User Behavior in Asynchronous Slow Search

Ryan Burton
University of Michigan School of Information
105 S. State St.,
Ann Arbor MI, 48109
ryb@umich.edu

Kevyn Collins-Thompson
University of Michigan School of Information
105 S. State St.,
Ann Arbor MI, 48109
kevynct@umich.edu

ABSTRACT

Conventional Web search is predicated on returning results to users as quickly as possible. However, for some search tasks, users have reported a willingness to wait for the perfect set of results. In this work, we present the first study to analyze users' willingness to wait and their search success, when given a Web search system that embodies characteristics of *slow search*, where speed can be traded for an improvement in quality. We conducted a between-subjects user study involving tasks that required multiple queries to complete, providing a Web search system that gave users the option to additionally issue asynchronous queries for which results improve in relevance over time as users continued working. We analyze the resulting survey results and interaction log data to investigate how users spent their time while waiting, and how behavior and search outcomes changes when users are given the option of using a system with asynchronous slow search capabilities. We find that when given a slow search system, users are able to perceive the improvement in quality over time, and find tasks to be easier compared to a baseline conventional Web search system. Additionally, we find that users continue to issue their own queries and examine additional documents while the slow search queries are processed in the background, and use the slow search feature more effectively as they gain exposure to its behavior across tasks. Our study significantly advances our understanding of the benefits and tradeoffs involved in providing slow search scenarios for Web search.

Keywords

Search behavior; interactive information retrieval; user interfaces; slow search

1. INTRODUCTION

Current search systems are heavily optimized for speed: commercial search engines often conspicuously display the fraction of a second that it takes to return the list of results to a query. Traditional systems take numerous shortcuts for efficiency, such as making simplifying linguistic assumptions for query processing, document matching and ranking [16, 12]. As a result, much semantic richness is discarded in the process of retrieval, and much of the potential in terms of relevance quality may not be realized. The implicit time budget to which system developers must adhere also limits the scope and effectiveness of creative and useful extensions that may be considered for search processing and interfaces, such as enhanced personalization or novel ways of diversifying or summarizing results [10].

Slow search – the notion that a system may be able to "take its time" to process results for increased effectiveness – has been proposed, but only at the level of advancing the concept and exploring user attitudes to waiting for queries [16, 17, 8]. In this paper, we present a study that investigates the effect that an actual slow search system that supports asynchronous (background) query processing has on user behavior.

Search that focuses on speed, sometimes at the expense of quality, may be underserving users with particular needs or devices. For instance, the growth of mobile phone usage is outpacing that of desktop PCs—especially in developing countries—but there is a capability gap not only between phones and PCs, but between different phones as well. This may lead to lower levels of information seeking and engagement [14]. This study would therefore be useful to search engine implementers and interface designers targeting developing regions. Demonstrating the feasibility of this new slow search paradigm would also encourage implementers of conventional search engines to further explore the importance of the time–quality tradeoff, potentially leading to more systems that can automatically adjust their performance along a scale that effectively trades off urgency and quality.

The contributions we present in this paper include an extensive analysis of a search system that embodies characteristics of slow search. We are primarily interested in the practical value of trading speed for quality. To that end, we developed a novel system which improves the topic relevance of a query asynchronously over time while the user continues to work. This allows us to investigate the types of tasks for which users are willing to tolerate a delay in processing for more relevant search results. Using log data, we show how users behave when given asynchronous slow search capabilities and compare it to a baseline without these features. We also trained a logistic regression classifier to predict task success depending on the capabilities given to the user and interaction features. We also contribute an anonymized data set[1] to allow for analysis by other researchers.

As the primary purpose of this work is concerned with understanding patterns of interaction behavior when users have the ability to run a slow search in the background, we consider the following research questions:

SIGIR '16, July 17-21, 2016, Pisa, Italy

DOI: http://dx.doi.org/10.1145/2911451.2911541

[1] http://umich.edu/~ryb/slow

RQ1: What are the types of queries for which users initially report they would have a willingness to wait?
RQ2: How much time will users typically wait for results from a slow query?
RQ3: In terms of search activity, how do users spend their time while waiting for a slow query to finish?
RQ4: How does typical user behavior change when provided with the ability to run a slow query?
RQ5: Do users perform search tasks more effectively with slow search?

We address RQ1 in Section 3.2, and RQ2–RQ5 in Section 4.

2. RELATED WORK

The concept of slow search was introduced to the literature by Dörk et al. [8] and Teevan et al. [16]. With inspiration from other "slow" movements, including slow food, slow travel, and slow technology, the authors posit the changes in how individuals and groups approach the process of search if a system emphasized slowness over speed. Poirier and Robinson [15] described a model of how slow principles may be applicable to information behavior. These initial papers provide insight in how proposed slow systems might be built, or survey-based results on how users might be willing to use such systems.

Previous work has shown that users would be willing to engage in slow search for certain kinds of queries and tasks. A study by Teevan et al. [16] based on user surveys and empirical analysis of search query logs found that, while increased load times for search results led to increased abandonment for typical queries, for tasks in which result quality was poor, users were willing to wait for better results or try alternative methods of finding information. However, none of this previous work built or studied a working slow search system with real users: to our knowledge, our paper represents the first study of how users interact with an actual Web search scenario providing slow search features.

Asynchronous search has been studied previously, but primarily in the context of bandwidth limitations and without recognition to the notion of improving search results [5]. Prior work has examined the relationship between time delays and user behavior [3, 16, 13], but these were in the context of conventional search systems, where users have the expectation of rapid responses, and with no benefit to waiting.

Slow search also has parallels with question answering systems. Aperjis et al. [2] found that users wait longer to get an additional answer after receiving a small number of responses on Yahoo! Answers. By analogy, a user performing a search may be willing to repeatedly check in on the results of a slow search as it builds a final result set, and decide whether to stop the search or continue waiting. However, the authors do not show how the number of answers for a question relate to their quality or the times in which they arrive. Liu et al. [11] demonstrated in a field experiment that the frequency, quality and time of solutions to tasks on the crowdsourcing site Taskcn reflect strategic decisions depending on the reward level, the existence of a reserve (i.e., a prior high-quality solution), and expertise of crowdworkers. The authors however do not investigate the waiting behavior of the requester in light of the solutions. In other words, for slow search, we are interested in a mix of the two—examining when a user believes the information received is "good enough" to stop waiting for additional information.

Büttcher et al. [4] compared the effectiveness of different systems while accounting for the CPU time involved in query processing. Generally, the systems that used more CPU time showed better results in effectiveness. In the efficiency task, comparing each system's best run to its fastest run, the differences in ms/query can be quite appreciable. This shows that there is often a benefit to extra processing time, and a system that takes advantage of this time when appropriate could satisfy users better, provided they are willing to wait.

There is also increasing recognition of time as an important factor in the evaluation of search systems. Clarke and Smucker [6] proposed a metric of time-based gain to measure an information retrieval system's effectiveness to reflect the value that a user gains over time in interacting with the system. For slow search, this metric is applicable to the value gained from waiting as the system works to provide better results. A recent user study by Crescenzi et al. looked at a design somewhat contrary to ours, namely, the effect on search behavior when users were given *less* time to search [7].

Compared to the existing literature, this work presents a working system that embodies the principles of slow search and directly improves the relevance of search results, while investigating the relationship between types of tasks, user impatience, and quality improvement over time.

3. METHOD

To measure user behavior characteristics, we designed an extension for the Chrome Web browser that works in conjunction with Web search engines to capture the current query and send it to a server for extended processing when the user clicks a "Work Harder" button to the right of the main search editbox on the search engine page, as shown in Figure 1. Doing this adds the query to a sidebar (a) on the search engine result page, which shows a progress bar (b) as well as the top three results at any given time (c). We call this extended-time background query a 'slow' query. The user may click on the "(more results)" link (d) at the bottom of the sidebar to view the full list of re-ranked results, as they are improved and updated asynchronously by potentially adding new documents to the list and re-ranking them. This page also displays a progress bar, and may be left open while the user continues to search on the main search page.

This 'slow' query processing occurs as a background process, during which users are free to continue performing their own searching and query reformulations in the main interface while the 'slow' query completes. In this study, we allow at most one slow query at a time, which may be cancelled before its progress is complete and removed from the sidebar. The extension also serves to log interaction through queries, clicks, and mouse movements.

3.1 Study participants

Our study consisted of 44 participants (18 Male, 26 Female; mean age = 23.5 SD = 5.9), recruited through the University of Michigan School of Information. Most were undergraduates (n = 18) or holders of an undergraduate degree (n = 12). The majority reported being very experienced with search engines: we asked about their familiarity on a scale from one to five; the mean response was 4.6 (SD = 0.6). Additionally, 38 reported using search engines more than once per day, while 5 reported using them more than once per week. The remaining participant reported using them more than once per month. We also asked participants to report their confidence in their abilities to find the information they need while searching on a scale from one to five; the mean response was 4.36 (SD = 0.65).

3.2 Background Survey

To better understand tasks for which people might be willing to wait for a better answer (RQ1), we asked participants to provide a description of the last search task they performed in which they failed in satisfying their information need. We report these tasks as well as their anticipated willingness to wait for the perfect results below.

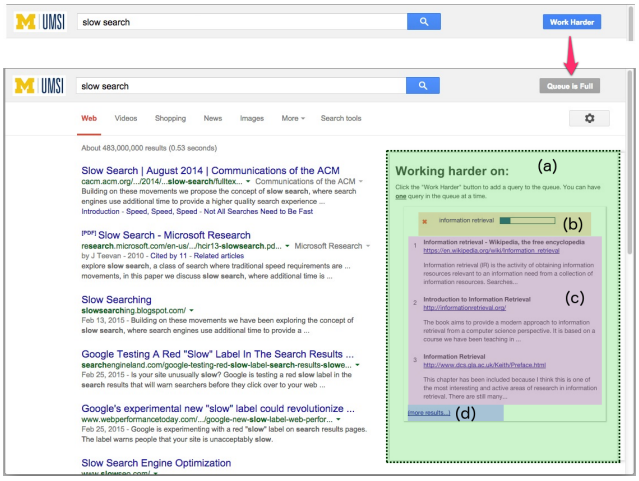

Figure 1: Interface with "Work Harder" button and sidebar (a). Colors added for illustration. Clicking the "Work Harder" button in the upper right adds the current query to the queue (b). The top three results at any moment are presented below (c), and a full list of re-ranked results is available by clicking on (d). These interface additions are always present.

3.2.1 *Prior tasks users reported as difficult*

We first asked participants the following question:

Think back to the last time you had trouble finding information with a Web search engine. What was the information that you were trying to find? Please be as specific as you can, as best as you can remember.

We coded responses by topic, summarized in Table 1. As most participants were students, the majority had issues finding information for classes or assignments. The common trends for difficult education-related needs involve finding new and novel information (e.g., finding articles on a topic that has not been seen before), finding reliable scholarly articles on a topic, and expressing the problem in the right way for the search engine to yield useful results ("It was difficult to search for because I wasn't sure what I was searching for.").

For other topics, users had difficulty finding a specific item, such as a person, product, or song, most commonly among the Career, Entertainment, and Shopping topics. The main issue in these cases involved expressing the right criteria to find these items. For instance, one subject tried to find a particular drawer slide, but was not able to use the right search terms. Instead, he had to iteratively search related topics in order to pick up more useful search terms.

Topic	Count
Education	16
Shopping	6
Entertainment	5
Health	5
Career	3
Technology/Troubleshooting	3
Food	2
Sports	2

Table 1: Topics of tasks reported as difficult.

We also categorized participants' reported tasks according to the nature of information they were seeking. Overall, 16/44 (36%) of difficult/unsatisfied needs involved searching for specific items or facts that satisfied multiple attributes; 10/44 (22%) were questions seeking a specific factual answer; 4/44 (9%) needs were for the latest version of information; 4/44 (9%) involved searching for a person. The remaining needs involved more vaguely-defined needs, more exploratory research needs, or procedural information on how to solve a problem. This predominance of multi-attribute search needs, the nature of which we can find examined in [10], motivated our design of tasks for slow search as described in Sec. 3.3.1.

3.2.2 *User willingness to wait*

As part of studying the time-quality tradeoffs that users might find acceptable in a search engine (RQ2), we asked participants:

Given your experience, if a search system was able to provide the perfect results, how long would you be willing to wait for the search engine to process your query while you did other tasks and you were notified when it found these results?

We are interested in what users self-report as acceptable, not only to calibrate our experiment, but also to see if their actual behavior matches the expected behavior (which we compare in Sec. 4.1). Users reported a willingness to wait 9.5 minutes on average (SD = 13.2). An exponential decay in acceptable waiting time is evident from our analysis: with the survey response data, we fitted an exponential decay model $w = \exp(-at)$ to estimate the empirical probability w that a typical user would be willing to wait at least t minutes. The fitted exponential parameter was $a = 0.11$, meaning that for every additional minute of waiting time, about 10% of remaining users were not willing to continue waiting. We note that this rate of decay 'impatience factor' is in accord with that reported by Teevan et al. [16] that asked a similar question about willingness to wait for perfect results.

3.3 Experiment Design

For the purpose of this study, we focused on investigating the effects of an improvement in relevance for multi-attribute tasks. To that end, we implemented a server that communicates with the Chrome extension to simulate an improvement in relevance over the course of five minutes for each slow query submitted. For each task that a user may choose to tackle, we manually selected five to ten high-quality documents and snippets that, collectively, allow a participant to correctly solve the problem posed by the task. When the "Work Harder" button is used, the server selects documents from the pool to insert into the ranking every twenty seconds. Similarly, another process on the server periodically moves high-quality documents closer to the top of the ranking over the course of the five minute period, until these documents reach the top of the ranking.

We randomly assigned participants to one of three conditions. In the baseline condition ($n = 16$), the interface resembles a conventional Web search engine, with no "Work Harder" button or sidebar. In the "*static gain*" condition ($n = 15$), the interface adds a persistent "Work Harder" button and sidebar to the conventional interface. Furthermore, the system inserts highly-relevant documents in the middle of the ranking "below the fold" of the re-ranked results page and the rank position of these documents stays the same over the course of the five minutes. Finally, in the "*dynamic gain*" condition ($n = 13$), the interface is the same as in the "*static gain*" condition, but the system inserts documents at

the last position of the re-ranked list and then continuously increases the position of documents at 20 second intervals, over the five minute time window, until they finish at the top of the ranking. In this study, we used a dynamic gain that was linear with respect to time. With this design, we introduce the two new capabilities of an improved result list and a dynamic ranking. We chose to contrast the "*static gain*" condition with the "*dynamic gain*" condition to determine whether users actually perceived the improved relevance as well as to study the effect of the dynamic ranking.

3.3.1 Description of Search Tasks

Participants were presented with a list of four topics, with each topic having three tasks within it. Each participant was required to select one task from two separate topics. We allowed participants to choose tasks and topics of interest to them with the goal of increasing their intrinsic motivation to complete each task.

We prepared the total set of twelve tasks such that each task was presented in the form of a question to be answered, and each task called for the participant to find five items that satisfy multiple attributes specified within the problem. We did this to control the cognitive effort required for each task – users were expected to find a set of candidate answers and verify that each of them satisfied all constraints in order to receive the full reward. For any particular item that the user submitted in their answer, we considered it "correct" if and only if it satisfied all of the required constraints. We believed that having a slow search system which reduced this high expected effort would encourage use of that capability when available.

Local Businesses (32 tasks completed)
Name five I.T. companies in Ann Arbor with at least 50 employees.
Entertainment (28 tasks completed)
Name five video games in which Pharrell Williams's music has been featured.
Education (21 tasks completed)
Who are the five most influential professors in the United States in the field of sociology?
Shopping (7 tasks completed)
What are five smartphones that are thinner than a standard No. 2 pencil and usable on AT&T?

Table 2: Examples of search tasks and their topics.

In Table 2 we present examples of search tasks for each of the four topics, along with the number of task completions by topic. Local Businesses had the most interest, with its tasks being chosen 32 times in total. Conversely, Shopping received the least attention, which users choosing these tasks only 7 times.

3.3.2 Study Procedure

The user study took place in a laboratory setting at the University of Michigan School of Information. Participants volunteered to attend one of eight study sessions, with each session lasting a maximum of 90 minutes. Each participant was placed at a computer set up with the Chrome extension, which in turn was randomly associated with one of the three study conditions (Baseline, *static gain*, and *dynamic gain*). Participants completed two search tasks, with each task lasting a maximum of thirty minutes.

To introduce participants to the capabilities of the system before they began the first task, users were asked to perform an exploratory search task—in this case to explore the topic 'snow leopards' —as a warmup for five to ten minutes.

As motivation to finish the task within the allotted thirty minutes, we compensated users based on their performance in answering each question, which called for an answer that addressed each attribute of the problem, as well as three relevant documents that they found useful in solving the problem. This gave us a way of verifying whether participants found the documents inserted into the ranking, and explicit relevance feedback of these documents. An answer that perfectly met the criteria of the task led to a bonus of $2 with partial credit being possible, and giving relevant documents led to a bonus of $1 per URL – this served as motivation to give explicit relevance feedback.

3.4 Data Preparation

Missing Data. For each request made by the extension to log interaction data, the system associated a session ID with a particular interaction event, with this session ID being linked to the user's ID, which is randomly generated when a user begins the study. After the data collection was complete, there were 34 out of 1149 clicks without session IDs in our log database, and hence, we were unable to associate these clicks with a task, user ID, or condition. We therefore manually inspected the click data in an attempt to re-associate each click with a session ID. We were able to re-associate all clicks but one due to ambiguity in candidate tasks: 13 of 34 clicks were recoverable from session IDs included in page URLs, and 20 of 34 clicks could be manually recovered based on analyzing clicks with session IDs from closely associated contemporaneous queries.

Relevance Judgements. As a part of completing the task, we asked users to provide three relevant documents that helped in answering the task's question. We made final judgements on these documents in the process of calculating bonuses – if the document indeed provides information relevant to answering the question correctly, then the document was deemed relevant for the bonus. Otherwise, we considered the document not relevant.[2]

4. EXPERIMENT RESULTS

In this section we conduct an analysis of user activity, addressing the remaining research questions (RQ2–RQ5) and examining users' behavior in more detail.

4.1 How Long Participants Waited For Results

Our second research question (RQ2) concerns the amount of time users typically wait for results. In our background questionnaire described in Section 3.2.2, our participants expressed a mean willingness to wait 9.5 minutes for the perfect results for their difficult information needs. Our system imposes a maximum wait time of five minutes, which was not explicitly communicated to our participants. Five minutes was selected in order to give users more time to submit multiple slow queries within a single task session. 2/15 participants in the *static gain* condition and 2/13 participants in the *dynamic gain* condition explicitly mentioned without prompting that the waiting time was too long when asked about their impressions of using the system in post-task questionnaires.

Analyzing the behavioral data, we find that in both the *dynamic gain* and *static gain* conditions, approximately twice as many participants used slow search in the second task ($n = 11$) compared to the first task ($n = 5$). Table 3 presents expected wait times for each condition and task, representing the time for which each slow query is processed until

[2]If the document only provides information that would result in answering the question incorrectly, as, for example, only providing a Web page for a coffee shop outside of the location we ask, the page was considered not relevant.

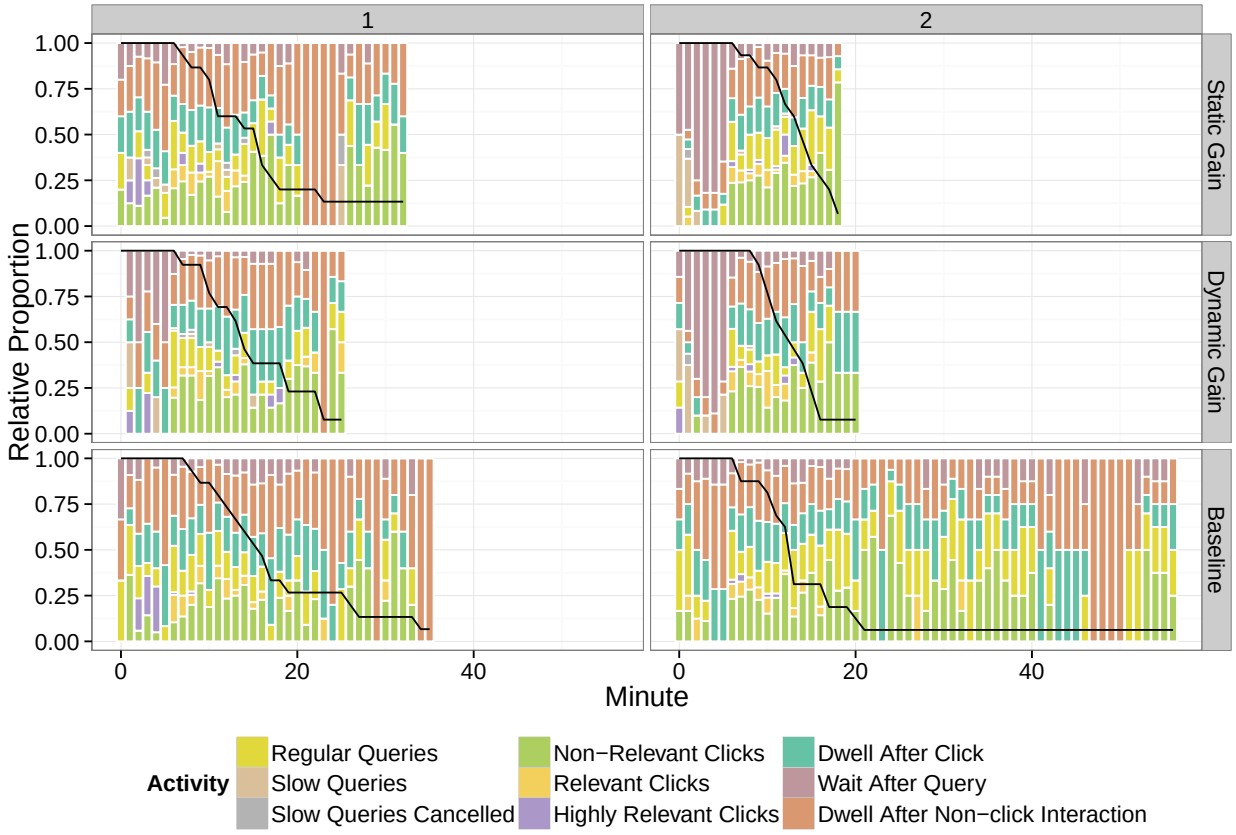

Figure 2: The actions that users perform over the course of the two tasks by condition. The black lines show the proportion of remaining participants in the session.

Condition	Task	Query Processing Time (sec)	SD	Users
dynamic gain	1	227.3	109.8	4/13
dynamic gain	2	227.2	82.4	3/13
static gain	1	136.1	145.6	5/15
static gain	2	266.6	85.0	1/15

Table 3: Mean slow query processing times by task, with standard deviation (SD) and fraction of users who cancelled their slow query.

either completion, or cancellation by a user. Users in the *dynamic gain* condition waited an average of 227 seconds in both tasks 1 and 2. By comparison, for users in the *static gain* condition, their wait times increased from an expected 136.1 seconds to 266.6 seconds: participants cancelled more queries in the first task than in the *dynamic gain* condition, but by the second task, these users cancelled fewer than in the *dynamic gain* condition, which likely contributed to the increase in mean slow processing time observed for the static gain condition. An independent two-group t-test shows that the difference between the mean wait time in the first and second tasks of the *static gain* condition is statistically significant ($t(9.7949) = -2.318$, $p < 0.05$).

4.2 How Participants Spent Their Time

To obtain an overview of user activity as participants progressed through each task, we aggregated the actions that users performed and averaged across users for each minute of activity. The resulting plot of these aggregated actions is shown in Figure 2. Generally, participants took slightly longer to complete their first task than their second (944 seconds vs. 804 seconds on average). For the two tasks, the users in the dynamic gain condition had the shortest completion times (879 seconds for the first task, and 735 seconds for the second task). These differences were not statistically significant.

Differences between first and second session. In general, there appeared to be a period of slight acclimatisation as users in the slow search conditions made and cancelled slow queries throughout the session. By comparison, in the second task, users started by making slow queries and committed more to this decision rather than cancelling and restarting. More precisely, in the *static gain* condition, users made an average of 0.53 slow queries and cancelled 0.33 of them in the first task. By the second task, they made 0.93 slow queries and cancelled 0.07 of them. Similarly, in the *dynamic gain* condition, they made an average of 0.62 slow queries in their first task and cancelled 0.23 of them; by their second task, they made 1 slow query and cancelled 0.08 of them. Potentially, this small number of slow queries could reflect an optimal interaction strategy, which we will discuss further in Section 4.4. We present the results for comparison in Table 4.

Document assessment over time. In Figure 3, we plot the mean cumulative clicks performed during each task as time progressed. We can see that the conditions began with

Condition	Task	Submitted	Cancelled
dynamic gain	1	0.62	0.23
dynamic gain	2	1.00	0.08
static gain	1	0.53	0.33
static gain	2	0.93	0.07

Table 4: Slow queries submitted/cancelled by task.

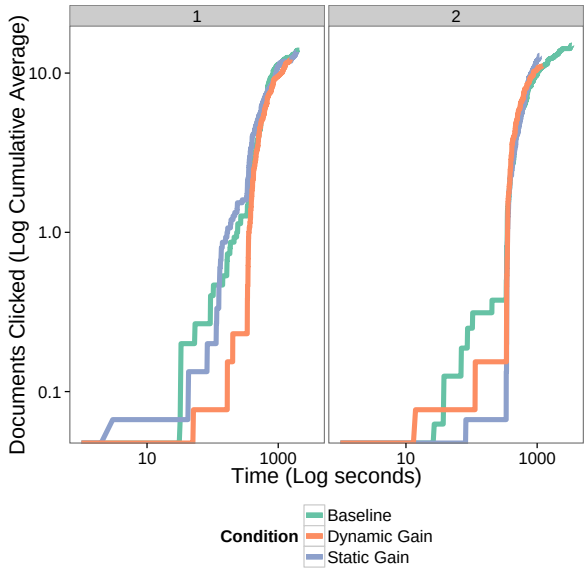

Figure 3: Average time-click curves by task. This includes non-relevant clicks.

rather different click trajectories at the beginning of each task, but eventually converged as the task continued. Users in the *dynamic gain* condition begin examining documents slightly later than in the baseline and *static gain* conditions in the first task. Comparing this to the second task, users in the three conditions examined fewer documents in the corresponding period at the beginning of the first task. Additionally, users in the baseline condition began examining more documents sooner than in the slow conditions.

4.2.1 Behavior While Waiting

Our third research question (RQ3) pertains to activity while waiting for slow search results to finish processing. To answer this question, we looked at how users continued to interact with the system after submitting a slow query. We note that for the conditions with the slow search button, many users spent their time waiting after submitting a slow query. This was especially pronounced in their first five minutes of each task, where more users waited on average after making a query than at subsequent time periods, Comparing the two slow search conditions in the first task, we see in Figure 2 that more users spend time waiting after querying in the first five minutes of *dynamic gain* than in *static gain*. However, the profiles are more similar by the second task: in both groups, users made heavy use of the "Work Harder" button initially, waiting before eventually clicking on results.

Figure 4 shows the activities that users performed while slow queries were processing for the *dynamic gain* and *static gain* conditions, i.e. after a slow query was submitted, and before the query either finished processing or was cancelled.

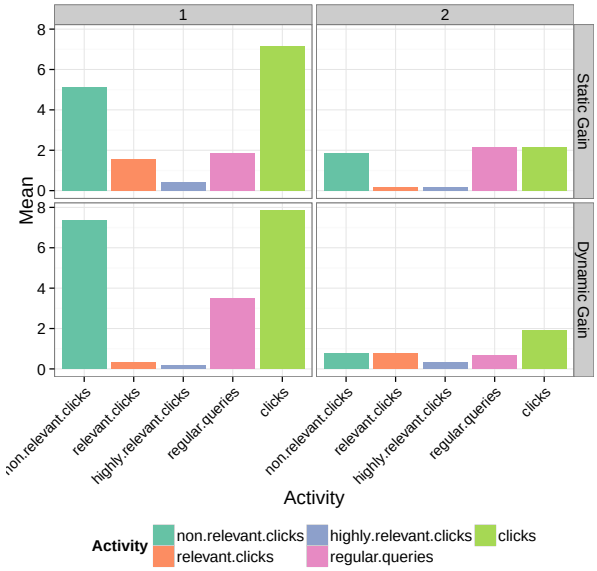

Figure 4: How users spend their time while waiting for slow queries to finish.

Most activity in this interval is focused on examining documents: the number of clicks is relatively high in the first task (6.6 for *static gain* and 7.6 for *dynamic gain*). In comparison, the number of queries is relatively low (2.2 for *static gain* and 3.8 for *dynamic gain*). By the second task, users do less in this interval, perhaps relying on the system more than examining documents and conducting additional queries themselves. In both conditions, the average number of queries and clicks both decrease (*static gain*: Queries = 1.55, Clicks = 3; *dynamic gain*: Queries = 0.91, Clicks = 3.09). Of additional note is that for *static gain*, the number of relevant clicks dropped from 1.2 to 0.18, while the number of highly relevant clicks stayed relatively consistent (0.6 in task 1 and 0.55 in task 2). This may indicate that users' time was better spent in the second task with regards to finding the most relevant documents to solving the given task. We can compare this to *dynamic gain*, where the number of relevant clicks increased (0.4 to 0.73), and also the number of highly relevant clicks increased from 0 to 0.36. Thus, while the number of non-relevant clicks and queries decreased, users made better use of their time in finding helpful documents.

4.3 Feature Analysis of Search Behavior

To investigate research questions RQ4 and RQ5, we computed a list of features characterizing search behavior, as inspired by previous studies such as [1]. The features we calculated are outlined in Table 5 and Table 6.

Dwell time (C_{IT}) was determined by calculating the time between a click and any subsequent interaction with a search page (mouse movements, scrolling, keyboard events, queries, or clicks). As we ask users to provide five correct items that satisfy multiple attributes for each task, we calculate Precision as the proportion of items included in their answer for a task that satisfy all attributes.

4.3.1 Behavioral Analysis of Searchers by Condition

For our fourth research question (RQ4), we investigate the types of changes seen in users' behavior when given asynchronous slow search capabilities. Having randomly assigned

Feature	SG	DG	Baseline	U Test
Baseline features				
Session length ($\Sigma\Delta t^*$, sec.)	839.47	807.23	961.00	-
Regular queries ($CntQ_R$)	6.50	4.58	8.81	-
Regular queries/sec (Q_RPS)	0.01	0.01	0.01	-
Slow features				
Slow queries ($CntQ_S$)	0.73	0.81	0.00	SG > B*; DG > B*
Slow queries per second (Q_SPS)	9.03×10^{-4}	9.99×10^{-4}	0.00	-
Slow queries cancelled ($CntQ_SC$)	0.33	0.35	0.00	SG > B*; DG > B*
Slow queries cancelled per second (Q_SCPS)	3.86×10^{-4}	3.65×10^{-4}	0.00	SG > B*; DG > B*
Query features				
Query word length (QWL)	4.48	5.20	5.72	-
Query character length (QCL)	27.14	31.20	34.88	-
Click features				
Pages in session ($CntR$)	13.27	11.62	14.48	-
Clicks per query (CPQ)	3.14	3.23	2.37	-
Time to first click (sec.) (Q_{DT})	24.93	25.53	24.46	-
Dwell Time (sec.) (C_{IT})	234.36	276.28	358.86	-
Outcomes				
Reward ($)	4.21	4.04	4.16	-
Reward Variance ($)	1.13	1.12	2.14	-
Precision	0.85	0.72	0.83	-
Click Relevance	4.03	3.08	3.55	-

Table 5: Comparison of behavioral features across conditions. SG = *Static Gain*; DG = *Dynamic Gain*. *p<0.05; **p<0.01; ***p<0.001.

users into a condition either with or without such capabilities, we compare the session-level features for each condition. We present the values of these features in Table 5.

Compared to the two slow search conditions, users in the baseline condition on average were the slowest in completing a session ($\Sigma\Delta t = 961$ seconds), issued the highest number of queries ($CntQ_R = 8.81$) and the longest queries ($QWL = 5.72$; $QCL = 34.88$), and had the longest dwell time ($C_{IT} = 358.86$). These differences were not statistically significant, but they may reflect a greater degree of effort for users in this condition, as users take more time to examine and possibly evaluate documents, and conduct more queries to address the various facets of the problem. We also found that users did indeed make use of slow search when given the option: features that quantify the use of slow search such as $CntQ_S$, Q_SPS, $CntQ_SC$, and Q_SCPS were significantly greater than zero in the *dynamic gain* and *static gain* conditions ($p < 0.05$). For most of these features (that is, $CntQ_S$, Q_SPS, and $CntQ_SC$), the values were highest in *dynamic gain*, though not significantly more so than in *static gain*. In contrast, Q_SCPS was highest in the *static gain* condition, though this was not statistically significant. No other differences were significant (with all tests here based on paired Mann-Whitney U tests with Bonferroni correction). Cognitive factors such as the evolving degree of user trust in result quality for the slow search conditions may contribute to these cross-condition differences and exploring these is a topic for future work.

Feature	NS	S	U Test
Baseline features			
Session length ($\Sigma\Delta t^*$, sec.)	847.69	891.10	NS < S*
Regular queries ($CntQ_R$)	7.53	6.20	-
Regular queries/sec (Q_RPS)	0.01	0.01	-
Slow query features			
Slow queries ($CntQ_S$)	0.39	0.57	-
Slow queries per second (Q_SPS)	5.22×10^{-4}	6.72×10^{-4}	-
Slow queries cancelled ($CntQ_SC$)	0.14	0.27	-
Slow queries cancelled/sec (Q_SCPS)	2.97×10^{-4}	1.65×10^{-4}	-
Query features			
Query word length (QWL)	5.01	5.23	-
Query character length (QCL)	28.90	32.67	-
Click features			
Pages in session ($CntR$)	12.44	13.75	-
Clicks per query (CPQ)	2.55	3.14	-
Dwell Time (sec.) (C_{IT})	284.20	296.36	-
Time to first click (sec.) (Q_{DT})	13.65	32.86	NS < S*
Outcomes			
Reward ($)	3.23	4.74	-
Reward Variance ($)	1.70	0.36	-
Precision	0.52	1.00	NS < S***
Click Relevance	3.42	3.69	-

Table 6: Comparison of behavioral features by success level. NS = Not Successful; S = Successful. *p<0.05; **p<0.01; ***p<0.001.

4.3.2 Behavioral Analysis of Successful Searchers

Our fifth and final research question (RQ5) investigates whether users perform tasks more effectively with the help of slow search. As Table 5 shows, users in the *dynamic gain* condition received the smallest reward, and had the lowest precision. This raised the question of what factors played a part in increased performance. We compared user features for successfully completed tasks to those of the remaining tasks. We define success as a precision of 1 for a particular task, such that the answers given satisfied all the criteria set by the task's question. We present the features computed based on success in Table 6. We performed Mann-Whitney U tests to determine whether there were significant differences by success.

We found significant differences in session length ($\Sigma\Delta t$; $p < 0.05$) and time to first click ($p < 0.05$). The average session length for successful tasks (891.10 seconds) was significantly higher than that for less successful tasks (847.69 seconds). Despite this, the number of queries issued is not significantly different, though the average time to first click is significantly higher for the successful (343 seconds) than for the less successful (260.44 seconds). This indicates that the time spent examining the search results was a major factor in success, as we also notice that the dwell times were not significantly different. We also investigated click relevance, as described in Section 4.2. Users who were successful had an expected click relevance of 3.69, compared to a click relevance of 3.42 for the rest of the users. However, this difference was not statistically significant.

We also performed logistic regression to predict user success using the above feature set. We found that the intercept ($\beta = -6.524$) and the clicks per query ($\beta = 0.887$) were sig-

nificant predictors of success ($p < 0.05$). For the intercept, a participant is not likely to be successful, with all other predictors held constant. An additional click per query increases the odds of success by 143%. Other predictors that were marginally significant ($p < 0.1$) include the rate of regular queries (Q_RPS, $\beta = -0.009$, $p < 0.1$) and the effect of session length when the condition is *static gain* ($\beta = -0.009$, $p = 0.1$). In the case of the query rate, an increase in this rate predicts an increase in the odds of success, whereas an increase in the session length in the *static gain* condition predicts a decrease in the odds of success. No other terms were significant predictors.

Overall, the logistic regression analysis shows that making good use of one's time is the main factor in success. That is, searching and examining documents in a short period of time usually means that the user will be successful. The interaction between the session length and using the *static gain* system also suggests that, as a longer session length implies difficulty in satisfying an information need, not being able to take adequate advantage of the system's assistance decreases the likelihood of success.

4.4 Analysis of Interaction Strategies

Because the ability to perform a slow search was a new feature for participants – the training period built into the start of the study not withstanding – we examined how participants' choice of search strategies changed across sessions as users became more familiar with the feature.

In particular, we were interested in how users in the two slow conditions adapted their decision-making and use of the feature in relation to more optimal strategies. Each of the slow conditions could be considered to have an optimal strategy in terms of the number of regular queries issued, the number of snippets examined, the time taken to invoke the "Work Harder" button for the first time, and the waiting time for slow processing.

For the *dynamic gain* condition, we consider one optimal strategy to be the following: 1. Issue a query; 2. Click "Work Harder"; 3. Wait for 5 minutes as the results automatically improve to maximum effectiveness; 4. Examine the first 10 slow results[3]. In comparison, the *static gain* condition has a very different strategy: 1. Issue a query; 2. Click "Work Harder" 3. Examine the first 30 slow results immediately. The differences stem from the fact that the *static gain* condition happens to improve relevance immediately, but to a much lesser degree than in the *dynamic gain* condition at 100% completion. Thus, for the *static gain* condition, it is in the user's best interest not to wait, but this is not evident from the interface.

To examine how behavior changed relative to these strategies, we analyzed whether these strategic components shifted toward optimality from the first task to the second task, in each condition. To do this, we estimated the number of snippets examined by using time on page from our baseline condition to determine the time to examine one snippet (8s), and used the time on page from the improved results page with slow results with the assumption that the times to examine a snippet are comparable. This value is capped at the number of results on the page (50). We employed a bootstrap hypothesis testing procedure [9], and present our findings in Tables 7 and 8.

Table 7 shows for that for *static gain* users, the time it took for users to first use slow search significantly decreased from the first to second task: from 386 s to 130 s ($p < 0.01$).

[3]The times taken for each step would be as short as possible, and a user might elect to do other things, including regular searches, during the waiting interval.

[4]Insufficient data due to lack of use of this feature during the session.

Component	Task 1	Task 2	p-value
Queries	6.4	6.6	0.479
Time to "Work Harder" (sec.)	386	130.1	0.005 **
Wait time (sec.)	136.1	277.3	0.002 **
Snippets examined	NA[4]	22.5	

Table 7: Changes in interaction in relation to optimal strategies for *static gain*. The number of snippets examined is estimated. *p<0.05; **p<0.01; ***p<0.001.

Component	Task 1	Task 2	p-value
Queries	5.38	3.68	0.131
Time to "Work Harder" (sec.)	190.8	135.9	0.04 *
Wait time (sec.)	266.6	277.1	0.048 *
Snippets examined	21.2	20.1	0.5

Table 8: Changes in interaction in relation to optimal strategies for *dynamic gain*. The number of snippets examined is estimated. *p<0.05; **p<0.01; ***p<0.001.

Wait time increased significantly from 136.1 s to 277.3 s ($p < 0.01$). We see users moving closer to the optimal strategy for when to invoke slow search, but not for wait time. The number of queries did not change significantly, but increased slightly from 6.4 to 6.6. This could suggest that users did not know to take advantage of the fact that the preselected documents were always in the middle of the ranking, and continued to search on their own even as they waited more for an effect.

Table 8 shows that in the *dynamic gain* condition, users invoked slow search much sooner for the second task (190.8 s to 135.9 s; $p < 0.05$), and significantly increased their waiting time (266.6 s to 277.1 s; $p < 0.05$). Additionally, although not statistically significant, we notice a decrease in the number of queries issued (5.38 to 3.68). This seems to suggest that these users had begun adjusting their behaviors toward the optimal strategy, as they developed a better mental model of how the system responded to their use of the "Work Harder" button.

4.5 Post-task Survey Results

After each task, we asked users about their experience using the system. We also asked participants in conditions having the "Work Harder" button to give their impressions on whether the button made the task easier, whether they noticed an improvement in the quality of results, as well as to write about their thoughts on the usefulness and ease of use of the system.

Figure 5 shows the mean ratings on a five point Likert scale of participants' experiences of using the system by condition. The error bars represent the standard errors of the means. We performed ANOVAs on these results to see if exposure to the different conditions affected the ratings given to whether they were able to find the information they were looking for, their productivity, the effort extended, if they liked using the system, if the button made the task easier, if the button improved the quality of results, if the progress bars were useful, and if the ability to check the intermediate results was useful. Of these, we saw significant differences between conditions in response to the button making the task easier ($F(1, 54) = 5.324$, $p < 0.05$), the button improving the quality of the results ($F(1, 54) = 4.529$, $p < 0.05$), and the progress bars being useful ($F(1, 54) = 5.146$, $p < 0.05$).

In general, users' perceptions of the slow search features (the latter four in Figure 5) were higher in the *dynamic gain*

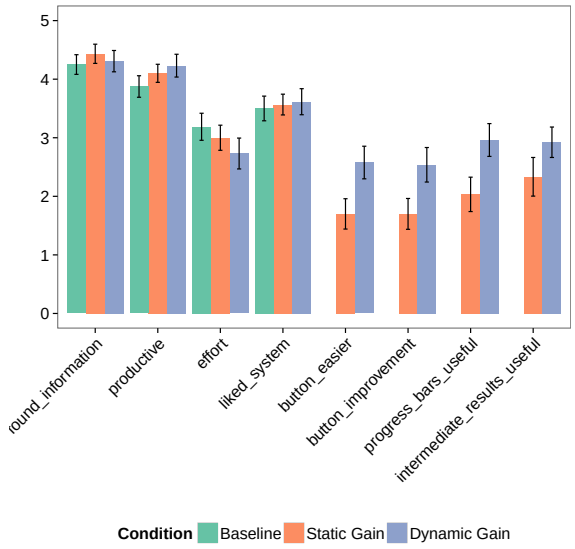

Figure 5: Post-task survey scores by condition.

condition than in the *static gain* condition. Among the more general experiential questions (the former four), we see that users in the *static gain* condition gave the highest rating for whether they found the information they were looking for ($M = 4.43$, $SD = 0.90$), while in the *dynamic gain* condition they gave the lowest rating ($M = 4$, $SD = 1.17$). For productivity, we see that having the button improved users' perceptions over the baseline ($M = 3.88$, $SD = 1.04$), with the *static gain* condition having the slight edge ($M = 4.1$, $SD = 0.84$) over the *dynamic gain* condition ($M = 4.07$, $SD = 1.01$). Users in the baseline condition also reported exerting the most effort ($M = 3.19$, $SD = 1.31$), which might have been reflected in their interactions, with users in this condition taking longer on average to complete tasks, perform more queries, and examine documents. Compared to the other conditions, users in the *static gain* condition reported liking the system the most ($M = 3.57$, $SD = 0.97$). Indeed, among these former four questions, the *static gain* condition has the highest ratings, though, once again, the differences were not statistically significant.

5. DISCUSSION AND IMPLICATIONS

Our study provides insights about how users engaged with a slow search system that provided an asynchronous query capability with improvements in search result quality over time. We now discuss our main findings and implications.

Users are willing to wait for multi-attribute queries (RQ1). We found through our background survey that many of the tasks and queries that users typically have trouble with are multi-attribute queries in which various constraints of a query must be satisfied (Section 3.2.1). This justifies our use of such queries in our study, and the use of slow search after users gain familiarity with the system shows that multi-attribute queries are a good fit for a slow search system.

Users will typically wait for results (RQ2). Our background survey revealed that users reportedly are willing to wait for a mean of 9.5 minutes for "perfect" results (Section 3.2.2). Placing users under time pressure and imposing a maximum time of five minutes for query processing also led to users waiting, as was seen in Section 4.1. Interestingly, in both *dynamic gain* and *static gain* conditions, users typically submitted more slow queries, waited more, and cancelled fewer slow queries by the second task. A future study may manipulate the processing time for these slow queries to examine users' tolerance for waiting, and whether users will wait under tighter time constraints. Future studies may also look at impatience under greater uncertainty.

Users spent time looking for additional documents while waiting (RQ3). As illustrated in Section 4.2, users in both the *dynamic gain* and *static gain* conditions spent their time performing queries and clicking on documents in the interval while a slow query was processing. By the second task, these activities were reduced, but not in a statistically significant sense. We also showed that users in both of these conditions performed more slow queries in the second task and also waited more after performing these queries instead of clicking on documents or cancelling. This may indicate that users could still have been learning to use the system by the second task despite the training period and using the system for the first task. The reasons why users appeared to make more effective use of slow search by the second task require further study: the change could be due simply to their experience with the system in the first task, or it could be due to their increased awareness of the feature due to our explicitly asking users about their experience in using the "Work Harder" button between tasks. A future study may extend the training period to ensure that users are not only familiar with the system, but that they are also confident in predicting what the system will do. We also plan to do a longer-term online study in which users interact with the system for an extended period of time, which will help us to determine how long it takes for user behavior to stabilize and what it looks like when it does. Such a study will also help to understand usage in different scenarios without artificial constraints.

User search behavior did not significantly change with additional slow search capabilities (RQ4). Our analysis in Section 4.3.1 showed that user behavior in terms of search interactions was similar across conditions, with users in the two slow conditions making significant use of the "Work Harder" button. We observed that users in the baseline condition took longer to complete sessions, conducted more and longer queries, and clicked on fewer documents per query (Table 5). Users, by the end of the study, may have still not yet fully understood the capabilities of the system. However, these results may also indicate that slow search systems should cater to similar types of queries as current search systems, and support the kinds of interactions that users have grown accustomed to. A future study may serve to tease out these differences by looking at users who have become familiar with the system and users without such a system.

Users did not achieve higher final effectiveness with slow search, but showed evidence of higher efficiency (RQ5). For the tasks we evaluated, users achieved comparable final rewards across the three conditions, with the *baseline condition* showing slightly higher average reward, but overall differences were not statistically significant. However, as we note above, users obtained these rewards in less overall time for both slow search conditions compared to the baseline condition, giving some evidence of higher efficiency. In addition, users in the *dynamic gain* condition did indeed report that they noticed more of a difference in the improvement in search results than users in the *static gain* condition, and gave higher ratings for the usefulness of the progress bars. This may have been because it would have been clearer in the *dynamic gain* condition that the results were changing, and continued to change during the five minute duration. In contrast, users in the *static gain* condition may have not noticed the change between the unmodified and the modified results. Regardless, this shows that users are able to notice the difference when the results change, suggesting there

is some utility in having future systems expose progressive improvements in ranking to users.

We note that users found slightly fewer relevant documents on average in the *dynamic gain* condition compared to the *static gain* condition. One explanation for this difference is that in the *static gain* condition, the system inserts all of the same documents that would have been inserted in the *dynamic gain* condition, but in the lower quarter of the ranking. As a result, *static gain* users had the opportunity to gain access to these documents more quickly if they were willing to look for them in the ranking. In the *dynamic gain* condition, however, users had to wait longer to see the same highly relevant results, since the system begins with those documents at the bottom of the initial ranking and improves their position steadily over the course of five minutes.

While our dataset and corresponding analysis has allowed us to gain insight into the research questions we posed, we also recognize a number of limitations in our current study. Our findings, particularly that of RQ1, would be more robust with a larger sample of users. A future study in a more natural setting may also reduce experimental demand effects that might have influenced user behavior, and users' choices of tasks may have also affected their performance.

For future work, there are multiple possible avenues in exploring user interaction with slow search systems The 'Work Harder' button might be removed altogether and replaced with a background process that can automatically find and attempt to improve results for failed or abandoned search sessions. A user with low time pressure and a high degree of trust in a slow search system may submit a query to be processed in the background while performing non-search tasks, especially in the transition between devices, in which case supporting the examination of intermediate results or performing more queries in the interim would not be vital. In another instance, a user may use the system as as a supporting agent in a search task: the system would gather additional relevant documents and present them to the user as they continue to search. The participants' tendency in this study to continue searching shows that this is a useful capability to have.

The concept of "slowness" could additionally be applied to many different scenarios in which other aspects of retrieval 'quality' may be improved. This study focused on improving relevance for multi-attribute queries—a difficult class of queries for many existing systems, but in principle a system could also improve intrinsic diversity, employ crowdsourcing to augment algorithms, or summarize and organize results.

6. CONCLUSION

We reported on a user study that investigated five research questions about user interaction with a slow search system that offered users the option of running a 'slow' query in the background, showing progressive results in a sidebar. Using surveys and log data, we analyzed users who interacted with the system in one of three between-subjects conditions: a 'dynamic gain' condition that steadily improved search result quality of the optional slow query over the course of five minutes, a 'static gain' slow query that inserted relevant documents immediately with no additional ranking improvements over time, or a baseline condition giving conventional Web search results. Our findings suggest that users elected to perform slow search queries when given the opportunity. Additionally, we show that users are willing to wait for multi-attribute queries (RQ1), users will indeed wait for results when using slow search (RQ2), and users continued to search while waiting for results (RQ3). User behavior did not significantly change with additional slow search capabilities (RQ4), and users did not achieve higher final effectiveness with slow search, but did finish in less time (RQ5) on the tasks we evaluated. Followup studies may explore different mechanisms to improve quality for slow search and further investigate the nature of time-quality tradeoffs and user choice.

Acknowledgements. We thank Eytan Adar, Yan Chen, and the anonymous reviewers for their feedback.

References

[1] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein. Find it if you can: a game for modeling different types of web search success using interaction data. In *Proceedings of SIGIR 2011*, pages 345–354. ACM, 2011.

[2] C. Aperjis, B. A. Huberman, and F. Wu. Human speed-accuracy tradeoffs in search. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–10. IEEE, 2011.

[3] J. Brutlag. Speed matters for Google web search. *Google*, June 2009. URL http://bit.ly/1Oonkyz.

[4] S. Büttcher, C. L. Clarke, and I. Soboroff. The TREC 2006 Terabyte track. In *TREC 2006 Notebook*, volume 6, page 39. NIST Special Publication, 2006.

[5] J. Chen, S. Amershi, A. Dhananjay, and L. Subramanian. Comparing web interaction models in developing regions. In *Proceedings of the First ACM Symposium on Computing for Development*, page 6. ACM, 2010.

[6] C. L. Clarke and M. D. Smucker. Time well spent. In *Proceedings of the 5th Information Interaction in Context Symposium*, pages 205–214. ACM, 2014.

[7] A. Crescenzi, D. Kelly, and L. Azzopardi. Time pressure and system delays in information search. In *Proceedings of SIGIR 2015*, pages 767–770, New York, NY, USA, 2015. ACM.

[8] M. Dörk, P. Bennett, and R. Davies. Taking our sweet time to search. In *Proceedings of CHI 2013 Workshop on Changing Perspectives of Time in HCI*, 2013.

[9] R. Gould. Bootstrap hypothesis testing. *Stats 110A*, 2002. URL http://bit.ly/1ncz67z.

[10] Y. Kim, K. Collins-Thompson, and J. Teevan. Using the crowd to improve search result ranking and the search experience. *ACM TIST: Special Issue on the Crowd in Intelligent Systems*, 7(4):50, 2016.

[11] T. X. Liu, J. Yang, L. A. Adamic, and Y. Chen. Crowdsourcing with all-pay auctions: A field experiment on taskcn. *Proceedings of the American Society for Information Science and Technology*, 48(1):1–4, 2011.

[12] M. L. Mauldin. Retrieval performance in FERRET: A conceptual information retrieval system. In *Proceedings of SIGIR 1991*, pages 347–355. ACM, 1991.

[13] D. Maxwell and L. Azzopardi. Stuck in traffic: how temporal delays affect search behaviour. In *Proceedings of the 5th Information Interaction in Context Symposium*, pages 155–164. ACM, 2014.

[14] P. M. Napoli and J. A. Obar. The emerging mobile internet underclass: A critique of mobile internet access. *The Information Society*, 30(5):323–334, 2014.

[15] L. Poirier and L. Robinson. Informational balance: slow principles in the theory and practice of information behaviour. *Journal of Documentation*, 70(4):687–707, 2014. . URL http://dx.doi.org/10.1108/JD-08-2013-0111.

[16] J. Teevan, K. Collins-Thompson, R. W. White, S. T. Dumais, and Y. Kim. Slow search: Information retrieval without time constraints. In *Proceedings of HCIR 2013*, page 1. ACM, 2013.

[17] J. Teevan, K. Collins-Thompson, R. W. White, and S. Dumais. Slow search. *Communications of the ACM*, 57(8): 36–38, 2014.