SearchX: Empowering Collaborative Search Research

Sindunuraga Rikarno Putra
Delft University of Technology
Delft, The Netherlands
sindunuragarikarnoputra@student.tudelft.nl

Felipe Moraes∗
Delft University of Technology
Delft, The Netherlands
f.moraes@tudelft.nl

Claudia Hauff
Delft University of Technology
Delft, The Netherlands
c.hauff@tudelft.nl

ABSTRACT

Collaborative search has been an active area of research within the IR community for many years. While for “single-user” research a variety of up-to-date open-source search systems exist, few “multi-user” search tools are open-source and even fewer are being maintained. In this paper, we present SearchX, an open-source collaborative search system we are currently developing—and using for our research. We designed and built SearchX using the modern Web stack (and are thus not siloed by an operating system or a particular browser type), enabling efficient research across platforms (Desktop, mobile) and with online users (e.g. crowdworkers). A video, describing the demo can be found at https://www.youtube.com/watch?v=uf24m6p3vts.

ACM Reference format:
https://doi.org/10.1145/3209978.3210163

1 INTRODUCTION

Collaborative search, i.e. the retrieval and sensemaking of information by multiple users with the same information need, has been an active area of research for many years—workshops focusing on collaborative search explicitly have started appear in 2008, e.g. [11] and continue to be important for the development of the field today, e.g. [3].

In contrast to single-user search and retrieval (where many up-to-date and open-source tools exist such as Terrier, Lucene or Elasticsearch), our community has few software resources available to kickstart research in collaborative search as seen in Table 1. Of the six most well-known collaborative search systems, only one—Coagmento [7]—is open-sourced and that requires users to install a browser plugin or a mobile app. In contrast, we present SearchX, a collaborative search system we developed with modern Web stack in mind; it does not require users to install additional software (like a plugin/app) and is accessible to anyone with a modern Web browser, no matter their platform. This in turn enables us to employ SearchX effortlessly in large-scale online experiments with for instance crowdworkers.

In order to bring down the programmatic hurdle of collaborative search research (which ideally enables quick prototyping of ideas), we open source SearchX1 and describe here the current state of development, how it fares compared to existing software efforts and how our users (crowdworkers) experienced it in a recent deployment.

2 COMPARISONS WITH EXISTING SYSTEMS

Table 1 provides an in-depth comparison of SearchX and the most well-known existing collaborative search tools [2, 5–7, 9, 10]. We split the dimensions along which we compare the tools into three broad categories: awareness (features that raise searchers’ awareness of each other’s activities), knowledge sharing (features that enable searchers to relay their findings to each other), and division of labour (features that enable searchers to divide the work). Lastly, we also consider the availability of the tools to the wider community.

We find that Coagmento [7] and Querium [6] have the largest number of collaborative features implemented, with SearchX lagging in particular in the knowledge sharing category, which is the focus of the ongoing development effort. Interestingly, the division of labour category is the most sparsely populated across all systems, with only SearchTogether offering all of the features.

The information in the availability category provides the main motivation for our programming effort: most tools were only described in papers, but never open-sourced. Coagmento, the only tool that is open-sourced, requires an additional installation step (either a browser plugin or an iOS/Android app) that is not feasible to enable research with a large population of online users. In contrast, SearchX is a pure open-source Web application, whose architecture we describe next.

3 SEARCHX ARCHITECTURE

The high-level architecture of SearchX is shown in Figure 1. We took the single-user pineapple search framework [4]2 as a starting point, updated it, included provisions for user studies and extended it into the collaborative search realm.

Front-end. The front-end (shown in Figure 3) is written in JavaScript and makes use of React, a popular and well-maintained JavaScript front-end library. As the front-end is a standard Web application, any user with a modern browser (to be more specific a modern browser that implements ES6 functionalities as all major browser

---

1The first two authors contributed equally.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGIR’18, July 8–12, 2018, Ann Arbor, MI, USA, July 8–12, 2018 (SIGIR’18), 4 pages.

https://doi.org/10.1145/3209978.3210163

2We open-sourced SearchX at https://github.com/felipemoraes/searchx-frontend (front-end) and https://github.com/felipemoraes/searchx-backend (back-end).

The authors kindly provided us with their source code.
Table 1: Feature comparison of existing collaborative search systems (ordered by publication year of the first paper describing
the system) with SearchX. SearchX features marked with (√) are currently under development. A dash − indicates that this
information is not available. Language and Platform abbreviations: JS=JavaScript, BP=Browser Plugin, IE=Internet Explorer,
FF=Firefox, GC=Google Chrome. †The Coagmento iOS app is only available in Apple’s US app store.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Awareness</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Query History</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Colour Coding</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page Views</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Knowledge Sharing</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bookmarks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Page Rating</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page Annotation</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group Notes</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group Summary</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Division of Labour</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chat</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Results Split</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recommend / Share</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Availibility</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Functioning</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Last Update</td>
<td>2009</td>
<td>-</td>
<td>-</td>
<td>2017</td>
<td>-</td>
<td>-</td>
<td>2018</td>
</tr>
<tr>
<td>Open Source</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>PHP &amp; JS</td>
<td>-</td>
<td>PHP</td>
<td>JS</td>
</tr>
<tr>
<td>Platform</td>
<td>BP (IE)</td>
<td>Windows</td>
<td>-</td>
<td>BP (FF, GC) &amp; iOS†</td>
<td>Web</td>
<td>Web</td>
<td>Web</td>
</tr>
</tbody>
</table>

We follow React’s programming model and thus each front-end component is implemented in a modular life cycle
manner and communicates with the back-end via the RESTful API or Socket. io, depending on how instantaneous the information
needs to arrive at the collaborators’ interfaces. As we are employing SearchX for research purposes, the front-end also contains com-
ponents which are geared towards user studies and facilitate the deployment of SearchX for online experiments with crowdwork-
ers. Those components include questionnaires, a task bar, alerts (to handle crowdworker compliance), an interactive interface guide
(Figure 2), and a logger to enable the detailed logging (not just clicks,
but also hovers and scrolls) of users’ interactions with the interface. When users form a group of collaborators (determined by the back-end), the front-end also includes a number of shared components (i.e. all collaborators see them in the same state): bookmarks, query history, and chat widget.

Extensions. As this layer is implemented based on the React library, it is easily extensible. We are currently extending the interface with page view information (which collaborator viewed which document), page ratings (how useful did a collaborator find the document) and page annotations.

Back-end. The back-end makes use of the node.js server framework and is written in JavaScript. It provides the core components of our collaborative search system which we broadly categorise into four types:

- search tasks components: examples are search task assignment (tasks are assigned at random or based on users’ prior knowledge levels) and group formation (users are allocated to an available group to collaboratively search);
- retrieval components include retrieval back-end components (at the moment we rely mainly on the Bing search API but can also plug in an Elastic instance) and filters (e.g. to remove certain search results before forwarding them to the front-end);
- logging components: these include caching of search results (to serve users consistently the same results for the same issued query), a document crawler based on Puppeteer (a tool to access Headless Google Chrome) that crawls all documents shown in a SERP and user trace logging with all logs being stored in a MongoDB instance.
- the document renderer provides a document viewing experience that does not require the user to leave the SearchX interface (in contrast to clicking on the SERP and opening a new browser tab), thus allowing us to also log what users do (dwell-time and scrolling) within documents they view.

Extensions. Due to the modular nature of SearchX, the back-end can be extended to cater for a larger variety of search APIs, additional user study components and more fine-grained logs if desired.

4 USER FEEDBACK

In a recent collaborative search experiment (pairs of users, conducting searches for a learning task for 20+ minutes at a time) involving all components of SearchX we collected feedback from fifty crowdworkers in total.

At the end of the task we presented them with a questionnaire that included a set of questions elaborating on their satisfaction with the different collaborative features (chat, bookmarks and query history). Specifically, we asked “Did you find the collaborative features useful?” and expected answers on a five point Likert scale (strongly disagree to strongly agree). Figure 4 summaries the crowdworkers’ feedback in this user study. More than 60% of our users considered the shared query history and bookmarks to be useful in a collaborative search setting, while only 24% said the same about the chat feature. A possible explanation can be found in our user population (crowdworkers) which have no intrinsic motivation to excel at a task.

Among the qualitative feedback (elaborations on each of the collaborative features) the following are some of the—admittedly selected—positive statements we received:

- “We used the group chat to discuss what we had learned and the areas that we had to focus on more and what we didn’t understand.”
- “I mainly used the group chat to summarize and see if my partner saw anything wrong with my summary- to test my knowledge”
- “I knew what pages were already deemed useful by my partner based on which pages were already bookmarked.”
- “I looked at what my partner had searched [in the query history] so that I could search different terms to broaden our search”
- “I knew which base information would already be covered by my partner due to the shared query history.”

5 DEMONSTRATION

Figure 3 shows the type of demonstration we plan to show off at the SIGIR conference. We will provide at least two laptops and let participants solve fun search puzzles in a collaborative manner. The search puzzles are collected from A Google a Day3, following the template in Figure 5. This type of task has also been employed in prior interactive (collaborative) IR studies [1, 8]. We are able to artificially increase the difficulty of each task by playing around with the type of search results we return from the back-end (e.g. instead of results 1-10 we can also return results ranked 91-100).

6 CONCLUSIONS

We have presented SearchX, a system for collaborative search research, recently deployed in a crowdsourcing study with up to 60% of our users rating the collaborative features as useful to their search needs. SearchX is meant to a be a continuous effort; we plan to include more user interfaces components, in particular shared workspaces at scale (i.e. those supporting many collaborators at once). We open source SearchX (cf. footnote 1) and will continue developing it for at least three years.

ACKNOWLEDGEMENTS

This research has been supported by NWO projects LACROSSE (612.001.605) and SearchX (639.022.722).

REFERENCES


3http://www.agoogleaday.com/
Figure 3: SearchX Collaborative search interface. [A] bookmarking information including who bookmarked, [B] shared query history, [C] shared bookmarks, [D] chat. All features employ colour coding to visually represent the searcher who performed the search, added the bookmark and so on.


Figure 4: User feedback overview.

Together with your group find the answer to this puzzle, using SearchX only:

What album was released three years after the death of the artist that’s tattooed on the upper left arm of the actor who played “Irish” Micky Ward in a 2010 film?

Figure 5: Task template for our demonstration.