Predicting User Engagement with Direct Displays Using Mouse Cursor Information

Ioannis Arapakis*
Eurecat
Barcelona, Spain
arapakis.ioannis@gmail.com

Luis A. Leiva†
Sciling
Valencia, Spain
llt@acm.org

ABSTRACT

Predicting user engagement with direct displays (DD) is of paramount importance to commercial search engines, as well as to search performance evaluation. However, understanding within-content engagement on a web page is not a trivial task mainly because of two reasons: (1) engagement is subjective and different users may exhibit different behavioural patterns; (2) existing proxies of user engagement (e.g., clicks, dwell time) suffer from certain caveats, such as the well-known position bias, and are not as effective in discriminating between useful and non-useful components. In this paper, we conduct a crowdsourcing study and examine how users engage with a prominent web search engine component such as the knowledge module (KM) display. To this end, we collect and analyse more than 115k mouse cursor positions from 300 users, who perform a series of search tasks. Furthermore, we engineer a large number of meta-features which we use to predict different proxies of user engagement, including attention and usefulness. In our experiments, we demonstrate that our approach is able to predict more accurately different levels of user engagement and outperform existing baselines.

CCS Concepts

•Information systems → Task models; •Human-centered computing → User studies;

Keywords

web search; knowledge module; direct displays; mouse cursor tracking; user engagement

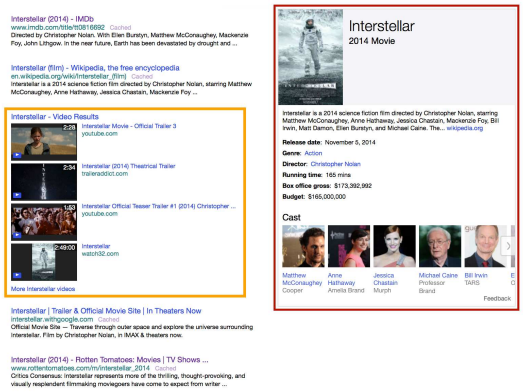


Figure 1: The KM display (in red border) and video snippets (in orange border) on the Yahoo SERP for the query "interstellar".

1. INTRODUCTION

In recent years, direct displays (DDs) have become a standard component on the search engine result pages (SERPs) of all major web search engines, where they display vertical search results, i.e., focused, specific content. One such prominent example is the knowledge module (KM) display (Fig. 1), which provides users with information about the named entities they are searching for as part of their search tasks. The content presented in the KM display is typically obtained in a semi-structured format from curated entity databases, such as Freebase or Wikipedia, and often includes both quantitative and qualitative information (e.g., domain-specific knowledge) about the queried entity. This raw information can be further enriched by the search engine, e.g., by showing a ranking of related entities, accompanied with explanations of their relationship. Often, the KM display is complemented with additional content, such as multimedia or social media content associated with the entity, typically obtained from third-party data sources.

In practice, DDs serve two main purposes. First, they provide the users with a well-structured summary of information which, otherwise, would be difficult or time-consuming to access. That is, the information is made available within the SERP itself and thus the user is not required to actively look for it and navigate away from the SERP. Second, DDs help tidy up the SERP section that contains the universal search results (i.e., the mix of main web search results, vertical search results and various DDs). For example, image and

*Work conducted while affiliated with Yahoo Labs.
†Work partially conducted while affiliated with the PRHLT Research Center at the Universitat Politècnica de València.

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

ISBN 978-1-4503-4069-4/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2911451.2911505

video results can now be displayed under the KM display, making the interaction with textual results an easier task.

In this context, most research has focused on general backend system tasks, the most important being knowledge base construction [6, 11, 14, 24, 41], or more specific backend tasks, such as related entity recommendation [9, 10]. With the exception of a recent query log analysis on exploratory search [34], so far little has been done to understand the way web search users interact with the frontend system and the embedded DDs. Our work attempts to understand how users engage with a prominent DD like the KM display in entity-centric search tasks. In particular, we are interested in predicting user engagement with a DD in the absence of explicit feedback (e.g., self-report data). To this end, we consider three proxies of engagement with online content: (1) attention, (2) usefulness and (3) perceived task duration, which all together reflect relevant cognitive and emotional processing.

Existing modelling techniques make a simplifying assumption when analysing web search log data: the user is assumed to be equally engaged with all parts of the SERP. However, in practice this assumption is not always true. For example, a user may click on certain links on the page, but not all links. Similarly, she may read a certain result snippet in the SERP, but not necessarily the entire list of results. She may even ignore the SERP content completely and focus only on the images shown in the KM display or other DDs. In sum, user interests inferred from the parts of the SERP the user did not engage with may not reflect the true reason for user's interests in the SERP. We believe that the users' interests can be better captured by analysing their within-content activity and, for this reason, we need to go beyond the existing user modelling techniques that assume a uniform user engagement with the page content. To meet this objective, one potential approach is to analyse the mouse cursor data of the user to identify the DDs that captured user's attention and lead to a deeper engagement with the SERP content. The proposed method relies on the use of simple yet highly discriminative features that lack the computational complexity of other methods, such as those that necessitate the extraction of cursor motifs. More importantly, it offers better granularity of user interactions, on a DD-level basis, especially in the absence of explicit user feedback (e.g., clicks or dwell time).

2. RELATED WORK

There has been an enormous body of research investigating user interactions from mouse cursor data. Mouse cursor tracking is considered today as a low-cost yet scalable proxy for user's attention, and so we focus our review of related work in this research area. Concretely, we will focus on web search and interaction mining as foundational precursors of user engagement research.

Early works considered simple, coarse-grained features derived from mouse cursor data to be surrogate measurements of user interest, such as amount of cursor movements [17, 39] or cursor travel time [12]. More recently, fine-grained cursor features have been taking into consideration, which have been proved more useful. For example, Guo and Agichtein [18] found differences in cursor travel distances between informational and navigational queries, and could classify the query type using cursor movements more accurately than using clicks. Guo and Agichtein [19] also used interactions such as cursor movement, hovers and scrolling to accurately infer search intent and interest in search results. They focused on automatically identifying a searcher's research or purchase intent based on features of the interaction. Huang et al. [26] sought to understand result relevance and search abandonment by mining cursor behaviour on SERPs. This work was extended by Diriye et al. [13] to investigate the use of cursor interactions for classifying the reason why a user abandoned a query, whether it was because they were satisfied because they found the information they were seeking or dissatisfied at the point of abandonment. Finally, Huang et al. [25] and Speicher et al. [40] modelled user cursor interactions on SERPs by extending click models to compute more accurate relevance judgments for the search results.

Mouse cursor data have also been used for more practical tasks. For example, to investigate the usability of online forms [5], prototype website redesigns [4, 28] cluster documents according to users' interactions [30] and predict page-level measurements such as dwell time, number of clicks or scroll reach [31].

In sum, previous approaches that relied on mouse cursor feature engineering have been directed at predicting general-purpose web search tasks like document relevance [20], search success [21] or searcher's frustration [16]. However, little work has been done on predicting user engagement within page content. The following are the most prominent works in this regard. Arapakis et al. [1, 2] extracted mouse *gestures* to measure within-content engagement on news pages and predict reading experiences, and Arapakis et al. [3] conducted a preliminary study which revealed the potential benefits of the KM display and its overall utility w.r.t. user experience. Lagun et al. [27] introduced the concept of frequent cursor subsequences (namely *motifs*) in the estimation of result relevance, which is a more general approach but does not address the problem user engagement prediction. Finally, Liu et al. [32] have applied the motifs concept to SERPs in order to predict search result utility, searcher effort and satisfaction at a search task level. However, it has been always assumed a uniform engagement with all parts of the page. In contrast, our work is the first to investigate user engagement within particular components of SERPs, in this case the KM display.

3. CROWDSOURCING STUDY

To understand how web search users engage with DDs like the KM display, we conducted a crowdsourcing study and collected feedback from participants who performed short, entity-centric search tasks using the Yahoo web search engine. With this study, we aim to predict: (1) when a user notices the KM display on the SERP, (2) if it is perceived as a useful aid to their search tasks and (3) whether interacting with the KM display alters the users' perception of how fast they complete the search tasks.

Crowdsourcing offers several advantages not available in other experimental settings [33], such as access to a large and diverse pool of participants with stable availability, as well as collection and analysis of real usage data at a large scale. Another advantage of crowdsourcing is the low cost of the tasks, which makes it a preferable solution over the more expensive laboratory-based experiments. On the downside, a limited range of parameters can be explored in a controlled manner and experimenters have to account for potential threats to ecological validity, distractions in the physical environment of the user, and privacy issues, to name a few.

Table 1: Examples of search query patterns. Asterisks represent multi-answer questions to increase the difficulty of the search tasks.

Category	Question Pattern	Question Example	Suggested Query
People	Who is X <wife | husband | sons>	Who is X's husband|wife?	Who is Orlando Bloom's wife?
People	What is X doing now	What is X's current occupation?	What is Gisele Bundchen doing now?
People	Where was X born	Where was X born?	Where was Ernest Hemingway born?
*People	X movies	Name two movies that X plays in	Peter Seller's movies
Movies	How long is the movie X	How long is the movie X?	How long is the movie Tron?
Movies	When is X coming out	When is movie X coming out?	When is Mad Max Fury Road coming out?
Movies	What is X rated	What is movie X rated?	What is Anchorman rated?
Movies	What is X about	What is movie X about?	What is Taxi Driver about?
Athletes	X salary	What is X's salary?	Lebron James salary
Athletes	X draft	What is X's draft?	Andrian Peterson draft
Athletes	X weight	What is X's weight?	Jimmy Graham weight
Athletes	X team	In which team X plays?	Joe Thomas team
Sport Teams	X head coach	Who is the head coach of the team X?	Miami Heat head coach
*Sport Teams	X players	Name two players of the team X	Houston Rockets players
Sport Teams	X official website	What is the official website of the team X?	Sacramento Kings official website
Sport Teams	X record	What is the record of the team X?	Oakland Raiders record

```
cursor timestamp xpos ypos event xpath attrs extras
0 1405605225834 390 195 mousemove /html/body/p[2] {"P":{}} {"tr":451,"tl":277,"br":473,"bl":313,"mid":317}
...
```

Figure 2: Example of a mouse cursor log, including the distance of the cursor to 5 control points of the KM display: `tr` (top right corner), `tl` (top left corner), `br` (bottom right corner), `bl` (bottom left corner), `mid` (center).

In our study, we used the Amazon Mechanical Turk service. All of the aforementioned limitations were taken into consideration and preventive measures were put into practice to discount low-quality responses. Also, strict selection criteria were applied to exclude unsuitable participants (e.g., HIT approval rate $\geq$ 98%, number of HITs approved $\geq$ 1,000).

3.1 Experimental Design

The experiment had a repeated measures design with one independent variable: KM display (with two levels: "visible" or "hidden"). The KM display visibility was controlled with client-side scripting, removing the KM display from the SERP in the "hidden" condition. The dependent variables (Section 3.4) were: (i) KM display noticeability, (ii) KM display usefulness and (iii) perceived task accomplishment speed. The experiment consisted of two short search tasks that were completed using the Yahoo search engine, one task with the KM display on the SERP and one without it. To control for order effects, we counterbalanced task assignments using a Latin square design.

Participants accessed the search engine through a custom proxy which did not alter the original look and feel of the SERPs. This allowed us to instrument the browsed pages on the fly and capture user interactions with the SERP without interfering with the actual web search engine interface in production. The proxy had a common entry page for all participants. For each search task, participants were presented with a question and were suggested a search query to begin with. The suggested queries were all picked from a pool of queries that triggered the KM display on the SERP, independent of the KM display visibility (Section 3.2).

3.2 Search Query Sample

Our query set consisted of 32 unique query patterns that were selected after a large-scale query log analysis. All queries would trigger the KM display on the Yahoo SERP, so we could ensure that in all tasks the KM display would be displayed on the SERP, thus allowing us to choose between leaving it visible or hiding it, depending on the control (hidden) and experimental (visible) conditions.

The selected query patterns belonged to four different topics (celebrities, movies, athletes, sport teams) and required either single or multiple answers. An example of a single-answer query is "Who is the head coach of the team X?" while an example of a multi-answer query is "Who are X's children?". To diversify our search query pool, we produced three questions per query pattern, as can be seen in Table 1, while we introduced some additional multi-answer questions (marked with *) to increase the difficulty of the search tasks. In total, our query set included 144 different queries.[1] In the study, the query set was repeated as many times as needed to accommodate all participants. Each query was answered under each condition by at least two participants and at most six participants.

3.3 Mouse Cursor Tracking

As previously stated, all users performed the search tasks through a web proxy. This allowed us to automatically instrument all browsed pages with mouse cursor tracking. For this, we used EVTRACK,[2] an open source JavaScript event tracking library that is part of the smt2ϵ system [31]. EVTRACK makes it possible to specify what browser events should be captured and *how* they should be captured, i.e., via event listeners (the event is captured as soon as it is fired) or via event polling (the event is captured at fixed-time intervals). Concretely, we captured all regular browser events (e.g., `load`, `click`, `scroll`) via event listeners and only `mousemove` via event polling (at 150 ms), since this event may introduce unnecessary overhead both while recording on the client side and while transmitting the data to the server [29].

Whenever an event was recorded, we logged the following information (Fig. 2): cursor id (0 for desktop browsers, a number identifying the touch point for mobile browsers), mouse cursor position (x and y coordinates), timestamp,

[1] `http://personales.upv.es/luileito/kme/queries.tsv`
[2] `https://github.com/luileito/evtrack`

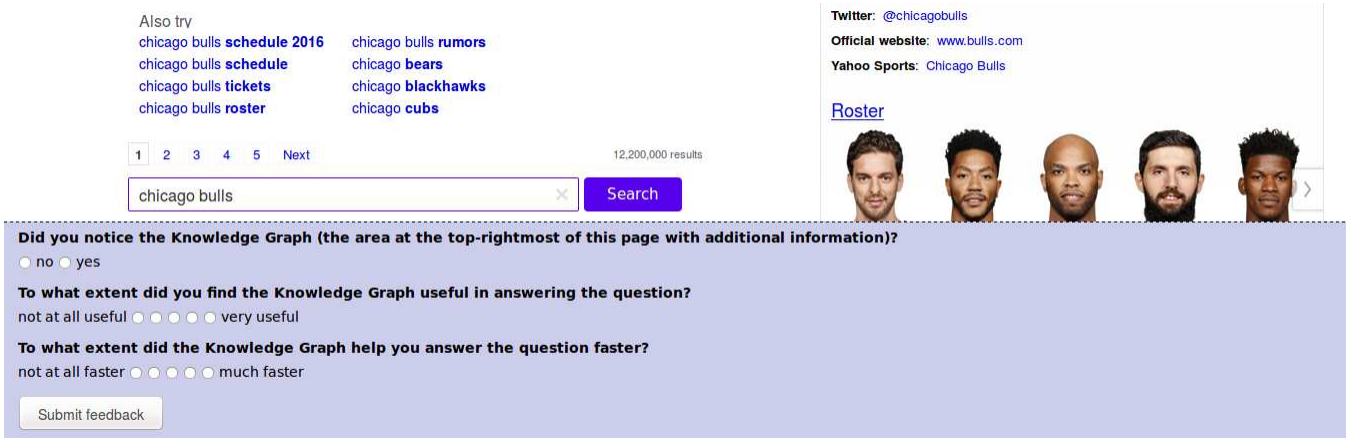


Figure 3: Example of the mini-questionnaire inserted at the bottom of the instrumented SERPs.

event name, xpath of the DOM element that relates to the event, DOM element attributes, and the Euclidean distance to 5 control points of the KM display (Fig. 2). The data were saved as CSV files, one browsed page at a time.

3.4 Self-Reported Measures of Engagement

In addition to recording mouse cursor data, the web proxy inserted a mini-questionnaire on the SERPs where the KM display was visible (experimental condition), in order to gather ground truth labels for the mouse cursor data; see Fig. 3. The mini-questionnaire was initially hidden, in order not to interfere with regular browsing, and was shown to the user just before leaving the SERP. The mini-questionnaire comprised 3 questions:

1. *Did you notice the knowledge module?* [yes/no]
2. *To what extent did you find the knowledge module useful in answering the question?* [1–5 Likert-type scale]
3. *To what extent did the knowledge module help you answer the question faster?* [1–5 Likert-type scale]

The labels of Likert-type scale was 1: not at all useful/faster, . . . , 5: very useful/faster (Fig. 3).

Our ground truth thus consists of three user engagement proxies, all of them being considered intrinsic components of engagement, as traditionally measured using questionnaires [36].

3.5 Participants

We recruited 612 participants through Amazon Mechanical Turk. From this initial sample, we approved assignments for 533 participants (female = 226, male = 307), aged from 18 to 66. Participants were of mixed nationality (e.g., American, Belgian, British, Finnish, German) and had varying educational backgrounds: 29.98% had a high school diploma, 18.98% had a college diploma, 41.56% had a BSc degree, 7.97% had an MSc and 1.52% had a PhD. All participants were proficient in English, 98.31% being native speakers. Finally, the majority were full-time (60.41%) or part-time (9.76%) employees while the remaining were either full-time students (7.69%), pursuing further studies while working (10.69%), performing home duties (6.75%) or other (4.69%).

3.6 Procedure

To begin, participants were informed about the terms and conditions of the study, followed by a short description of the SERP. The study had to be done in a single session. Participants could opt out at any moment, in which case they would not be compensated. Participants were asked to "evaluate two different backend systems of Yahoo web search by performing two search tasks". For each task, participants had to answer a question by searching for relevant information on the proxified search engine. As previously mentioned, in one task the KM display would be hidden (control condition) and in the other task it would be visible (experimental condition). The order of the tasks was randomized for each participant. Participants were also presented with a suggested query to begin their search, although they were free to submit additional queries (e.g., if the suggested query did not lead to the answer) and examine as many results as necessary to complete the search task. We used informational, entity-centric queries to introduce a common starting point across all participants. Upon finishing each task, participants were instructed to submit their answer and complete a post-task questionnaire. The study concluded with a demographics questionnaire. The payment for participation was $1.20 and each participant could take the study only once.

4. MODELLING USER ENGAGEMENT

In this section, we present our methodological approach for predicting the three user engagement proxies discussed in Section 3.4. By demonstrating that we can successfully model within-content interactions with DDs, using an inexpensive and scalable feedback like mouse cursor information, will allow us to derive more accurate and valid signals for predicting user engagement. In addition, with the proposed method we could infer with higher granularity which parts of a SERP or a web page the user truly notices and engages with, while lacking the additional cost of computationally expensive techniques for mouse cursor analysis.

4.1 Mouse Cursor Data

From our initial sample of 533 participants, we conclude to a subset of 300 participants, after excluding those cases which had incomplete mouse cursor logs. Our final dataset consists of 115, 699 cursor positions, collected during 600 search task sessions. Out of those 600 search task sessions, we further analyse the 300 cases that correspond to the experimental condition with the visible KM (Section 3.1) in the SERP. We note that cases are generally balanced, with 176 users having reported noticing the KM display. As a last step,

Table 2: Features used in the classification task to predict user engagement.

Base features	Meta-features	Aggregate functions*
Viewport (width, height)	# Moves (towards, away) KM	$x_{\min}$, $x_{\max}$
Cursor positions and timestamps	# Moves (towards, away) KM within dist. d	Σ, μ, $\tilde{x}$
Unique cursor positions	# Clicks (inside, outside) KM	σ^2, σ_χ, SST
Normalised viewport positions	Time to first click on KM	$\sum$ intra-distances of cursor positions w.r.t. KM
Unique normalised viewport pos.	# Preceding clicks to KM	Shannon entropy
Subsequent points' distance	# Hovers over KM	Permutation entropy ($w \in \{2, \ldots, 5\}$)
Subsequent points' duration	# Hovers over other elements	Weighted Permutation entropy ($w \in \{2, \ldots, 5\}$)
Cursor distance from KM	# Hovers over KM vs. other elements	Approximate entropy ($w \in \{2, \ldots, 5\}$)
Cursor speed	# Preceding hovers over other elements	FFT: i_{th} most powerful frequency ($i \in \{1, \ldots, 5\}$)
Cursor normalised speed	Time to first hover (KM, other elements)	Multivariate KL div. (symmetric, non-symmetric)
Cursor acceleration	Time hovering (KM, other elements)	Earth mover's distance
Cursor normalised acceleration	Distance traversed overall	Hausdorff distance
Cursor position status wrt. KM	Distance traversed (inside, outside) KM	
Vector angles	Distance from KM (corners, center)	
	# Cursor positions within distance d from KM	

* These functions are computed for most base and meta-features.

we normalise the values for each computed feature in the range $[0, 1]$ so that feature values that fall in greater numeric ranges do not dominate those in smaller numeric ranges.

4.2 Feature Engineering

Our task is to predict user engagement solely on the basis of inexpensive, easy-to-acquire user interaction signals. To this end, we explored a large number of basic as well as high-level meta-features that we engineered from the mouse cursor data we collected. We treated each data sequence as a time series and examined the statistical, spectral and temporal properties through the application of a number of aggregate functions. Each mouse cursor log was encoded as a feature vector of 638 components. In Table 2 we summarise these features under different categories and also list the aggregate functions applied to them. In what follows, we provide a brief description of the most important feature categories.

Temporal. Previous works [22, 35] have shown that accounting for the temporal characteristics of mouse cursor interactions can improve user profiling and prediction methods. Similarly, we considered the temporal dimension and measured (in milliseconds) the duration of cursor movements, the duration of hovering events and the total time up to the first click or hover, among others, both inside and outside of the KM display.

Spatial. Spatial features include the distance that the cursor has traveled overall. We considered both the Euclidean distance and the per-pixel travel distance on the x and y axes. In addition, we recorded the Euclidean distance of the mouse cursor to five reference points of the KM (Section 3.3). Finally, we recorded the status of the cursor position with respect to the KM display (inside, outside).

Direction. For every three subsequent mouse cursor positions, we determined the vectors they form and computed their angle in degrees. We also computed the direction of the mouse cursor movement with respect to the KM display.

Speed. The speed of mouse cursor movements has discriminative characteristics and can help disambiguate user intent [20]. For example, slow movements may indicate that the cursor is resting while the user is engaged in a cognitively demanding task such as reading carefully. On the other hand, ballistic movements suggest that the user is performing a quick scan to locate an information of interest in the text. In our analysis, we computed the speed for the distance that the mouse cursor travelled between subsequent pairs of positions. We considered both the Euclidean distance and the per-pixel travel distance on the x and y axes.

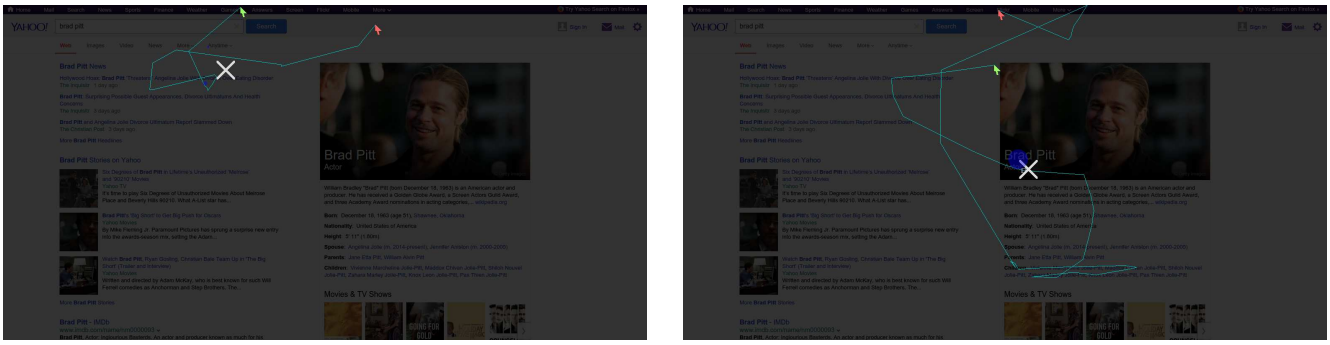
Acceleration. We also measured the acceleration for the distance that the mouse cursor travelled between three subsequent positions. As previously, we considered both the Euclidean distance and the pixel travel distance on the x and y axes.

Clicks. We considered the number of clicks performed inside and outside of the KM display, their ratio, as well as the number of preceding clicks prior to the first click on the KM display. The number of clicks has been used extensively in web search for decades, and is considered one of the most prominent features in user engagement research.

Descriptive statistics. This type of statistics provide simple, quantitative summaries of the data, such that patterns can emerge. In our analysis, we employed two general categories of descriptive statistics: measures of central tendency and measures of spread, motivated by the visualisations of the gathered mouse cursor trails (Figs. 4 and 5). The first category describes the central position of a frequency distribution for a dataset. Such examples in our feature set are the mean and the median. The second category describes how spread the scores in a dataset are distributed. To determine this spread, we computed the variance, sum of squares, standard deviation, kurtosis and skewness. Finally, we considered other simpler statistics, like the min, max and sum of values for each mouse cursor trail.

Distribution. These features include the sum of the mouse cursor positions' intra-distances, both inside and outside the KM display as well as overall, which indicate how compact or dispersed is the distribution of mouse cursor positions.

Shannon entropy. In information theory, entropy measures the disorder or uncertainty associated with a discrete, random variable, i.e., the expected value of the information in a message. The Shannon entropy [38] allows to estimate the average minimum number of bits needed to encode a string of symbols in binary form (if log base is 2) based on the alphabet size and the frequency of the symbols. Given



(a) Did not notice the KM display

(b) Noticed the KM display

Figure 4: Examples of the mouse cursor data by a user who did not notice the KM display (a) and a user who noticed the KM display (b). Green cursors represent the entry coordinate (first point of the mouse cursor trail). Red cursors represent the exit coordinate (last point of the mouse cursor trail). White crosses represent the centroid of the mouse cursor trail. Blue circles represent clusters of interacted areas.

a finite time series $X(t) = (x_t : 1 \leq t \leq T)$, the Shannon entropy can be expressed as

$$H(X) = -\sum_t p(x_t) \log p(x_t), \tag{1}$$

In our prediction task, we used Equation 1 to characterise the complexity of our mouse cursor trails and applied it to several meta-features. As we demonstrate later, this feature is useful in discriminating between engaged vs. non-engaged users, depending on the level of uncertainty exhibited by the mouse cursor data.

Permutation entropy. Permutation entropy [7] provides a fast and robust method for estimating the complexity of time series, by considering the temporal order of the values. More specifically, it calculates the variety of different permutations appearing at the components of a time series such as our mouse cursor trails.

Denoting S_n the set of all possible $n!$ permutations π of order n for a time series $X(t)$, the relative frequency for each $\pi \in S_n$ is defined as

$$p(\pi) = \frac{\#\{t \mid 0 \leq t \leq T - n, (x_{t+1}, \ldots, x_{t+n}) \text{ has type } \pi\}}{T - n + 1} \tag{2}$$

Then, the permutation entropy of order $n \geq 2$ is defined as

$$H(n) = -\sum_{\pi \in S_n} p(\pi) \log p(\pi) \tag{3}$$

This feature can be calculated for arbitrary real-world time series, and particularly in the presence of dynamical and observational noise. We computed the entropy for all permutations of order $n = 2, \ldots, 5$.

Weighted Permutation entropy. The Weighted Permutation entropy [15] extends the concept of Permutation entropy and addresses certain limitations such as its inability to differentiate between distinct patterns and their sensitivity.

This feature is computed in two steps. First, the weighted relative frequencies for each mouse cursor trail:

$$p_w(\pi_i^{n,\tau}) = \frac{\sum_{j \leq N} 1_{u:\text{type}(u)=\pi_i}(X_j^{n,\tau}) \cdot w_j}{\sum_{j \leq N} 1_{u:\text{type}(u)\in\prod}(X_j^{n,\tau}) \cdot w_j} \tag{4}$$

where $1_A(u)$ denotes the indicator function of set A defined as $1_A(u) = 1$ if $u \in A$ and $1_A(u) = 0$ if $u \notin A$. The Weighted Permutation entropy is then computed as

$$H_w(n, \tau) = -\sum_{i:\pi_i^{n,\tau} \in \prod} p_w(\pi_i^{n,\tau}) \log p_w(\pi_i^{n,\tau}) \tag{5}$$

where n and τ denote respectively the embedding dimension and time delay. The Weighted Permutation entropy is different from the Permutation entropy in the sense that the former is suitable for cursor trails with considerable amplitude information. For the range of trails that do not satisfy this property, the Permutation entropy might be a better alternative. We computed the entropy for all permutations of order $n = 2, \ldots, 5$.

Approximate entropy. The approximate entropy [37] expresses the (logarithmic) likelihood of similar patterns to be followed by similar observations. In other words, it quantifies the amount of regularity and the unpredictability of fluctuations in a time series. A low entropy indicates that the time series is deterministic, whereas a high value indicates randomness.

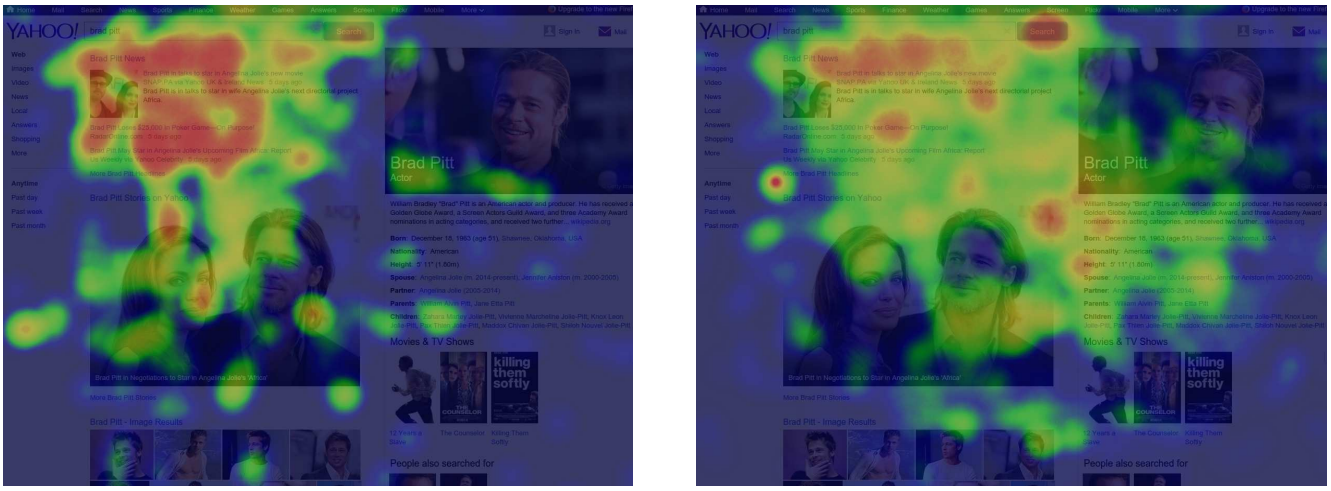
Denoting m the length of the compared data and r a filter factor (vector-wise comparison distance), we can define the approximate entropy ApEn of a time series $X(t)$ as

$$\text{ApEn}(m, r) = \Phi_m(r) - \Phi_{m+1}(r) \tag{6}$$

We computed the entropy for $m = 2, \ldots, 5$. We set $r = 0.2\,\text{SD}$, where SD is the standard deviation of the sequence values. The approximate entropy can be computed for any time series, chaotic or otherwise, at a low computational cost, and even for small data samples ($T < 50$).

Fast Fourier Transform. We performed a spectral analysis to determine the frequency components of our time series data. We used the fast Fourier transform (FFT), which is a more efficient way to compute the discrete Fourier transform (DFT). Given a time series $X(t)$ of length T and assuming a period T, the FFT computes two $T/2 + 1$ point frequency domain signals

$$X_k = \sum_t x_t \, e^{-i2\pi k \frac{t}{T}}, \quad k = 0 : T - 1 \tag{7}$$



(a) Did not notice the KM display

(b) Noticed the KM display

Figure 5: Heatmaps of mouse movements for all users who did not notice the KM display (a) and all users who noticed the KM display (b). The SERP shown here is an example to illustrate its parts and structure.

The two signals in the frequency domain are the real part and the imaginary part, and hold the amplitudes of the cosine and sine waves respectively. This frequency representation indicates how much of the variability of the data is due to low or high frequencies. In our analysis, we computed the amplitudes of all frequencies in our time series data and used their rankings as features, i.e., first most powerful frequency, second most powerful frequency, and so on.

Multivariate Kullback-Leibler divergence. We computed the multivariate Kullback-Leibler (MKL) divergence between 2 joint distributions:

$$D_{KL}(\mathcal{N}_1 \| \mathcal{N}_2) = \frac{1}{2}\Bigg[\log \frac{|\Sigma_1|}{|\Sigma_2|} + \mathrm{tr}(\Sigma_1^{-1}\Sigma_2) + (\mu_1 - \mu_2)^T \Sigma_1^{-1}(\mu_1 - \mu_2) - l \Bigg] \tag{8}$$

where μ is the mean vector, l is the length of the mean vector and Σ is the covariance matrix. The way we employed the multivariate KL divergence in our experiments is as follows. Initially, we used a set of training data (see Section 4.3) to learn the joint distributions of x and y mouse cursor positions for both engaged and non-engaged users. The browser viewport was normalised in $[0, 1]$ and discretised into 20 bins of size 0.05. This way, the cursor positions were split across the bins so that we could derive their distributions. Next, we computed the symmetric and non-symmetric KL distances between these two joint distributions and the training examples. Finally, we used these KL distances as features in our prediction task.

Earth Mover's Distance. Another feature we considered is the Earth Mover's distance (EMD), which is a measure of distance between two probability distributions. More specifically, the distributions are sets of weighted features that capture the distributions and the EMD is defined as the minimum amount of work needed to change one sequence into another. The notion of work is based on a unit of ground distance. Similarly to the KL divergence, we computed the EMD between the normalised mouse cursor positions in our training examples and the distributions of both engaged and non-engaged users.

Hausdorff Distance. The Hausdorff distance (HD) is the maximum distance of a set to the nearest point in the other set. More formally, the HD from set X to set Y is a maximin function defined as

$$d_H(A, B) = \max\left\{ \sup_{x \in X} \inf_{y \in Y} d(x, y),\ \sup_{y \in Y} \inf_{x \in X} d(x, y) \right\} \tag{9}$$

where sup is the supremum (least upper bound of a set), inf is the infimum (greatest lower bound of a set), x and y are coordinates of sets X and Y respectively and $d(x, y)$ can be any distance function (e.g., Euclidean) between these coordinates. We computed the HD for the different distributions of mouse cursor positions in the same manner as described in the previous two features.

4.3 Prediction Task

In this section we demonstrate that our modelling approach can predict successfully the three proxies of user engagement discussed in Section 3.4, and that it outperforms the standard baseline methods. The value of our prediction task lies in the fact that we use highly discriminative yet low-cost features. For each user engagement proxy, we trained a random forest (`RF`) classifier using the feature set described in Section 4.2.

As a first step, we performed a correlation analysis and excluded from our feature set those features that are highly correlated ($r \geq .80, p < .05$) and/or linearly dependent. Then, prior to training our models, we performed feature selection using a wrapper method that uses recursive feature elimination until finding the optimal feature set that maximises model's performance. To get performance estimates that account for the variation due to feature selection, we applied a 10-fold cross-validation and used the Area Under Curve (AUC) as the performance measure to optimise. By applying this step, we avoided overfitting our models and excluded noisy features that would not contribute to the accurate prediction of our classes.

Table 3: Top-10 features of each user engagement proxy according to the mean decrease in accuracy (MDA).

Attention (52 feat.)	MDA	Usefulness (103 feat.)	MDA	Task Duration (81 feat.)	MDA
CD_minCursorDistFromKGMiddle	.0105	CD_medianCursorDistFromKGTopRight	.0056	CD_meanCursorDistFromKGMiddle	.0063
CD_minCursorDistFromKGBottomRight	.0066	CD_meanCursorDistFromKGMiddle	.0054	E_consecutivePointsDistXYAPE2	.0053
H_noHoversKGOverNoHoversOtherElements	.0062	emd_None	.0042	CD_medianCursorDistFromKGTopRight	.0049
CD_cursorDistYOverCursorDistXYInsKG	.0061	emd_Notice	.0041	CD_meanCursorDistFromKGTopRight	.0044
CD_minCursorDistFromKGTopRight	.0057	S_sdCursorSpeed	.0033	H_noHoversKGOverNoHoversOtherElements	.0039
H_timeHoverKG	.0049	CD_cursorDistX	.0032	DL_noMovesTowardsKGWithinDistX	.0038
CP_noCursorPosInsKGOverNoCursorPosOutKG	.0045	E_consecutivePointsDistXYAPE2	.0031	emd_Notice	.0033
CD_cursorDistXOverCursorDistYInsKG	.0042	MKL_SymmetricNoneIn	.0030	MKL_SymmetricNoticeIn	.0024
H_timeHoverKGOverHoverOtherElements	.0042	MKL_SymmetricNoticeIn	.0028	E_consecutivePointsDistXYAPE4	.0021
S_medianCursorSpeedInsKG	.0041	S_meanCursorSpeedOutKG	.0026	MKL_SymmetricNoneIn	.0021

Next, we performed a 10-fold cross-validation using stratified sampling, to create balanced splits of the data that preserve the overall class distribution. In each fold, we used 90% of the data for training and 10% for testing. Additionally, we held out a validation set from the training set for fine-tuning the classifier's hyperparameters (e.g., ϵ-threshold, number of trees). We then applied the optimal parameter values to our final model and evaluated its performance against the test set, and in comparison to several baselines.

Our choice of baselines was informed by existing research as well as current practices in industry [19, 20, 25, 26, 42]. More specifically, we considered if the user has clicked on the KM display (`hasClickedKM`, binary classifier), if the mouse cursor has hovered over the KM display (`hasHoveredKM`, binary classifier), and the time spent on the page (`dwellTime`) as a feature to the same `RF` classifier. For assessing the models' performance, we considered the standard IR metrics of precision, recall and accuracy. Traditionally, the most frequently used metrics are accuracy and error rate. However, metrics like accuracy can be deceiving in certain situations and are highly sensitive to changes in data [23]. Therefore, we also computed the F-Measure, which combines precision and recall as a measure of the effectiveness of classification in terms of the weighted importance on either recall or precision as determined by the β coefficient (we use $\beta = 1$). Last, because F-Measure is sensitive to data distribution, we used as an additional performance criterion the AUC.

4.4 Results

In what follows, we report the results of our prediction task for each user engagement proxy. A Kruskal-Wallis test for stochastic dominance was statistically significant in all cases. We therefore conducted a *post-hoc* analysis involving multiple pairwise comparisons, for which we corrected the level of significance to control the false discovery rate by using the Benjamini-Hochberg correction [8]. We note that the final performance values reported in Table 4 are the macro-averages across all ten folds. We highlight all cases where our model performs significantly better than the different baselines and provide the corresponding Z statistic.

4.4.1 Attention

The first proxy of user engagement we predicted is the noticeability of the KM display. More specifically, we were interested in detecting accurately if the KM display captures the user attention even at the absence of events such as clicks or hovers. For training our model, we used a subset of 52 features (some examples are shown in Table 3) from our initial feature set of 638 features. The top section of Table 4 reports the performance of our classification model in comparison to the different baselines. As we can observe, our model's predictive performance is better than any of the competitor baselines. This improvement is evident across all performance metric that we considered, but more importantly with respect to F-Measure, where our model introduces an improvement over all baselines of 13.8% (`hasClickedKM`), 15.6% (`hasHoveredKM`), 24.6% (`dwellTime`) and 9% (joint model) respectively. Another encouraging result is our model's performance with respect to the AUC which, given existing research conventions, can be considered as excellent/good. On the other hand, the AUC achieved by the baselines indicates no discrimination between the predicted classes.

4.4.2 Usefulness

The next proxy of user engagement that we considered is the usefulness of the KM display. The highly subjective nature of this engagement metric suggests that it is a more challenging task, something which is made evident by the overall performance degradation observed in the `hasClickedKM` baseline. Nevertheless, our model appears to be the best performer in comparison to the other baselines and maintains its advantage (using an extended set of 103 features shown in Table 3), as indicated by the reported measures. From the results shown in the middle section of Table 4, we can see that the difference between our model and the other baselines in terms of F-Measure is widened to 44.1% (`hasClickedKM`), 9.8% (`hasHoveredKM`), 7.2% (`dwellTime`) and 2% (joint model) respectively.

4.4.3 Perceived Task Duration

We conclude our prediction task with a final proxy of user engagement: the perceived task duration. Here, we aimed to predict whether users felt that they completed their task faster, given that this belief is attributed to the examined DD (in our case the KM display). For this task, we learned a `RF` model using the 81 top-ranked features shown in Table 3. We performed the same 10-fold cross-validation to ensure that our model's performance is not over-optimistic. As shown in the bottom section of Table 4, the performance of our model did not change much despite the prediction challenge that the targeted concept posses. The differences in the F-Measure scores between our model and the baselines (39% for `hasClickedKM`, 14.3% for `hasHoveredKM`, 9.5% for `dwellTime` and 10% for the model that combines all baselines) indicate once more the capacity of the proposed method to capture this aspect of engagement and provide accurate predictions on future user engagement for a DD.

4.5 Computational Complexity

As a side contribution, we comment on the computational complexity of recent approaches to mouse cursor analysis, in terms of temporal cost, and compare them with our proposed method. On the one hand, the Mouse Gestures technique [2]

Table 4: Performance metrics for the proposed method and the different baselines. Scores in parentheses are Dunn's pairwise Z statistic. A bold typeface denotes the best result in a row.

	Performance Metric	Our method	hasClickedKM	hasHoveredKM	dwellTime	All baselines
Attention	(Weighted Avg.) Precision	**.77**	.62 (3.08)**	.67 (1.96)	.50 (4.81)***	.74 (0.56)
	(Weighted Avg.) Recall	**.76**	.62 (3.21)*	.61 (3.58)***	.52 (5.38)***	.69 (1.57)
	(Weighted Avg.) F-Measure	**.76**	.58 (3.86)***	.60 (3.48)***	.51 (5.23)***	.68 (1.53)
	(Weighted Avg.) Accuracy	**.76**	.62 (3.21)*	.61 (3.58)***	.52 (5.36)***	.67 (1.51)
	AUC	**.86**	.42 (5.31)***	.51 (4.18)***	.53 (3.59)***	.72 (1.51)
Usefulness	(Weighted Avg.) Precision	**.74**	**.74** (0.43)	.69 (0.86)	.62 (2.76)*	.65 (2.23)
	(Weighted Avg.) Recall	**.74**	.33 (4.86)***	.61 (3.02)**	.73 (0.39)	**.74** (0.64)
	(Weighted Avg.) F-Measure	**.74**	.30 (5.56)***	.64 (2.78)**	.66 (2.07)	.68 (2.05)
	(Weighted Avg.) Accuracy	**.74**	.33 (4.86)***	.61 (3.02)**	.73 (0.39)	**.74** (0.64)
	AUC	**.71**	.45 (4.80)***	.58 (2.22)*	.57 (2.47)*	.60 (2.24)*
Task Dur.	(Weighted Avg.) Precision	.74	**.77** (-0.96)	.63 (1.88)	.62 (2.01)	.64 (1.55)
	(Weighted Avg.) Recall	**.73**	.40 (4.77)***	.57 (2.81)**	.70 (0.86)	.66 (1.28)
	(Weighted Avg.) F-Measure	**.73**	.34 (5.46)***	.59 (2.67)**	.64 (2.16)*	.63 (2.16)*
	(Weighted Avg.) Accuracy	**.73**	.40 (4.77)***	.57 (2.81)**	.70 (0.86)	.66 (1.28)
	AUC	**.77**	.41 (5.60)***	.55 (3.41)**	.62 (2.25)*	.59 (2.76)**

Significance levels (two tails, corrected for multiple comparisons): $^{*}p < .05$; $^{**}p < .01$; $^{***}p < .001$.

relies on principal component analysis (PCA) preprocessing and k-means clustering. The cost of PCA is $\mathcal{O}(p^2N + p^3)$ (covariance matrix computation + eigen value decomposition) with p features and N data points, whereas k-means has $\mathcal{O}(i\,c\,N)$ with i iterations and c clusters. On the other hand, the algorithms used in Cursor Motifs [27, 32] use both dynamic time warping (DTW), which has cost $\mathcal{O}(N^2)$, and k-nearest neighbours (kNN), which has cost $\mathcal{O}(N^2k^2)$ with k neighbours. Our method has computations of $\mathcal{O}(N)$ (linear) or $\mathcal{O}(N \log N)$ (quasilinear) cost, instead of the cubic and quadratic cost associated to the other approaches. Most important, our method is straightforward to implement and highly discriminative. As such, we expect that it may be of practical value in several problems that make use of mouse cursor analysis.

5. DISCUSSION AND CONCLUSIONS

With the rising presence of direct answers in search results where no click is required to acquire relevant information, the need to understand the utility of within-page interactions becomes critical. Whether a DD like the KM display is useful is hard to determine, particularly if the information the user is seeking is on the module itself. To this end, this work has examined the impact of DDs in web search, providing empirical evidence of their overall utility. We conducted a crowdsourcing study that revealed the potential benefits of using mouse cursor data to predict user engagement with DDs. In particular, we showed that our feature selection model outperforms the standard baselines to measure three user engagement proxies with the KM display.

With respect to the noticeability proxy, our initial results clearly suggest that it is possible to predict when the user attention is captured by a DD like the KM display using only a simple, yet highly discriminative, set of features derived from mouse cursor activity. This is an important finding considering that existing online user engagement metrics assume a uniform engagement with the web page content and do not distinguish well enough between attended and ignored DDs. If we can predict accurately if a DD was truly noticed although it was not clicked or hovered, then we can be more confident that the user engaged with it and improve our *true negative* prediction rate. On the other hand, if we can predict when a DD was indeed not noticed, although it was clicked or hovered, we can reduce our *false negative* rate.

Regarding our second user engagement proxy, usefulness of the KM display, we observe that the `dwellTime` model managed, to some extent, to capture how users engage with the DDs by considering the amount of time they spend on the page. In fact, the model that combines the three baselines performs equally good as ours in terms of accuracy and recall. However, it lacks the precision of our model. In other words, when that model predicts the positive class it will be less often correct and, as a result, will not provide trustworthy judgements. Our model's ability to predict more accurately when a user finds a DD useful has important implications on the methodology for understanding the impact of launching a new DD, modifying its existing design, and how that change may affect web search UIs.

In the final proxy of user engagement, perceived task duration, our model was able to discriminate more accurately than any baseline when users felt that they have completed their task faster, given that this belief is attributed to the examined DD. This information, combined with the previous grounds truths that we predicted successfully allows us to understand better, for example, how users engage with ads, images or videos; something which existing click-based models do not address adequately due to their current biases and limitations.

While our prediction tasks may be useful for descriptive analysis, the main practical use of our models is perhaps to automatically select or lay out the DDs. This is an interesting research avenue, because DDs are optional for the SERPs and so the user behaviour could provide signals about whether DDs should be shown or not in particular queries. Ultimately, modelling user engagement with DDs has wide-ranging applications in web search ranking, evaluation and interface design. Therefore we anticipate that further research on this topic may have an important impact on future web search interfaces.

Acknowledgments

We thank B. Barla Cambazoglu and Marios Koulakis for fruitful discussions. This work was partially supported by the International Excellence VLC/CAMPUS program.

6. REFERENCES

[1] I. Arapakis, M. Lalmas, B. B. Cambazoglu, M.-C. Marcos, and J. M. Jose. User engagement in online news: Under the scope of sentiment, interest, affect, and gaze. *JASIST*, 65(10), 2014.

[2] I. Arapakis, M. Lalmas, and G. Valkanas. Understanding within-content engagement through pattern analysis of mouse gestures. In *Proc. CIKM*, 2014.

[3] I. Arapakis, L. A. Leiva, and B. B. Cambazoglu. Know your onions: Understanding the user experience with the knowledge module in web search. In *Proc. CIKM*, 2015.

[4] E. Arroyo, S. Sullivan, and T. Selker. CarCoach: A polite and effective driving coach. In *Proc. CHI EA*, 2006.

[5] R. Atterer, M. Wnuk, and A. Schmidt. Knowing the User's Every Move - User Activity Tracking for Website Usability Evaluation and Implicit Interaction. In *Proc. WWW*, 2006.

[6] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *Proc. ISWC*, 2007.

[7] C. Bandt and B. Pompe. Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.*, 88, Apr 2002.

[8] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate - a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, 57(1):289–300, 1995.

[9] B. Bi, H. Ma, B.-J. P. Hsu, W. Chu, K. Wang, and J. Cho. Learning to recommend related entities to search users. In *Proc. WSDM*, 2015.

[10] R. Blanco, B. B. Cambazoglu, P. Mika, and N. Torzec. Entity recommendations in web search. In *Proc. ISWC*. 2013.

[11] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proc. SIGMOD*, 2008.

[12] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *Proc. IUI*, 2001.

[13] A. Diriye, R. W. White, G. Buscher, and S. Dumais. Leaving so soon?: understanding and predicting web search abandonment rationales. In *Proc. CIKM*, 2012.

[14] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proc. SIGKDD*, 2014.

[15] B. Fadlallah, B. Chen, A. Keil, and J. Príncipe. Weighted-permutation entropy: A complexity measure for time series incorporating amplitude information. *Phys. Rev. E*, 87(2), 2013.

[16] H. A. Feild, J. Allan, and R. Jones. Predicting searcher frustration. In *Proc. SIGIR*, 2010.

[17] J. Goecks and J. Shavlik. Learning users' interests by unobtrusively observing their normal behavior. In *Proc. IUI*, 2000.

[18] Q. Guo and E. Agichtein. Exploring mouse movements for inferring query intent. In *Proc. SIGIR*, 2008.

[19] Q. Guo and E. Agichtein. Ready to buy or just browsing? Detecting web searcher goals from interaction data. In *Proc. SIGIR*, 2010.

[20] Q. Guo and E. Agichtein. Beyond dwell time: estimating document relevance from cursor movements and other post-click searcher behavior. In *Proc. WWW*, 2012.

[21] Q. Guo, D. Lagun, and E. Agichtein. Predicting web search success with fine-grained interaction data. In *Proc. CIKM*, 2012.

[22] Q. Guo, D. Lagun, D. Savenkov, and Q. Liu. Improving relevance prediction by addressing biases and sparsity in web search click data. In *WSCD Workshop*, 2012.

[23] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE TKDE*, 21(9):1263–1284, 2009.

[24] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Intell.*, 194, 2013.

[25] J. Huang, R. W. White, G. Buscher, and K. Wang. Improving searcher models using mouse cursor activity. In *Proc. SIGIR*, 2012.

[26] J. Huang, R. W. White, and S. Dumais. No clicks, no problem: using cursor movements to understand and improve search. In *Proc. CHI*, 2011.

[27] D. Lagun, M. Ageev, Q. Guo, and E. Agichtein. Discovering common motifs in cursor movement data for improving web search. In *Proc. WSDM*, 2014.

[28] L. A. Leiva. Restyling website design via touch-based interactions. In *Proc. MobileHCI*, 2011.

[29] L. A. Leiva and J. Huang. Building a better mousetrap: Compressing mouse cursor activity for web analytics. *Inf. Process. & Manage.*, 51(2), 2015.

[30] L. A. Leiva and E. Vidal. Assessing user's interactions for clustering web documents: a pragmatic approach. In *Proc. Hypertext*, 2010.

[31] L. A. Leiva and R. Vivó. Web browsing behavior analysis and interactive hypervideo. *ACM TWEB*, 7(4), 2013.

[32] Y. Liu, Y. Chen, J. Tang, J. Sun, M. Zhang, S. Ma, and X. Zhu. Different users, different opinions: Predicting search satisfaction with mouse movement information. In *Proc. SIGIR*, 2015.

[33] W. Mason and S. Suri. Conducting Behavioral Research on Amazon's Mechanical Turk. *Behav. Res. Methods*, 44(1), 2010.

[34] I. Miliaraki, R. Blanco, and M. Lalmas. From "Selena Gomez" to "Marlon Brando": Understanding explorative entity search. In *Proc. WWW*, 2015.

[35] V. Navalpakkam and E. Churchill. Mouse tracking: measuring and predicting users' experience of web-based content. In *Proc. CHI*, 2012.

[36] H. L. O'Brien and E. G. Toms. The development and evaluation of a survey to measure user engagement. *JASIST*, 61(1), 2010.

[37] S. M. Pincus. Approximate entropy as a measure of system complexity. *PNAS*, 88(6), 1991.

[38] C. Shannon. A mathematical theory of communication. *Bell Syst. Tech.*, 27(3), 1948.

[39] 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 *Proc. SAC*, 2006.

[40] M. Speicher, A. Both, and M. Gaedke. Tellmyrelevance! predicting the relevance of web search results from cursor interactions. In *Proc. CIKM*, 2013.

[41] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge. In *Proc. WWW*, 2007.

[42] D. Warnock and M. Lalmas. An exploration of cursor tracking data. In *arXiv:1502.00317*, 2015.