Improving Searcher Models Using Mouse Cursor Activity

Jeff Huang
Information School
University of Washington
sigir2012@jeffhuang.com

Ryen W. White
Microsoft Research
Redmond, WA 98052
ryenw@microsoft.com

Georg Buscher
Microsoft Bing
Bellevue, WA 98004
georgbu@microsoft.com

Kuansan Wang
Microsoft Research
Redmond, WA 98052
kuansanw@microsoft.com

ABSTRACT

Web search components such as ranking and query suggestions analyze the user data provided in query and click logs. While this data is easy to collect and provides information about user behavior, it omits user interactions with the search engine that do not hit the server; these logs omit search data such as users' cursor movements. Just as clicks provide signals for relevance in search results, cursor hovering and scrolling can be additional implicit signals. In this work, we demonstrate a technique to extend models of the user's search result examination state to infer document relevance. We start by exploring recorded user interactions with the search results, both qualitatively and quantitatively. We find that cursor hovering and scrolling are signals telling us which search results were examined, and we use these interactions to reveal latent variables in searcher models to more accurately compute document attractiveness and satisfaction. Accuracy is evaluated by computing how well our model using these parameters can predict future clicks for a particular query. We are able to improve the click predictions compared to a basic searcher model for higher ranked search results using the additional log data.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

Keywords

cursor data, scrolling, searcher models, click data, user interactions, search result examination

1. INTRODUCTION

Web search engines allow users to search and retrieve relevant documents from billions of Web pages. The user issues a query and the search engine returns a list of results, generally ranked in order of relevance. The relevance of the results is based on a number of factors: how well a document matches the query, a document's reputation, and more recently, implicit feedback in the form of past behaviors for that query from many users. Records of these user behaviors are commonly sourced from query logs containing users' queries and the corresponding clicks, if any. Query logs are easy to collect, since they typically already exist as web server access logs without modification to the search engine.

Being able to compute relevance scores from implicit feedback allows a search engine to better rank the search results for future queries. Clicks (in the aggregate) provide a clear signal that users were attracted to the search result, and numerous studies have used click data in searcher models to infer relevance scores. These searcher models (e.g., [4, 7, 34]) track the user's state as they examine search results and use the observable events (e.g., clicks) to infer search result attractiveness and document relevance. However, query logs possess inherent limitations, some of which have been noted in the literature [11, 24]. They are unable to reveal actual user intent, provide little data about uncommon queries, and omit many interactions. Furthermore, they are uninformative for queries that have no clicks, i.e., abandoned queries.

In this paper, we introduce richer interaction data that can be used to supplement query and click data. This richer data comprises cursor movements and scrolling on the search engine results page (SERP), data which is not collected by commercial search engines but may be potentially useful. We believe cursor movements and scrolling can be additional implicit signals of relevance. These interactions can be captured at scale and can be recorded without disrupting the user, as shown in Huang et al. [21]. Actions such as cursor hovering and scrolling can be translated into implicit relevance feedback when overlaid on the SERP. In this work, we explore techniques to extend searcher models by using cursor hovering and scrolling activity to reveal latent variables in these searcher models to more accurately infer search result attractiveness and document relevance. As far as we are aware, this is the first study that explores the potential of cursor and scrolling interactions for use in searcher models.

Our primary contribution in this work is our study of extending a popular searcher model by adding hover and scroll data, informed by our analysis of replays of user interactions on the search results page. We find qualitative evidence that from a human observer's perspective, hovering and scrolling provide insight into the user's intentions and attention as they examine the SERP. We find that we

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

can improve searcher models by estimating whether a search result was viewed based on cursor hover and scroll behavior.

The remainder of the paper is structured as follows. In Section 2 we describe related work that characterizes cursor interactions with SERPs and searcher models. Section 3 describes the cursor data that we used in our study. We present an initial exploratory analysis of the data, which was useful in informing decisions about model features, in Section 4. In Section 5 we describe an extension to the searcher model using the cursor hover and scroll data, and present results of experiments using them in Section 6. We discuss the findings, their implications, and limitations of the method in Section 7, and conclude in Section 8.

2. RELATED WORK

Two lines of prior research are related to the work described in this paper: (i) studies characterizing how people interact with search result pages using their cursor, and (ii) searcher models primarily using click data.

2.1 Cursor Tracking on the Web and in Search

Buscher et al. investigated the use of gaze tracking to predict salient regions of Web pages [2] and the use of visual attention as implicit relevance feedback to personalize search [3]. To avoid eye-tracking studies, cursor tracking has been evaluated as an alternative to eye tracking for determining attention on the Web page. Initial studies established a close correspondence between gaze and cursor position [16, 21, 29, 30]. An early study by Chen et al. [5] showed that the distance between gaze and cursor was markedly shorter in regions of encountered pages to which users attended. More recent work has focused on the relationship between cursor and gaze on search tasks. In a study involving 32 subjects performing 16 search tasks each [29, 30], Rodden et al. identified a strong alignment between cursor and gaze positions. They found that the distance between cursor and gaze positions was longer along the x-axis than the y-axis, and was generally shorter when the cursor was placed over the search results. Guo and Agichtein [16] reported similar findings in a smaller study of ten subjects performing 20 search tasks each. Like Rodden et al., Guo and Agichtein noticed that distances along the x-axis tended to be longer than the distances along the y-axis. They could predict with 77% accuracy when gaze and cursor were strongly aligned using cursor features. Huang et al. [21] similarly found a correlation in cursor and gaze positions, and noted that the correlation was stronger on SERPs than on other Web pages. Huang et al. [20] determined when alignment occurred based on factors of time and cursor behavior. They showed that the alignment between cursor and gaze is stronger as the user is more active with the cursor.

Guo and Agichtein [14] captured cursor movements using a modified browser toolbar and found differences in cursor travel distances between informational and navigational queries. Furthermore, a decision tree could classify the query type using cursor movements more accurately than using clicks. Guo and Agichtein also used interactions such as cursor movement, hovers, and scrolling to accurately infer search intent and interest in search results [15]. They focused on automatically identifying a searcher's research or purchase intent based on features of the interaction. Rodden et al. [30] identified four general uses of the cursor in Web search—neglecting the cursor while reading, using the cursor as a reading aid (either horizontally or vertically), and using the cursor to mark interesting results. In a large-scale study, Huang et al. [21] conducted an analysis of cursor activity (including clicks on hyperlinks, clicks on non-hyperlinks, hover behavior) and its relation to Web search behavior. They also showed how cursor activity could be used to estimate the relevance of search results and to differentiate between good and bad search abandonment. Rather than tracking the cursor unobtrusively via cursor tracking, Lagun and Agichtein [25] presented a method to estimate gaze position by blurring the SERP and only revealing a region proximal to the cursor. They found that result viewing and clickthrough patterns agree closely with unrestricted viewing of results, as measured by eye-tracking equipment.

One line of related research has explored the use of cursor movements, clicks, and gaze as implicit indicators of interest on Web pages. In early work, Goecks and Shavlik modified a Web browser to record themselves browsing hundreds of Web pages [10]. They found that a neural network could predict variables such as the amount of cursor activity on the SERP, which they considered surrogate measurements of user interest. Claypool et al. [6] developed the "curious browser", a custom Web browser that recorded activity from 75 students browsing over 2,500 Web pages. They found that cursor travel time was a positive indicator of a Web page's relevance, but could only differentiate highly irrelevant Web pages. They also found that the number of clicks on a page did not correlate with its relevance. Hijikata [19] used client-side logging to monitor five subjects browsing a total of 120 Web pages. They recorded actions such as text tracing and link pointing using the cursor. The findings showed that these behaviors were good indicators for interesting regions of the Web page, around 1.5 times more effective than rudimentary term matching between the query and regions of the page. Shapira et al. [31] developed a special Web browser and recorded cursor activity from a small number of company employees browsing the Web. They found that the ratio of cursor movement to reading time was a better indicator of page quality than cursor travel distance and overall length of time that users spend on a page. Liu and Chung [26] recorded cursor activity from 28 students browsing the Web. They noticed patterns of cursor behaviors, including reading by tracing text. Their algorithms predicted users' cursor behaviors with 79% accuracy.

2.2 Searcher Models Using Click Data

Searcher models have been developed from two main hypotheses that are commonly used as assumptions in the models. Since users are biased towards clicking search results that are more highly ranked [23], the **examination hypothesis** is used to isolate a search result's attractiveness from its position. This hypothesis, originally formulated in Richardson et al. [28], states that the likelihood that a user will click on a search result is influenced only by 1) whether the user examined the search result and 2) its attractiveness. In other words, a user must examine a search result before potentially clicking that result. By making this assumption, a search result's attractiveness can be computed independent to its position in the ranking, i.e.,

$$P(C_i = 1) = P(E_i = 1)P(C_i = 1|E_i = 1),$$

where the term $P(E_i = 1)$ is the position bias and the term $P(C_i = 1|E_i = 1)$ is the search result's attractiveness.

To determine whether a user examined the search result, some searcher models draw from the **cascade hypothesis** [7] which dictates the search results a user has examined. The cascade hypothesis states that a user always examines search results sequentially and goes from top-to-bottom on the SERP. A user decides whether to click a result before examining the next result, preventing scenarios where the user returns to a higher-ranked search result after passing it by. Therefore, if users do not examine a particular search result, they will not examine any search results below it, i.e.,

$$P(E_1 = 1) = 1,$$
$$P(E_{i+1} = 1 | E_i = 0) = 0.$$

While the original cascade model stipulated that once a user clicked, they would no longer examine any search results, extensions of this hypothesis have sidestepped that assumption. The Dependent Click Model [13] allows for query sessions comprising multiple clicks: it possesses a parameter representing the probability that the clicked document is irrelevant and that the user returns to examining more search results. The Click Chain Model [12] and Dynamic Bayesian Network Model (DBN) [4] both extend this by adding an additional parameter representing the probability that a user abandons a query session without clicking, thus circumventing the cascade hypothesis's side effect that users are assumed to examine every search result in abandoned queries. Later in the paper, we will describe our extension to the DBN model, which we selected because of its popularity in the literature and its good performance. Additionally, the inference step could be simplified and the model itself was conducive to the inclusion of cursor data due to a separate examination state; these are two more reasons this model was particularly suited to our study.

Other searcher models such as the User Browsing Model [9] and the Partially-Observable Markov Model [34] avoid the cascade hypothesis entirely by allowing that the user jumps between search results non-sequentially in their examination. However, these models must possess more parameters representing the probabilities of transitions between search result positions. This makes the inference particularly difficult when there are fewer query sessions with clicks from which to learn.

The research presented in this paper intersects the two categories of prior work presented in this section by extending **searcher models** using **cursor tracking**.

3. CURSOR DATA

We recorded interaction data directly on the SERP of the Bing Web search engine. Log data were gathered over a period of 13 days between May 26, 2011 and June 7, 2011 during an external experiment on a small fraction of user traffic, primarily from English-speaking countries. We sampled by user, storing every query from each user in the experiment. In total, our data comprised around 1.8 million queries, averaging eight queries per searcher (median = 3 queries).

To record user interactions with the SERP at scale without the need to install any browser plugins, we used an efficient and scalable approach similar to that used by Huang et al. [21]. We implemented entirely JavaScript-based logging functions that were embedded into the HTML source code of the SERP. To obtain a detailed understanding of user interactions with the SERP, we deployed methods to measure and record a variety of interactions with the page as well as page characteristics, such as the layout of elements on the page. We recorded information on cursor movements, clicks, scrolling, as well as bounding boxes of certain components on the SERP and the browser's viewport size.

↓ Time	
	Query: lady gaga concert tickets
	Cursor moves from top to hover over 3rd search result
	Cursor pauses for 3 seconds
	Text "Tour Dates Only" is hovered with the cursor
	Cursor moves to the 4th search result, pausing 1s
	User scrolls to the 5th search result, pausing 3s
	Cursor returns to the 4th search result and clicks
	Click: Result 4 [http://gaga.com/tix/]

Figure 1: A user searches for "lady gaga concert tickets", examines the first page of results, and clicks the 4th search result. Typical query logs contain only query and click data (bold).

↓ Time	
	Query: flourless cake recipe
	Cursor moves to the bottom-right over whitespace
	No activity for 4 seconds
	Cursor moves over to the scrollbar
	User scrolls down half a screen
	No activity for 2 seconds
	User scrolls down half a page
	Cursor makes left-right motions over the 6th result
	User scrolls to the bottom of the page
	User quickly scrolls back up to the top
	Cursor moves to the top-right over the page
	User closes the page

Figure 2: A user searches for "flourless cake recipe" and scrolls to the bottom of the page, then scrolls back up and closes the window.

Figure 1 presents a fictional query along with the corresponding click data and client-side interactions. In this and many other cases, the cursor and scrolling data reveals more information about the user's intent. In the above scenario, the query logs only show that a query was issued, and that some time later, the 4th result was clicked. This is useful information, but the interaction data supplement this by showing that the user was active the whole time examining several results and that the user likely examined the 5th result and returned to the 4th result, indicating the 1–3 and 5th results may have been less relevant than the 4th result.

Figure 2 presents a fictional query that has no clicks. In typical query logs, the only recorded data would be the query text itself. The richer cursor and scrolling data here shows that the user did indeed scroll all the way to the bottom. We also see that the user paused to read through the results[1]. In this particular case, it seems reasonable to assume the user abandoned the query because they did not find what they were seeking. Thus, this query can be labeled as unsatisfying in a user-centered analysis of the logs.

[1]Query logs can compute the dwell time of a click, but only if another recorded event occurs after the click.

When logging any additional type of user interaction data beyond clickthrough, a tradeoff has to be made between: (i) level of detail (e.g., temporal and spatial resolution), (ii) the impact of any additional JavaScript code on page weight, page load time, and therefore the user experience, which can be sensitive to even small increases in load time, and (iii) the amount of data transferred (and hence bandwidth consumed) between the client and the remote server, as well as log volume created on the backend server. We negotiated a tradeoff between these dimensions by: (i) reasonably coarsening the log resolution, (ii) compressing the JavaScript code down to around three kilobytes, and (iii) compressing the log data as well as using a buffering approach for its transferal via Ajax to the backend server. We now describe in more detail the fields recorded in our log data and the methods used to record them.

3.1 Cursor Positions

The JavaScript function for logging cursor positions periodically checked the cursor's x- and y-coordinates within the Web page relative to its top-left corner of the page every 250 milliseconds. Whenever the cursor had been moved more than 8 pixels away from its previously logged position, its new coordinates were sent to the backend server. Eight pixels correspond to approximately half a line of text on the SERP. Since cursor tracking was relative to the document, we captured cursor alignment to SERP content regardless of how the user got to that position (e.g., by scrolling, or keyboard). Therefore, this approach was compatible with other behaviors such as scrolling or keyboard input. In previous cursor tracking studies, the cursor position was polled at particular time intervals, such as every 50 milliseconds (ms) [15] or every 100ms [29]. This is impractical at a large scale because of the large amount of data to transfer from the user's computer to the server. Our approach is similar to Huang et al. [21], who found that a 40ms pause provided a reasonable tradeoff between data quantity and granularity of the recorded events. However, we elected to record the cursor position every 250ms since we were sending data to the remote server every eight pixels of cursor movement, rather than every two seconds. As such we wanted to minimize the data gathered and transmitted to avoid adversely affecting the user experience with delays associated with log data capture and data transmission to the remote server.

3.2 Clicks

Clicks were recorded using the JavaScript onMouseDown event handling method. Thus, the backend server received log entries with location coordinates for every click, no matter whether the click occurred on a link or elsewhere on the page (even on white space containing no content that appears adjacent to or between SERP elements). In order to identify clicks on hyperlinks and differentiate them from clicks on inactive page elements, we also extracted and logged unique hyperlink identifiers that were embedded in the SERP, along with the corresponding URL of the hyperlink. The URL helped identify the actual search result because different query sessions could have different search results or the same search results ranked differently.

3.3 Scrolling

We also recorded the current scroll position, i.e., the y-coordinate of the uppermost visible pixel of the SERP in the browser viewport. This coordinate was checked three times per second and was recorded whenever it had changed more than 40 pixels compared to the last logged scroll position. Forty pixels correspond to the height of about two lines of text. From this coordinate we were able to gain a number of insights into scrolling activity, including whether the user was scrolling up or down, and the maximum scroll depth in the result page, in order to understand how far down the page the user had scrolled.

3.4 Page Layout

Simply logging the text of what was displayed on the SERP is insufficient for reconstructing its layout since SERPs vary per query (depending on what kinds of SERP elements are shown, etc.), font sizes, and other browser preferences. To reconstruct the exact SERP layout as it was rendered in the user's browser, we recorded the positions and sizes of certain regions. The specific regions in which we were interested in were as follows: (i) top and bottom search boxes, (ii) left rail and its contained related searches, search history, and query refinement areas, (iii) mainline results area and its contained result entries, including advertisements and instant answers, and (iv) right rail.

For each region bounding box, we determined and logged the coordinates of its upper left corner as well as its width and height in pixels. Using this information, we could later map the positions of cursor positions and clicks to specific regions of the page. The recorded data also contained the size of the user's Web browser window, which combined with the scrolling activity could deduce information about the parts of the page that were visible at a particular time during the query session.

Figure 3 presents a screenshot of the page layout of a reconstructed SERP taken from a query session replay. Important components are outlined in light blue boxes; the user's cursor position is shown as a gray pointer; and the green area represents parts of the Web page not visible in the browser window on the user's screen at the current time.

4. EXPLORATORY ANALYSES

Before constructing any searcher models, we wanted to obtain a deeper understanding of the recorded cursor and scrolling activity since this type of data was relatively unexplored. This included both a qualitative perspective and a more traditional quantitative analysis of the data. The findings here inform the approach we take in enhancing the searcher model.

4.1 Qualitative Observations

We began by reconstructing the SERP layout from the recorded logs, and developed a tool to replay the entire sequence of cursor interactions on the page in great detail. This included an outline highlighting the viewable area of the Web page (based on the dimensions of the Web browser viewport), since this would change according to the users' screen resolution and their scrolling. One of the authors then visually investigated a random sample of over a hundred replays of interactions on the search result pages made by real users. During the replays, the author put himself in the users' place to determine their intent. His judgments of their intents were informed by the cursor behaviors described in prior literature. These qualitative observations were a rich

Figure 3: The reconstructed SERP during a query session replay. Light blue boxes outline important components, a grey pointer represents the user's cursor position, and the green area overlays off-screen portions of the Web page. The number in the top-left is the time elapsed since the start of the query session.

way of truly understanding the data and provided a number of key insights that were difficult to quantify.

First, we saw that many users could only view a small portion of the Web page initially, which only displayed ads or an "Answer" element (such as the PhonebookAnswer, which shows local results and contact information, in Figure 3); these users would often scroll down a bit to view at least a couple of search results. The time spent pausing after a scroll suggested that they indeed examined those newly revealed search results. We were less confident that the user had examined all the visible search results if they did not scroll, since they often clicked a link or abandoned the query immediately after the page loaded.

Second, while we could not see where the user was actually looking, the cursor would commonly move around the page from top-to-bottom while hovering over particular areas, and then move to the scrollbar to reveal more search results, corresponding nicely with the *linear traversal hypothesis* [23]. This behavior seemed to suggest that whenever a user hovered over a search result, they had at least examined that result and the search results above it.

Third, we observed some users moving their cursor back-and-forth horizontally which we believed to be them following the cursor as they read text; some users would do this quite frequently in a single query session. This corroborates previous work that observed this behavior in lab settings [20, 26, 30], and suggests that this behavior is specific to individual users.

Finally, we observed many sessions in which the user would move their cursor quickly and directly from the search box to the first search result, without scrolling down to view any of the lower-ranked search results. This happened often in navigational queries, so this provoked the question whether interaction data would be more or less useful in navigational queries; we explore this later in Section 6.2.2.

4.2 Quantitative Summary

As described in an earlier section, the raw interaction events comprised cursor positions, clicks, window scrolls, and page layout. Statistics specific to the different types of interaction data logged include:

Cursor: Users hovered over multiple search result captions (mean = 2.6, median = 2), even for navigational queries when it was clear that a single search result will suffice. This pattern of behavior has been observed in previous studies of eye tracking [8], as well as previous work on large-scale cursor tracking [21].

Clicks: Mouse clicks were collected regardless of whether they were navigational clicks (clicks on a hyperlink) or interactive clicks on controls in the page (17.1% of clicks). 64.7% of all clicks were hyperlink clicks and 35.2% were non-hyperlink clicks, including re-query events (estimated from clicks on the upper or lower search boxes) totaling around 11% of all queries.

Scrolling: Window scrolling is a client-side interaction that is rarely captured in the context of Web search. Of the queries in our set, 29.7% contained at least one scroll event. 61.8% of logged interaction sequences for a query ended on a downwards scroll. As expected, there were more downward scrolls than upward scrolls, and the majority of scrolled queries (54.8%) comprised only downward scrolls. This suggests that most queries do not result in the user returning to the top of the SERP to examine search results that may be hidden following scrolling.

Figure 4 contrasts queries in which the user has scrolled with queries where the user did not scroll. As expected,

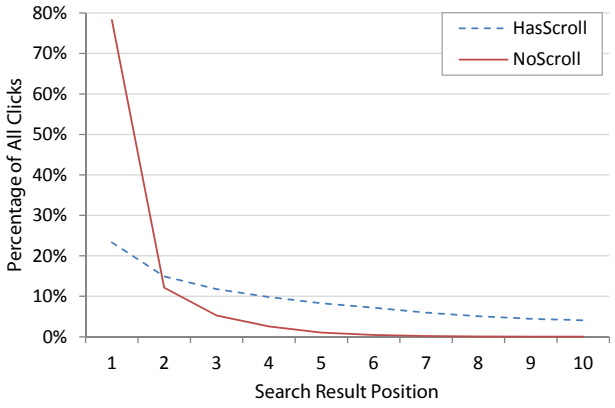

Figure 4: The click distributions for cases in which the user does not scroll, and when there is at least one scroll event during the query session. The distribution is heavily skewed when there is no scrolling, and almost linear when the user has scrolled.

when the user has not scrolled, the click distribution is significantly more skewed towards higher-ranked search results. The difference between the click distributions is quite drastic: in query sessions where the user has scrolled, search results in lower-ranked positions have a fairly good chance of being selected. This is consistent with the examination hypothesis mentioned earlier, and supports a hypothesis that scrolling towards a set of search results makes it likely the user has examined those results.

The observations informed the following two hypotheses that we wanted to apply to searcher models: 1) when a user scrolled down, they have already examined the search results in their viewing area and those above it, and 2) when a user hovered over a search result, they have examined it and the search results above it. In the next section, we validate these hypotheses by implementing them in a traditional searcher model that uses only clicks.

5. EXTENDING A SEARCHER MODEL

Searcher models are structured based on theoretical knowledge of a user's search examination process. Their internal parameters are inferred from observable data, which in turn can be applied to compute relevance label scores for search results. Label scores are position-independent and computed from the model for every query × search result. The search results can then be re-ranked using these labels for future occurrences of the same query. Thus, a better searcher model can compute more accurate relevance labels for search ranking.

We replicate the Dynamic Bayesian Network (DBN) model [4] as the baseline model to which we compare against. The DBN model is the most cited searcher model since the Cascade Model (which compared favorably to all models before it). It compares favorably to the Cascade Model [4, 36], and fares well compared to other models (e.g., [36, 37]). Thus, the DBN model serves as a solid baseline for our purposes; it provides an example searcher model in which we can focus on whether cursor data can improve a model, rather than outperforming all models, i.e., more of an analysis of the value of cursor data than strictly model development.

The DBN model is a graphical model where the nodes represent states of the user examining the search results. The model is represented formally as follows:

- E_i: the user examined the search result
- C_i: the user clicked the search result
- A_i: the search result attracted the user
- S_i: the landing page satisfied the user (relevance)

$$
\begin{aligned}
A_i = 1, E_1 = 1 &\Leftrightarrow C_i = 1 \\
P(A_i = 1) &= a_u \\
P(S_i = 1 | C_i = 1) &= s_u \\
C_i = 0 &\Rightarrow S_i = 0 \\
S_i = 1 &\Rightarrow E_{i+1} = 0 \\
P(E_{i+1} = 1 | E_i = 1, S_i = 0) &= \lambda \\
E_i = 0 &\Rightarrow E_{i+1} = 0
\end{aligned}
$$

In this model, users examine search results from top to bottom, assessing at each result whether or not it is attractive enough to click (cascade hypothesis), which depends only on the attractiveness of the link a_u (examination hypothesis). If they click, there is some probability s_u they will be satisfied and stop the search process; if they are not satisfied, they either return to the search results page to examine the next search result with probability λ, or abandon the search. Figure 5 enumerates the user states and decisions in the DBN model; the "hover above and scroll towards" state was a new observable event generated from the cursor data. Examining a search result could emit this event, but the events are not a precondition of examining a result.

During the exploratory analysis phase, we saw that scrolling towards a set of search results led to a higher chance of those results being examined. Additionally, hovering over a search result similarly suggested that result and those above it were examined. These assumptions were incorporated into the searcher model by adding the following constraint:

$$(\exists h \in H : i \leq h) \vee i \in V \Rightarrow P(E_i = 1),$$

where H is the set of search result positions the user hovered over, and V is the set of all search results shown when the user scrolled. These events would reveal that the user had examined the search results, but a user examining a search result would not necessarily emit a corresponding hover or scroll event.

We reimplement the DBN model with $\lambda = 1$, labeled *Algorithm 1* in Chapelle and Zhang [4], to simplify the inference of latent variables. Then we incorporate the additional examination constraint to validate the observations in the exploratory analyses.

6. EXPERIMENT

In this section, we describe an experiment comparing the baseline DBN model with the modified DBN model incorporating cursor data for computing relevance labels. We define the click perplexity metric used to evaluate the model and report the experiment and results.

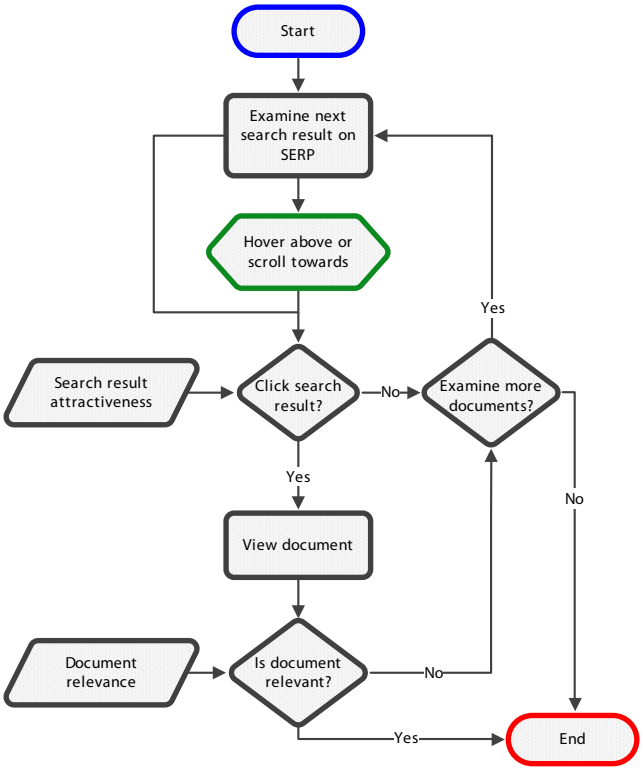

Figure 5: Flow diagram of the users' states in the modified Dynamic Bayesian Network model enhanced with cursor hover and scrolling data. The hexagon represents the new potentially observable events that can be captured in interaction logs.

6.1 Evaluation

While we cannot evaluate unobserved events in a searcher model, we can test how well the model predicts clicks, the observable events. Click perplexity was evaluated in a number of other searcher model studies [9, 12, 35, 36, 37] as a measure of predicting click-through rates. Our evaluation used a similar methodology as the past studies in literature: query sessions were divided evenly into training and test sets, each comprising at least 5 query sessions; we only accepted one query session from each user for a particular query to prevent a small number of users from dominating the data. There were 7,341 queries in which at least 10 unique users issued the query; this filtered out queries with insufficient data.

We compared *the DBN model with only click data*, as it is implemented in the literature, with *the DBN model with click and cursor data* from our logs. These data were used to train the searcher model, and the trained model was used to predict clicks in the test set[2]. Better prediction of clicks in the test set implies that the searcher model (and its inferred parameters) better reflects the result examination process. The click perplexity quantifies how much the test data surprises the trained model; it is computed for each combination of query and position as,

$$p_i = 2^{-\frac{1}{N}\sum_{n=1}^{N}(C_i^n log_2 q_i^n + (1-C_i^n) log_2 (1-q_i^n))}$$

where p_i is the perplexity in the ith position, N is the number of links, and q_i^n is the predicted click probability for the nth query session. The exponent represents the cross-entropy estimated from a probability distribution. The lowest perplexity is 1, meaning the trained model perfectly predicted the test data, while a larger perplexity means the model was less accurate in predicting the test data. Because the lower bound of the perplexity depends on the click-through rate of the query, the perplexity varies substantially depending on the position of the search result. Therefore, we computed a separate perplexity value for each of the top ten rank positions.

6.2 Results

We now report on the results of our experiments. Figure 6 shows the computed perplexities for each position on the SERP. The baseline searcher model comprising only click data did not perform as well as the searcher model incorporating both click and scrolling data. The latter model was further improved when incorporating hover data as well, although the improvement was small since there is an overlap between the search results the user scrolls to, and the search results at or above that which the user hovers above.

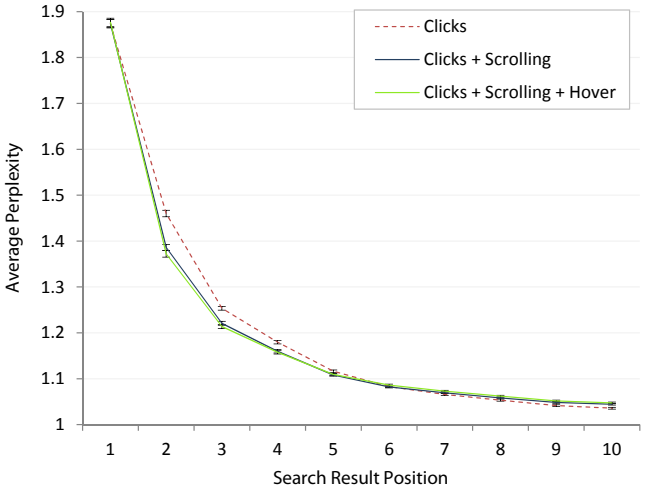

Figure 6: A comparison between 3 variations of the DBN model: 1) the baseline model using only click data, 2) a modified model also incorporating scrolling data, and 3) a modified model incorporating clicks, scrolling, and hover data. Lower click perplexity represents better prediction. Error bars represent the standard error of the mean.

It is clear that the additional cursor data improves relevance labels for search results in positions 2–5, but the prediction is slightly worse for search results in positions 6–10. However, users benefit more from the better prediction accuracy for the search results in higher positions since users consider them more important, so we believe there is an overall improvement. More accurate click predictions signify that the document relevance and search result at-

[2]The cursor data was only used for training the searcher model, and not for testing, i.e., we did not try to predict cursor movements and scrolling.

tractiveness labels are more likely to be close to true objective values of these parameters. For search results in positions 6–10, there appears to be a slight decrease in accuracy for predicted clicks in the models incorporating cursor data. We are unsure whether this is due to overfitting or noise in the data. The difference in perplexity for using clicks only compared to clicks + scrolling was significant at the $t(7340) \geq 8.26, p < 0.001$ level at positions 2–5. The difference in perplexity for using clicks + scrolling data compared to clicks + scrolling + hover data was significant at the $t(7340) \geq 3.07, p < 0.002$ level at positions 2 and 3, as the perplexity drops from 1.46 to 1.37 in position 2 and from 1.25 to 1.21 in position 3. Both differences were significant even after applying the Bonferroni correction.

We performed two additional sets of analyses of our results designed to better understand the nature of our gains. We studied the distribution of gains and losses across the top 10 rank positions. We also studied the effect of query types (navigational versus non-navigational) on our click prediction accuracy. We now report the findings of each.

6.2.1 *Gains and Losses*

For each of the 7,341 queries in our set, and for each rank position, we determined whether the DBN model with full cursor data (clicks + scrolling + hovers) outperformed the model with only clicks. We then computed the percentage of queries for which the model with cursor data attained a perplexity value above, below, or equal to the clicks-only model. Note that to simplify the analysis, we ignored the magnitude of difference between the models for a query. Figure 7 highlights the change in click prediction performance for different positions.

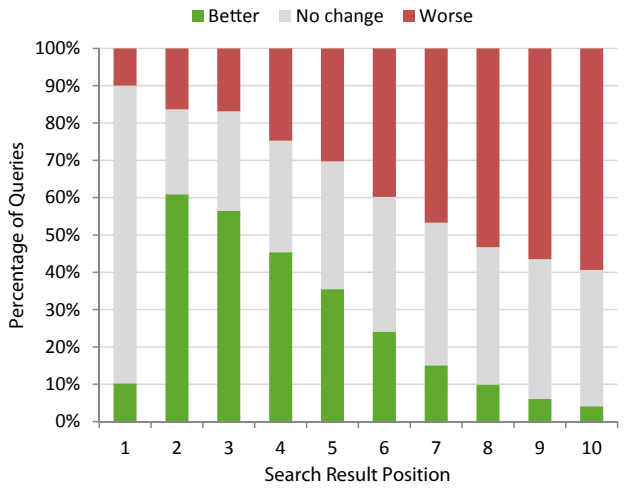

Figure 7: The percentage of queries whose click perplexity was helped or hurt by adding cursor hover and scroll data to the DBN searcher model.

The findings reveal a number of things. First, there was generally no change for the first rank position. Second, the biggest gains from the cursor model came at position 2, where over 60% of queries were benefited by using the additional cursor data. Third, the fraction of queries for which the cursor model performed best decreased fairly rapidly with rank, ending with only 5% of queries benefiting from cursor data for at rank position 10. One possible explanation is that since users only scroll for a third of queries, we possess less hover and scrolling evidence from which to learn user preferences at lower ranks. Finally, for the second rank position onwards, the fraction of queries for which there is no change remains fairly constant in the 20–35% range (increasing gradually with rank). Although there were no immediately noticeable patterns in those queries, they need more analysis since they may represent an opportunity for additional gains, especially further down the ranking, where they represent a sizeable fraction of queries.

6.2.2 *Effect of Query Intent*

We also segmented the queries into navigational and non-navigational query types to see if performance differences existed between different query intents in our models. Teevan et al. adopted a metric of click entropy as a threshold to classify navigational and non-navigational query types [33]. They showed that navigational queries classified in this manner exhibited differences in user behavior. We used the same method to segment the queries in our set, and identified 2,407 navigational queries and 3,509 non-navigational queries. The remaining 1,426 queries had a click entropy value between the navigational and non-navigational thresholds, and were removed from this part of the analysis.

Originally, we hypothesized that cursor and scrolling data may be less useful for navigational queries, since the clicks can be determined more easily. However, our findings (summarized in Figure 8) showed that click prediction improved when cursor data was added in both navigational and non-navigational queries, particularly in higher-ranked positions; the click predictions were almost evenly improved in navigational queries as in non-navigational queries. Differences in click perplexity between all four combinations—navigational and non-navigational queries, with and without cursor data—were statistically significant at the $t(5915) \geq 7.63, p < .001$ level after applying the Bonferroni correction. We also inspected the individual queries for which the additional cursor data helped and hurt click prediction; the queries exhibited no discernible pattern. It appears that the improvements were uniform and not particular to one type of query.

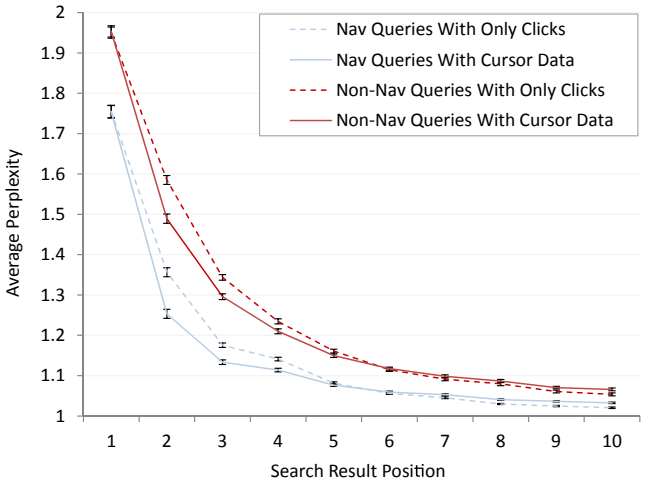

Figure 8: Comparison in click prediction between navigational and non-navigational queries, with and without cursor data. Lower click perplexity represents better prediction.

7. DISCUSSION

Initial exploratory findings have given us key insights. The replays and aggregate analyses have suggested that users seem to examine the search results they scroll towards, and the search results they hover over. We were able to convert these findings into constraints in the DBN model, a popular searcher model. An experiment comparing the DBN model with only click data to the DBN model with hover and scrolling as examination signals showed that the additional data helped predict clicks. After further disentangling the queries into navigational and non-navigational query intents, we found little difference between the two types of queries, and visually inspecting the queries themselves sustained our confidence that better relevance labels can be inferred across all types of queries. Our experiments have shown that by augmenting query logs with richer interaction data attainable at scale (in our case, cursor hover and scrolling), we can realize improvements in existing searcher models. The better searcher models can compute more accurate attractiveness and relevance labels for pairs of query $\times$ search result, which in turn lead to position-independent search result scores that can be ranked.

This work has implications for the design of search systems. Search companies have been processing query logs for some time now, and the amount of query logs that can be collected is limited—they cannot obtain more query and click data from a fixed number of users. However, companies can scalably and efficiently collect more search data such as cursor movements and scrolling. These can be used to improve searcher models by generating more accurate attractiveness and relevance labels for search results. Search engineers can then leverage these labels to supplement existing scoring factors for ranking the search results, such as document and link analyses algorithms. Additionally, the labels can be used for analytics and to answer questions such as "Which search results are highly attractive but are not actually relevant?" or "For which queries are users likely to abandon because of unattractive search results?"

In other data-intensive computer science areas such as natural language processing and data mining, there has been evidence that collecting and mining additional data can be more useful than improving algorithms. A study by Banko and Brill [1] compared various learning algorithms for disambiguating natural language. They showed that increasing the amount of data by 10-fold would make even the worst algorithm better than the best algorithm. A recent article published by Google researchers about "The Unreasonable Effectiveness of Data" [17] highlights the power of web-scale data for machine translation. Rajaraman presented anecdotal evidence to argue that, "adding more, independent data usually beats out designing ever-better algorithms to analyze an existing data set" in an article titled "More data usually beats better algorithms" [27]. A similar phenomenon may be occurring in Web search, where growing dependence on user-generated search logs makes it more important going forward to collect more independent data. Cursor interactions and scrolling activity are new types of data that are more difficult to interpret but can supplement existing query logs. We may be able to improve ranking in the future, as well as user assistance interface features, by modeling interaction on different interface features.

We acknowledge several limitations in our study which may be addressed in future work. Our searcher model was fairly simple since our goal was to demonstrate the value of new interaction data rather than develop the best overall searcher model. Our searcher model does not take advantage of several metrics that have been shown to improve searcher models: click order [22], the duration between clicks [18], and temporally changing relevance of search results [32]. Another limitation is that SERPs in commercial search engines are becoming richer and more interactive. While SERPs have greatly evolved from 10 blue links, features such as *hover preview* and interactive customized displays for many types of search results have changed how users behave. Finally, the extensions we made to the searcher model were informed by an exploratory analysis of replays of user sessions; the results show that there can be benefit to incorporating non-click data in searcher models, but we have not investigated what other forms of data may similarly improve them. These limitations can be addressed in future work, perhaps with searcher models beyond the DBN model, incorporating previously mentioned factors such as click order, duration, and temporal relevance.

8. CONCLUSIONS

We have conducted exploratory analyses of recorded user interactions on the SERP with both qualitative and quantitative approaches. These analyses suggested that scrolling towards a set of search results and hovering over search results are related to whether a user examined them. By adding these interactions as constraints in a popular searcher model, we were able to infer attractiveness and relevance labels for search results. We evaluated newly generated labels from the revised searcher model by comparing it to the searcher model using only click data. The searcher model with the additional cursor data was able to better predict future clicks, implying that the labels are more accurate.

Click data in query logs are but one source of user behavior data that can be used for ranking. We have shown that collecting richer data in the form of client-side interactions can be useful. Cursor and scrolling activity is only a subset of recordable interaction data, and using new independent data may be a fruitful avenue to pursue. As we previously referenced, researchers in data mining and machine translation have found that simply adding more data can result in an order of magnitude of greater improvement in the system than making incremental improvements to the processing algorithms. We have yet to see if richer data will make such an impact in the application of ranking from searcher models, but our experiment suggests the possibility.

9. ACKNOWLEDGMENTS

The authors thanks Katherine Ye for editing earlier drafts of the paper. The first author of the paper is supported by a Google Research Award and Facebook Fellowship.

10. REFERENCES

[1] M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of ACL*, pp. 26–33, 2001.

[2] G. Buscher, E. Cutrell, and M. R. Morris. What do you see when you're surfing?: using eye tracking to predict salient regions of web pages. In *Proceedings of CHI*, pp. 21–30, 2009.

[3] G. Buscher, A. Dengel, and L. van Elst. Eye movements as implicit relevance feedback. In *CHI Extended Abstracts*, pp. 2991–2996, 2008.

[4] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of WWW*, pp. 1–10, 2009.

[5] M. C. Chen, J. R. Anderson, and M. H. Sohn. What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In *CHI Extended Abstracts*, pp. 281–282, 2001.

[6] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *Proceedings of IUI*, pp. 33–40, 2001.

[7] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of WSDM*, pp. 87–94, 2008.

[8] E. Cutrell and Z. Guan. What are you looking for?: an eye-tracking study of information usage in web search. In *Proceedings of CHI*, pp. 407–416, 2007.

[9] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of SIGIR*, pp. 331–338, 2008.

[10] J. Goecks and J. Shavlik. Learning users' interests by unobtrusively observing their normal behavior. In *Proceedings of IUI*, pp. 129–132, 2000.

[11] C. Grimes, D. Tang, and D. M. Russell. Query logs alone are not enough. In *WWW Workshop on Query Log Analysis*, 2007.

[12] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos. Click chain model in web search. In *Proceedings of WWW*, pp. 11–20, 2009.

[13] F. Guo, C. Liu, and Y. M. Wang. Efficient multiple-click models in web search. In *Proceedings of WSDM*, pp. 124–131, 2009.

[14] Q. Guo and E. Agichtein. Exploring mouse movements for inferring query intent. In *Proceedings of SIGIR*, pp. 707–708, 2008.

[15] Q. Guo and E. Agichtein. Ready to buy or just browsing?: detecting web searcher goals from interaction data. In *Proceedings of SIGIR*, pp. 130–137, 2010.

[16] Q. Guo and E. Agichtein. Towards predicting web searcher gaze position from mouse movements. In *Proceedings of CHI*, pp. 3601–3606, 2010.

[17] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8 –12, 2009.

[18] Y. He and K. Wang. Inferring search behaviors using partially observable markov model with duration (pomd). In *Proceedings of WSDM*, pp. 415–424, 2011.

[19] Y. Hijikata. Implicit user profiling for on demand relevance feedback. In *Proceedings of IUI*, pp. 198–205, 2004.

[20] J. Huang, R. White, and G. Buscher. User see, user point: gaze and cursor alignment in web search. In *Proceedings of CHI*, pp. 1341–1350, 2012.

[21] J. Huang, R. W. White, and S. Dumais. No clicks, no problem: using cursor movements to understand and improve search. In *Proceedings of CHI*, pp. 1225–1234, 2011.

[22] S. Ji, K. Zhou, C. Liao, Z. Zheng, G.-R. Xue, O. Chapelle, G. Sun, and H. Zha. Global ranking by exploiting user clicks. In *Proceedings of SIGIR*, pp. 35–42, 2009.

[23] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of SIGIR*, pp. 154–161, 2005.

[24] R. Kohavi, R. Longbotham, D. Sommerfield, and R. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, 2009.

[25] D. Lagun and E. Agichtein. Viewser: Enabling large-scale remote user studies of web search examination and interaction. In *Proceedings of SIGIR*, pp. 365–374, 2011.

[26] C.-C. Liu and C.-W. Chung. Detecting mouse movement with repeated visit patterns for retrieving noticed knowledge components on web pages. *IEICE - Trans. Inf. Syst.*, E90-D(10):1687–1696, October 2007.

[27] A. Rajaraman. More data usually beats better algorithms. `http://anand.typepad.com/datawocky/2008/03/more-data-usual.html`, 2008.

[28] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of WWW*, pp. 521–530, 2007.

[29] K. Rodden and X. Fu. Exploring how mouse movements relate to eye movements on web search results pages. In *SIGIR Workshop on Web Information Seeking and Interaction*, pp. 29–32, 2007.

[30] K. Rodden, X. Fu, A. Aula, and I. Spiro. Eye-mouse coordination patterns on web search results pages. In *CHI Extended Abstracts*, pp. 2997–3002, 2008.

[31] B. Shapira, M. Taieb-Maimon, and A. Moskowitz. Study of the usefulness of known and new implicit indicators and their optimal combination for accurate inference of users interests. In *Proceedings of SAC*, pp. 1118–1119, 2006.

[32] R. Srikant, S. Basu, N. Wang, and D. Pregibon. User browsing models: relevance versus examination. In *Proceedings of KDD*, pp. 223–232, 2010.

[33] J. Teevan, S. T. Dumais, and D. J. Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In *Proceedings of SIGIR*, pp. 163–170, 2008.

[34] K. Wang, N. Gloy, and X. Li. Inferring search behaviors using partially observable markov (pom) model. In *Proceedings of WSDM*, pp. 211–220, 2010.

[35] Y. Yang, X. Shu, and W. Liu. A probability click tracking model analysis of web search results. In *Proceedings of ICONIP*, pp. 322–329, 2010.

[36] Y. Zhang, D. Wang, G. Wang, W. Chen, Z. Zhang, B. Hu, and L. Zhang. Learning click models via probit bayesian inference. In *Proceedings of CIKM*, pp. 439–448, 2010.

[37] F. Zhong, D. Wang, G. Wang, W. Chen, Y. Zhang, Z. Chen, and H. Wang. Incorporating post-click behaviors into a click model. In *Proceedings of SIGIR*, pp. 355–362, 2010.