User Interaction Sequences for Search Satisfaction Prediction

Rishabh Mehrotra*
University College London
London, United Kingdom
r.mehrotra@cs.ucl.ac.uk

Imed Zitouni[1], Ahmed Hassan Awadallah[2],
Ahmed El Kholy[1], Madian Khabsa[2]
[1]Microsoft, [2]Microsoft Research
Redmond, WA, USA
{izitouni,hassanam,ahkhol,makhab}@microsoft.com

ABSTRACT

Detecting and understanding implicit measures of user satisfaction are essential for meaningful experimentation aimed at enhancing web search quality. While most existing studies on satisfaction prediction rely on users' click activity and query reformulation behavior, often such signals are not available for all search sessions and as a result, not useful in predicting satisfaction. On the other hand, user interaction data (such as mouse cursor movement) is far richer than just click data and can provide useful signals for predicting user satisfaction. In this work, we focus on considering holistic view of user interaction with the search engine result page (SERP) and construct detailed *universal interaction sequences* of their activity. We propose novel ways of leveraging the universal interaction sequences to automatically extract informative, interpretable subsequences. In addition to extracting frequent, discriminatory and interleaved subsequences, we propose a Hawkes process model to incorporate temporal aspects of user interaction. Through extensive experimentation we show that encoding the extracted subsequences as features enables us to achieve significant improvements in predicting user satisfaction. We additionally present an analysis of the correlation between various subsequences and user satisfaction. Finally, we demonstrate the usefulness of the proposed approach in covering abandonment cases. Our findings provide a valuable tool for fine-grained analysis of user interaction behavior for metric development.

KEYWORDS

satisfaction; interaction sequences; subsequences; hawkes process

1 INTRODUCTION

As increasingly larger proportions of users rely on search engine interactions to satisfy their information needs and as retrieval systems advance, the need for good evaluation metrics increases. As such, developing better understanding of how users interact with search engines becomes increasingly important for improving user's search experience. Since obtaining explicit feedback from users is prohibitively expensive and challenging to implement in real-world retrieval systems, commercial search engines have exploited implicit feedback signals derived from user activity. While users interact with a search engine, they leave behind fine grained traces of interaction patterns. These interaction patterns contain valuable information, which could be useful for predicting user satisfaction as well as developing metrics for search engine evaluation to assist rapid experimentation.

Recent work has extensively studied implicit feedback measures (e.g., mouse scrolling, gaze tracking, physiological signals, etc.), and verified their effectiveness in predicting search satisfaction (or dissatisfaction) [11, 13, 21, 31]. Compared to coarser models of clicks alone, such user interactions provide additional insight into searchers' behavior. Despite recent progress in utilizing such implicit interaction signals, developing metrics around these signals often involves intensive manual effort [11, 20] to gain insights about the data, and to make use of it for practical applications. Also, existing models around utilizing detailed user interaction data are not interpretable and require deep investigation to extract meaningful insights. For example, popular approaches like visualizing areas of high cursor activity via heatmaps require manual inspection and lacks detail about sequences of user activity.

In this work, we focus on considering a holistic view of user interaction with the search engine result page (SERP) and construct detailed *universal interaction sequences* of their activity. We propose novel ways of leveraging the universal interaction timelines to automatically extract informative, interpretable subsequences. In addition to proposing ways to extract frequent and discriminative subsequences, we propose an interleaved subsequence extraction method which is able to jointly leverage discriminatory and frequent aspects of subsequences to extract subsequences which allow for noisy actions to interleave amidst more meaningful and informative signals.

Further, while the subsequences extracted are able to capture informative user interactions, they ignore the temporal spread of actions. To this end, we incorporate time in a principled manner when modeling user interaction sequences and leverage recent advancements in point process models to do so. We propose a Hawkes process formulation of interaction sequences which enables us to weigh the extracted subsequences based on the inter-activity times. We present a large scale evaluation of the proposed approach using crowdsourced judgments as well as weakly labeled data and demonstrate that including the proposed subsequences significantly improves user satisfaction prediction performance. We additionally show that the proposed techniques work in abandonment cases too and can be further explored to develop sophisticated methods for detecting and predicting good abandonment. Our findings provide a valuable tool for fine-grained analysis of user interaction behavior for developing metrics and gauging user satisfaction.

*Work conducted while at Microsoft Research.

SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan
© 2017 ACM. 978-1-4503-5022-8/17/08…$15.00
DOI: 10.1145/3077136.3080833

2 RELATED WORK

As search systems become more sophisticated, developing techniques for evaluating their performance plays an increasingly pivotal role. The current research builds upon and advances research in three directions: (i) User Satisfaction prediction, (ii) Implicit feedback, and (iii) Subsequence mining.

User Satisfaction

The concept of satisfaction was first introduced in IR researches in 1970s according to Su et al. [38]. A recent definition states that ”satisfaction can be understood as the fulfillment of a specified desire or goal” [24]. However, search satisfaction itself is a subjective construct and is difficult to measure. Some existing studies tried to collect satisfaction feedback from users directly. For example, Guo *et al.* 's work [13] on predicting Web search success and Feild et al.'s work [6] on predicting searcher frustration were both based on searchers' self-reported explicit judgments. Differently, other researchers employed external assessors to restore the users' search experience and make annotations according to their own opinions. For example, Guo et al.fis work [14] on predicting query performance was based on this kind of annotations. Recently, simplistic user feedback signals have been used to gauge user satisfaction. For instance, it has previously been shown that clicks followed by long dwell times are correlated with satisfaction [9]. Hassan et al. [16] propose to use query reformulation as a negative indicator of search success and thus satisfaction and show how an approach based on query features outperforms an approach based on click features, with the best performance being achieved by a combination of the two. Kim et al. [25] consider three measures of dwell time and evaluate their use in detecting search satisfaction. Lagun *et al.*[29] consider scroll and viewport features for predicting satisfaction in mobile search.

Subsequence & Timeseries mining

Sequential pattern mining was first introduced by Agrawal *et al.* [2] in the context of market basket analysis, which led to a number of other algorithms for frequent sequence mining. Frequent sequence mining suffers from pattern explosion: a huge number of highly redundant frequent sequences are retrieved if the given minimum support threshold is too low. We refer the interested reader to Chapter 11 of [1] for a survey of frequent sequence mining algorithms. There has also been some existing research on probabilistic models for sequences, especially using Markov models. Gwadera et al. [10] use a variable order Markov model to identify statistically significant sequences. More recently, Fowkes *et al.* [8] proposed a subsequence interleaving model based on a probabilistic model of the sequence database.

Gestures for Relevance & Satisfaction

Traditional evaluation techniques relied on classical methodologies that use query sets and relevance judgments. More recently, a number of different interaction behaviors have been taken into consideration in the prediction of search user satisfactions including both coarse-grained features (e.g. clickthrough based features in [14]) and fine-grained ones (e.g. cursor position and scrolling speed in [13]). Mouse movement information like scroll and hover have proven to be valuable signals in inferring user behavior and preferences [11, 21, 35], search intent [12], search examination [32] and predicting result relevance [22]. However, none of these studies tried to extract mouse movement patterns and adopt them to predict search satisfaction. Arapakis et al. [3] extracted mouse gestures to measure within-content engagement. Lagun et al. [28] introduced the concept of frequent cursor subsequences (namely motifs) in the estimation of result relevance. Different from their work, we focus on how to extract informative subsequences from user interactions to help predict search satisfaction a search impression level instead of result level.

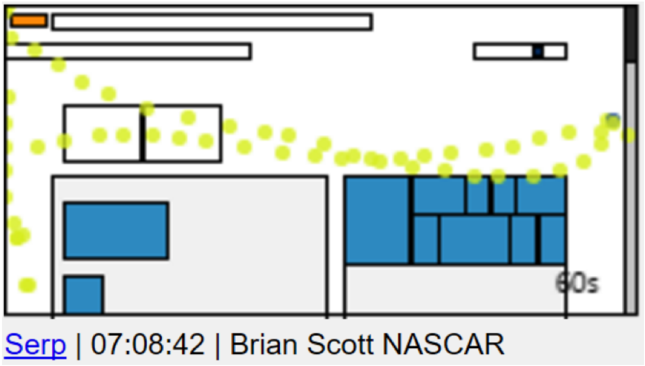


Figure 1: Example of user interaction with the SERP elements rendered for the query *Brian Scott NASCAR*. The sequence of green dots denotes the user's cursor position over a period of time.

User action sequences have been used to predict user satisfaction [15], graded satisfaction [23] and to study search engine switching behavior [36, 39]. Sequential user actions have also been used to explore developing search trails composed of query sequences for enhancing search support [17, 37]. Our work differs from these works in two aspects. First, while past work considered only high-level actions, we consider more detailed fine-grained actions and in the process, propose few novel user actions. Second, most existing work uses the entire action sequences to make predictions while we focus on a slightly different problem of extracting meaningful subsequences which are most representative of user interaction and predictive of user satisfaction.

The closest work to ours is Liu *et al.* [31] which estimate the utilities of search results and the efforts in search sessions with motifs extracted from mouse movement data on search result pages (SERPs). Our work is different since we aim at a slightly different problem of extracting interpretable subsequences and focus on informative action sequences rather than mouse movement coordinates for predicting search satisfaction.

3 EXTRACTING USER INTERACTION SEQUENCES

Our goal in this work is to extract informative and interpretable subsequences from user interaction data which best predict user satisfaction. In this section, we define key concepts used throughout the paper (3.1), describe a way to first construct and then use the *Universal Timeline* of user interaction to extract interaction sequences (3.2), and finally analyse the interaction sequences (3.3) and characterize generic trends in user interaction with the SERP.

Action	Description
Click_algoX	Click on the X-th algorithmic result
Click_Ans	Click on any answer (non-image) result
Click_IMG	Click on any image result
MouseRead	horizontal line across a result snippet of length > 50px and duration > 100 ms that goes from left to right which starts and ends inside an algo-result, or advertisement or an answer result
Scroll	page scroll recorded on the search engine result page
Move	any cursor movement of length > 10px and duration greater than > 50 ms
pause	smallPause: no cursor movement on the SERP for time < 5 seconds mediumPause: no cursor movement on the SERP for 5s < time < 20s longPause: no cursor movement on the SERP for 20s < time < 40s veryLongPause: no cursor movement on the SERP for time > 40s
Resize	change in the size of the window/screen encompassing the result page
IssueQuery	user movement to the Search Box on the SERP and typing of text in the query box
dwellTime	smallDwellTime: dwell time on a clicked result URL with time spent < 10s mediumDwellTime: dwell time on a clicked result URL with 10s < time < 40s longDwellTime: dwell time on a clicked result URL with time spent > 40s
QuickBack	click on a SERP URL followed by returning back to the SERP within 5s

Table 1: Examples of actions considered along with their description used to create the user interaction sequence.

Example Sequences
Scroll → smallPause → Move → Move-algo-1 → smallPause → Move-algo-2 → Move-ans → Move → mediumPause → Move → Move-algo-2 → smallPause → Click-algo-2
smallPause → Move → Click-IMG → longDwellTime
mediumPause → Scroll → Move_algo9 → Move_algo8 → veryLongPause → Move → Click_algo3 → QuickBack → Move_algo3 → smallPause

Table 2: Example of sequences extracted.

3.1 Definitions

We first define what constitutes a sequence and use this definition to define subsequence.

Sequence: Given a search impression and a list of possible user actions, a sequence is defined as a time-ordered list of actions performed by the user when interacting with the search result page.

Subsequence: is a subset of continuous or interleaved actions extracted from the sequence.

Informative Subsequence: is a subsequence which helps a system predict user satisfaction with the search result page.

Interpretable Subsequence: is a human readable, comprehensible set of actions which are easy for system designers to understand and develop metrics around.

Such Informative and Interpretable subsequences represent common user- and query-invariant subsequences which would be difficult to identify or describe by manual inspection or feature engineering. Table 2 presents three examples of user interaction sequences extracted from real world search traffic. With this background, we formally define the problem of extracting subsequences as:

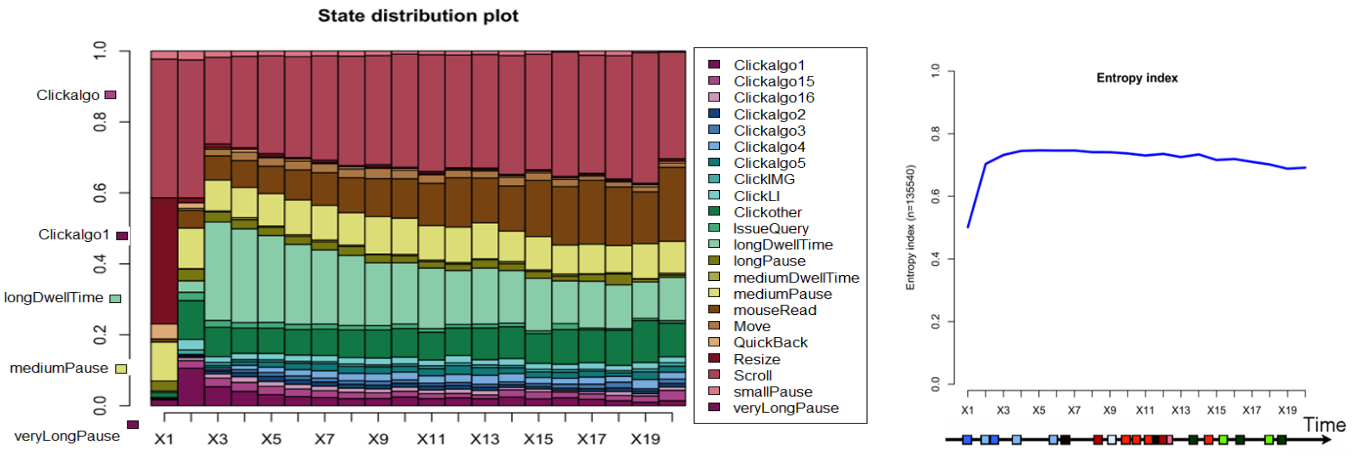
Subsequence Extraction: Given a labeled set of interaction sequences with satisfaction labels, our aim in this work is to infer a set of informative and interpretable subsequences which best predict user satisfaction with the search result page.

A robust subsequence extraction system helps us extract a set of meaningful patterns that are useful for helping a human analyst understand the important properties of user interaction, that is, subsequences should reflect the most important patterns in the interaction data, while being sufficiently concise and non-redundant that they are suitable for manual examination. These criteria are inherently qualitative, reflecting the fact that the goal of subsequence mining is to build human insight and understanding for subsequent metric development and satisfaction prediction.

3.2 Universal Timeline Creation

The richness of the result page rendered in response to a user query allows users to interact with SERPs in myriad ways, including clicking results, scrolling, expanding task panels, hovering over images, pausing to read and absorb content, among others. While most existing work has considered click based interaction signals or mouse movement features, these signals either lack coverage or are often abstracted at high SERP-level aggregates, which blinds the model to finer level user interaction signals. Our aim here is to analyze user interaction with the SERP (as depicted in Figure 1) and extract an interpretable interaction sequence (as shown in Table 2). To do so, we construct a universal action sequence timeline from the following three different timelines:

1. **Viewport Timeline**: Viewport is defined as the position of the webpage that is visible at any given time to the user. Viewport timeline allows us to consider user actions concerning the viewport, for example, scroll on the result page and resize of the screen.
2. **Cursor Timeline**: The cursor timeline provides us with all the cursor related user activity. Backend search logs record detailed user mouse activity which helps us track the mouse movement and link the corresponding cursor activity to the different elements on the SERP. Cursor timeline provides a major portion of the activities we consider in our work.



(a) State distribution plot of actions across positions

(b) The entropy plot of actions spread over positions in interaction sequence.

Figure 2: Analyzing the extracted sequences: (a) State distribution plot & (b) The entropy plot.

(3) **Keyboard Timeline**: The keyboard timeline records all keyboard related user activity (for example, text enter).

For each search impression, we log the three timelines with corresponding user actions along with the timestamp. Based on these three timelines, we generate one holistic universal action sequence timeline describing all user activity on the SERP by temporal sorting of individual timelines followed by stacking up the three timelines, and then interleaving them based on timestamps of the recorded actions. This provides us with a universal sequence of user interaction, examples of which are shown in Table 2. We next take a more detailed look at the actions considered to construct the timelines.

Actions Considered: In order to construct the three timelines, we considered a number of actions which includes different types of interactions performed by the users. For click based actions, we associate the cursor information with the corresponding element on the SERP and recorded the joint action-element pair as an action, for example, click_algo1 signified a click on algorithmic result at position 1. Beyond clicks, we considered a range of cursor movement actions ranging from simple Move (denoting a mouse movement across different SERP element) to more sophisticated and intentional cursor movements like a *MouseRead*. We define a *MouseRead* as a horizontal line across a result snippet of length > 50px and duration > 100 ms that goes from left to right which starts and ends inside an algo-result, or advertisement or an answer result. Beyond cursor movement actions, we considered inter-activity time as pauses and categorized a pause into one of three types based on the duration of the pause: (i) short pause (time), (ii) medium pause (time) and (iii) longPause (time). We additionally considered issuing query and scroll related activities. Table 1 lists the major actions considered.

3.3 Analyzing Interaction Sequences

As shown in Table 2, the set of actions and the total sequence length differs across the different sequences, which hints at an inherent diversity across the different sequences. While the dataset considered is described in detail in Sections 6.1 & 7, we consider a subset of user interactions from a major US commercial search engine and analyze the extracted interaction sequences based on a number of factors including sequence length, state distribution and state entropy.

3.3.1 Sequence Length. Sequence length counts the number of actions present in the sequence. To some extent, sequence length approximates user engagement with the SERP with longer sequences highlighting detailed and richer interaction with the result page. We observe that most sequences are between 2 - 10 actions long, with a strong peak at 4-5 actions, denoting that most sequences average 4-5 actions. The sequences of length 0 represent abandonment cases wherein the user didn't interact with the result page at all. We observe a low rate of abandonment in the data considered.

3.3.2 State Distribution. We next analyze the observed sequences in terms of the actions and the position in the sequence where the actions appear. Figure 2a shows the action distribution at each position of the sequence. For each position, we plot the proportion of sequences which contain a particular action at a particular position in the sequence. Indeed, as expected, certain actions are more likely to happen at the start of interaction, for example, click on the first algorithmic result is expected to happen at the start of the user interaction. *longDwellTime* is another action whose proportion drops significantly as we go to the right. As expected, long dwell time is positively correlated with search satisfaction. Since higher ranked documents result in more satisfying clicks, we notice the presence of *longDwellTime* earlier in the sequence. Additionally, we observe that the occurrence of certain actions decreases towards the right - which implies that certain actions are more likely to occur at later stages of user interaction than at the start. Finally, certain actions are equally likely throughout the sequence, for example, clicks on image results, *mediumPause* etc.

3.3.3 State Entropy. In order to measure the diversity of actions observed at different positions of interactions, we compute the

Shannon entropy of the action distribution at different positions. Letting p_i denote the proportion of sequences having action i at the considered time sequence position, the entropy is: $h(p_1,p_2,...,p_n) = -\sum_{i=1}^{n} p_i log(p_i)$ where s is the size of the alphabet. The entropy is 0 when all sequences contain the same action and is maximal when the same proportion of sequences contain each action. The entropy can be seen as a measure of the diversity of states observed at the considered sequence position. Figure 2b presents the entropy scores across the different positions in the interaction sequence. Higher entropy implies more diversity in the user actions at that particular position in the action sequence. We observe lower entropy at the start of interaction sequences, which implies that there is low diversity at start of interactions; and the trend continues to exhibit high diversity at user interacts more. With lower entropy at the start, we expect low discriminability of the sequence.

4 EXTRACTING INFORMATIVE SUBSEQUENCES

So far we have looked at how to obtain detailed user interaction sequences from their interaction with the result page and looked at some analysis characterizing the observed sequences. Next, we leverage the interaction sequences to extract informative subsequences which are most predictive of user satisfaction. We next describe three ways of extracting the subsequences.

4.1 Frequent Subsequences

Our first method is based on frequent subsequence mining, which aims at capturing the most common subsequences present in the observed interaction sequences. The subsequences are generated by maintaining a sliding window of a given length and shifting it across the entire sequence for all the sequence. We considered subsequences with varying sizes from unigrams (individual actions) to 4-gram subsequences. Table 3 shows the top subsequences of size 2 and above.An advantage of considering frequent subsequences is the fact that most sequences would contain these subsequences and hence they don't suffer from low coverage. We use these extracted frequent subsequences along with most frequent actions and action bigrams as features when predicting user satisfaction.

4.2 Discriminative Subsequences

An important and desired characteristic of subsequences is their discriminatory power. Indeed, the more discriminatory a subsequence is, the better it helps us differentiate satisfying interactions from unsuccessful search interactions. To this end, we propose our second subsequence extraction method which, given a dataset of observed subsequences and their corresponding class labels (SAT/DSAT), aims at extracting the most discriminatory subsequences which help us best differentiate between SAT subsequences from DSAT subsequences.

Assuming that we are given a set of sequences with the corresponding satisfaction label (SAT/DSAT), we make use of the well-established chi-square test to compute the discriminatory power of a given subsequence. The Chi-square test calculates the probability of getting the experimental result on the basis that the null hypothesis is true. In our case, we state the Null and Alternative Hypothesis as follows:

Null Hypothesis (H0): there is no detectable difference between two classes
Alternative hypothesis (HA): there is a detectable difference between the two classes.

The statistical principle behind any discrimination test should be to reject a null hypothesis (H0) that states there is no detectable difference between two classes. Given the observed sequences and the corresponding class label (SAT/DSAT) as input, the Chi-Square test computes the residuals based on:

$$\chi^2 = \sum \left[\frac{(O-E)^2}{E} \right] \tag{1}$$

where O = observed frequency; E = expected frequency. Table 3 presents the top subsequences ordered by their discriminatory power for the set of sequences from a labeled dataset (described in detail in Section 6.1). We observe that a Click → *longDwellTime* is ranked high in terms of discriminatory power which confirms established notion that SAT interactions usually have a satisfying click, i.e., a click followed by a long dwell time. Additionally, we observe that the extracted discriminatory subsequences picks up new potentially SAT subsequences, for example: Scroll → *Mouseread* → *Move*. We also observe few potential DSAT interactions, as expressed by the interaction susequence: (*smallPause* → *Move* → *mediumPause* → *Move*) which highlights the user moving around on the SERP.

An important point to note here is that these discriminatory subsequences are not that popular - the Index column highlights the ranked index of the corresponding subsequences based on their frequency. This suggests that while these subsequences are discriminatory, they are not that popular and hence might not be more generally useful in discriminating SAT/DSAT interactions.

4.3 Interleaved Subsequences

One key limitation of the frequent and discriminative methods of extracting subsequences is that they only look at adjacent actions while extracting subsequences, while failing to consider interleaved subsequences. In this section, we introduce an alternate perspective on subsequence mining, in which we develop subsequences by interleaving a group of subsequences. We formulate the problem of identifying a set of *important* sequences that are useful for explaining the observed interaction sequences. We define *important* subsequence as those subsequences that best *explain* the observed sequences under an interleaved model of subsequences.

Quantifying Importance:
One can think of *importance* score as the weight of the subsequence in the model: the higher the score, the more supported sequences the subsequence explains. Thus importance score provides a more balanced measure than just frequency and discriminatory aspects alone, at the expense of missing some frequent subsequences that only explain some of the observed sequences they support. We jointly encode the notion of discriminability of the subsequence together with its support while defining the importance score of a

Frequent Subsequences		Discriminative Subsequences			
Subsequence	**Coverage**	**Subsequence**	**Index**	**Residual0**	**Residual1**
(Clickother → longDwellTime)	0.226767	(smallPause → Move → Click	138	-2.21152	2.123184
(Scroll → smallPause)	0.215066	(Scroll → Move → smallPause → Move → Click-algo3 → longDwellTime	735	-2.19005	2.102566
(Clickalgo1 → longDwellTime)	0.169337	(Clickalgo1 → longDwellTime)	15	-1.78004	1.708939
(smallPause → Clickother)	0.159562	(Move → smallPause → Move → mediumPause → Move)	265	1.950145	-1.87225
(smallPause → mediumPause)	0.131312	(Scroll → smallPause → Move → ClickLI)	569	-1.98225	1.903068
(smallPause → Clickother → longDwellTime	0.125173	(Move → Clickalgo1)	70	-1.84755	1.773754
(mediumPause → smallPause)	0.120038	(smallPause → mediumPause → Move)	105	1.861069	-1.78673
(smallPause → Clickalgo1)	0.114195	(Scroll → mouseRead → Move)	376	-1.9366	1.859247
(smallPause → mouseRead)	0.114129	(smallPause → Move → mouseRead → Move)	134	-1.68427	1.616994

Table 3: Frequent & Discriminative subsequences extracted. Coverage represents the fraction of sequences which contain the frequent subsequences. Index refers to the rank in terms of frequency of the subsequence, while Residual0 and Residual1 denote the discriminatory power of the subsequence for predicting DSAT and SAT respectively.

subsequence. More formally,

$$imp(S) = \varphi(S)\frac{\sum_{i=1}^{N}[z_S^i \geq 1]}{supp(S)} \tag{2}$$

where $\varphi(S)$ denotes the dsicriminability score of a subsequence obtained from Section 4.2, $\sum_{i=1}^{N}[z_S^i \geq 1]$ counts the number of sequences *explained* by the subsequence S and $supp(S)$ is the support of the subsequence S. The variable z_S^i counts the number of times the subsequence S appears in the i-th sequence in an interleaved manner. The primary intuition while selecting subsequences being to prefer subsequences which are discriminative while at the same time cover a larger number of observed sequences.

We next describe the approach for generating new candidate subsequences that are to be considered for inclusion in the set of *interleaved* subsequences. We initialize the set of *interleaved* subsequences with singleton actions along with their supports. We maintain a priority queue ordered by the importance score and sort the current set I by decreasing order of importance score. The algorithm then iteratively selects all ordered pairs $S_1, S_2 \in I$ and generates a new candidate $S = S_1 S_2$ and adds the candidate to the priority queue. Finally, we pull the top-k highest ranked candidate based on their importance score to compose the set of interleaved subsequences.

An important information which we haven’t utilized so far is the temporal aspect of the user interaction. We next describe a principled approach of incorporating time while using the extracted subsequences in the SAT prediction model.

5 INCORPORATING TIME VIA HAWKES PROCESS

While the subsequences extracted so far are able to capture informative user interactions, they ignore the temporal spread of actions. So far, time has been modeled in terms of few time-related aspects like pause and dwell time. In this section, we aim at incorporating time in a principled manner when modeling user interaction sequences. We leverage recent advancements in point process models and propose a Hawkes process formulation of interaction sequences. We first give a brief background on Hawkes process (5.1) and later describe how we use the Hawkes process model parameters for weighing the extracted subsequences (5.3).

5.1 Hawkes Process

A point process N is a random measure on a completely separable metric space S that takes values on $N \cup \{\infty\}$. A point process is typically characterized by prescribing its conditional intensity $\lambda(t)$, which represents the infinitesimal rate at which events are expected to occur around a particular time t, given the history of the process up to t, $H_t = t_i : t_i < t$ [33, 34] Thus, in a point process, $N(t)$ counts the number of points (i.e., occurrences of events) in $(-\infty, t]$, and the conditional intensity function $\lambda(t|H_t)$ denotes the expected instantaneous rate of future events at timestamp t depending on H_t, the history of events preceding t.

An important example of a point process is the Poisson process, which always has a deterministic conditional intensity $\lambda(t)$. We say that a point process N is self-exciting if

$$Cov[N(t_1, t_2), N(t_2, t_3)] > 0 \tag{3}$$

for any $t_1 < t_2 < t_3$. This means that if an event occurs, a successive event becomes more likely to occur locally in time and space. This is, however, not true for a Poisson process which has independent increments, hence $Cov[N(t_1, t_2), N(t_2, t_3)] = 0$.

The Hawkes process is a specific class of self- or mutually-exciting point process models [18, 19]. A univariate Hawkes process $\{N(t)\}$ is defined by its intensity function

$$\lambda(t) = \mu(t) + \int_{-\infty}^{t} \mu(t-s)dN(s) \tag{4}$$

where $\lambda_0 : \mathfrak{R} \rightarrow \mathfrak{R}^+$ is a deterministic base intensity, $\mu : \mathfrak{R}^+ \rightarrow R^+$ is a kernel function expressing the positive influence of past events on the current value of the intensity process. In terms of discrete time intervals, we can re-write the intensity function as:

$$\lambda(t) = \lambda_0(t) + \sum_{t_i < t} \mu(t-s) \tag{5}$$

The process is well known for its self-exciting property, which refers to the phenomenon that the occurrence of one event in the past increases the probability of events happening in the future. Such a self-exciting property can either exist between every pair of events, as assumed in a normal univariate Hawkes process, or only exist between limited pairs of events. We next formulate the extracted interaction sequence in a multi-variate version of Hawkes processes and later describe a way of weighing the subsequences using the Hawkes process parameters.

5.2 Modeling Sequences

Given a set of user interaction sequences composed of (maximum) M distinct actions, we model the sequences using a Multivariate Hawkes process model. A Multivariate Hawkes process can be defined as an d-dimensional Hawkes process wherein events can occur along any of the d-dimensions at any given time. Each one-dimensional Hawkes process can be influenced by the occurrence of events of other types. Without loss of generality, we will consider that these mutual excitations take place along the edges of an unweighted directed network G = (V, E) of d nodes and adjacency matrix $A \in \{0,1\}^{d \times d}$. The intensity function in a d-dimensional multivariate Hawkes process can be defined as:

$$\lambda(t) = \lambda_m(t) + \sum_{m:t_m<t} A_{u_m u} \kappa_{u_m u}(t - t_m) \tag{6}$$

where $\lambda_m(t) \geq 0$ is the natural occurrence rate of events of type m (i.e. along that dimension) at time t, and the triggering kernel function evaluation $\kappa_{uv}(t - t_m) \geq 0$ determines the increase in the occurrence rate of events of type u at time t_i, caused by an event of type v at a past time.

In our case, the number of dimensions corresponds to the number of actions considered (M); the intensity function for each dimension can be interpreted as a rate at which that action occurs. The summation in the second term is over all the action events that have happened up to time t. $\lambda_m(t)$ describes the background rate of action occurrence that is time-independent, whereas the second term describes the mutual-excitation part, so that another action in the past increases the probability of observing this action in the (near) future. The natural occurrence rates and triggering kernels are usually inferred by means of log-likelihood maximization, the details of which are beyond the scope of the current work. We refer interested readers to Embrechts *et al.* [5].

5.3 Kernel Weighting of Subsequences

The Hawkes process model described above enables us to compute the base rates for each action ($\lambda_0(m)$) as well as the triggering kernel ($\kappa_{uv} \in R^{M \times M}$). The triggering kernel value (κ_{uv}) capture the mutually exciting property between the actions u and v. Intuitively, it captures the dynamics of influence of events occurred in the action v to the action u. As such, larger value of κ_{uv} indicates that action u is more likely to trigger an occurrence of action v. We propose to use the triggering kernel to score each subsequence extracted by the techniques proposed earlier. Hawkes kernel weighting of the subsequences allows us to incorporate temporal aspects of the interaction; thereby weighing temporally relevant subsequences more than others. Given a subsequence, we look at adjacent pairwise actions in the subsequence, compute the kernel trigger score for the corresponding pair, and average it across all the adjacent action pair in the subsequence. We then use the obtained averaged kernel weight of the subsequence as a feature for SAT prediction experiment.

6 EXPERIMENTAL EVALUATION

Estimating user satisfaction from user behavior signals is of critical importance to web search engines. In this section, we demonstrate how automatically extracted subsequences can be used to improve estimation of user satisfaction. We conduct a number of experiments using crowdsourced judgments as well as real world search engine traffic and compare the proposed approach to a number of baselines.

6.1 Crowdsourced Judgments

Our data consists of a random sample of user sessions from a major US commercial search engine engine during a week in June 2016. We randomly sampled user sessions with substantial user activity, and included all queries and the search result page rendered for all search impressions from that user in the timeframe. Additionally, detailed user activity on the result page was logged for model development. Crowdsourced judgments have commonly been used to obtain labeled data [40, 41]. For each search impression, we obtained human labeled judgments on whether the user interaction was satisfying (labeled SAT) or not (labeled DSAT).

The labeling was conducted using an in-house micro-tasking platform that outsources crowd work to vendors, similar to CrowdFlower, and provides access to judges who regularly perform relevance judgment tasks. Workers were under NDA and all data containing personal identifiable information (PII), such as names, phone numbers, addresses, or social security numbers, were removed. The internal human annotation platform was used to design, publish and manage human annotation tasks.

Detailed guidelines were issued to the judges to describe the task and a number of examples were shown explaining how to judge for satisfaction. To ensure the quality of the judging results, we apply a series of quality control methods. One of the methods is creating 'gold hits' that you already know the answer of, then measure the judges by comparing how far off their answers are from the gold hits answers. We also measure the quality of the judgments with the amount of consensus reached which required overlap on the hits, i.e. the same hit to be judged by multiple judges.

In order to provide relevant information to the judges, we provided a detailed summary of user interaction with the SERP. The judges were provided a link to the SERP shown to the user alongside details like number of clicks, time spent on the SERP and scroll information. Additionally, for all the clicked documents, we provided URL level details which included the exact URL, the position on the SERP where it was shown and the total dwell time on each URL.

We randomly sampled over 2100 user sessions and over 450 judges provided judgments for about 6820 search impressions, resulting in over 20460 judgments. Among the first two judgments collected for each query, the judges agreed on the label 74% of the time. We measured inter-rater agreement using Fleiss' Kappa [7], which allows for any number of raters and for different raters rating different items. This makes it an appropriate measure of inter-rater agreement in our study since different judges provided labels for different items. A kappa value of 0 implies that any rater agreement is due to chance, whereas a kappa value of 1 implies perfect agreement. In our data, $\kappa = 0.64$, which, according to Landis and Locke [30], represents substantial agreement.

6.2 Baselines

We consider a number of baselines from recent published literature.

- Baseline 1 (click with dwell time): This baseline is based on the common approach in the literature as labeling satisfaction as occurring if a user clicks on a search result and then spends a minimum of t seconds on a page and does not follow the query up with a reformulation. Spending

Method	Accuracy	Pos P	Pos R	Neg P	Neg R
Baseline 1 (Clicks + DwellTime)	0.56	**0.99**	0.559	0	0
Baseline 2 (Click based actions)	0.59	0.58	**0.99**	0	0
Baseline 3 (Mouse Movement)	0.606	0.643	0.741	0.587	0.454
Baseline 4 (Scroll & Viewport)	0.586	0.679	0.703	0.521	0.364
Baseline 5 (Reading Pattern Signals)	0.596	0.652	0.771	0.564	0.415
All Actions	0.62*	0.653	0.775	0.562	0.414
Action Bigrams	0.582	0.594	0.93	0.521	0.096
Frequent Subsequences	0.603	0.626	0.811	0.52	0.29
Discriminative Subsequences	0.592	0.61	0.886	0.56	0.18
Interleaved Subsequences	0.613	0.631	0.862	0.572	0.22
Actions + Frequent + Discriminative Subsequences	0.622*	0.65	0.8	0.581	0.39
Hawkes weighted Actions	0.631*	0.663	0.781	0.57	**0.462***
Hawkes-weigthed-Actions + Hawkes-weigthed-Subsequences	**0.672***	0.677	0.86	**0.65***	0.37

Table 4: Measurements of prediction quality based on different methods on all user study data. $*$ indicates statistical significant ($p \leq 0.05$) using paired t-tests compared to the corresponding best performing baseline.

a minimum amount of time on a webpage is known as a long dwell click and has been shown to be correlated with satisfaction [26]. In this study, we set t = 30 seconds.

- Baseline 2 (click based actions): This baseline is based on predicting satisfaction based on clickthrough based features [13].
- Baseline 3 (Mouse movement): This baseline is based on recent work aimed at predicting satisfaction using mouse movement patterns.
- Baseline 4 (Scroll & Viewport): This baseline is based on the recently proposed scrolling and viewport features [41]
- Baseline 5 (Reading pattern signals): This baseline is based on the reading pattern signals from Kiseleva *et al.*[27]

We additionally consider variants of the proposed techniques: (i) All Actions, (ii) action bigrams, (iii) the three different types of extracted subsequences, (iv) hawkes weighted actions and (v) combinations of different features.

6.3 SAT Prediction

Based on the obtained judgments, we aim at predicting user satisfaction for each impression. We used Gradient Boosted Decision Trees (GBDT) as the classifier with 5-fold cross-validation for all the results reported. Each extracted subsequence was used as a feature, first with a binary label marking its presence and then with the temporal Hawkes weights. We compare the proposed subsequence based features with a number of standard techniques used in state-of-the-art user satisfaction prediction systems. Table 4 presents the prediction results for the crowdsourced data.

6.3.1 Quantifying gains of detailed actions. We begin by investigating the gains obtained by considering fine-grained user actions over traditionally used clickbased and dwell time based features. Clicks and dwell times have been shown to accurately predict user satisfaction [26] and power a number of industrial experimentation metrics [42]. We observe a 6% increase in SAT prediction accuracy over dwell time based features and ~ 3% increase over click based actions. We also observe that considering all actions are more predictive than individual scroll based or mouse movement based signals. This suggests that adding fine grained user actions indeed helps improve prediction performance, and that other types of actions beyond clicks, are also informative and should be considered by developing metrics.

6.3.2 Comparing subsequences. Having shown the utility of considering fine grained actions beyond clicks, we investigate the performance of the various proposed subsequence extraction techniques. We observe that the action bigrams perform the worst among the extracted subsequences, with all three frequent, discriminatory and interleaved subsequences performing better. It is interesting to note that the discriminatory subsequences do not perform as well as the frequent ones, despite that the fact that they're discriminatory by definition. Such poor performance of discriminatory might stem from the fact that most of the discriminatory subsequences are ranked very low in terms of frequency; hence, despite them being helpful in discriminating the SAT interactions form DSAT ones, they may not always be present and hence impact the prediction accuracy to a less extent. Additionally, the joint encoding of discriminability and importance enables the Interleaved subsequences to perform the best among the extracted subsequences. Such subsequences, when combined with actions, results in over 6% improvement over click and dwell time baseline.

6.3.3 Quantifying Benefits of Incorporating Temporal Aspects. We observe that incorporating temporal aspects of user interaction helps. As can be seen from the improved performance of the Hawkes process weighted actions, the SAT prediction accuracy increases by ~ 5% over the action bigrams and performs slightly better than other subsequence extraction techniques as well as the best baseline performance. When the temporal aspects of all extracted subsequences are incorporated via Hawkes weighting scheme, we observe an improvement of over 12% in terms of prediction accuracy over traditionally used click and dwell time based signals. Indeed, time plays a major role in differentiating satisfying user interactions from DSAT interactions.

6.4 Feature Correlation Analysis

One major motivation for the current work is to extract meaningful informative and interpretable subsequences from user interactions

Top Positively Correlated		Top Negatively Correlated	
Subsequence	**Correlation**	**Subsequence**	**Correlation**
longDwellTime	0.213252	Scroll, mediumPause, smallPause	-0.05476
Clickalgo1	0.109037	smallPause, mediumPause, Move	-0.05183
Clickalgo1, longDwellTime	0.102703	Move, smallPause, mediumPause, Move	-0.03799
Move, Clickalgo1	0.102603	Move, IssueQuery	-0.03356
smallPause, Move, ClickLI, longDwellTime	0.053584	longPause, Move	-0.02571
smallPause, Move, mouseRead, Move	0.050649	IssueQuery	-0.02363

Table 5: Correlation analysis of the different subsequences.

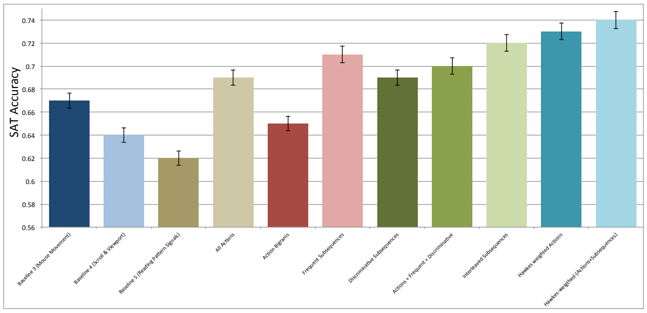


Figure 3: Impact on Abandonment Cases

for developing satisfaction metrics. To this end, we present the feature correlation analysis in Table 5 to gauge the impact of the different subsequences on predicting user satisfaction. We compute the Pearson correlation between the user SAT label and the subsequences and show the top most informative subsequences.

The correlation results re-establish known facts - clicks and long dwell time features are strongly correlated with predicting SAT. Beyond clicks and dwell time, we observe that frequent subsequences like Scroll → Move → Pause are also predictive of SAT label. The discriminatory subsequence: smallPause → Move → MouseRead → Move is strongly correlated with the satisfaction label. Additionally, subsequences highlighting general random user movement in the SERP, for example: Move → smallPause → mediumPause → Move, is negatively correlated with satisfaction. These correlation results could provide insight into which kind of interaction signals should be detected and used for gauging user satisfaction.

6.5 Impact on Abandonment Cases

Often, there are cases where a user may not click on any search result but still be satisfied. This scenario is referred to as good abandonment and presents a challenge for most approaches measuring search satisfaction, which are usually based on clicks and dwell time. As our final crowdsourcing experiment, we consider the utility of the proposed methods in the abandonment cases. We consider all impressions which had no click activity in the user interaction sequence and repeat the user satisfaction prediction experiment on the considered sequences. Figure 3 presents the results. We observe that the same approach works well in abandonment cases too. SAT prediction accuracy increases as we incorporate signals and actions beyond clicks and dwell time. We observe that a combination of different subsequences and temporal aspects via Hawkes process outperforms all other approaches and boosts the SAT prediction accuracy to over 74%. We leave the detailed exploration of interaction signals for good abandonment cases for future work.

7 LARGE SCALE PSEUDO-LABELLED DATA

Owing to the limited scale of experimentation possible with crowdsourced judgments as well as the differences in opinion of crowdsourced judges and actual users, we may have insufficient data and labels to reliably evaluate the performance of the proposed subsequence extraction techniques. To resolve this problem, we build a pseudo-labeled dataset comprised of large-scale query logs wherein we randomly extracted over 148000 query impressions from the query logs of the same search engine that provided the data for the crowd-sourced assessment. To assign pseudo satisfaction labels to search interactions, we assume that a click followed by a query reformulation is a dissatisfied click, while a click with a dwell time of ≥ 30 seconds not followed by a query reformulation is a satisfied click. A query reformulation is the act of submitting a follow up query to modify a previous search query in hope of retrieving better results. Post-click query reformulation is considered a strong DSAT predictor and has been used as a predictor of search satisfaction in previous work [16, 26]. The intuition here is that dissatisfied users will reformulate their queries, while satisfied users will not. This is a crude estimate of user satisfaction but it allows us to easily generate large numbers of pseudo labeled instances without leveraging any information about the click itself (which could lead to confounding).

To identify query reformulations we use a method similar to that described in Boldi *et al.*[4], where features of query similarity (e.g. edit distance, word overlap, etc.) and time between queries are used to identify query reformulations. Using these assumptions, we randomly collected 14,670 user sessions with 148561 search impressions from the search logs of the engine described earlier.

Table 6 presents measurement of prediction quality on this dataset. We observe a similar trend with respect to the relative performance of the actions and subsequences for satisfaction prediction. We find that fine grained actions perform better than the baseline methods, and the interleaved subsequences performing the best among the extracted subsequences. Finally, adding temporal aspect via Hawkes weighting scheme helps boost predictive power by over 7% compared to the best performing baseline.

8 DISCUSSION

Predicting user satisfaction plays an instrumental role in designing and experimenting with search systems. We adopted a novel way of considering user interactions, and constructed universal interaction sequences and leveraged them via three methods to extract informative subsequences which performed better than generic click and dwell time based action signals for predicting user satisfaction. Further, we proposed a Hawkes process model to incorporate temporal

Method	Accuracy	Pos P	Pos R	Neg P	Neg R
Baseline 1 (Clicks + DwellTime)	0.522	**0.965**	0.511	0.22	0.18
Baseline 2 (Click based actions)	0.59	0.602	0.3	0.531	**0.79**
Baseline 3 (Mouse Movement)	0.583	0.592	**0.92**	0.531	0.11
Baseline 4 (Scroll & Viewport)	0.605	0.61	0.81	0.51	0.28
Baseline 5 (Reading Pattern Signals)	0.563	0.581	0.742	0.54	0.13
All Actions	0.62	0.655	0.771	0.562	0.4
Action Bigrams	0.595	0.59	0.937	0.512	0.09
Frequent Subsequences	0.603	0.611	0.812	0.51	0.283
Discriminative Subsequences	0.591	0.6	0.88	0.55	0.17
Interleaved Subsequences	0.613^{*}	0.645	0.761	0.572	0.32
Actions + Frequent + Discriminative Subsequences	0.622^{*}	0.64	0.802	0.582	0.38
Hawkes weighted Actions	0.635^{*}	0.657	0.77	0.565^{*}	0.41
Hawkes-weigthed-Actions + Hawkes-weigthed-Subsequences	**0.667^{*}**	0.665	0.872	**0.656^{*}**	0.37

Table 6: Measurements of prediction quality based on different methods on large scale pseudo-labelled dataset. ∗ indicates statistical significant ($p \leq 0.05$) using paired t-tests compared to the corresponding best performing baseline.

aspects of user interactions while modeling the extracted subsequences, which when combined with other signals, outperforms all the baselines considered.

The promising results call for deeper investigation into such detailed user activity. Our investigation suggests three concrete areas of further investigation: (i) the type of search result shown, (ii) user type analysis to detect user groups for personalized interaction modeling and (iii) development of sequential models around the extracted universal interaction sequences.

REFERENCES

[1] Charu C Aggarwal and Jiawei Han. 2014. *Frequent pattern mining*. Springer.

[2] Agrawal and Srikant. Mining sequential patterns. In *ICDE 1995*.

[3] Ioannis Arapakis, Mounia Lalmas, and George Valkanas. Understanding within-content engagement through pattern analysis of mouse gestures. In *CIKM 2014*.

[4] Paolo Boldi, Francesco Bonchi, Carlos, Debora, Aristides, and Sebastiano. The query-flow graph: model and applications. In *CIKM 2008*.

[5] Paul Embrechts, Thomas Liniger, Lu Lin, and others. 2011. Multivariate Hawkes processes: an application to financial data. *Journal of Applied Probability* (2011).

[6] Henry A Feild, James Allan, and Rosie Jones. Predicting searcher frustration. In *SIGIR 2010*.

[7] Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* (1971).

[8] Jaroslav Fowkes and Charles Sutton. 2016. A Subsequence Interleaving Model for Sequential Pattern Mining. *arXiv preprint arXiv:1602.05012* (2016).

[9] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. 2005. Evaluating implicit measures to improve web search. *ACM TOIS* (2005).

[10] A Gaivoronski. 2005. Markov models for identification of significant episodes. (2005).

[11] Qi Guo and Agichtein. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *WWW 2012*.

[12] Qi Guo and Eugene Agichtein. Exploring mouse movements for inferring query intent. In *SIGIR 2008*.

[13] Qi Guo, Dmitry Lagun, and Eugene Agichtein. Predicting web search success with fine-grained interaction data. In *CIKM 2012*.

[14] Qi Guo, Ryen W White, Susan T Dumais, Jue Wang, and Blake Anderson. 2010. Predicting query performance using query, result, and user interaction features. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*.

[15] Ahmed Hassan, Rosie Jones, and Kristina Lisa Klinkner. Beyond DCG: user behavior as a predictor of a successful search. In *WSDM 2010*.

[16] Ahmed Hassan, Xiaolin Shi, Nick Craswell, and Bill Ramsey. Beyond clicks: query reformulation as a predictor of search satisfaction. In *CIKM 2013*.

[17] Ahmed Hassan and Ryen W White. Task tours: helping users tackle complex search tasks. In *CIKM 2012*.

[18] Alan G Hawkes. 1971. Point spectra of some mutually exciting point processes. *JSTOR* (1971).

[19] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* (1971).

[20] Jeff Huang, Ryen White, and Georg Buscher. User see, user point: gaze and cursor alignment in web search. In *CHI 2012*.

[21] Jeff Huang, Ryen W White, Georg Buscher, and Kuansan Wang. Improving searcher models using mouse cursor activity. In *SIGIR 2012*.

[22] Jeff Huang, Ryen W White, and Susan Dumais. No clicks, no problem: using cursor movements to understand and improve search. In *CHI 2011*.

[23] Jiepu Jiang, Ahmed Hassan Awadallah, Xiaolin Shi, and Ryen W White. Understanding and predicting graded search satisfaction. In *WSDM 2015*.

[24] Diane Kelly. 2009. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval* (2009).

[25] Youngho Kim, Ahmed Hassan, White, and Zitouni. Comparing client and server dwell time estimates for click-level satisfaction prediction. In *SIGIR 2014*.

[26] Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. Modeling dwell time to predict click-level satisfaction. In *WSDM 2014*.

[27] Julia Kiseleva, Kyle Williams, Hassan Awadallah, Crook, Zitouni, and Anastasakos. Predicting user satisfaction with intelligent assistants. In *SIGIR 2016*.

[28] Dmitry Lagun, Ageev, Qi Guo, and Agichtein. Discovering common motifs in cursor movement data for improving web search. In *WSDM 2014*.

[29] Dmitry Lagun, Chih-Hung Hsieh, Webster, and Navalpakkam. Towards better measurement of attention and satisfaction in mobile search. In *SIGIR 2014*.

[30] J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics* (1977).

[31] Yiqun Liu, Ye Chen, Jinhui Tang, Jiashen Sun, Min Zhang, Shaoping Ma, and Xuan Zhu. Different users, different opinions: Predicting search satisfaction with mouse movement information. In *SIGIR 2015*.

[32] Yiqun Liu, Wang, Ke Zhou, Nie, Zhang, and Ma. From skimming to reading: A two-stage examination model for web search. In *CIKM 2014*.

[33] Yosihiko Ogata. 1988. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical association* (1988).

[34] Y Ogata. 1989. Statistical model for standard seismicity and detection of anomalies by residual analysis. *Tectonophysics* (1989).

[35] Kerry Rodden, Xin Fu, Anne Aula, and Ian Spiro. Eye-mouse coordination patterns on web search results pages. In *CHI 2008*.

[36] Denis Savenkov, Dmitry Lagun, and Liu. Search engine switching detection based on user personal preferences and behavior patterns. In *SIGIR 2013*.

[37] Adish Singla, Ryen White, and Jeff Huang. Studying trailfinding algorithms for enhanced web search. In *SIGIR 2010*.

[38] Louise T Su. 1992. Evaluation measures for interactive information retrieval. *Information Processing & Management* (1992).

[39] Ryen W White and Susan T Dumais. Characterizing and predicting search engine switching behavior. In *CIKM 2009*.

[40] Ryen W White, Matthew Richardson, and Wen-tau Yih. Questions vs. queries in informational search tasks. In *WWW 2015*.

[41] Kyle Williams, Julia Kiseleva, Aidan C Crook, Zitouni, Awadallah, and Khabsa. Detecting good abandonment in mobile search. In *WWW 2016*.

[42] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. Beyond clicks: dwell time for personalization. In *RecSys2014*.