Exploring Client-Side Instrumentation for Personalized Search Intent Inference: Preliminary Experiments

Qi Guo and Eugene Agichtein
Mathematics & Computer Science Department
Emory University
{qguo3,eugene}@mathcs.emory.edu

Abstract

Clickthrough on search results have been successfully used to infer user interest and preferences, but are often noisy and potentially ambiguous. The reason mainly lies in that the clickthrough features are inherently a representation of the majority of user intents, rather than the information needs of the individual users for a given query instance. In this paper, we explore how to recover *personalized search intent* for each search instance, using a more sensitive and rich client-side instrumentation (including mouse movements) to provide additional insights into the intent behind each query instance. We report preliminary results of learning to infer query intent over rich instrumentation of search result pages. In particular, we explore whether we can automatically distinguish the different query classes such as navigational vs. informational queries. Our preliminary results confirm our intuition that client-side instrumentation is superior for personalized user intent inference, and suggest interesting avenues for future exploration.

Introduction

Recovering user intent is an important yet very difficult problem. For example, it would be helpful for a search engine to know if a query intent is primarily *navigational* (that is, to find a particular known website) or *informational* (that is, to find information about a topic). While traditional, server-side approaches typically assign a small set of most popular intents to a query, it has been shown by previous work that user goals vary a great deal, even if expressed with the same keyword query.

In this paper, we begin to explore the issues related to automatically determining query intent based on *client-side* instrumentation, that allows an unprecedented access to "observe" user actions as she is performing a search, in real time. In particular, we wish to understand what information about client-side instrumentation is most useful for intent inference, and how it compares with the more exclusive, but less precise, server-side instrumentation currently performed by modern web search engines.

As a first natural step in our exploration we focus on the basic task of query intent (or user goal) classification into the "traditional" categories of navigational, informational, and transactional queries, originated by Broder (Broder 2002), and refined by (Rose and Levinson 2004):

- *Navigational*: A user has in mind a known, specific site, and searches for it using the whole or part of the site domain name.
- *Informational*: A user searches for any information about a topic
- *Transactional/Resource*: A user is trying to perform a transaction, such as buying a book online. This is related to the navigational queries, but differs in that the user does not (explicitly) specify the website where the transaction would take place.

In fact, there are many exceptions and ambiguities in this general classification. As (Rose and Levinson 2004) pointed out, sometimes the same query may have multiple meanings. For example, the query "obama" could be informational, navigational or even transactional. People may search to know more about Barak Obama, or to visit his official website, or perhaps the user goal is to donate money online to support Mr. Obama's campaign (i.e., a resource/transactional query). Therefore, to classify the *query* into a single intent would be incorrect. What is really necessary is to classify user goal for each *query instance* – that is, the particular search done by the user. (Note that exploiting personal user models directly may not solve this problem, as user goals vary between search sessions).

Unfortunately, query instance classification is a far more difficult problem than query type classification (i.e., to assign a dominant intent to all instances of same keyword query). To address this problem, we explore client-side instrumentation to capture the user interaction in real time, thus allowing us to predict the intent of the individual query instance. Not surprisingly, our experiments confirm that in some cases the intent of the same keyword query varies drastically by user, and more importantly, that we can *automatically* distinguish between some intent types simply by properly modeling the client-side behavior, while knowing nothing about the user's prior history or expectations. We also explore the benefits of combining the server-side and client-side features to make the prediction more accurate.

In summary, our contributions include:

- **Practical lightweight client instrumentation for web search**: We present CSIP, a practical, deployed lightweight implementation of client-side instrumentation that allows unprecedented access to fine-grained user interactions (Section 3).
- **Results of experimentation with real user interactions**: We demonstrate the feasibility of our approach by showing that CSIP achieves high accuracy intent classification, with limited amounts of training data (Section 5)
- **Preliminary result analysis exploring the benefits of client vs. server-side instrumentation**: We report our analysis of the results focusing on the difficult and ambiguous cases that we believe can be successfully tackled with our approach (Section 6).

Next we briefly review related work to set the context for our paper.

Related Work

The origins of user modeling research can be traced to library and information science research of the 1980s. In 1982 Belkin et al., (Belkin, Oddy, and Brooks 1982) introduced an influential user model ASK (for Anomalous States of Knowledge). An excellent overview of the traditional "pre-Web" user modeling research is available in (Belkin 1997). With the explosion of the popularity of the web, and with increasing availability of large amounts of user data, the area of modeling users, user intent, and in general web usage mining has become an active area of research in the information retrieval and data mining communities.

In particular, inferring user intent in web search has been studied extensively, including references (e.g., (Rose and Levinson 2004; Lee, Liu, and Cho 2005; Agichtein et al. 2006; White and Drucker 2007; White, Bilenko, and Cucerzan 2007)). Taxonomies of web search and user goals have been relatively stable since Broder's classic paper classifying intent into Navigational, Transactional and Informational (Broder 2002). Recently, topical commercial query classification was presented in (Rose and Levinson 2004).

Previous research on user behavior modeling for web search focused on aggregated behavior of users to improve web search (Mobasher et al. 2002; Agichtein, Brill, and Dumais 2006; Clarke et al. 2007) or to study other general aspects of behavior (White, Bilenko, and Cucerzan 2007; Downey, Dumais, and Horvitz 2007). However, it has been shown that user goals and experience vary widely (e.g., (White and Drucker 2007)) and have significant effects on user behavior. Hence, methods for personalization of user models have been proposed (Mobasher, Cooley, and Srivastava 2000; Shen, Tan, and Zhai 2005; Sieg, Mobasher, and Burke 2007) that include more precise characterization of user profiles for more fine-grained modeling and more effective implicit relevance feedback.

These studies have primarily focused on indicators such as clickthrough to disambiguate queries and recover intent and model user goals. Recently, eye tracking has started to emerge as a useful technology for understanding some of the mechanisms behind user behavior (e.g., (Joachims et al. 2007; Cutrell and Guan 2007).

In this paper we begin to explore using *mouse movements* to disambiguate, classify, and infer intent of queries. However, there have been indications that mouse movements (e.g., page scrolling) correlate with user interest (Fox et al. 2005), and could be used for better implicit feedback. Previous work on mouse movements has shown a correlation between eye movement and mouse movements (e.g., (Rodden and Fu 2006; Phillips and Triggs 2001)). In other work, researchers have shown the value of mouse movement tracking for usability analysis (Mueller and Lockerd 2001) and (Atterer, Wnuk, and Schmidt 2006) and activity tracking. In other work, protocols were proposed to track all user actions (including mouse movements and text of web pages), accompanied by talk-aloud qualitative analysis of user behavior (Card et al. 2001). However, we are not aware of previous work on using mouse movements to *automatically* infer user intent, or to automatically classify queries into broad classes such as navigational vs. informational.

In particular, we posit that automatically modeling user behavior with rich *client-side* instrumentation can allow to distinguish true user intent where server-side instrumentation does not have sufficient information. For example, people often repeat web searches, most often to re-find information they have seen in the past (Teevan et al. 2007). While these queries might appear to be informational to the server-side models, in fact it is almost trivial to identify such searches as navigational when considering client-side instrumentation, which we describe next.

CSIP: Client-Side Intent Predictor

We now describe our system. Our goal is to capture as much information as possible about the user interactions, while remaining lightweight (that is, not to negatively impact the user's experience).

Client-side instrumentation using LibX

For our research, we developed a minor modification of the Firefox version of the open source LibX toolbar [1] In particular, we used a simple javascript code to sample the mouse movements on the pre-specified web search result pages, and other interactions. The mouse move events (and other events, such as printing a page) were encoded into a string and when the buffer of the events was exceeded, are sent to the server.

These toolbars were installed on the public-use shared terminals in the Emory University library; Furthermore, all the users opted in to participate in our study, and no directly identifiable user information was stored, protecting the privacy of the participants. As we will see, our instrumentation still captures interesting interaction patterns, despite not being tied to particular user identity or profile.

Our approach is to represent client-side interactions as *feature vectors* and then apply standard machine learning/classificaiton methods to classify query instances according to user intent. Naturally, the information sources, and the feature representation user are crucial for accurate

[1] Available at `www.libx.org`.

classification, and in the rest of the section we focus on these issues.

In general, our goal is to perform *full* client side instrumentation for query intent inference, which would combine the interactions with the user profile and server-side information about other users. The general structure is illustrated in Figure 1. While we plan to incorporate additional information sources in the future, our current implementation focuses on the *Query Text*, *Server-Side/Clickthrough* and *Client-Side/Real-time interactions*. Most notably, we are not yet modeling user history, which is the topic of our future work.

Representing user interactions

We now describe our information sources captured and the corresponding feature representations.

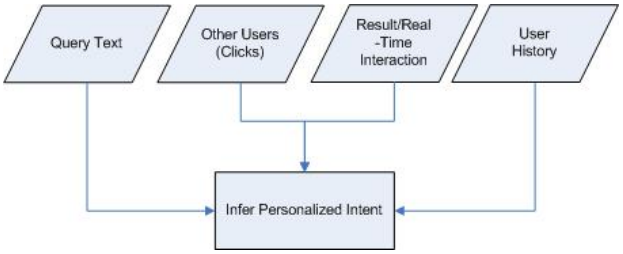


Figure 1: Overview of CSIP

Query Text Query text is the most intuitive and simple feature for inferring user intent - generally, an informational query is likely to contain more terms than a navigational or a transactional query. Therefore, we include query length (number of terms) as a feature for all the classifiers presented in this paper.

Other User/Server-Side clickthrough For the server-side information, the additional features include click distribution (the fraction of the result with most clickthrough over all the clicks), average deliberation time (i.e., time until the first click on any of the results) and similarity between a clicked search result URL and the query (i.e., whether the query is a substring of the URL with the clicked URL, potentially indicating a navigational query).

Real-Time Interaction/Client-side instrumentation For the client-side information, we primarily focus on the mouse movements and the corresponding mouse move trajectories.

CS–Client:Simple: First, we consider a naive representation, where we use simple mouse movement features such as the length, vertical range, and horizontal range of trajectories. Our reasoning behind this simple representation is based on the observation that the range and length of mouse trajectories will differ between purely navigational and purely information queries (e.g., information queries are likely to require longer and wider range of mouse movements to find and click on a relevant result). In our experiments, this limited representation corresponds to the **CS** for Client:Simple intent classifier.

CF–Client:Full: As we will show, the naive representation above is not rich enough to capture the possible information hidden in the mouse movements. Our second representation (to which we will refer as **CF**–for Client:Full–) attempts to capture the physiological characteristics of the mouse trajectories, inspired by the work of (Phillips and Triggs 2001). In particular, we attempt to capture properties such as *speed*, *acceleration*, *rotation* and other precise characteristics of the mouse movements.

To distinguish the behavioral patterns in different stages of the mouse movements, we split each mouse trajectory into five parts: initial, early, middle, late, and end. Each of the five parts contains 20% of the sample points of the trajectories. Next, we approximate each part as a segment and for each segment we computed the average speed, average acceleration, slope and the rotation angle between the current segment and the segment connecting the beginning and the end (the click position) of the trajectories. In this version, we simplified each part of trajectories as a segment and represent the mouse movement along each segment as a constant velocity.

Our hypothesis for this representation is that for informational queries, the mouse is more likely to switch between speed up (when the user finds something interesting and moves the mouse towards it)and slow down (when the user begins reading or is about to click) several times and is more likely to go back and forth (rotation angles change several times) than for a navigational query. Similarly, other characteristics like the slope of the trajectory may also vary. More details about the features we have used for these two representations are given in Table 1.

Learning to classify intent

For our initial exploration we use standard supervised machine learning classification techniques. In particular, we use the Weka [2] implementation of the standard classifiers such as Support Vector Machines and decision trees. In future work we plan to explore more specialized machine learning methods, since our interactions are inherently temporal. In particular, we plan to better represent the mouse trajectories, which can be modeled more accurately without the binning/discretization step of considering only the fixed segment features. Nevertheless, even standard classifiers are able to demonstrate the feasibility and the benefits of using client-side instrumentation over the more traditional server-side models. We describe our empirical studies next.

Experimental Setup

In this section, we describe how we gathered and labeled the datasets user for our experiments, the precise definition of the intent classification tasks we use for our case study, and the evaluation metrics used to compare the different methods.

[2] At `http://www.cs.waikato.ac.nz/ml/weka/`.

Feature	*Specification*	*Server*	*Client*
TopFraction	Given a query, the fraction of its most frequent clicked URL over all its clicked URLs	Y	
IsSubstring	True if the query is a substring of its most frequent clicked URL, False if not; Unknown if no clicks	Y	
DeliberationTime	The time before a user first time clicked a result, -1 if no clicks	Y	
QueryLength	Number of terms of the given query	Y	Y
TrajectoryLength	The length of the mouse move trajectories		Y
VerticalRange	The vertical range of the mouse move trajectories		Y
HorizontalRange	The horizontal range of the mouse move trajectories		Y
Segments	We split each trajecotries evenly into five segments and represent each segment with its avgSpeed, avgAcceleration, slope and rotationAngle		Y
AvgSpeed	the distance between two end points of the given segments, over the time elapsed between these two points.		Y
AvgAcceleration	the average acceleration is computed by assuming that each segment represents a Constant Acceleration Motion		Y
Slope	the slope of each segment		Y
RotationAngle	the rotation angle between the current segment and the segment connecting the beginning and the end (the click position) of the trajectories		Y

Table 1: Feature Specification

Dataset

The data was gathered from mid-January 2008 until mid-March 2008 from the public-used machines in the Emory University libraries. The dataset statistics are reported in Table 2. The population was primarily undergraduate college students who agreed to opt-in for this study. The identify of the participants is unknown.

For this preliminary study we focused on only *initial queries* that is, avoiding follow-up queries in same search session. The dataset statistics are summarized in Table 2), consisting of around 1500 initial query instances, with their Google general search result page, next URLs, and mouse move trajectories.

From this set, we randomly sampled 300 query instances (without replacement, only including the first instance of each query) into our sample. The number was chosen as a reasonable initial pilot study – large enough to be interesting, and small enough to allow careful human labeling of the "correct" classification of the intent, according to the tasks, defined next.

Finally, we complemented our local dataset with server-side interactions obtained from a large log (15 million queries and corresponding interactions) from a commercial search engine, which contained the query and the click-through from all the users who issued the query. No identifiable user information was available, so there was no way to determine if queries were issued by same or different users.

Statistic	*Total*
Number of users	860
Number of search sessions	1,597
Number of queries	3,214
Average trajectory length (px)	1,068
Average vertical range (px)	324
Average horizontal range (px)	537

Table 2: Dataset statistics

Specific intent classification tasks

In this paper, we focus on the following four tasks:

- **Task 1**: Classify a query instance into Navigational / Informational / Transactional. This is the "traditional" intent classification task.
- **Task 2**: Same as Task 1, but do not distinguish between Transactional and Navigational queries. As we will see, Transactional queries actually are quite similar to Navigational queries, and there is often ambiguity between the two goals even for a human annotator. So, we re-label all transactional queries as navigational.
- **Task 3**: Same as Task 2, but consider *re-finding* queries (i.e., those queries where the user is using a query as a "bookmark" to return to previously found site) as navigational. This task is more exploratory, and we discuss it in more detail in Section 6.
- **Task 4**: Same as Task 3, but identify and ignore *likely Failed* queries (i.e., queries with none of the results were clicked). Similar to Task 3, this task is perhaps even more subjective as the annotators had to guess whether a (real) user was satisfied or not with the result set. Nevertheless, this task is quite interesting, and we explore it further in Section 6.

In summary, our main experimentation focuses on Tasks 1 and 2 – both traditional query classification tasks with established annotation guidelines to distinguish the query classes. We describe the manual annotation process next.

Creating manual "truth" labels

To manually classify query instance intent, we "replayed" the user interactions with the results for each query in the sample drawing the corresponding mouse trajectories, query terms, and next URL (often the URL of a clicked result), on a snapshot of the result page. Using these clues and our intuition we then labeled the query intent into one the classes, also marking searches that had ambiguous intent (e.g., we

could not determine whether a query was navigational or informational). To illustrate the input we used to label the queries manually, Figure 2 illustrates a sample of two navigational queries; Figure 3 reports a sample of two informational queries; and Figure 4 illustrates a sample of two transactional queries.

The labeled dataset statistics are reported in Table 3. Note that 14% of the searches in the sample were ambiguous, and an additional 3% of the searches could not be "replayed" as the original search result page could not be recovered. These 17% of the searches were discarded as we did not have reasonable way of labeling the corresponding "correct" query intent.

Label	*Number*	*Percentage*
Navigational	89	29.67%
Informational	147	49.00%
Transactional	13	4.33%
Error	9	3.00%
Ambiguous	42	14.00%

Table 3: Distribution of search intent in 300 query sample

Metrics

Having obtained a set of our best guesses at the intent as described above, we can compare the prediction accuracy of various methods. In particular, we use standard information retrieval and classification metrics:

- **Accuracy**: The fraction of all the query instances that were correctly assigned the query intent label (compared to manual label).
- **F1**: Macro-averaged F1 measure computed for each class, averaged across all classes. This complementary metric can help capture the difference in performance for skewed class distributions (where Accuracy might be misleading). The F1 measure for each class is computed as $2 \cdot PR/(P+R)$ where P is precision (i.e., fraction of predicted class instances that are correct) and R is recall (fraction of all true class instances correctly identified).

These two metrics give a complete picture of overall performance as well as performance for each intent class.

Methods Compared

We summarize the main methods used for intent prediction.

- **S**: Server-side instrumentation only (e.g., query text, URL, clickthrough), trained by identifying instances of the manually labeled queries described above that were submitted to the Microsoft Live Search engine.
- **CS**: Simple client-side features only (e.g., mouse move range)
- **CF**: More sophisticated, full client-side features (e.g., mouse trajectory representation)
- **CSIP**: Combination of both full client-side instrumentation (CF) and the server-side instrumentation (S), thereby using all the available information.

Results

We now report the main experimental results for this study. First, we experimented with the different classification methods (as our focus is on the behavior representation, we simply wanted to choose the best "off-the-shelf" classifier for our task). We experimented with many of the available Weka classifier implementations to find the most accurate classifier for each feature set/representation. As as result of these preliminary experiments we chose the Weka implementation of the C4.5 classifier (J48) as the most accurate classifier for the S method, and the SVM implementation (SMO) for the client-side methods (CS, CF, and CSIP).

First, we consider the accuracy of the different methods on the original Task 1. The results, produced using 4-fold cross-validation, are summarized in Table 4. In this case, CS consistently outperforms S, for a modest gain on all metrics. However, CF (full client) performs substantially better than CS, indicating the benefit of our fine-grained mouse trajectory analysis. Finally, the integrated CSIP system (that combines both full client and server side analysis) has the highest accuracy and F1 measure of all systems. Interestingly, the improvement of CSIP over CF is not large, suggesting that the most benefit comes from the query-instance client-side behavior, and not from the server-side information aggregated across all users issuing the same query – allowing CSIP to have higher accuracy than S by as much as 17%.

Method	*Accuracy (%)*	*F1*			
		Nav	*Info*	*Trans*	*Macro Average*
S	65.46	46.20	76.60	0	40.93
CS	67.70 (+3%)	57.90	76.3	0	44.73(+9%)
CF	75.50 (+15%)	69.00	82.40	0	50.47 (+23%)
CSIP	**76.31 (+17%)**	**71.30**	**83.10**	0	**51.47(+26%)**

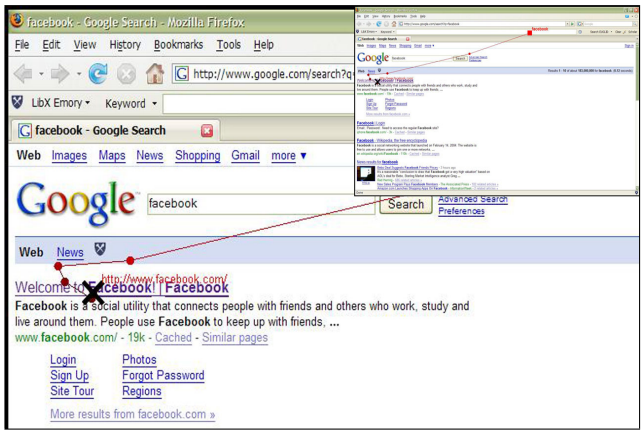
Table 4: Accuracy and F1 for different methods (Task 1)

As we discussed, transactional queries are very similar to navigational queries – both in intent and in resulting behavior. Table 5 reports the accuracy of the different methods if transactional queries are re-labeled as navigational, as has been done in previous work (e.g., (Lee, Liu, and Cho 2005). Nor surprisingly, the accuracy of all the methods increases for this "easier" task. The gain of the CF and the CSIP methods over the baseline server-side or simple client classifiers remains consistent and substantial.

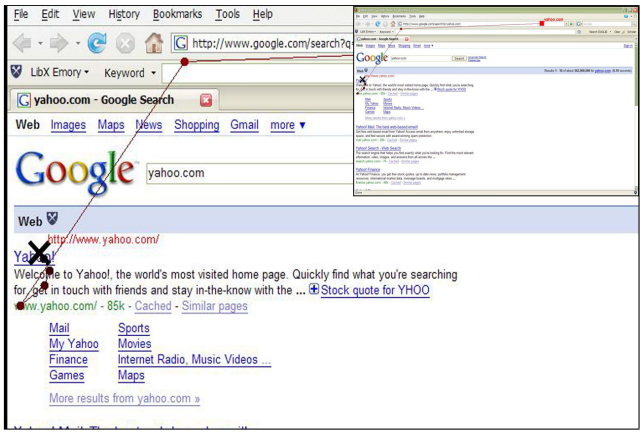
Method	*Accuracy (%)*	*F1*		
		Nav	*Info*	*Macro Average*
S	67.87	49.40	76.50	62.95
CS	70.28 (+4%)	68.60	71.80	70.20 (+12%)
CF	78.71 (+16%)	72.30	**82.70**	77.50 (+23%)
CSIP	**79.92 (+18%)**	**76.60**	82.40	**79.50 (+26%)**

Table 5: Accuracy and F1 for different methods (Task 2)

To better understand the contribution of the different features we report the information gain of each feature (computed for Task 2) in Table 6. As we can see, the most important features represent different aspects of mouse trajec-

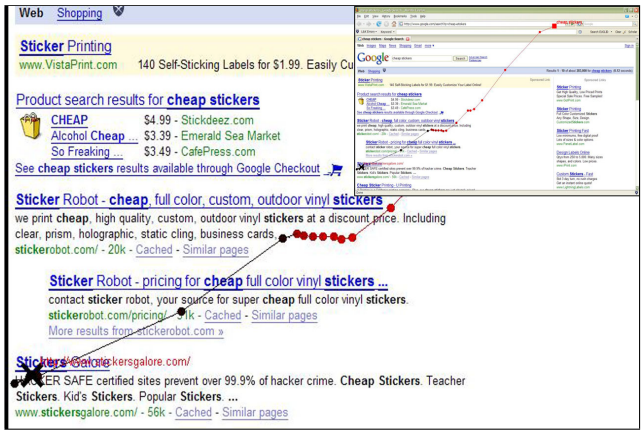


Query: “facebook”

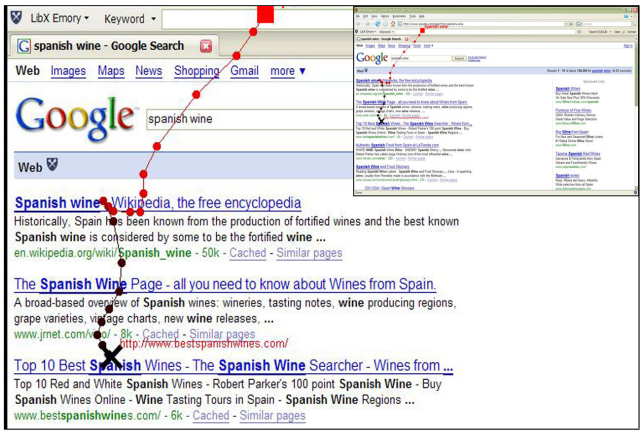


Query: “yahoo.com”

Figure 2: Two examples of navigational intent

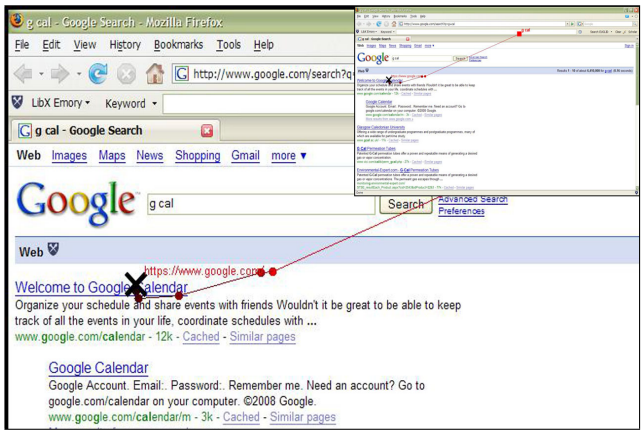


Query: “cheap stickers”

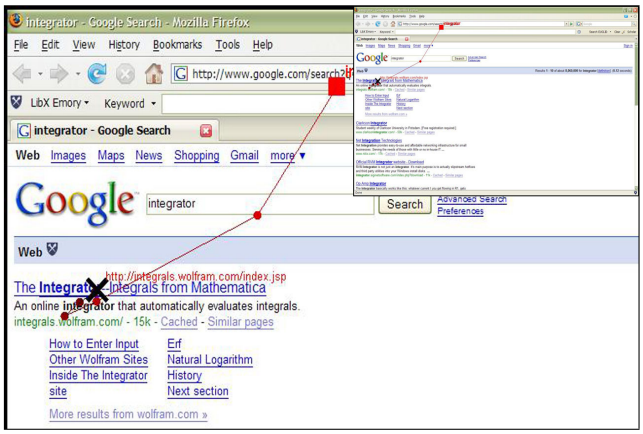


Query: “spanish wine”

Figure 3: Two examples of informational intent



Query: “g cal”



Query: “integrator”

Figure 4: Two examples of transactional intent

tories (e.g., speed, acceleration, rotation) but also include query length and deliberation time – more traditional user modeling features.

Information Gain	*Feature*
0.2043	AvgAcceleration (segment 3)
0.197	AvgAcceleration (segment 2)
0.1705	AvgSpeed (segment 3)
0.1509	AvgSpeed (segment 4)
0.1451	VerticalRange
0.1449	AvgAcceleration (segment 4)
0.1425	AvgAcceleration (segment 1)
0.1275	TrajectoryLength
0.1146	TopFraction
0.1125	RotationAngle (segment 0)
0.0922	AvgSpeed (segment 2)
0.0843	QueryLength
0.0781	IsSubstring
0.075	AvgAcceleration (segment 0)
0.0708	DeliberationTime

Table 6: Most important CSIP features (ranked by Information Gain)

In summary, we have shown that our fine-grained client side instrumentation (CF) and the integrated client- and server-side method (CSIP) exhibit promising performance, resulting in substantially higher accuracy than server-side or naive client instrumentation. We now analyze our results in more depth, considering other query intent tasks.

Discussion

We now consider in more detail some of the ambiguous queries that we discarded from experiments in the previous section. We re-examined the ambiguous queries and attempted our best guesses at their intent. We believe that 27 of the ambiguous queries are probably *re-finding* queries – that is, queries that appear informational based on the text of the query, but are really "bookmarks" to re-retrieve previously found website. Our guess is based on the observation that in these cases the users did not even read the results before the click, which is very similar to the user behavior of typical navigational queries. Although according to the query text, a query instance might look like informational, such as "rpi rankings" and "emory financial aid"(illustration of these two queries are given in Figure 5), it is very likely that the user intent was actually navigational since he had visited the page or he assumed that there were such a page. As a result, we labeled these 27 of the ambiguous queries as likely re-finding queries. For the other 15 queries, as we could not determine the intent, we continue to discard them for the remaining experiments. Two examples of such ambiguous queries are illustrated in Figure 6. We labeled the query "opus emory" is ambiguous because the user checked many results before she clicked on the promising top one result due to some unknown intent. And we labeled the query "canada energy" as ambiguous because there is not enough information to tell whether the intent was navigational or informational: it is possible that the user wanted to learn about "canada energy" and accidentally clicked on the "canada energy sector" page, and it is also possible that the intent really was to find the home page of "canada energy sector". To clearly label this search, more information is required.

We will also re-examine the cases of the *failed* queries (i.e., those with no click on any result). These presumed Failed queries were originally included in our experiments in the previous section. The statistics of the presumed occurrences of the re-finding and failed queries in our labeled dataset are reported in Table 7. As we can see, while refinding queries are relatively rare (9% of our sample), the failed queries are quite frequent (28.33% of our sample).

Label	*Number*	*Percentage*
Ambiguous (Refinding)	27	9.00%
Ambiguous (Unknown)	15	5.00%
Failed (No clicks)	85	28.33%

Table 7: Distribution of the presumed Re-finding and Failed searches in labeled dataset

Re-Finding queries

First we explore the refinding queries in more detail (which we refer to as Task 3). An example of a re-finding query is shown in Figure 5. If we consider re-finding queries to be navigational in intent (e.g., the user has found the site before using the same query), and relabel them accordingly, the behavior of classifiers changes drastically. We report the results in Table 8. In particular, the *client-side-only* method, CF, substantially outperforms the combined CSIP method (79.71% accuracy for CF vs. 77.53% accuracy for CSIP). This result illustrates that when query intent is indeed *personalized* – that is, for the current user, the normally informational query is actually navigational – then the client-only classifier is more accurate, and incorporating the "majority" intent in fact degrades performance.

However, further investigation is needed to distinguish these likely refinding queries from just "easy" queries (i.e., the search engine results are so good that the user does not need to read the results before a click). To address this problem, we plan to use User History. In this paper, we incorporate this part of queries as navigational and try to figure out whether the client-side instrumentation can help identify this kind of queries. As the result shows in Task 3 and 4, the gap of the performance between the Client-side based classifiers and the Server-side based classifiers is enlarged; and because of its integration of both Client-side and Server-side features, CSIP perform a little bit worse than the pure Client-side Full classifier.

Method	*Accuracy (%)*	*F1*		
		Nav	*Info*	*Macro Average*
S	64.49	49.00	72.80	60.90
CS	71.38(+11%)	72.70	70.00	71.35(+17%)
CF	**79.71(+24%)**	**78.50**	**80.80**	**79.65(+31%)**
CSIP	77.53(+20%)	77.40	77.70	77.55(+27%)

Table 8: Accuracy and F1 for different methods (Task 3)

Failed queries

The other prominent case is that of "Failed" queries. We can easily identify these by construction (that is, by defining Failed queries to be those with no click). Two examples of possible failed queries are illustrated in Figure 7. One difficulty of recovering intent of this type of query lies in that sometimes the behavioral pattern of a failed query appears similar to a navigational query. For example, if a user misspells the query, or none of the results appear relevant, the user will immediately click the "did you mean" feature, or refine the query or even give up. Alternatively, the user may have gotten the needed information from the result summaries, which is the real reason that there is no click on a result. In contrast, if a query intent is navigational but the desired result does not return on the top or even does not return, the user might spend much time reading, which is similar to informational query. However, we believe that the reading pattern of navigational query and informational query should be different due to the different intent of reading - for navigational, reading is more likely a glance at the title while for informational query, reading is more likely to be scrutinizing on the snippets. Further investigation on the disambiguation of these cases will be very important in improving the intent prediction.

For the sake of exploration, suppose we discard all failed queries from our dataset (to which we refer as Task 4). The results for this (easier) task are reported in Table 9. As we can see, the Accuracy and F1 of *all* methods increase substantially. Interestingly, the CF classifier is the most accurate (achieving Accuracy of 83.59% vs. accuracy of 82.56% achieved by CSIP), indicating that when re-finding queries are treated according to the *individual* user intent (i.e., as navigational queries) and when the failed queries are not considered, server-side information (i.e., information about behavior of other users) is not helpful for individual/query-instance identification.

Method	*Accuracy (%)*	*F1*		
		Nav	*Info*	*Macro Average*
S	68.21	76.20	52.30	64.25
CS	76.41(+12%)	70.10	72.80	75.30(+17%)
CF	**83.59(+23%)**	**86.70**	**78.70**	**82.70(+29%)**
CSIP	82.56(+21%)	86.00	77.00	81.50(+27%)

Table 9: Accuracy and F1 for different methods (Task 4)

Error Analysis

Finally, to gain a better insight about the prediction results, we conducted case studies and compare the server-side and full client-side classification results, to identify cases where one classifier outperformed the other and cases that neither of the classifier predicted correctly. Our findings are summarized in Table 10. As indicated in the table, the difficult queries for our CSIP method are mainly shorter easy/refinding informational queries (eg. "asters") and rare/unknown navigational queries with possible reading behavior (lower speed in some stages, eg. "aiesec"). The trajectories of the two such difficult queries are shown in Figure 8.

In summary, CSIP can identify navigational intent for relatively rare queries, including re-finding queries and navigational queries for obscure websites (either of these two cases are not likely to have substantial clickthrough information in the query logs). As another promising feature, CSIP can identify informational intent for queries that resemble navigational queries (for example, the query coincides with a name of a web site), but is actually an informational query.

Conclusions

We presented a preliminary exploration of using rich client-side instrumentation, in combination with server-side query logs, to infer and disambiguate search intent. As we have shown, some queries, while they may appear to be navigational or informational, are in fact ambiguous – and the mouse trajectories and other client-side information can be successfully used to identify such cases and ultimately to help infer the underlying user intent. Our results are promising and suggest interesting directions for future work, namely to develop tailored machine-learning algorithms for our query intent prediction task, and to apply our methods to other intent prediction tasks such as user satisfaction or predicting query performance.

Acknowledgments

We thank Microsoft Research for providing the MSN Search query logs and for partially supporting this research. We also thank Arhur Murphy, Selden Deemer, and Kyle Felton of the Emory University Libraries for their support with the data collection and many valuable discussions.

References

Agichtein, E.; Brill, E.; Dumais, S.; and Ragno, R. 2006. Learning user interaction models for predicting web search result preferences. In *Proc. of SIGIR*, 3–10.

Agichtein, E.; Brill, E.; and Dumais, S. 2006. Improving web search ranking by incorporating user behavior information. In *Proc. of SIGIR*, 19–26.

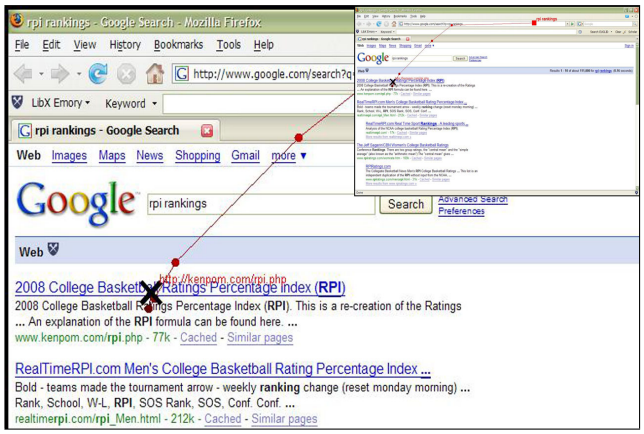
Atterer, R.; Wnuk, M.; and Schmidt, A. 2006. Knowing the users every move: user activity tracking for website usability evaluation and implicit interaction. In *Proc. of WWW*, 203–212.

Belkin, N.; Oddy, R. N.; and Brooks, H. M. 1982. Information retrieval: Part ii. results of a design study. *Journal of Documentation* 38(3):145–164.

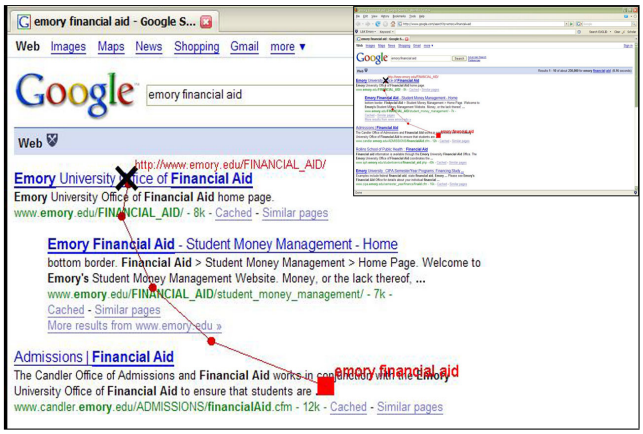
Belkin, N. J. 1997. User modeling in information retrieval. *Tutorial presented at the Sixth International Conference on User Modelling (UM97)*.

Broder, A. 2002. A taxonomy of web search. *SIGIR Forum* 36(2).

Card, S. K.; Pirolli, P.; Wege, M. V. D.; Morrison, J. B.; Reeder, R. W.; Schraedley, P. K.; and Boshart, J. 2001. Information scent as a driver of web behavior graphs: results of a protocol analysis method for web usability. In *Proc. of CHI*, 498–505.

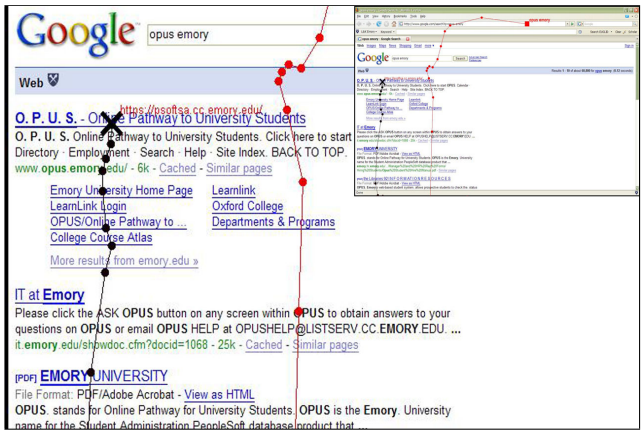


Query: "rpi rankings"



Query: "emory financial aid"

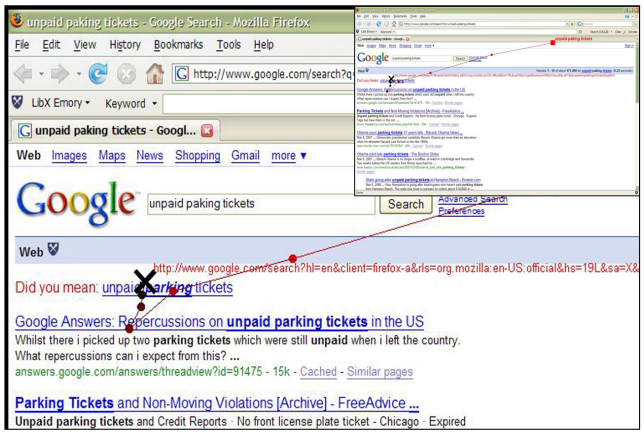
Figure 5: Two examples of easy/refinding intent



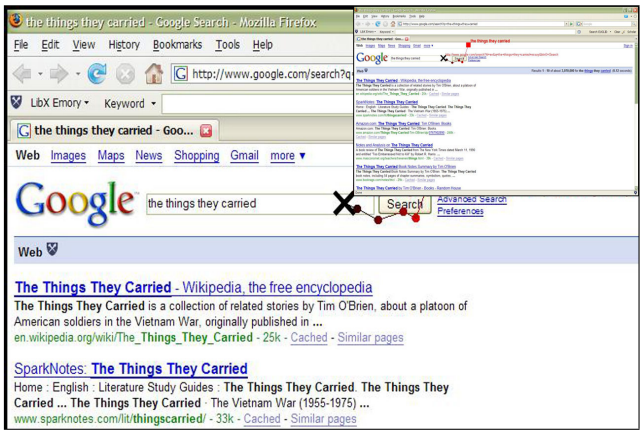
Query: "opus emory"

Query: "canada energy"

Figure 6: Two examples of ambiguous intent



Query: "unpaid paking ticket"

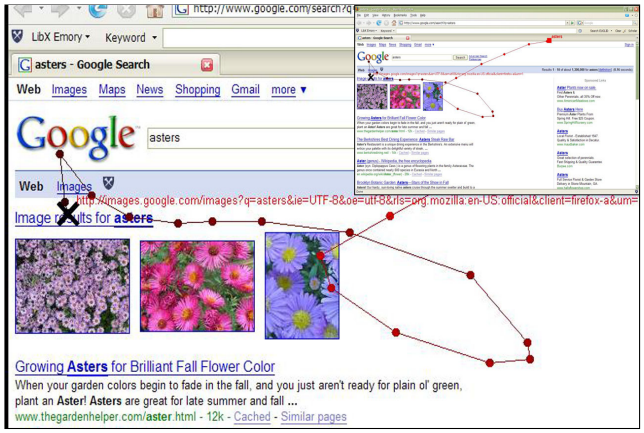


Query: "the things they carried"

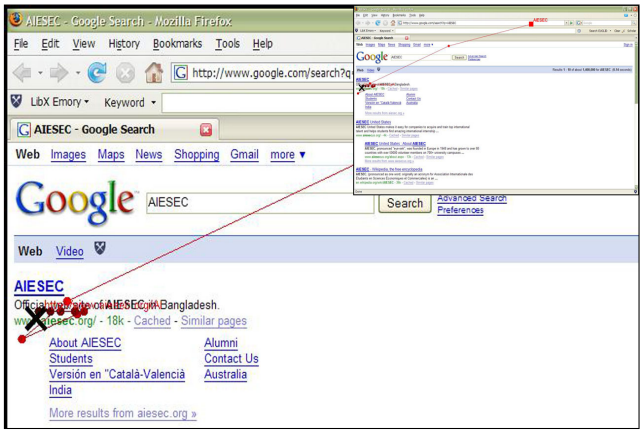
Figure 7: Two examples of probably failed informational query: misspell, refine without click

Method	*Correctly Classified Cases*	*Incorrectly Classified Cases*
S	1. long informational queries	1. rare/unknown navigational queries 2. short informational queries
CF	1. navigational queries 2. relatively difficult informational queries	1. easy/refinding informational queries 2. navigational queries with possible reading behavior
CSIP	1. navigational queries 2. long or relatively difficult informational queries	1. short easy/refinding informational queries 2. rare/unknown navigational queries with possible reading behavior

Table 10: Summary of Error Analysis



Query: "asters"



Query: "aiesec"

Figure 8: Two types of difficult queries for CSIP

Clarke, C. L. A.; Agichtein, E.; Dumais, S.; and White, R. W. 2007. The influence of caption features on clickthrough patterns in web search. In *Proc. of SIGIR*, 135–142.

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

Downey, D.; Dumais, S. T.; and Horvitz, E. 2007. Models of searching and browsing: Languages, studies, and application. In *Proc. of IJCAI*, 2740–2747.

Fox, S.; Karnawat, K.; Mydland, M.; Dumais, S.; and White, T. 2005. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems* 23(2):147–168.

Joachims, T.; Granka, L.; Pan, B.; Hembrooke, H.; Radlinski, F.; and Gay, G. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.* 25(2).

Lee, U.; Liu, Z.; and Cho, J. 2005. Automatic identification of user goals in web search. In *Proc. of WWW*, 391–400.

Mobasher, B.; Dai, H.; Luo, T.; and Nakagawa, M. 2002. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery* 6(1):61–82.

Mobasher, B.; Cooley, R.; and Srivastava, J. 2000. Automatic personalization based on web usage mining. *Commun. ACM* 43(8):142–151.

Mueller, F., and Lockerd, A. 2001. Cheese: tracking mouse movement activity on websites, a tool for user modeling. In *Proc. of CHI*, 279–280.

Phillips, J. G., and Triggs, T. J. 2001. Characteristics of cursor trajectories controlled by the computer mouse. *Ergonomics* 44(5):527–536.

Rodden, K., and Fu, X. 2006. Exploring how mouse movements relate to eye movements on web search results pages. In *Web Information Seeking and Interaction Workshop*.

Rose, D. E., and Levinson, D. 2004. Understanding user goals in web search. In *Proc. of WWW*, 13–19.

Shen, X.; Tan, B.; and Zhai, C. 2005. Implicit user modeling for personalized search. In *Proc. of CIKM*, 824–831.

Sieg, A.; Mobasher, B.; and Burke, R. 2007. Web search personalization with ontological user profiles. In *Proc. of CIKM*, 525–534.

Teevan, J.; Adar, E.; Jones, R.; and Potts, M. A. S. 2007. Information re-retrieval: repeat queries in yahoo's logs. In *Proc. of SIGIR*, 151–158.

White, R. W., and Drucker, S. M. 2007. Investigating behavioral variability in web search. In *Proc. of WWW*, 21–30.

White, R.; Bilenko, M.; and Cucerzan, S. 2007. Studying the use of popular destinations to enhance web search interaction. In *Proc. of SIGIR*, 159–166.