Exploring Searcher Frustration

Henry A. Feild

November 9, 2009

Abstract

When search engine users have trouble finding what they are looking for, they become frustrated. Across two user studies, we found that roughly a third of queries submitted end with users being moderately to extremely frustrated. By modeling searcher frustration, search engines can predict the current state of user frustration, tailor the search experience to help users find what they are looking for, and avert them from switching to another search engine. We describe several reoccurring causes of frustration and present several models to predict frustration—using query log and physical sensor features—that significantly outperform the baseline.

1 Introduction

In this work, we investigate *searcher frustration*. We consider a user frustrated in the context of information retrieval (IR) when their search process is impeded. A frustration model capable of predicting how frustrated searchers are throughout their search is useful retrospectively to collect statistics about the effectiveness of a search system. More importantly, it allows for real-time system intervention of frustrated searchers, hopefully preventing users from leaving for another search engine or just giving up. Evidence from users' interactions with the search engine during a task can be used to predict a user's level of frustration. Depending on the level of frustration and some classification of the *type* of frustration, the system can change the underlying retrieval algorithm or the actual interface. For example, we posit that one common cause or type of frustration is a user's inability to formulate a query for their otherwise well defined information need.

One way that a system could adapt to address this kind of frustration is to show the user a conceptual break down of the results; rather than listing all results, group them based on the key concepts that best represent them. Using a well worn example, if a user enters 'java', they can see the results based on 'islands', 'programming languages', 'coffee', etc. Of course, most search engines already strive to diversify result sets, so documents relating to all of these different facets of 'java' are present, but they might not be clear to some users, causing them to become frustrated.

An example from the IR literature of a system that adapts based on a user model is work by White, Jose, and Ruthven (2006). They used implicit relevance feedback to detect changes in users' information needs and alter the retrieval strategy based on the degree of change. Our work is similar, but we want to detect frustrated behavior, and adapt the system based on the type of frustration, regardless of the information need itself.

While automatic frustration modeling has not been specifically investigated in the IR literature, it has in the area of intelligent tutoring systems (ITS) research. When a system is tutoring a student, it is helpful to track that student's affective state, including frustration, in order to adapt the tutoring process to engage the student as much as possible. Our research borrows heavily from the tools used in and insights gleaned from the ITS literature.

The goals for our line of research are as follows: first, determine how to detect a user's level of frustration; second, determine what the key causes or types of frustration are; and third, determine

the kinds of system interventions that can reduce different types of frustration. This work explores the reasons users become frustrated and the question of whether frustration can be accurately predicted using features derived from query logs and physical sensors.

The remainder of this paper is organized as follows. In Section 2 we will discuss two bodies of work that were critical in the formation of this research. Then we will describe the setup of two user studies and a high-level analysis of the collected data, such as the causes of frustration reported by users, in Sections 3 and 4. Next, we will introduce several models to predict frustration and a discussion of how they performed in Section 5. Finally, we will wrap up with our conclusions and describe our next steps in this vein of research.

2 Related Work

Our research is based heavily on two bodies of work: one from the IR literature and the other from the intelligent tutoring systems (ITS) literature. Before we describe those, we will describe how frustration fits in with user satisfaction in the IR literature.

2.1 Frustration and satisfaction

Recall that we define frustration in the context of IR as the impediment of search progress. Frustration has not been directly studied in the field of IR, but *searcher satisfaction* has. Satisfaction in search can have different meanings. Several studies have left the meaning largely up to the subjects; Fox et al. (2005) and Huffman and Hochster (2007) asked participants to rate their satisfaction with the entire search process on a scale. Another approach is to collect searcher satisfaction scores for a range of search attributes, such as the accuracy and coverage of search results (Al-Maskari et al., 2007).

We define satisfaction as the fulfillment of a need or want, which in the case of IR is a user's information need. While satisfaction and frustration are closely related, they are distinct. As a consequence, searchers can ultimately satisfy their information need (i.e., be satisfied), but still have been quite frustrated in the process (Ceaparu et al., 2004).

In previous work, satisfaction has been examined at the task or session level[1] (Al-Maskari et al., 2007; Fox et al., 2005; Huffman and Hochster, 2007). In other words, the overarching information need and the collection of queries used to address that need are considered, rather than individual queries themselves; these satisfaction models only cover user satisfaction *after* a task has been completed, not *while* a task is in progress.

Searcher satisfaction models are useful for retrospective analysis and improvement. However, this only helps the user experience in future searches and does nothing for dissatisfied searchers—there is no place for adaptive retrieval models other real-time solutions. However, with a frustration model that is defined throughout a search, these real-time solutions are available.

2.2 Predicting searcher satisfaction

Fox et al. (2005) conducted a study to determine if there is an association between features derived from query logs[2] and explicit user satisfaction. In addition, they explored which implicit measures are most highly associated with satisfaction and what patterns of user interaction are associated with different levels of satisfaction (they refer to this as *gene analysis*).

This study used an Internet Explorer browser plugin to log the data. The subjects were Microsoft employees and all tasks were user generated. The explicit measures were collected via two pop-up windows. The first was displayed to users after they navigated away from a non-search engine page; it

[1] We will consider *task* and *session* interchangeable in this research.

[2] Query logs, also referred to as transaction logs, contain information about users' interactions with a search system, such as the queries they enter and the results on which they click.

asked users to indicate one of the following concerning the page: 1) *I liked it*, 2) *It was interesting, but I need more information*, 3) *I didn't like it*, or 4) *I did not get a chance to evaluate it (broken link, foreign language, etc.)*.

The second dialog prompted users after every task (i.e., after the user was finished searching for some information need) and they were asked to mark one of the following with respect to the task: 1) I was satisfied with the search, 2) I was partially satisfied with the search, or 3) I was not satisfied with the search.

Fox et al. found there exists an association between query log features and searcher satisfaction, with the most predictive features being click through, the time spent on the search result page, and the manner in which a user ended a search. They found many interesting patterns. For example, of the search tasks that consisted of the user entering a query, visiting one result, and then ending the task, 81% resulted in the user being satisfied, while 10% ended in partial satisfaction and 7% in dissatisfaction. In another pattern where the user enters a query and then clicks on four or more results for that query, 51% end in user dissatisfaction, while 35% are partially satisfied and only 13% are completely satisfied.

2.3 Detecting ITS user emotion

Cooper et al. (2009) describe a study in which students using an intelligent tutoring system were outfitted with four sensors: a mental state camera that focused on the student's face, a skin conductance bracelet, a pressure sensitive mouse, and a chair seat capable of detecting posture. The goal of the study was to ascertain if using features drawn from the sensor readings in combination with features extracted from user interaction logs with the ITS could more accurately model the user's affective state than using the interaction logs alone.

The emotional states considered were *interest*, *excitement*, *confidence*, and *frustration*. To get a grounding for truth, the students were prompted every five minutes[3] with the question, *How [interested / excited / confident / frustrated] do you feel right now?* At each prompting, students were only asked about one emotion, which was randomly chosen. Students were asked to respond by rating their level of the respective affective state on a five-point scale, with the middle point being neutral and either end the extreme (e.g., *anxious* versus *very confident*).

Cooper et al. found that across the three experiments they conducted, the mental state camera was the best stand-alone sensor to use in conjunction with the tutoring interaction logs for determining frustration. However, using features from all sensors and the interaction logs performed best. They used step-wise regression to develop a model for describing each emotion. For frustration, the most significant features were from the interaction logs and the camera, though features from all sensors were considered in the regression.

Using this model, Cooper et al. mapped the five-point emotion ratings into the range -1 (the emotion level was not high) to 1 (was high) and performed a leave-one-out classification. For frustration, this resulted in an accuracy of 89.7%; the baseline—guessing that the emotional state is always low—resulted in an accuracy of 85.29%. While the baseline is high, in another study using the same sensors, but different features, Kapoor, Burleson, and Picard (2007) created a model that was capable of classifying when the user of an ITS was going to click an *I'm frustrated!* button with 79% accuracy and a chance accuracy of 58%.

3 User Search Studies

To understand and model frustration in the IR context, it is necessary to have labeled data from users of search systems. While there are several sets of query logs available from various search companies (e.g., MSN, AOL, and UpToDate), these sets lack three features that are key to our research goals. First, these are all server-side logs—logs that were collected by the respective search company, which can only see

[3] Students were not prompted if they were solving a problem (Cooper et al., 2009).

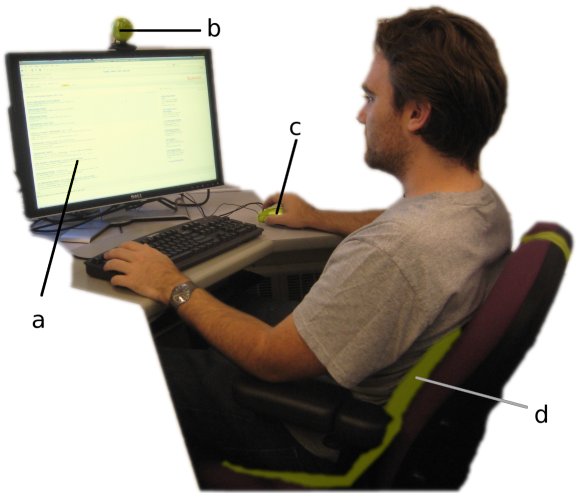


Figure 1: The setup used for both user studies. The equipment is as follows: a) a Web browser plugin to log user interactions during each task; b) a camera, which reports confidence values for six affective states; c) a pressure sensitive mouse; and d) a pressure sensitive seat cushion.

what a user does on their search results pages. These logs do not capture if and when users switched to another search engine, so there is potentially a chunk of each user's search session missing. Second, they do not have explicit feedback from the users concerning their current state of frustration and third, there were no physical sensors present to monitor the users while they searched. To collect these three key features, we decided to conduct user studies, where subjects could be monitored by sensors, we could collect rich, client-side logs of their interactions with the Web, and we could ask them for explicit feedback about their frustration during a search.

We conducted our user studies in two phases. We first ran a pilot study with fifteen users in late July 2009. After analyzing much of the data and fixing bugs with the logging software, we conducted a thirty-person study in the middle of October 2009. We will first outline the general study structure that is common to both studies and then describe the aspects unique to each study.

3.1 General Setup

Figure 1 shows the equipment used for the studies. Lettered a–d are the Web browser plugin to log users' interactions during each task, a camera, a pressure sensitive mouse, and a pressure sensitive chair, respectively. Next, we will describe what is recorded by these devices.

3.1.1 Firefox Plugin

To log users' interactions with search engines, we modified the Firefox[4] plugin that comes with the Lemur Toolkit Query Log Toolbar[5]. Among the interactions that are recorded in the logs are the queries users enter to a major search engine (i.e., Google, Yahoo!, Bing, and Ask.com); the pages visited and any query that led to that page being visited, the dwell time, or amount of time spent on a page; and several other details about scrolling and window focusing. The plugin was also designed to prompt the user to report feedback for each page, query, and task. The exact dialogs changed between the two studies, so the details for each can be found in Sections 3.2 and 3.3 for the pilot and user studies, respectively.

[4]http://www.mozilla.com

[5]http://www.lemurproject.org/querylogtoolbar/

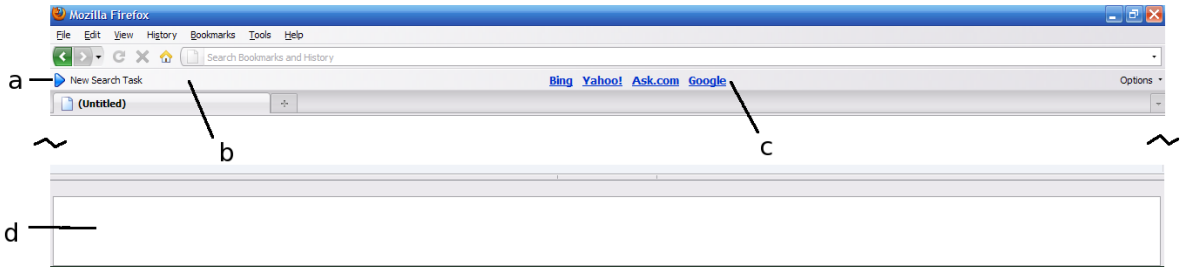


Figure 2: A screenshot of the Firefox plugin before a task has started. The areas of interest are a) the "Start Search Task" button, b) an area for a timer (only displayed during a task), c) the search engines from which users may choose, and d) a text box at the bottom of the browser where notes or responses can be recorded.

The query log produced by this plugin is a *client-side* log. This is in contrast to many publicly available query logs, which are sever-side, as we mentioned above. This means we can log information about pages visited that were not clicked from a results page. Client-side logs are richer and allow for more accurate user modeling. The features extracted from the client-side logs are reported in Section 5.

A screenshot of the plugin before a task is started is shown in Figure 2. To start a task, users must click the "Start Search Task" button (a). Once a task has begun, a timer appears where (b) is. The four search engines from which users could choose are linked to in the center of the toolbar (c). In the pilot study, the search engine links were fixed in place, which did not control for ordering effects. In the user study, we programed the plugin to randomize the ordering at the start of every task. Finally, at the bottom of the browser is a text area (d). This portion of the toolbar had different uses in the two studies: in the pilot study, it was used to take notes and in the user study, it was used to respond to tasks.

3.1.2 Sensors

The three sensors are the same sensors used by Cooper et al. (2009) (see Section 2.3 for more details on the research). The data recorded by each is as follows:

Mouse sensor: reports pressure readings from six individual pressure sensors: two on each side and two on the top.

Camera: tracks facial expressions, emitting confidence values for six mental states—agreeing, concentrating, disagreeing, interested, thinking, and unsure.

Chair sensor: reports pressure readings from six individual pressure sensors: three on the seat and three on the back.

We extracted the same features used by Cooper et al. (2009), which are listed in Section 5.

3.2 Pilot Study

There are two key differences between the data collected in the pilot and user studies: the tasks given to the users and the feedback requested during searches. The next two sections elaborate on the exact tasks given to and the feedback requested from the users.

In the pilot study, each user was asked to search for six tasks. They were asked to spend no more than ten minutes per task, though we made it clear that the time limit was a soft boundary—participants could take more or less time if they wanted. Tasks were given to users in a pre-determined order consistent with a Latin squares setup. A Latin square is an $n \times n$ matrix, where n is the number of treatments (in our case, six tasks) to give to users. Its purpose is to remove the variable of treatment ordering

Label	Task Description
Thailand	You are considering taking a trip to Thailand. Search the web to make a list of pros and cons of such a trip. Consider the price of travel, lodging, food, and entertainment. Also take into account reviews in blogs and forums from others who have taken vacations there.
Anthropology	Your friend would like to attend graduate school to study anthropology and wants your help to find some candidate schools. Find several schools that offer decent anthropology programs and pros and cons for each. Also consider that your friend currently lives in Ohio with family and would like to go to school as close to them as is possible.
GRE score	You would like to attend graduate school for computer science. What is the minimum GRE score you need to get into the majority of the top 25 ranked programs?
MS Word	You have recently bought Microsoft Office 2008 for Mac. In MS Word, you created a document and set the background to a grid pattern and saved it. When you opened the document later, the background no longer had the grid pattern, but was a solid color. Find if this error in saving a documents background is a known problem for MS Office for Mac and three sources that offer potential fixes.
Hangar menu	You and your friends want to go out for dinner in Amherst tonight. To pick a place, you want to look at the menus on-line. Find the menu for the Hangar Pub and Grill. Note the URL for the page that contains it.
Computer virus	Find three sources describing the next big computer virus or worm and how computer users can defend against it.

Table 1: The search tasks given to users in the pilot study.

from a controlled study. The square guarantees that each treatment will be placed at the beginning of the treatment sequence exactly once. One Latin square is an n-sequence sample from the $n!$ possible permutations of treatments. For the pilot study, we used three Latin squares, as we had fifteen users (each user's sequence was a specific row from one of the three Latin squares).

The tasks are described in Table 1. We chose the tasks more or less randomly, though some were influenced by our personal experiences. Four of the six were more research oriented, such as Thailand. We hypothesized that such tasks would take more time, thus increasing the chance of users becoming frustrated, and would encourage users to pursue subtasks in which they would have more inherent interest (e.g., following up on a lead about restaurants in Thailand). Two of the tasks were posed with the expectation that they would causes searcher frustration: MS Word and Hangar menu. We consider MS Word to be hard because it is a very specific debugging question and it is not clear how the information need should be formulated as a query. We felt the Hangar menu would be difficult because the Hangar Pub and Grill does not have a Web site. Rather, one needs to navigate to the UMass Wiki[6], which has the menu under the name "Wings".

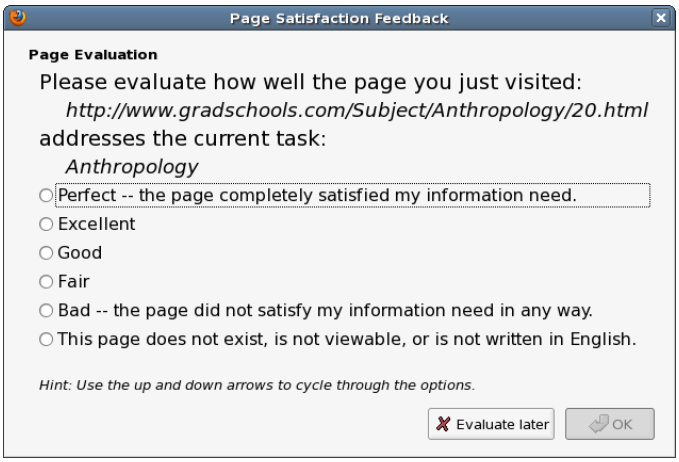
When participants arrived to the study, they were given a set of instructions and a packet of paper slips; each slip contained one task description and were stapled in the order determined by the Latin squares. For each task, users were asked to press the "Start Search Task" button and select their next task from a drop down menu. The "Start Search Task" button then turned into an "End Task" button and a timer appeared next to it. As stated, users were asked to spend no more than ten minutes per task, and this timer kept track of how munch time they had remaining.

The plugin waited until a query was entered; once triggered, the plugin would prompt the user with dialog shown in Figure 3-a after navigating away from any non-search page. This prompt asked users to rate how well the page they just visited satisfied the current task's information need on a scale of 1 (Bad) to 5 (Perfect). They could also select that the page was not viewable.

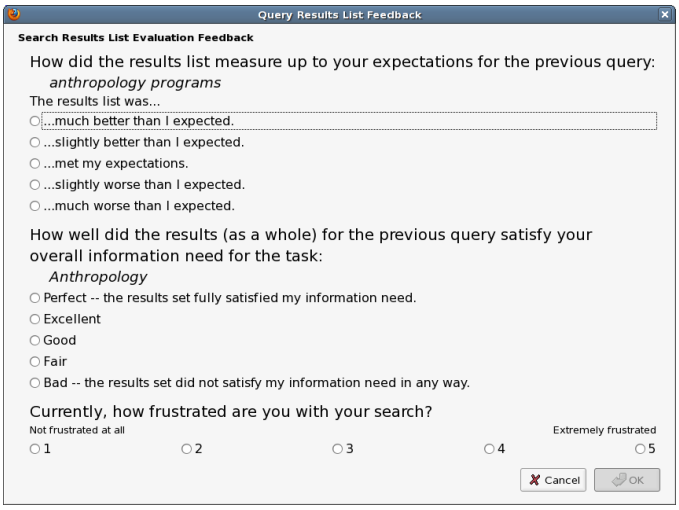
After at least one query was entered, whenever a new query was entered or the participant clicked

[6]http://www.umasswiki.com/wiki/Wings

a.



b.



c.

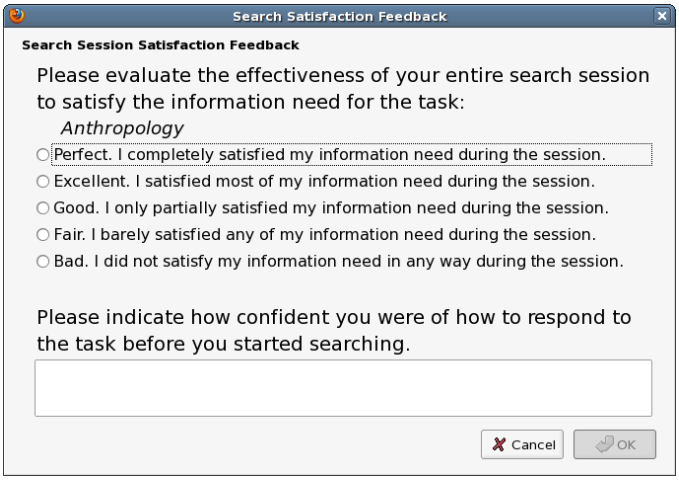


Figure 3: The feedback prompts displayed to users during the pilot study at the a) page level, b) results list level, and c) task level.

the "End Task" button, we knew that user was finished with their previous query. The plugin would then display the prompt in Figure 3-b. This asked the user to report how well the results returned by their previous query met their expectations and satisfied their information need, and how frustrated they were with the entire search up until now, all on a scale of 1–5 (1 being low or not frustrated at all, 5 being high or extremely frustrated).

Finally, when a user clicked on the "End Task" button, they were prompted to enter feedback about the task as a whole (i.e., across all queries) (Figure 3-c). This asked them to rate how well the whole session satisfied their information need and to state how confident they were in the task at the beginning.

3.3 User Study

In the full user study, we decided to use a pool of twelve tasks, some with multiple versions (see Table 2). For the tasks with multiple versions, there is a general framework for the task and one aspect—a place name in all with one instance—that can have three or four different values. We used four 12×12 latin squares to generate sequences of tasks. Each user was asked to complete the first seven tasks in their ordering[7] and to spend no more than about seven minutes per task. The tasks fall into two broad categories: fact-finding, informational tasks (e.g., Bridges) and real-world navigational queries (e.g., Chipotle). The particular version of a task depended on the physical computer at which a participant sat. There were five computers used and no statistical methods were used to decide which computer would get which version of a task—they were manually randomized. The purpose of the versions was to inject some variety among otherwise identical tasks so we could observe the differences.

As before, the tasks themselves were chosen more or less randomly. These are more focused than the pilot study tasks, as several participants of the pilot study complained about the open-endedness of the tasks. For some tasks, such as Temperature, we used city names that are ambiguous without the state name, anticipating that this may cause some frustrating interactions between the user and search engine. Many of the real-world navigational tasks are similar to the types of tasks that can be answered using the "deep web" results provided by the more commonly used search engines. In the event that the search engine did not provide results with the first user query, we expected the user to become frustrated because they had to exert more effort than they anticipated. For example, if we were responding to the Chipotle task for Amherst, MA, we might submit the query *chipotle near amherst, ma* to Google. The top result shows a map of Chipotle in Worcester, MA along with the address. However, if that map did not show up (either because of a poorly formulated query or the location specified in the query was not sufficiently close a Chipotle), a user may become frustrated.

For the user study, we also modified the prompts and added new ones. We decided to automate the task-selection query, so a user only needed to click the "New Search Task" button and the next task would automatically be served to them. At the beginning of a task, we asked the user to rate how confident they were in what the task was asking and how much of the answer they already knew. Figure 4-a shows a screenshot.

Each time a new query was entered, the participant was prompted to describe their expectations for the query, as shown in Figure 4-b. We decided to use a free-write box here because it provided more information than a check list of predefined expectations based on a small informal study.

For each page that was visited, the same dialog that was presented in the pilot study was shown to users in the user study; see Figure 4-c.

After each query-level search, users where prompted with the dialog in Figure 5-a. Users were asked to enter what actually occurred during their search, compared to their expectations. Unfortunately, many users misinterpreted this to mean "What did you expect?", and rewrote what they had entered for the query expectation dialog. As in the pilot study, users were asked to report how well the search satisfied the current task's information need and how frustrated they were with the task so far. In addition, if the

[7] Originally, each user was to do eight tasks, but after the first two subjects spent longer than anticipated, we reduced the number to seven.

Label	Task Description	Substitution values
Temperature	What is the average temperature in ⟨PLACE⟩ for winter? Summer?	Dallas, South Dakota / Albany, Georgia / Springfield, Illinois
Bridges	Name three bridges that collapsed in the USA since 2007.	
Drought	In what year did the USA experience its worst drought? What was the average precipitation in the country that year?	
Pixels	How many pixels must be dead on a MacBook/MacBook Pro before Apple will replace the laptop? Assume the laptop is still under warranty.	
Concert	Is the band ⟨BAND⟩ coming to Amherst/Northampton within the next year? If not, when and where will they be playing closest?	Snow Patrol / Greenday / State Radio / Goo Goo Dolls / Counting Crows
TV	What was the best selling television set of 2008? Specify brand and model.	
PetsMart	Find the hours of the PetsMart nearest ⟨PLACE⟩.	Wichita, Kansas / Thorndale, Texas / Nitro, West Virginia
Dow	How much did the Dow Jones Industrial Average increase/decrease at the end of yesterday?	
Coffee shops	Find three coffee shops with WI-FI in ⟨PLACE⟩.	Staunton, Virginia / Canton, Ohio / Metairie, Louisiana
Chipotle	Where is the nearest Chipotle restaurant with respect to ⟨PLACE⟩?	Manchester, MD / Brownsville, Oregon / Morey, Colorado
Verizon	What's the help line phone number for Verizon Wireless in Massachusetts?	
Inspection	Name four places to get a car inspection for a normal passenger car in ⟨PLACE⟩.	Hanover, Pennsylvania / Collinwood, Tenessee / Salem, North

Table 2: The search tasks given to users in the user study. For certain tasks, there are place holders in the description; the proper nouns that were substituted are found in the third column. In the user study, the exact replacement depended on the computer on which the browser was running.

user selected a frustration level of 2 or greater, they were asked what made them frustrated. Again, we elected to use a free-write box here to obtain more information from the user.

Finally, at the end of the task, users were shown the dialog in Figure 5-b. They were asked to assess how well the task was satisfied during the entire search session. In addition, they were asked to pick their most useful query, what changes they would make to the search engine, and what other resources they would normally have consulted to respond to the current task.

4 Study Analysis

In this section, we will describe the high-level findings from the pilot and user studies, including participant demographics, causes of frustration, and aspects of the studies that can be improved in future implementations.

4.1 Pilot Study

The pilot study consisted of fifteen participants from the University of Massachusetts Amherst. The mean age was 25.5 years. Two-thirds of participants were male. Most (eleven) were graduate students; the other four were undergraduates. Fourteen users rated their experience with Web search as a 5 on a 1–5 scale (1 being 'none', 5 being 'I search several times a day'). One user rated their experience as a 4. All fifteen users described their area of interest as computer science, with three listing a joint major: mathematics, microbiology, or biochemistry. A total of 90 tasks were completed, 351 queries were entered, and 705 pages were visited.

Aggregating across all users and tasks, the responses to "Currently, how frustrated are you with your search?", asked at the query level, are distributed as follows. Recall that 1 means *not frustrated at all* and 5 is *extremely frustrated*.

Query Frustration	None				Extreme
Feedback value:	1	2	3	4	5
Frequency:	113	111	78	31	18

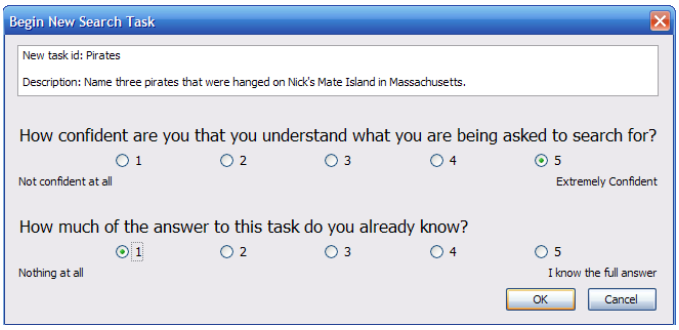
Note that users were moderately to extremely frustrated (3–5) for 36% of queries. If we instead binarize the feedback such that 1 is *not frustrated* and 2–5 is *frustrated*, then users were frustrated after 68% of queries.

The query level feedback for satisfaction and expectation fulfillment are below. Recall that at the query level, satisfaction was determined by asking participants, "How well did the results (as a whole) for the previous query satisfy your overall information need?", where the options were in the range 1 (*bad—the results set did not satisfy my information need in any way*) to 5 (*perfect—the results set fully satisfied my information need*). Users' information needs were moderately to highly satisfied (3–5) for 45% of queries, and were not satisfied at all (1) by 29% of queries. The distribution is as follows.

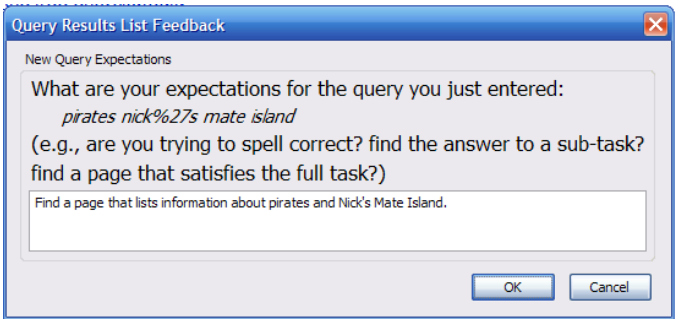
Query Satisfaction	Bad	Fair	Good	Excellent	Perfect
Feedback value:	1	2	3	4	5
Frequency:	101	97	80	40	33

Recall that we measured how well a query met the user's expectations by asking, "How did the results list measure up to your expectations for the previous query?". The response options were in the range from 1 (*bad—much worse than I expected*) to 5 (*perfect—much better than I expected*). Met expectations are represented by a 3. Users' expectations were met or exceeded (3–5) for 45% of queries. Only 19% of queries were much worse (1) than users' expectations. The distribution aggregated over all users and tasks is as follows:

a.



b.



c.

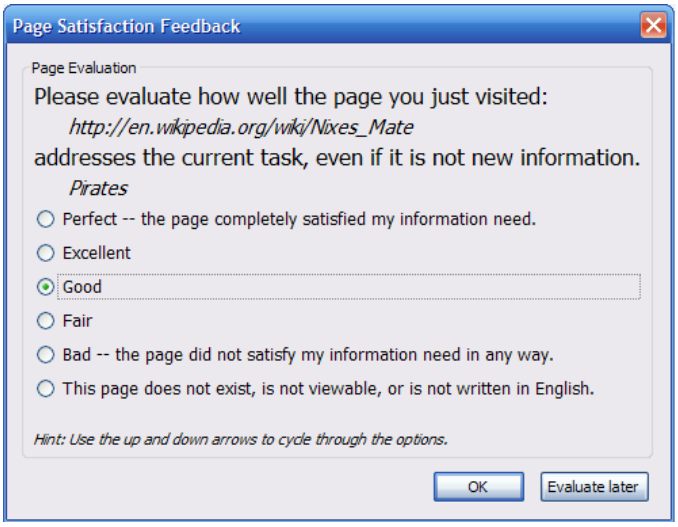
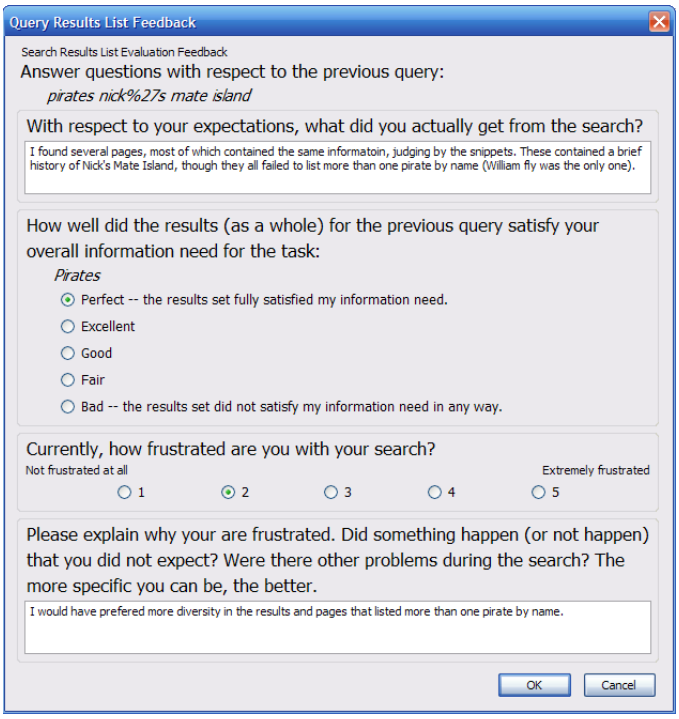


Figure 4: Three of the five feedback prompts displayed to users during the user study a) at the beginning of a task, b) at the start of each query-level search, and c) after every page is visited.

a.



b.

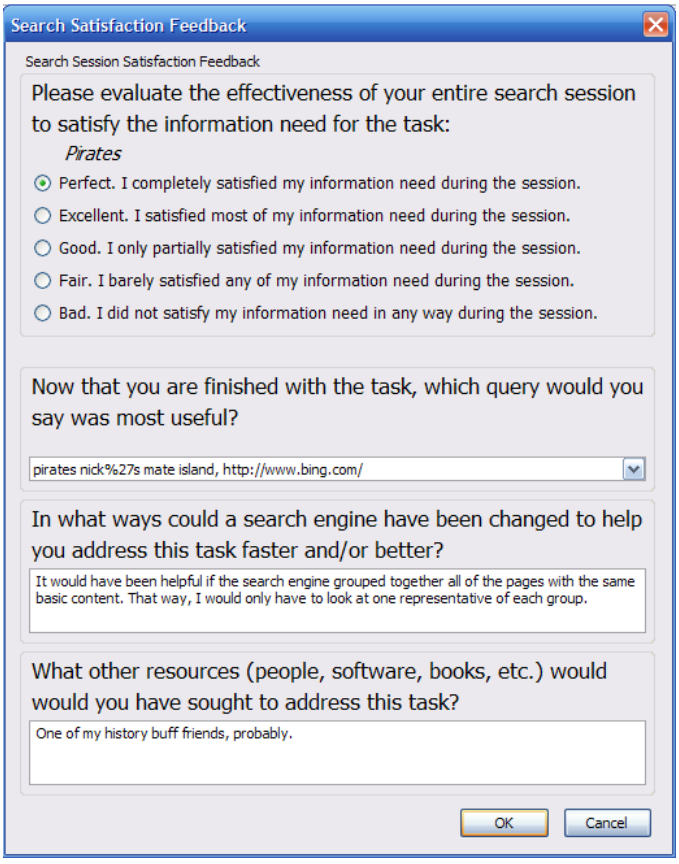


Figure 5: Two of the five feedback prompts displayed to users during the user study a) at the end of each query-level search and b) at the end of each task.

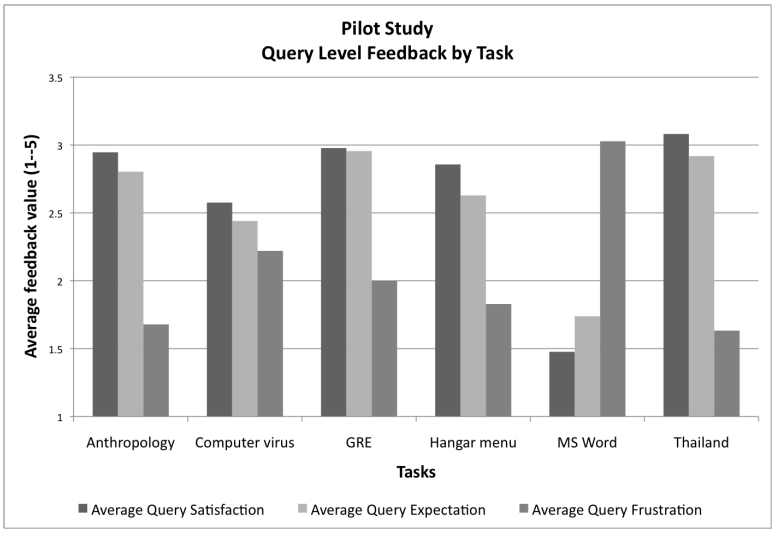


Figure 6: *Pilot Study.* The average feedback score for how well a set of results met a user's expectations, satisfied the information need, and how frustrated users were so far in there task across users by task.

Query Expectation Fulfillment	Bad	Fair	Good	Excellent	Perfect
Feedback value:	1	2	3	4	5
Frequency:	66	129	108	33	15

At the end of each task, participants were asked how well the entire search session satisfied their information need. Unfortunately, a bug in our logging software caused the feedback values of fair (2) and good (3) to be conflated. Nonetheless, 32% of the ninety tasks were perfectly satisfied and 92% were at least somewhat satisfied.

Task Satisfaction	Bad	Fair&Good	Excellent	Perfect
Feedback value:	1	2&3	4	5
Frequency:	7	34	20	29

The pilot study was most useful in helping us find shortcomings in both our logging software and our experimental setup. The logging software failed to capture certain events, such as scrolling. For the setup, we realized we needed more information from the searchers. For example, we originally intended to be able to code causes of frustration from the feedback in the pilot study. However, without understanding what the user was explicitly thinking, coding was very difficult.

The pilot study was also useful for finding some general trends and helping us hypothesize models for predicting frustration. The latter is discussed in Section 5. We will discuss a couple of the more interesting trends now. Figure 6 shows the average feedback levels for satisfaction, how expectation was met, and how frustration broke down by task. We see that the most frustrating tasks—the ones where the green or right-most bar is highest—are MS Word and Computer virus. The MS Word task also had the lowest satisfaction and least met expectations. On the other hand, tasks with at least moderate satisfaction and met expectations had lower frustration. This shows that there is likely a relationship between query expectation, search satisfaction, and frustration. This also shows that the Hangar menu task was not as difficult as anticipated.

Another trend we found in the pilot study data is that users become more frustrated as they enter more queries for a given task, as shown in Figure 7.

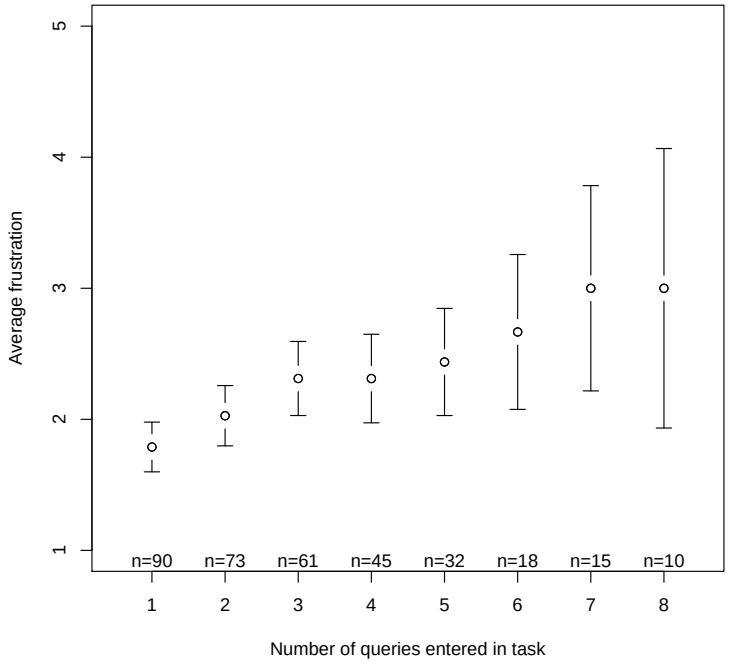


Figure 7: *Pilot Study.* The number of queries entered so far for a task versus average frustration. The x-axis shows the number of queries entered so far for a task. The y-axis shows the frustration for each particular x value averaged across all users and tasks. n is the number of instances for each value of x over which y is averaged. The 95% confidence bars are set around each mean (so we are 95% confident that the true mean falls somewhere between the bottom and top bars). For example, when 4 queries have been entered for a task, the average frustration across all users and tasks is just above 2 (somewhat frustrated), and we are 95% confident that the true mean is somewhere between 2 and 2.5. The general trend is that frustration increases with the number of queries entered for a task.

4.2 User Study

The user study consisted of 30 students from the University of Massachusetts Amherst. The mean age of participants was 26. Most participants were computer science or engineering graduates, though a few were from English, kinesiology, physics, chemical engineering, and operation management. Two participants were undergraduates. All but three users reported a 5 on the search experience scale; two reported a 3, and one a 4. Seven participants were female. A total of 211 tasks were completed, feedback was provided for 463 queries, and 711 pages were visited.

The number of pages visited is nearly the same as in the pilot study—711 versus 705, respectively—despite the user study having twice the number of users. A likely reason for the difference is the type of tasks used in the studies. While the pilot study had mostly long, research-oreiented tasks, the user study consisted of mostly shorter, exact-answer tasks. This is supported if we consider the average number of queries submitted per task in either study: 2.2 in the user study versus 3.9 for the pilot study.

Aggregating across all users and tasks, the frustration feedback is distributed as follows. Recall that 1 means *not frustrated at all* and 5 is *extremely frustrated*.

Query Frustration	None				Extreme
Feedback value:	1	2	3	4	5
Frequency:	235	128	68	25	7

In this study, the frustration feedback was skewed towards the *not frustrated* end compared with the pilot study. Users were moderately to extremely frustrated after 22% of queries. If we binarize the feedback such that 1 is *not frustrated* and 2–5 are *frustrated*, then users were frustrated after 50% of the queries, as compared with 68% in the pilot study. This may have been an effect of the tasks, which were different in the two studies.

According to the query satisfaction feedback, users' information needs were moderately to highly satisfied for 60% of queries (compared to 45% in the pilot study), and were not satisfied at all by 20% of queries (compared to 29% in the pilot study). In general, users information needs were satisfied to a higher degree in the suer study. The distribution is as follows.

Query Satisfaction	Bad	Fair	Good	Excellent	Perfect
Feedback value:	1	2	3	4	5
Frequency:	94	90	102	65	112

Feedback about how well each query met users' expectations was not collected for this study. Rather, users were asked to enter a description of the performance of the query given their expectations. Accordingly, we cannot provide a distribution for expectation fulfillment.

For task satisfaction, the same bug from the pilot study caused the feedback values of fair (2) and good (3) to be conflated. Of the 211 tasks, the feedback was consistent with the pilot study results: 39% were perfectly satisfied and 93% were at least somewhat satisfied.

Task Satisfaction	Bad	Fair&Good	Excellent	Perfect
Feedback value:	1	2&3	4	5
Frequency:	14	66	48	83

The user study data was used to train and test frustration models (see Section 5) and to determine the causes of frustration. Here, we will discuss the most prominent causes of frustration found in the data. Causes were coded by one author looking at: 1) the previous queries entered by the same user for the same task, 2) the query expectation, 3) the result (i.e., how the query actually performed with respect to the expectation), 4) the level of satisfaction, 5) the level of frustration, and 6) the reason the user gave for being frustrated. Future work should refine the coding list generated and have it corroborated by multiple annotators.

The most frequent causes of frustration and an example of each is presented in Table 3. The examples are excerpts from actual user feedback for the question of why they were frustrated.

Cause	Example
Results off-topic	"I'm searching TV sets, but there are only a few pages ranked at the top are about TV sets, but many pages about autos."
More effort than expected	"I was frustrated because I expected the search engine to find the answer in a quicker way. However, I was not able to find the answer to the question at all."
Results too general	"Found some info comparing droughts and info abour PDI's, but not the actual worst drought ever"
Unconfirmed answer	"No answer from official website"
Answer did not seem to exist	"no information about store hours at all"

Table 3: Most frequent causes of frustration with example user feedback for each.

5 Modeling Searcher Frustration

Our primary reason for conducting this research is to determine if frustration can be automatically detected using query logs and physical sensors. In this section, we describe the features we extracted from the logs and the subset of features we decided to use to detect searcher frustration. Then we will provide a summary of the classifiers used. Finally, we will discuss how the models actually performed.

5.1 Features and Feature Selection

We extracted forty-one features from the query logs, many of which were influenced by the features used by Fox et al. (2005). Eighteen of them are at the query level, meaning they are aggregated over a single query and its results, and are listed in Table 4. The other twenty-three are at the task level, meaning they are aggregated over the task from the beginning up through the current query, and are listed in Table 5.

We extracted forty features from the sensors, which are listed in Table 6. These are the same features Cooper et al. (2009) used for the three sensors. The camera features are all straight forward, as they correspond to the raw output from the camera. The mouse and chair features require derivation from the raw data, however, as described in Cooper et al. (2009).

We looked at four ways of aggregating sensor readings over which the features were extracted. The first was to consider only the time spanning the immediately preceding query search. The second was the same, but removing sensor readings collected while prompts were displayed to the user. The third considers the entire task leading up to the point at which frustration is being reported. The forth and final is like the third, but ignores sensor readings taken while the user was responding to prompts. Our analysis showed that using the third and forth were most effective. Between the two, there does not appear to be much difference, so henceforth we will refer only to the third method.

One of our goals was to determine what features are useful in modeling searcher frustration. We used the data collected from the pilot study to analyze what features correlated most strongly with the frustration reported by users. When performing this analysis, we looked at each user individually and compared across users. In order for us to judge a particular feature as important, it had to correlate well with frustration across a large number of users. Figure 8 show an example of a stronger feature—the mean average dwell time in a task (that is, the average time spent viewing a page for a query, averaged over all queries seen so far for a task), which appears negatively correlated with frustration. Specifically, most of the high frustration points correspond to shorter dwell times, while lower level of frustration are spread across many dwell times. One way of interpreting this is that frustrated searchers discard pages quickly, whereas non-frustrated searchers are more likely to spend a significant amount of time on pages.

The left side of Table 7 lists the sensor features that were the most correlated with frustration and the right side lists the features selected from the query logs. We will refer to the selected sensor features as *SensorModel* and the features from the query logs as *QueryLogModel*. The union will be referred to as

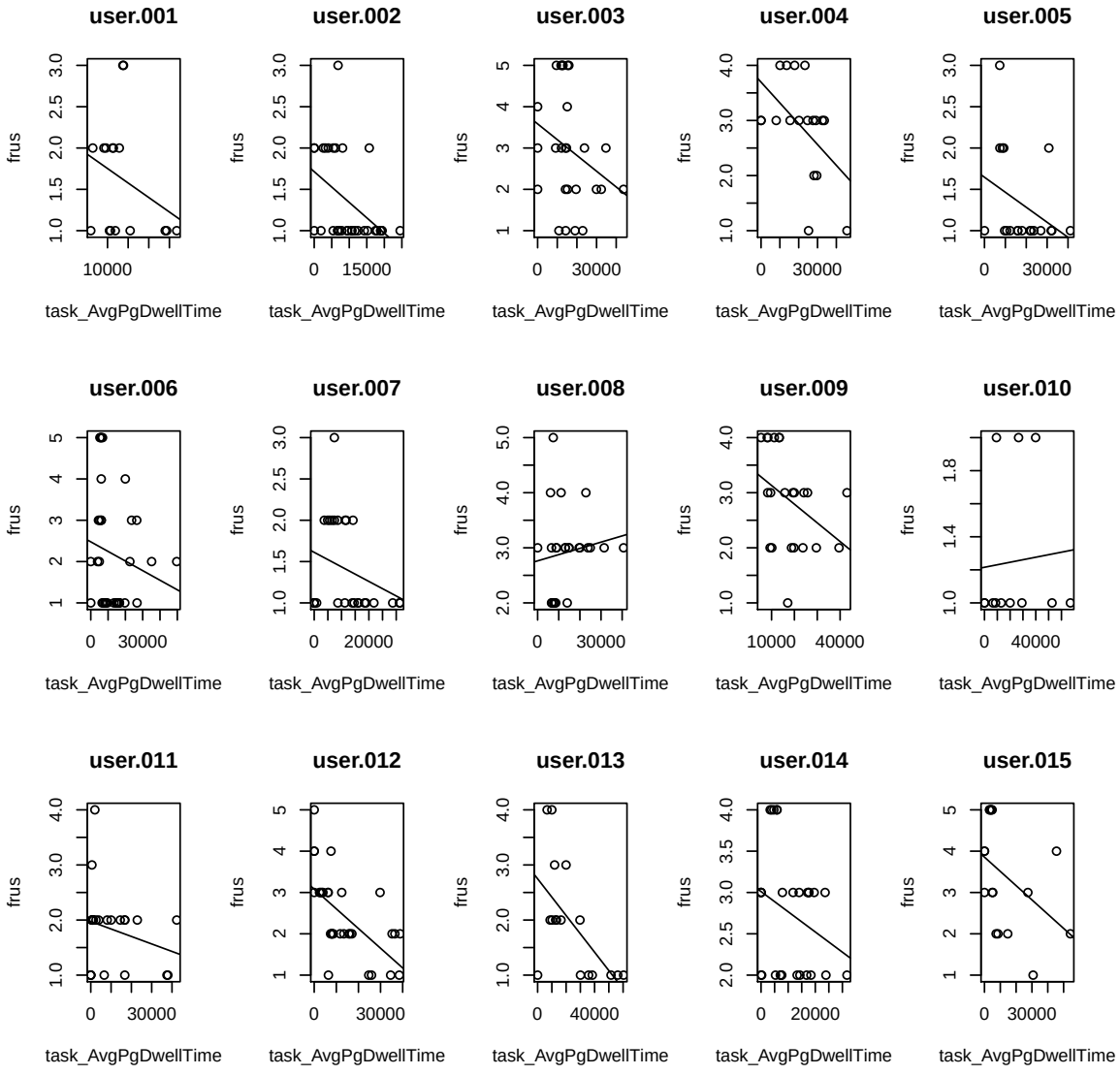


Figure 8: Plots of the mean average dwell time from the start of a task (that is, the average time spent viewing a page for a query, averaged over all queries seen so far for a task) versus frustration for the fifteen users in the pilot study. The lines are linear regression fits.

Feature	Description
Query level	
JaccardDistFromPrevQry	The Jaccard distance between the current and previous query.
IsPrefix	Binary; true if the current query is a prefix of the previous query.
Prefixes	Binary; true if the previous query is a prefix of the current query.
IsSuffix	Binary; true if the current query is a suffix of the previous query.
Suffixes	Binary; true if the previous query is a suffix of the current query.
IsContainedIn	Binary; true if the current query is contained within the previous query.
Contains	Binary; true if the previous query is contained within the current query.
RsltsClck	The number of results visited directly from the results page.
RsltsVisitedPrev	The number of results listed that were visited previously in this task.
RsltPgsViewed	The number of result list pages viewed.
RsltsReturned	The number of results returned for the current query.
PgsViewed	The number of pages viewed, including ones that were navigated to from listed results.
AvgVisitedRsltRnkOnPg	The average rank of a visited result on the result list page for this query.
AvgVisitedRsltRnkAbs	The average absolute rank of a visited page for this query.
AvgPgMaxScroll	The average maximum proportion of a page scrolled over during one scroll event for a page.
MeanAvgTimeBWScrolls	The mean of the average time between scrolls for each page.
AvgPgDwellTime	The average time spent viewing a page for this query.
AveragePropOfPgViewed	The average proportion of a page viewed.
SearchTime	The time the user spent on this query.

Table 4: The query level features extracted from the query logs.

SensorQueryLogModel.

5.2 Classifiers

For modeling frustration, we considered binary classification and regression. For binary classification, we binarized the data so that all instances where users reported a frustration level of 0, we labeled as *not-frustrated*; we labeled all other levels as *frustrated*. For regression, we used the frustration levels reported by the users.

For classification, we considered four classifiers: Majority Class, Random with Prior, Logistic Regression, and J48 Decision Tree. Majority Class always predicts the most frequent class label from the training set. Random with Prior randomly chooses between classes using the class label distribution from the training set as a prior. Logistic Regression (also called Maximum Entropy) and J48 were picked out of convenience.

For regression, we used: Mean Value, Linear Regression, and REP Tree. Mean Value predicts the mean of the training set values. Linear Regression is self-explanatory. REP Tree is a decision tree that uses the reduced error pruning algorithm (Quinlan, 1999). REP was chosen because it is one of only two regression-capable decision trees available in Weka (Hall et al., 2009) (version 3.7), which is the machine learning toolkit used for the experiments.

5.3 Results and Discussion

To obtain results, we used a leave-one-out setup relative to users. That is, we have $\langle feature, label \rangle$ instances from thirty users and we treat each user's instances as a unique fold. We then performed thirty-fold cross validation, using every possible set of twenty-nine folds for training and testing on the left out

Feature	Description
Task level: aggregated over queries q_1 through q_i for use with query q_i	
AvgJaccardDist	The average Jaccard distance between adjacent queries.
ProportionedIsPrefix	The proportion of queries so far for which IsPrefix is true.
ProportionedPrefixes	The proportion of queries so far for which Prefixes is true.
ProportionedIsSuffix	The proportion of queries so far for which IsSuffix is true.
ProportionedSuffixes	The proportion of queries so far for which Suffixes is true.
ProportionedIsContained	The proportion of queries so far for which IsContained is true.
ProportionedContains	The proportion of queries so far for which Contains is true.
QryCntUnq	The number of unique queries entered for this task up until now.
QryCntTot	The total number of queries entered for this task up until now.
QryPopUnq	The proportion of queries that are unique so far.
RsltPgingCnt	The number of times a user paged through a result list.
AvgRsltPgCnt	The average number of result list pages visited per query.
RsltsClck	The number of results clicked on up until now.
AvgRsltsClck	The average number of results click on per query.
PgsVisited	The number of pages, whether or not listed in results, visited.
AvgPgsVisited	The average number of pages visited.
AvgPgMaxScroll	The average maximum proportion of a page scrolled over during one scroll event for a page.
AvgPgDwellTime	The average time spent viewing a page.
AvgRsltsVisitedPrev	The average number of results listed that were visited in multiple result lists.
AvgVisitedRsltRnkOnPg	The average rank of a visited result on the result list page.
AvgVisitedRsltRnkAbs	The average absolute rank of a visited page.
MaxDupQryCnt	The maximum number of duplicate queries entered so far.
MaxDupQryProp	The proportion of queries effected by the most frequently reoccurring query.
TaskTime	The time the user has spent on this task so far.

Table 5: The task level features extracted from the query logs.

Feature	Mean	Std. Dev	Min	Max
Camera				
Agreeing	CmeanA	CdevA	CminA	CmaxA
Concentrating	CmeanC	CdevC	CminC	CmaxC
Disagreeing	CmeanD	CdevD	CminD	CmaxD
Interested	CmeanI	CdevI	CminI	CmaxI
Thinking	CmeanT	CdevT	CminT	CmaxT
Unsure	CmeanU	CdevU	CminU	CmaxU
Mouse				
Pressure	MmeanP	MdevP	MminP	MmaxP
Seat				
Sit forward	SmeanF	SdevF	SminF	SmaxF
Net seat change	SmeanS	SdevS	SminS	SmaxS
Net back change	SmeanB	SdevB	SminB	SmaxB

Table 6: The features extracted from the sensor feedback.

Concentrating–Mean	(CmeanC)
Thinking—Std.Dev.	(CdevT)
Interested—Std.Dev.	(CdevI)
Interested—Max	(CmaxI)
SitForward—Std.Dev.	(SdevF)
SeatChange—Mean	(SmeanS)

AvgPgDwellTime (at query level)
AvgJaccardDist
AvgPgDwellTime (at task level)
AvgVisitedRsltRnkOnPg
PgsVisited
QryCntTot
QryCntUnq
RsltPgingCnt

Table 7: On the left are the sensor features deemed most likely to be helpful in predicting frustration (*SensorModel*); on the right are the query log features deemed most helpful in predicted frustration (*QueryLogModel*).

fold. We can use three levels of metrics from this data: micro (aggregate an evaluation metric over all the predictions); macro (calculate metrics on each fold's predictions individually, then average over the folds); and weighted macro (same as macro, but weight each fold by the number of instances contained in that fold). We report all three.

The metrics we report for classification are accuracy and F1 (with respect to the *frustrated* label). Accuracy gives us a general picture of how well a classifier performs across all labels. F1 shows us how well a classifier performs with respect to a specific label, in our case, *frustrated*. F1 weights recall and precision equally, and thus a high F1 value is desirable.

For regression we use three metrics: correlation coefficient, mean absolute error (MAE), and root mean squared error (RMSE). A good regression model will have a high correlation coefficient and low MAE and RMSE.

To test for statistical significance, we use Fisher's randomization test (Cohen, 1995; Smucker et al., 2007) and consider a two-sided significance level of 0.05. We use 100,000 permutations of the two system's contingency tables, resulting in an error margin of 2% (± 0.001) at the 0.05 significance level, as described by Smucker et al. (2007).

Interestingly, classifiers using *QueryLogModel* performed better than both *SensorModel* and *SensorQueryLogModel*. Because of that, we only show the results for the classifiers using *QueryLogModel*. in Tables 8 (classification) and 9 (regression). However, classifiers using both *SensorModel* and *SensorQueryLogModel* outperformed the baseline; they just did not perform as well as using *QueryLogModel* alone.

The results make it clear that modeling frustration is a difficult problem. The simple baseline suggests that it would be more accurate to predict the *least* frequent class label and demonstrates a poor alignment between the training and test sets with respect to the most frequent class label. However, *QueryLogModel* in conjunction with the J48 classifier performed 175% better than the baseline for macro accuracy. J48 has the highest scores overall and is significantly better than both the baseline and Random. It is not statistically better than Logistic Regression, however, except with respect to micro F1.

For regression, Linear Regression is the frontrunner, with the highest correlation coefficients and lowest error. Its macro and weighted-macro correlation coefficients are quite high and demonstrate this model's strength. An interesting observation is that it is statistically better than the baseline (Mean) for all cases except micro and weighted macro RMSE. It is statistically better then REP with respect to just over half of the metrics (not statically significant for micro correlation coefficient and all levels of MAE). Given that Linear regression is statistically better in the majority of cases, it is clearly the better model.

Between the pilot study and the user study, we were able to find several features that appear to be moderately correlated with frustration. We determined the strength of the relationship between a feature and frustration by calculating the Pearson's correlation between that feature and frustration for each user and then averaging across users, yielding a macro average. We then ranked all the features by their absolute macro average. For both studies, the most correlated query log features remained stable, with

Classifier	Accuracy	F1
Majority Class	0.3853	0.1069
	0.3682	0.0468
	0.3853	0.0587
Random	0.4719	0.3646
	0.4766	0.3244
	0.4719	0.3384
Logistic Regression	0.6191	0.6036
	0.6354	0.5541
	0.6191	0.5597
J48	0.6342	0.6667
	0.6459	0.6048
	0.6342	0.6108

	Random		Logistic Regression		J48	
	Accuracy	**F1**	**Accuracy**	**F1**	**Accuracy**	**F1**
Majority Class	**0.0034**	**0**	**0**	**0**	**0**	**0**
	0.0002	**0**	**0**	**0**	**0**	**0**
	0.0034	**0**	**0**	**0**	**0**	**0**
Random		**<0.0001**	**0**	**<0.0001**	**0**	**0**
			0	**0**	**0**	**0**
			<0.0001	**0**	**<0.0001**	**0**
Logistic Regression					0.5956	**0.0135**
					0.6808	0.0640
					0.5956	0.0634

Table 8: Classifier performance metrics using *QueryLogModel* (top) and p-values between classifiers (bottom). The rows in each cell represent, from top to bottom, micro, macro, and weighted-macro levels. We consider p-values below the 0.05 level to be significant (shown in bold), meaning the two classifiers are significantly different with respect to the specified metric.

Regression Model	Corr. Coef.	MAE	RMSE
Mean	-0.4213	0.8113	0.9894
	<0.0001	0.8221	0.9630
	<0.0001	0.8113	0.9501
Linear Regression	0.2463	0.7634	0.9634
	0.3780	0.7609	0.9132
	0.3790	0.7634	0.9176
REP	0.1665	0.7858	1.0111
	0.2835	0.7698	0.9559
	0.2757	0.7858	0.9712

	Linear Regression			REP		
	Corr. Coef.	MAE	RMSE	Corr. Coef.	MAE	RMSE
Mean	**0**	**0.0031**	0.1204	**0**	0.1750	0.3083
	0	**0.0001**	**0.0032**	**<0.0001**	**0.0041**	0.7206
	0	**0.0031**	0.0809	**0.0001**	0.1750	0.3149
Linear Regression				0.0964	0.2108	**0.0188**
				0.0433	0.6178	**0.0341**
				0.0303	0.2108	**0.0071**

Table 9: Regression performance metrics using *QueryLogModel* (top) and p-values between classifiers (bottom). The rows in each cell represent, from top to bottom, micro, macro, and weighted-macro levels. We consider p-values below the 0.05 level to be significant (shown in bold), meaning the two classifiers are significantly different with respect to the specified metric.

only one change in the top five most correlated features. The most strongly correlated features are (all at the task level; see Table 5): the result paging count (RsltPgingCnt), the number of pages visited so far in the task (PgsVisited), the total number of queries entered so far in the task (QryCntTot), the number of unique query terms entered so far in the task (QryCntUnq), the proportion of queries entered so far that were duplicates (MaxDupQueryPorp), and the mean average page dwell time (AvgPgDwellTime). The macro averaged correlation coefficients were in the moderate correlation range.

The query level log features do not appear to be helpful predictors. One reason for this might be that frustration is cumulative over a task; while poor performance during a query may make the level of frustration go up or down, it does not have the scope to predict what the absolute frustration level will be. This suggests that query level features would make good predictors for the increase or decrease of frustration, but not as an absolute predictor. It is also important to understand that the query level features are encompassed as averages in the task level features.

Sensor-derived features were less correlated with frustration than the log features—all macro averaged correlation coefficients were less than 0.23. In addition, the strongest correlated features are different between the two studies. In the pilot study, the five most strongly correlated features were: unsure standard deviation (CdevU), interested standard deviation (CdevI), thinking standard deviation (CdevT), concentrating max (CmaxC), and sit forward standard deviation (SdevF). However, in the user study, the most strongly correlated features were: back-change max (SmaxB), thinking max (CmaxT), mouse pressure max (MmaxP), seat change max (SmaxS), and agreeing max (CmaxA). We tested classification models using the latter set of features, but no improvements were made. It is not clear why the features are not more correlated with frustration, as they are in other studies of frustration (Cooper et al., 2009; Kapoor et al., 2007). One possibility is that the way we have aggregated the features during a task is unhelpful; other studies have used thirty second windows, while we use windows several minutes long. We plan to examine this more closely in our future work.

6 Conclusion and Future Work

This research has provided some understanding about searcher frustration and how it can be modeled. The coding for causes of frustration discussed in Section 4 shows us that frustrating situations can be reduced to a set of key factors. With more data and analysis, we can further refine this coding scheme.

This research also shows that frustration can be predicted with more accuracy than the baseline using both binary classification and regression. Unexpectedly, the features from the physical sensors that seemed the most correlated to frustration in the pilot study did not augment the query log features as expected in predict frustration in the user study. However, there are many ways of splitting the user sessions and aggregating the sensor features. In future work, we will explore some of the more promising alternatives.

Another exciting path for future work is to consider pattern or sequence mining to see what sequences of interactions tend to lead to frustration and which ones do not. These could then be used as additional features, together with query log and sensor features.

A major contribution of this work is the rich data set collected from both the browser plugin and the sensors. More feedback was collected than was used directly in this study. This was done on purpose with the intention that other researchers interested in questions of search intent and satisfaction could also use the data.

One of the shortcomings of controlled user studies involving search is that the users are searching for pre-defined tasks. As such, they have no inherent interest in searching for a task other then to help the experimenters collect data. One way to collect data for real user tasks is to have volunteers download a version of our browser plugin and have them use it for some subset of their daily search tasks. This makes it difficult to use sensors, but with built-in web cameras becoming ubiquitous on many types of computers, it is worth investigating the practicality of at least using a camera sensor.

The next major step of this research is to begin investigating ways in which a retrieval system can be altered to address frustration. We gave some examples in Section 1, but there are many other methods we would like to explore, including the suggestions provided by participants in the user study.

References

A. Al-Maskari, M. Sanderson, and P. Clough. The relationship between IR effectiveness measures and user satisfaction. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 773–774. ACM Press New York, NY, USA, 2007.

I. Ceaparu, J. Lazar, K. Bessiere, J. Robinson, and B. Shneiderman. Determining Causes and Severity of End-User Frustration. *International Journal of Human-Computer Interaction*, 17(3):333–356, 2004.

P. R. Cohen. *Empirical methods for artificial intelligence*. MIT Press, 1995.

D. G. Cooper, I. Arroyo, B. P. Woolf, K. Muldner, W. Burleson, and R. Christopherson. Sensor model student self concept in the classroom. In *First and Seventeenth International Conference on User Modeling, Adaption, and Personalization*, Trento, Italy, June 2009.

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

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

S. Huffman and M. Hochster. How well does result relevance predict session satisfaction? In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 567–574. ACM Press New York, NY, USA, 2007.

A. Kapoor, W. Burleson, and R. Picard. Automatic prediction of frustration. *International Journal of Human-Computer Studies*, 65(8):724–736, 2007.

J. R. Quinlan. Simplifying decision trees. *Int. J. Hum.-Comput. Stud.*, 51(2):497–510, 1999. ISSN 1071-5819. doi: http://dx.doi.org/10.1006/ijhc.1987.0321.

M. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. pages 623–632, 2007.

R. White, J. Jose, and I. Ruthven. An implicit feedback approach for interactive information retrieval. *Information Processing and Management*, 42(1):166–190, 2006.