

# QueryCrumbs for Experts: A Compact Visual Query Support System to Facilitate Insights into Search Engine Internals

Jörg Schlötterer  
University of Passau  
joerg.schloetterer@uni-passau.de

Christin Seifert  
University of Passau / University of Twente  
c.seifert@utwente.nl

Michael Granitzer  
University of Passau  
michael.granitzer@uni-passau.de

**Abstract**—Search experts use advanced query language and search tactics to formulate their queries. However, the effectiveness of those advanced techniques depends on the search engine internals. We propose QueryCrumbs for Experts, a compact visualization, which facilitates insights to the search engine internals and therefore allows the searcher to determine effective search strategies. Treating the search engine as a black box, QueryCrumbs can be seamlessly integrated into existing search interfaces, guiding the user’s exploration and assessment of results. QueryCrumbs for Experts visualize the recent search history alongside with a simple and also a qualitative comparison of the result sets, from which conclusions about the search engine internals can be drawn. The evaluation shows that, by identifying specific patterns in the visualization, expert users can gain valuable insights into search engine internals, empowering them to adapt their search accordingly.

**Keywords**-Information Retrieval; Query History Visualization; Information Re-finding; Search History; Search Experts;

## I. INTRODUCTION

The search behaviour of expert users is known to differ from less advanced users [1]. Experts use advanced query language, such as quoted queries or boolean operators and apply a variety of different search tactics [2]. However, the effectiveness of those advanced techniques strongly depends on the internals of the search engine used. For example, for the advanced query language to take effect, the correct syntax has to be used. Also, reformulating a keyword query by concatenating all terms with a boolean AND will not change the result set, if this concatenation is already the default operation of the search engine. Thus, the searcher does not gain any new results. Similarly, reformulating a query with synonyms has no effect, if the search engine can map the original keyword to those synonyms and returns the results obtained by the synonym query already in the original result set of the first query. In addition, some of the search engine’s internal capabilities can be query (or even term) specific, i.e., a synonym mapping may be available for a particular query, while no mapping may be available for another query.

In previous work, we presented QueryCrumbs, a compact, interactive visualization for accessing, altering and

re-submitting previously issued queries [3]. In this paper, we extend the QueryCrumbs approach with a qualitative comparison, allowing search experts to gain insights about search engine internals<sup>1</sup> The concept is depicted in Fig. 1 and a detailed explanation is provided in Sec. III. The sequence of queries in the history is displayed with a corresponding sequence of query marks in the visualization. Moving the mouse over one of the query marks in the history displays the mark’s corresponding query. Clicking a mark means navigating back and re-issuing the corresponding query. A color coding scheme provides the simple comparison, i.e., the marks of queries with similar result sets have the same color. Hovering over a query mark invokes the qualitative comparison, which indicates the presence (and position) of results from the hovered query in other queries by coloring corresponding sub-marks dark gray. QueryCrumbs treat the search engine as black box, i.e., the only required information are queries and results, not how they were obtained. This renders the visualization search engine agnostic and allows for easy integration in existing search interfaces.

In a think-aloud study with search experts, we investigated, which conclusions can be drawn about search engine internals when using the QueryCrumbs visualization. The results revealed that experts can indeed gain insights into the search engine’s internals and capabilities. Based on these insights, they can adapt their querying behaviour to pursue effective strategies and formulations. Concretely, our contributions are the following:

- We propose QueryCrumbs for Experts, a search engine agnostic, compact and interactive visualization, supporting overview and navigation of the query history and providing simple and qualitative comparison of result sets across the query history
- With the evaluation, we show, that search experts can gain insights into the search engine internals and which conclusions they can draw, using QueryCrumbs

The rest of this paper is organized as follows: After review-

<sup>1</sup>A short demo video snippet taken from the evaluation screencasts (audio removed) is available alongside with the evaluation prototype at <http://doi.org/10.5281/zenodo.1249456>

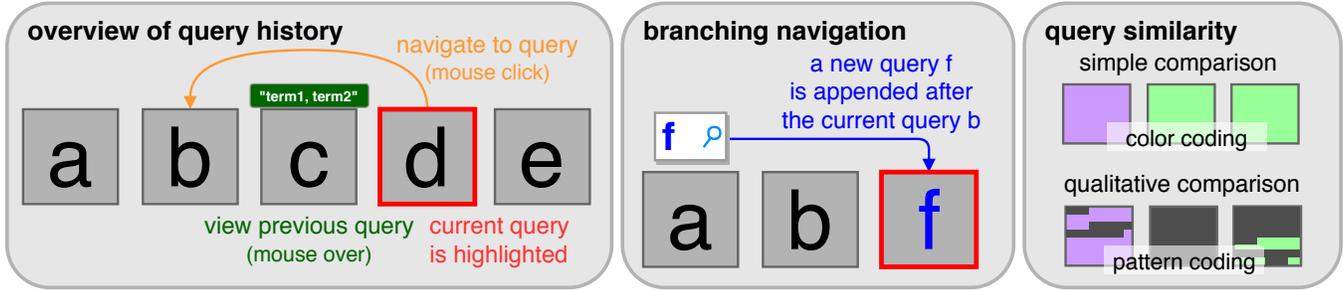


Figure 1. QueryCrumbs visualization concept. Left: A sequence of queries (a,b,c,d,e) is shown, the current query (d) is highlighted, while navigating back to a previous query (b) reissues the query. Hovering over a query (c) shows the query terms and the similarity to all queries. Center: Issuing a new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

ing related work in the next section, we present the concept of QueryCrumbs for Experts in Sec. III. Then, we present the evaluation with search experts in Sec. IV and discuss the results. We conclude the paper with a summary and an outlook on future work.

## II. RELATED WORK

In this section, we first provide background information on search expertise and search tactics and then present related search result and search history visualization approaches.

### A. Search Experts

In terms of query behaviour, search expertise has typically been investigated along the following two dimensions [4]: query attributes (choice of search terms, query length and syntax) and search strategies and tactics. Aula et al. [5] 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. Also, for log-based studies, the use of advanced query syntax (such as quoted queries) has been used to identify advanced searchers [1]. Bates has formulated a set of potentially useful search tactics [2], which have later been refined and extended by Smith [6]. 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, as in a library environment, the search tasks are usually more complex, sense-making tasks and therefore, an explorer-type behaviour [7] with longer sessions, containing more than the average two queries [8] is expected.

### B. Search Result and History Visualization

Beyond the standard result list, a variety of search interfaces has been developed to assist the user's exploration of the search results. For example, tile-based visualizations like TileBars [9] and HotMap [10] display the relative frequency of query terms in a retrieved document with the color

intensity of corresponding squares. Faceted search interfaces allow for organizing or filtering items throughout orthogonal categories, supporting users to find their way through large document collections and specifying their information need more precisely [11], [12]. A recent tool, called uRank [13], presents keyword-based summaries of the retrieved result set and allows to refine and reorganize the results based on selected keywords and their importance.

While Hearst points out the need for search history visualizations in general [14], Shneiderman et al. state the need for a search history to support the query re-formulation stage, and allow users to review, alter and re-submit previous searches [15]. Search history visualizations share many demands with browsing history visualizations [16], [17]. A commercial example for a search visualization is Google's Wonderwheel [18], a tool for interactively finding related queries in which a query is represented as a node in a graph. Similarly, the Footprints [19] tool exploits navigation paths of users to suggest potential paths through the information space. Web pages are represented as nodes in a graph representing most visited paths. Wonderwheel and Footprints visualize the complex information space and focus on exploration of the space generated by other users. QueryCrumbs in contrast focuses on exploitation of the user's own history. While WebComets [20] builds on the user's own history and visualizes web-page visits as glyphs, encoding topically coherent pages via the same color, the focus is on parallel browsing and dwell times. Komlodi et al. present design guidelines and examples for search history visualization based on a study with librarians [21], [22]. This work is similar to ours, in particular, since their target user group is the same (search experts). Their interface follows the information webspace concept [23], and therefore has richer interactions and is much more complex. QueryCrumbs provide a simple interface that combines search result and history visualization by comparing the search results across the history.

### III. QUERYCRUMBS FOR EXPERTS CONCEPT

In this section, we repeat the relevant parts of the original QueryCrumbs concept for the paper to be self-contained and present the according extensions towards QueryCrumbs for Experts. For a detailed discussion of the concept and its underlying human querying model, we refer the reader to [3]. Conceptually, QueryCrumbs for Experts visualizes the most recent path through the general querying graph, i.e., the user’s history of search queries, supporting the 4 user 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.
- **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.

Figure 1 shows the concept of the visualization and interaction design. 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. We chose to reissue the query instead of storing the results, since the latter would require a tighter coupling to the search engine: Means to display the stored results would be required, while in our implementation, QueryCrumbs do not require any knowledge of or interaction with the displayed search result list. Also, for queries far back in the history, the result set may have changed in the meantime.

We do not show the explicit branching, but rather visualize the query history in a linear fashion, unrolling any cycles. This means, if a user backtracks to a previous query and issues a new query from there, a new branch would be created. Instead of showing the explicit branching, we cut off the old branch and show only the current one. This decision on simplification is supported by a study on web search logs providing details on branching and backtracking behavior [24]. 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. This is also in line with the monitoring search tactic *check* [2], i.e., reviewing the original search intent and comparing it to the current search topic. Cutting the branch, the user backtracked from, means cutting the branch where the search intent has diverged.

#### A. Measures for Query Similarity

The two 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 different measures for query similarity capturing different levels of detail.

Typically, search engines return a ranked list of results for a query  $q_k$ . Let this ranked list be denoted by  $R_k = [r_k^1, \dots, r_k^i, \dots, r_k^n]$ , where  $r_k^i$  is the  $i$ -th result for query  $q_k$ . Because users of Web search engines only access the top items in the result list [25], [26], the similarity calculation is based on the top  $\tau$  items, yielding the ranked list  $R_k^\tau$ . The ranked lists can be directly used to perform the **qualitative comparison**, i.e., comparing which elements in two result lists are similar when viewed side-by-side. Here we are not only interested in whether a result  $r_k^i$  is contained in both lists, but also at which position, as its position might be different in each of the two lists.

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  $sim_{ij}$  of two queries  $q_i$  and  $q_j$  can then be calculated as the Jaccard coefficient [27] on the two result sets (as opposed to using the ranked lists for the detailed similarity).

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

$sim_{ij}$  expresses the percentage similarity between two queries. A binary indicator variable  $s_{ij}$  can be obtained by introducing a similarity threshold  $\theta \in [0, 1]$ , and is calculated as follows:

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

We further refer to  $s_{ij}$  as binary similarity. This similarity corresponds to the user task **simple comparison**.

#### B. Visualization Design

The basic design of the visualization incorporates a mark for each query. The currently selected query is outlined with a red border. Query similarity is encoded in the mark’s visual attributes and position is used to show the query sequence.

The binary similarity  $s_{ij}$  from Equation 2 is encoded by color (simple comparison). Similar queries have the same

color. 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).

For the qualitative comparison, the query mark is divided into  $\tau$  equally-sized sub-marks, one for each element in the result list. This is similar to Dense Pixel Displays [28], 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. 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 tests 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 highlighted in the query mark. Therefore, we decided to compare the hovered query also to itself, which colors the sub-marks for all results in the list dark gray. As an additional benefit, this also allows to easily determine, when less than  $\tau$  results are available (the query mark is not completely filled).

#### IV. EVALUATION

The visualization was designed to show different levels of similarity of search result lists. The assumption is that in particular the qualitative comparison can give search experts deeper insights into a search engine’s internals and querying process. With these insights, search experts can then adjust their query strategies to pursue potentially effective strategies. For example, rewriting a query with synonyms or concepts from a taxonomy does not yield additional results, if the search engine already expands the query with synonyms and maps the query terms to taxonomy concepts internally. Such a situation is easily identified via the qualitative comparison, when all sub-marks of two compared queries are colored dark (i.e., the result sets are the same). The goal of the expert user study was to qualify potential insights, experts can gain while interacting with QueryCrumbs during a search session. We further wanted to understand their reasoning for the insights and which patterns in the visualization indicate certain findings.

#### A. Participants and Test Material

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.

The evaluation setup comprised a search interface with an input field for query terms, the QueryCrumbs visualization and a search result list. We used the Europeana Search API<sup>2</sup> as search backend. Europeana is the European aggregator for cultural heritage, providing access to over 50 million artworks, artefacts, books, videos and sounds collected from over 3000 institutions across Europe. The similarity calculations were based on the 25 top-most search results ( $\tau = 25$ ). The query similarity threshold  $\theta$  was set to 0.1 for the binary similarity, which was determined as a good threshold for visually indicating similarity in preliminary experiments. A value of 0.1 for  $\theta$  may seem low, but we account this to the high diversity of items in Europeana. For providers with more domain specific content, we expect a higher threshold. Participants had the opportunity to reset the QueryCrumbs visualization, i.e., to delete the query history and every participant started with an empty history.

#### B. Procedure and Tasks

Participants were given an explanation of the visualization, with specific focus on the qualitative comparison and had time to get familiar with the interactions. After that, we introduced three tasks (see description below) and participants were asked to perform them while thinking aloud. The tasks took approximately 30 minutes to perform. 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. Participants were asked to perform the following tasks:

- **Task 1:** In this first task, participants were free to interact with the interface with queries of their own choice. That is, they could enter arbitrary queries and use QueryCrumbs to investigate the result sets.
- **Task 2:** In the second task, participants received a set of prepared queries to input. We prepared 7 sets of queries that we assumed to lead to interesting insights. The sets of queries are listed in table I.
- **Task 3:** In this optional third task participants could again use their own queries, if they expected more findings when trying to query again.

For all tasks, participants were asked to think aloud. In particular, they were encouraged to talk about what they observe, what conclusions they draw about the search engine (if any) and why they draw these conclusions. For the prepared query sets in **Task 2**, participants were asked to

<sup>2</sup><https://pro.europeana.eu/resources/apis/search> - accessed 22.05.2018

Table I  
PREDEFINED QUERY SETS FOR **TASK 2** AND CORRESPONDING FEATURES, ABOUT WHICH WE ASSUMED PARTICIPANTS TO GATHER INSIGHTS.

id	queries			feature
set 1	car	cars		stemming
set 2	car	automobile		synonyms
set 3	August Macke	Franz Marc	Paul Klee	concepts
set 4	August Macke	Macke		named entities
set 5	haystack series	"haystack series"		
set 6	loom	loom weaving		query syntax
set 7	ada lovelace	"ada lovelace"	ada and lovelace	
	ada or lovelace	ada AND lovelace	ada OR lovelace	

enter all queries of a particular set and investigate the result sets with QueryCrumbs. Then they were asked to proceed with the next set. Table I lists the prepared query sets for **Task 2** and the features, we assumed conclusions could be made about with the respective query set. However, we did not ask participants about specific features, but only asked, which conclusions they could draw about the search engine internals. 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. 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).

Figure 2 presents the corresponding visualization of QueryCrumbs for set 7. All queries in this set are similar,



Figure 2. QueryCrumbs Visualization for query set 7 from Table I.

thus all query marks have the same color. The mouse pointer is over the query mark for "ada and lovelace", hence this query is compared to all others and itself. The comparison with itself reveals, that less than 25 results have been retrieved, since the mark is not completely filled. The result sets of query 1 and 5 are the same, exhibiting the same patterns, when comparing the hovered query to them.

### C. Results

Table II provides an overview of the search engine features that experts have identified to be implemented (or not) during the evaluation. It is important to note, that we did

Table II  
OVERVIEW OF SEARCH ENGINE FEATURES IDENTIFIED BY EXPERTS USING QUERYCRUMBS. INDICATING THE NUMBER OF EXPERTS IDENTIFYING THE FEATURE (COLUMN 2), AND THE PREDEFINED SET OR EXAMPLE QUERIES USED.

Category	#users	Comment
<i>PRE-PROCESSING</i>		
Stemming	8	Set 1
Lemmatization	3	<i>child</i> and <i>children</i>
Stopword Removal	3	<i>to be or not to be</i> and <i>to or not to</i>
Case-sensitivity	1	Insensitivity assumed, not tested
<i>ADVANCED NATURAL LANGUAGE PROCESSING</i>		
Translation	2	<i>car</i> and <i>voiture</i>
Synonyms/Abbreviations	8	Set 2, <i>ww2</i> and <i>world war 2</i>
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	<i>star wars</i> and <i>wars star</i>
Query Term Proximity	1	Set 5

not ask participants about particular features, but about the conclusions they can draw about the search engine internals in general. Instead, we collected the features, which were found by the experts completely on their own during the evaluation and grouped them afterwards. 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). Most users (7 of 8) performed the optional third task to test for further insights. They used their own query sets, for example to identify whether the query term sequence has any influence (2 users) and whether the search engine uses translation of query terms (1 user).

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. They suggested further application areas, where list comparison is an inherent 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. Similarly, participants suggested to highlight overlapping results also in the result list. This means to highlight results in the currently displayed result list of the active query, whenever a comparison with a previous query is made by hovering and the results reappear. However, such a highlighting requires a tight integration of QueryCrumbs to the search interface. At the current state, QueryCrumbs require only little input (a query and identifiers of the corresponding result items - usually their URLs) and a hook to re-issue a query.

#### D. Discussion

For the evaluation, experts were provided with a set of predefined queries and asked to make their conclusions about the search engine explicit. We do not expect searchers to explicitly test for the search engine's capabilities in a real-life scenario. Rather, we expect search experts to naturally interact with the search interface, using QueryCrumbs as supportive tool to draw conclusions implicitly. For example, they might reformulate a query and by using QueryCrumbs quickly identify, that the result set has not changed. This means, the search engine had an internal mechanism to reformulate and/or expand the query, yielding the same results. Hence, no effort needs to be spent to investigate the result set again. Similarly, experts might use advanced query syntax in a format they are familiar with and recognize via QueryCrumbs whether the search engine supports that format or not. We envision the integration of QueryCrumbs into digital library interfaces as an additional supportive tool to guide expert users in the exploration and assessment of search results. The implementation is publicly available<sup>3</sup> and due to its search engine agnostic nature, QueryCrumbs are easily integrable to existing search interfaces.

<sup>3</sup><https://github.com/EEXCESS/QueryCrumbs>

## V. SUMMARY AND FUTURE WORK

In this paper, we presented QueryCrumbs for Experts, a compact, search engine agnostic visualization which facilitates insights into search engine internals. QueryCrumbs visualize the recent search history and provide a simple and a qualitative comparison of the retrieved results. Our 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. In real-world applications, we expect search experts to draw those conclusions implicitly and adapt their search strategies accordingly.

In future work, we plan to follow the suggestion of search experts in providing more details for the comparison of two elements. Further we plan to provide branches, users backtracked from on demand. These are currently cut and we display only the most recent path through the query history. The investigation of how search experts adapt their search strategies implicitly, based on log data, is also subject of future work.

## ACKNOWLEDGMENT

The presented 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.

## REFERENCES

- [1] R. W. White and D. Morris, "Investigating the querying and browsing behavior of advanced search engine users," in *Proceedings of the 30th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '07. ACM, 2007, pp. 255–262.
- [2] M. J. Bates, "Information search tactics," *Journal of the American Society for information Science*, vol. 30, no. 4, pp. 205–214, 1979.
- [3] C. Seifert, J. Schlötterer, and M. Granitzer, "Querycrumbs: A compact visualization for navigating the search query history," in *21st International Conference Information Visualisation (IV)*, July 2017, pp. 35–44.
- [4] R. W. White, S. T. Dumais, and J. Teevan, "Characterizing the influence of domain expertise on web search behavior," in *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, ser. WSDM '09. New York, NY, USA: ACM, 2009, pp. 132–141.
- [5] A. Aula, R. M. Khan, and Z. Guan, "How does search behavior change as search becomes more difficult?" in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 35–44.
- [6] A. G. Smith, "Internet search tactics," *Online Information Review*, vol. 36, no. 1, pp. 7–20, 2012.

- [7] R. W. White and S. M. Drucker, "Investigating behavioral variability in web search," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 21–30.
- [8] M. Kamvar, M. Kellar, R. Patel, and Y. Xu, "Computers and iphones and mobile phones, oh my!: A logs-based comparison of search users on different devices," in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 801–810.
- [9] M. A. Hearst, "Tilebars: visualization of term distribution information in full text information access," in *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 59–66.
- [10] O. Hoerber and X. D. Yang, "A comparative user study of web search interfaces: Hotmap, concept highlighter, and google," in *International Conference on Web Intelligence (WI '06)*. IEEE, 2006, pp. 866–874.
- [11] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst, "Faceted metadata for image search and browsing," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '03. New York, NY, USA: ACM, 2003, pp. 401–408.
- [12] C. Seifert, J. Jurgovsky, and M. Granitzer, "Facetscape: A visualization for exploring the search space," in *2014 18th International Conference on Information Visualisation*, July 2014, pp. 94–101.
- [13] C. di Sciascio, V. Sabol, and E. E. Veas, "Rank as you go: User-driven exploration of search results," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, ser. IUI '16. New York, NY, USA: ACM, 2016, pp. 118–129.
- [14] M. A. Hearst, *Search User Interfaces*, 1st ed. New York, NY, USA: Cambridge University Press, 2009.
- [15] B. Shneiderman, D. Byrd, and W. B. Croft, "Clarifying search: a user-interface framework for text searches," *D-lib magazine*, vol. 3, no. 1, pp. 18–20, 1997.
- [16] L. Tauscher and S. Greenberg, "Revisitation patterns in world wide web navigation," in *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '97. ACM, 1997, pp. 399–406.
- [17] M. Mayer, "Web history tools and revisitation support: A survey of existing approaches and directions," *Foundations and Trends in Human-computer Interaction*, vol. 2, no. 3, pp. 173–278, Mar. 2009.
- [18] D. Sondheim, G. Rockwell, M. Ilovan, M. Radzikowska, and S. Ruecker, "Interfacing the Collection," *Scholarly and Research Communication*, vol. 3, no. 1, 2012.
- [19] A. Wexelblat and P. Maes, "Footprints: History-rich tools for information foraging," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '99. ACM, 1999, pp. 270–277.
- [20] D. Cernea, I. Truderung, A. Kerren, and A. Ebert, "An interactive visualization for tabbed browsing behavior analysis," in *Computer Vision, Imaging and Computer Graphics – Theory and Applications*. Springer, 2014, pp. 69–84.
- [21] A. Komlodi, G. Marchionini, and D. Soergel, "Search history support for finding and using information: User interface design recommendations from a user study," *Information Processing and Management*, vol. 43, no. 1, pp. 10–29, Jan. 2007.
- [22] A. Komlodi, D. Soergel, and G. Marchionini, "Search histories for user support in user interfaces," *Journal of the Association for Information Science and Technology*, vol. 57, no. 6, pp. 803–807, 2006.
- [23] S. K. Card, G. G. Robertson, and J. D. Mackinlay, "The information visualizer, an information workspace," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '91. New York, NY, USA: ACM, 1991, pp. 181–186.
- [24] C. Eickhoff, J. Teevan, R. White, and S. Dumais, "Lessons from the journey: A query log analysis of within-session learning," in *Proc. of the 7th ACM International Conference on Web Search and Data Mining*, ser. WSDM '14. ACM, 2014, pp. 223–232.
- [25] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *Proc. of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '05. ACM, 2005, pp. 154–161.
- [26] P. Qvarfordt, G. Golovchinsky, T. Dunnigan, and E. Agapie, "Looking ahead: Query preview in exploratory search," in *Proc. of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '13. ACM, 2013, pp. 243–252.
- [27] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- [28] D. Keim, P. Bak, and M. Schäfer, "Dense pixel displays," in *Encyclopedia of Database Systems*, L. LIU and M. ZSU, Eds. Springer US, 2009, pp. 789–795.