware of the delete feature, they used it more often than the navigation feature. We assume that participants deleted queries without any relevant results in order to keep only relevant queries in their history. Removing irrelevant queries can be seen as a strategy between active (storing relevant results) and passive (not storing anything). That intermediate strategy requires less effort from the user than actively storing relevant results. By QueryCrumbs logging the query history automatically, potentially relevant results are stored even for passive users. Whether a query

returned no relevant results is easier to decide than whether a query returned (potentially - even if only in the future) relevant results, lowering the user's mental effort. The usage of this feature could also be used as an indicator for search engine evaluation. In search engine evaluation, typically clicks on a search result are seen as a positive signal. However, an abandoned query, i.e. a query where the user did not click on a result, does not need to be a negative signal. Li et al. [60] introduced "good abandonment", i.e. queries, where the user did not click on a result, but still could satisfy her information need. Such good abandonment occurs, when the search engine result page already satisfies the user's information need. The information need can be satisfied for example by answer boxes to factoid questions, or the result snippet might contain the required information. Hence, clicking a result is not necessary anymore. However, it is difficult to determine whether a user abandoned a query because she was satisfied or not. In such cases, the deletion feature could be used as an indicator for negative query abandonment.

Summing up, the visualization features were merely used as expected with the deletion feature having a higher usage than expected, indicating high value of that feature.

9. Discussion

We choose a *simplification of a human querying model* for the visualization that does not show the explicit branching, but rather visualizes the history in a linear fashion. Human querying models [2,6,7] indicate that backtracking occurs mostly when users arrive in a dead end, i.e. modifying query terms does not lead to relevant results anymore. A recent study on web search logs provides additional details on branching and backtracking behavior [61]. 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 interactions would result in a rather complex graph. Such graphs are hard to layout in a visually pleasing way and hard to navigate [62] and supporting small screens (e.g., mobile phones) would no longer be possible. Still, in the implementation of QueryCrumbs for the long-term study, we also store the backtracked branches without showing them. Storing the branches enables potential future extensions, such as for example visualizing backtracked branches on demand for individual queries.

The query history visualization is limited by the space given by the user interface influencing the *scalability* as the number of queries grow. A search session contains 4 queries on average [61], while 67% of the sessions contain 1 or 2 queries, and 33% of the sessions contain 3 or more queries [17]. Even with the limited space for 11 marks (as in our evaluations with first-time to expert users) 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. The implementation of QueryCrumbs for the long-

term study allows for 26 query marks on a full hd display (at least 6 search sessions on average). Again, as we store the complete query history, future extensions are easily possible. Those comprise for instance scrolling back in the history or a user-defined setting of the query mark size (which implicitly defines the amount of displayable query marks).

Questionnaire results from the expert evaluation indicate that while QueryCrumbs can increase the transparency of the search engine by issuing sensibly selected queries, result list comparison is not performed by search experts on a daily basis. Therefore search experts suggested to extend the QueryCrumbs visualization to other application areas, in which list comparison is a primary task, e.g. comparing friend lists in a social network. When comparing result lists, search experts suggested to provide more details for the comparison, e.g. additional visualizations such as a Venn-chart. Both, search experts and lay persons suggested to reflect details of the comparison also in the result list, e.g. by highlighting results that also occurred in other queries. However, the result list is not part of QueryCrumbs, but specific to a particular search engine (interface). Therefore, such an extension has to be implemented specific to the search engine, whereas the QueryCrumbs interface itself is search engine agnostic.

The differences we observed in the uptake of QueryCrumbs in the long-term study are mainly governed by the usage of the selected search engine. The small amount of usage (in days) by the majority of users does not reflect a low usage of QueryCrumbs, but a low usage of the selected search engine. However, for the long-term study, we were limited by implementation/extension capabilities of common general purpose search engines as discussed in Section 8.1. We opted for the best tradeoff between targeting a large audience and a consistent user interface experience. Still, we observed an uptake of QueryCrumbs for those users that used the selected search engine on a regular basis.

10. 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 and evaluated the layers intended for novices and intermediate users and experts. The formative user study confirmed that the first two layers were well understood and usable without instructions, and pointed towards which parts of the design could still be improved. A long-term deployment study with software-logging incorporating the first two layers of the visualization indicated actual uptake also outside the lab environment.

The expert evaluation revealed that using QueryCrumbs and sensibly selected query sets, experts can gain interesting insights in the behavior of the search engine, for instance about the applied pre-processing and the ranking of results. Those insights increase the transparency of search engine internals. We plan to integrate an automatic detection for a switch between layer two and three. When a user has successfully interacted with the visualization a specific number of times, we intent to notify her about the existence of the third layer. The users should still be in control of the choice of the layer, extending QueryCrumbs to an adaptable visualization [63].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Part of the work was developed within the East-Bavarian Centre of Internet Competence, Big and Open Data Analytics for Small and Medium-sized Enterprises (BODA), funded by the Bavarian Ministry of

Economic Affairs and Media, Energy and Technology. Part of the work was developed within the EEXCESS project funded by the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number 600601.

References

- [1] R. Capra, M. Pinney, Perez-Quinones, Refinding is Not Finding Again, Technical Report, Computer Science, Virginia Tech., 2005.
- [2] D. Morris, M. Ringel Morris, G. Venolia, Searchbar: a search-centric web history for task resumption and information re-finding, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08, ACM, New York, NY, USA, 2008, pp. 1207–1216, <https://doi.org/10.1145/1357054.1357242>.
- [3] J. Conklin, Hypertext: an introduction and survey, Computer 20 (9) (1987) 17–41, <https://doi.org/10.1109/MC.1987.1663693>.
- [4] M.A. Hearst, Search User Interfaces, 1st, Cambridge University Press, New York, NY, USA, 2009.
- [5] M.J. Bates, The design of browsing and berrypicking techniques for the online search interface, Online Rev. 13 (5) (1989) 407–424.
- [6] G. Marchionini, R. White, Find what you need, understand what you find, Int. J. Hum. Comput. Interact. 23 (3) (2007) 205–237, <https://doi.org/10.1080/10447310701702352>.
- [7] B. Shneiderman, D. Byrd, W.B. Croft, Clarifying search: a user-interface framework for text searches, D-lib Mag. 3 (1) (1997) 18–20.
- [8] A.J. Sellen, R. Murphy, K.L. Shaw, How knowledge workers use the web, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '02, ACM, New York, NY, USA, 2002, pp. 227–234, <https://doi.org/10.1145/503376.503418>.
- [9] J. Teevan, The re:search engine: simultaneous support for finding and re-finding, Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, UIST '07, ACM, New York, NY, USA, 2007, pp. 23–32, <https://doi.org/10.1145/1294211.1294217>.
- [10] J.-Y. Jiang, Y.-Y. Ke, P.-Y. Chien, P.-J. Cheng, Learning user reformulation behavior for query auto-completion, Proceedings of the 37th International ACM SIGIR Conference on Research, SIGIR '14, ACM, New York, NY, USA, 2014, pp. 445–454, <https://doi.org/10.1145/2600428.2609614>.
- [11] M. Levene, An Introduction to Search Engines and Web Navigation, 2nd ed., Wiley, 2010.
- [12] B. Shneiderman, Promoting universal usability with multi-layer interface design, Proceedings of the Conference on Universal Usability, CUU '03, ACM, New York, NY, USA, 2003, pp. 1–8, <https://doi.org/10.1145/957205.957206>.
- [13] J. Schlötterer, C. Seifert, M. Granitzer, Querycrumbs for experts: a compact visual query support system to facilitate insights into search engine internals, Proceedings of the 22nd International Conference Information Visualisation (IV), (2018), pp. 78–84, <https://doi.org/10.1109/IV.2018.00024>.
- [14] C. Seifert, J. Schlötterer, M. Granitzer, Querycrumbs: a compact visualization for navigating the search query history, Proceedings of the 21st International Conference Information Visualisation (IV), (2017), pp. 35–44, <https://doi.org/10.1109/IV.2017.23>.
- [15] A. Broder, A taxonomy of web search, SIGIR Forum 36 (2) (2002) 3–10, <https://doi.org/10.1145/792550.792552>.
- [16] J. Bae, V. Setlur, B. Watson, Graphtiles: a visual interface supporting browsing and imprecise mobile search, Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '15, ACM, New York, NY, USA, 2015, pp. 63–70, <https://doi.org/10.1145/2785830.2785872>.
- [17] B.J. Jansen, A. Spink, J. Pedersen, A temporal comparison of altavista web searching: Research articles, J. Am. Soc. Inf. Sci. Technol. 56 (6) (2005) 559–570, <https://doi.org/10.1002/asi.v56.6>.
- [18] B.J. Jansen, A. Spink, C. Blakely, S. Koshman, Defining a session on web search engines: research articles, J. Am. Soc. Inf. Sci. Technol. 58 (6) (2007) 862–871, <https://doi.org/10.1002/asi.v58.6>.
- [19] J. Huang, E.N. Efthimiadis, Analyzing and evaluating query reformulation strategies in web search logs, Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09, ACM, New York, NY, USA, 2009, pp. 77–86, <https://doi.org/10.1145/1645953.1645966>.
- [20] R.W. White, S.T. Dumais, J. Teevan, Characterizing the influence of domain expertise on web search behavior, Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM '09, ACM, New York, NY, USA, 2009, pp. 132–141, <https://doi.org/10.1145/1498759.1498819>.
- [21] A. Aula, R.M. Khan, Z. Guan, How does search behavior change as search becomes more difficult? Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10, ACM, New York, NY, USA, 2010, pp. 35–44, <https://doi.org/10.1145/1753326.1753333>.
- [22] R.W. White, D. Morris, Investigating the querying and browsing behavior of advanced search engine users, Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07, ACM, New York, NY, USA, 2007, pp. 255–262, <https://doi.org/10.1145/1277741.1277787>.
- [23] M.J. Bates, Information search tactics, J. Am. Soc. Inf. Sci. 30 (4) (1979) 205–214.
- [24] A.G. Smith, Internet search tactics, Online Inf. Rev. 36 (1) (2012) 7–20, <https://doi.org/10.1108/14684521211219481>.
- [25] R.W. White, S.M. Drucker, Investigating behavioral variability in web search, Proceedings of the 16th International Conference on World Wide Web, WWW '07,

- ACM, New York, NY, USA, 2007, pp. 21–30, <https://doi.org/10.1145/1242572.1242576>.
- [26] M. Kamvar, M. Kellar, R. Patel, Y. Xu, Computers and iphones and mobile phones, oh my!: a logs-based comparison of search users on different devices, Proceedings of the 18th International Conference on World Wide Web, WWW '09, ACM, New York, NY, USA, 2009, pp. 801–810, <https://doi.org/10.1145/1526709.1526817>.
- [27] A. Sutcliffe, M. Ennis, Towards a cognitive theory of information retrieval, *Interact. Comput.* 10 (3) (1998) 321–351, [https://doi.org/10.1016/S0953-5438\(98\)00013-7](https://doi.org/10.1016/S0953-5438(98)00013-7).
- [28] P. Mulhem, L. Nigay, Interactive information retrieval systems: from user centered interface design to software design, Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '96, ACM, New York, NY, USA, 1996, pp. 326–334, <https://doi.org/10.1145/243199.243280>.
- [29] D.A. Norman, *The Design of Everyday Things*, Basic Books, 2002.
- [30] J.a. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, *ACM Comput. Surv.* 46 (4) (2014) 44:1–44:37, <https://doi.org/10.1145/2523813>.
- [31] D. Sondheim, G. Rockwell, M. Ilovan, M. Radzikowska, S. Ruecker, Interfacing the Collection, *Sch. Res. Commun.* 3 (1) (2012) 60–78.
- [32] V.C.V.B. Segura, S.D.J. Barbosa, History viewer: displaying user interaction history in visual analytics applications, in: M. Kurosu (Ed.), *Human-Computer Interaction. Novel User Experiences*, Springer International Publishing, Cham, 2016, pp. 223–233.
- [33] V. Segura, S.D. Barbosa, Historyviewer: instrumenting a visual analytics application to support revisiting a session of interactive data analysis, Proceedings of the ACM on Human-Computer Interaction, 1 (2017), <https://doi.org/10.1145/3095813.11:1–11:18>.
- [34] L. Xu, Z.T. Fernando, X. Zhou, W. Nejdl, Logcanvas: visualizing search history using knowledge graphs, Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '18, ACM, New York, NY, USA, 2018, pp. 1289–1292, <https://doi.org/10.1145/3209978.3210169>.
- [35] D. Weidan, Personal web library: organizing and visualizing web browsing history, *Int. J. Web Inf. Syst.* 14 (2) (2018) 212–232, <https://doi.org/10.1108/IJWIS-09-2017-0065>.
- [36] P.H. Nguyen, K. Xu, A. Bardill, B. Salman, K. Herd, B.L.W. Wong, Sensemap: supporting browser-based online sensemaking through analytic provenance, Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST), (2016), pp. 91–100, <https://doi.org/10.1109/VAST.2016.7883515>.
- [37] I. ud Din, S. Khuroo, I. Ullah, A. Rauf, Semantic history: ontology-based modeling of users' web browsing behaviors for improved web page revisitation, in: R. Silhavy, P. Silhavy, Z. Prokopova (Eds.), *Intelligent Systems in Cybernetics and Automation Control Theory*, Springer International Publishing, Cham, 2019, pp. 204–215.
- [38] M. Carrasco, E. Koh, S. Malik, Pophistory: animated visualization of personal web browsing history, Proceedings of the CHI Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '17, ACM, New York, NY, USA, 2017, pp. 2429–2436, <https://doi.org/10.1145/3027063.3053259>.
- [39] K. Nagano, Y. Arakawa, K. Yasumoto, Trackthink: a tool for tracking a thought process on web search, Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers, UbiComp '17, ACM, New York, NY, USA, 2017, pp. 681–687, <https://doi.org/10.1145/3123024.3129267>.
- [40] A. Komlodi, G. Marchionini, D. Soergel, Search history support for finding and using information: User interface design recommendations from a user study, *Inf. Process. Manage.* 43 (1) (2007) 10–29, <https://doi.org/10.1016/j.future.2006.05.017>.
- [41] A. Komlodi, D. Soergel, Search histories for user support in user interfaces, *J. Am. Soc. Inf. Sci.* 57 (2006) 803–807, <https://doi.org/10.1002/asi.20297>.
- [42] S.K. Card, G.G. Robertson, J.D. Mackinlay, The information visualizer, an information workspace, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '91, ACM, New York, NY, USA, 1991, pp. 181–186, <https://doi.org/10.1145/108844.108874>.
- [43] B.B. Bederson, Interfaces for staying in the flow, *Ubiquity* 2004 (September) (2004), <https://doi.org/10.1145/1074068.1074069.1-1>.
- [44] T. Deng, L. Feng, A survey on information re-finding techniques, *Int. J. Web Inf. Syst.* 7 (2011) 313–332, <https://doi.org/10.1108/17440081111187538>.
- [45] H. Weinreich, H. Obendorf, E. Herder, M. Mayer, Off the beaten tracks: exploring three aspects of web navigation, Proceedings of the 15th International Conference on World Wide Web, WWW '06, ACM, New York, NY, USA, 2006, pp. 133–142, <https://doi.org/10.1145/1135777.1135802>.
- [46] H. Bruce, W. Jones, S. Dumais, Keeping and re-finding information on the web: What do people do and what do they need? *Proc. Am. Soc. Inf. Sci. Technol.* 41 (1) (2004) 129–137, <https://doi.org/10.1002/meet.1450410115>.
- [47] N. Jhaveri, K.-J. Räihä, The advantages of a cross-session web workspace, Proceedings of the CHI Extended Abstracts on Human Factors in Computing Systems, CHI EA '05, ACM, New York, NY, USA, 2005, pp. 1949–1952, <https://doi.org/10.1145/1056808.1057064>.
- [48] M. Dontcheva, S.M. Drucker, G. Wade, D. Salesin, M.F. Cohen, Summarizing personal web browsing sessions, Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology, UIST '06, ACM, New York, NY, USA, 2006, pp. 115–124, <https://doi.org/10.1145/1166253.1166273>.
- [49] K. Bharat, Searchpad: explicit capture of search context to support web search, *Comput. Netw.* 33 (1-6) (2000) 493–501, [https://doi.org/10.1016/S1389-1286\(00\)00047-5](https://doi.org/10.1016/S1389-1286(00)00047-5).
- [50] J. Heer, J. Mackinlay, C. Stolte, M. Agrawala, Graphical histories for visualization: Supporting analysis, communication, and evaluation, *IEEE Trans. Vis. Comput. Graph.* 14 (6) (2008) 1189–1196, <https://doi.org/10.1109/TVCG.2008.137>.
- [51] T.V. Do, R.A. Ruddle, The design of a visual history tool to help users refine information within a website, Proceedings of the 34th European Conference on Advances in Information Retrieval, ECIR'12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 459–462, https://doi.org/10.1007/978-3-642-28997-2_41.
- [52] T. Joachims, L. Granka, B. Pan, H. Hembrooke, G. Gay, Accurately interpreting clickthrough data as implicit feedback, Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, ACM, New York, NY, USA, 2005, pp. 154–161, <https://doi.org/10.1145/1076034.1076063>.
- [53] P. Qvarfordt, G. Golovchinsky, T. Dunnigan, E. Agapie, Looking ahead: query preview in exploratory search, Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, ACM, New York, NY, USA, 2013, pp. 243–252, <https://doi.org/10.1145/2484028.2484084>.
- [54] P.-N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, 1st ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.
- [55] M. Bostock, V. Ogievetsky, J. Heer, D3: data-driven documents, *IEEE Trans. Vis. Comp. Graph. (Proc. InfoVis)* 17 (12) (2011) 2301–2309.
- [56] C.A. Brewer, G.W. Hatchard, M.A. Harrower, Colorbrewer in print: a catalog of color schemes for maps, *Cartogr. Geogr. Inf. Sci.* 30 (1) (2003) 5–32, <https://doi.org/10.1559/152304003100010929>.
- [57] D. Keim, P. Bak, M. Schäfer, Dense pixel displays, in: L. LIU, M. ÖZSU (Eds.), *Encyclopedia of Database Systems*, Springer US, 2009, pp. 789–795, https://doi.org/10.1007/978-0-387-39940-9_1131.
- [58] C. Ware, *Information Visualization: Perception for Design*, 3, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2012.
- [59] J. Schlötterer, C. Seifert, M. Granitzer, Web-based just-in-time retrieval for cultural content, Proceedings of the 7th International ACM Workshop on Personalized Access to Cultural Heritage, PATCH, (2014).
- [60] J. Li, S. Huffman, A. Tokuda, Good abandonment in mobile and pc internet search, Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09, ACM, New York, NY, USA, 2009, pp. 43–50, <https://doi.org/10.1145/1571941.1571951>.
- [61] C. Eickhoff, J. Teevan, R. White, S. Dumais, Lessons from the journey: a query log analysis of within-session learning, Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM '14, ACM, New York, NY, USA, 2014, pp. 223–232, <https://doi.org/10.1145/2556195.2556217>.
- [62] I. Herman, G. Melancon, M.S. Marshall, Graph visualization and navigation in information visualization: a survey, *IEEE Trans. Vis. Comput. Graph.* 6 (1) (2000) 24–43, <https://doi.org/10.1109/2945.841119>.
- [63] B. Shneiderman, C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 5th, international ed., Addison-Wesley Publ. Co., Reading, MA, 2010.