Predicting Searcher Frustration

Henry Feild and James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
{hfeild, allan}@cs.umass.edu

Rosie Jones
Yahoo! Labs
4 Cambridge Center
Cambridge, MA 02142
jonesr@yahoo-inc.com

ABSTRACT

When search engine users have trouble finding information, they may become frustrated, possibly resulting in a bad experience (even if they are ultimately successful). In a user study in which participants were given difficult information seeking tasks, half of all queries submitted resulted in some degree of self-reported frustration. A third of all successful tasks involved at least one instance of frustration. By modeling searcher frustration, search engines can predict the current state of user frustration and decide when to intervene with alternative search strategies to prevent them from becoming more frustrated, giving up, or switching to another search engine. We present several models to predict frustration using features extracted from query logs and physical sensors. We are able to predict frustration with a mean average precision of 66% from the physical sensors, and 87% from the query log features.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process

General Terms

Experimentation, Measurement

Keywords

user modeling, searcher frustration, query logs, emotional sensors

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 as an effectiveness measure. More importantly, it allows for real-time system intervention to help frustrated searchers, hopefully preventing users from leaving for another search engine or abandoning the search altogether.

This work investigates what aspects of users' interactions with the search engine during a task can be used to predict frustration. Depending on the level of frustration, we may wish to change the underlying retrieval algorithm or the user interface. For example, one source of difficulty in retrieval is a user's inability to sift through the results presented for a query [12, 16]. One way that a system could adapt to address this kind of frustration is to show the user a conceptual breakdown of the results: rather than listing all results, group them based on the key concepts that best represent them [12]. 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 et al. [14]. They used implicit relevance feedback to detect changes in users' information needs and alter the retrieval strategy based on the degree of change. The focus of our work is to detect frustrated behavior, and adapt the system based on the type of frustration, regardless of the information need itself.

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 question of whether frustration can be accurately predicted and what features derived from query logs and physical sensors are the most useful in doing so.

Our contributions include (1) the first user study of frustration in web search (2) a publicly available data set of the data collected, and (3) a comparison of on-line models derived from sensor and query log data to predict frustration.

The remainder of this paper is organized as follows. In Section 2 we discuss related work from the IR, intelligent tutoring systems (ITS), and information science (IS) literature. We then describe the task and evaluation in Section 3, followed by a description of the user study we conducted is listed in Section 4. In Section 5 we describe the models used followed in Section 6 by a description and analysis of the experiments. We end with a summary and future work in Section 7.

2. RELATED WORK

In this section, we first describe frustration in the context of IR. We then detail four areas of related work: searcher satisfaction modeling, work carried out in the field of ITS where frustration has been modeled, and various work pertaining to user modeling in IR, such as predicting when users will switch to another search engine. These works helped to shape the user study we conducted and the models used to predict searcher frustration.

2.1 Frustration in Information Retrieval

We define frustration in the context of IR as the impediment of search progress. Xie and Cool [16] explored help-seeking or problematic situations that arise in searching digital libraries. They identified fifteen types of help-seeking situations that their 120 novice participants encountered. The authors' use of 'help-seeking situations' aligns well with our definition of frustration, since the issues encountered by the subjects impeded their search progress. The authors created a model of the factors that contribute to these help-seeking situations from the user, task, system, and interaction aspects. The qualitative nature of the study is useful in designing general help systems for digital library systems. However, there was no attempt to model frustration using logged interaction data, which is the goal of our work.

In a study examining how children search the Internet, Druin et al. [7] found that all of the twelve participants experienced frustration while searching. The authors point out that children make up one of the largest groups of Internet users, making frustration a major concern. In a similar study, Bilal and Kirby [2] compared the searching behavior of graduate students and children on Yahooligans! They found that over 50% of graduate students and 43% of children were frustrated and confused during their searches. In addition, they found that while graduate students quickly recovered from "breakdowns" (where users were unable to find results for a keyword search), children did not.

2.2 Satisfaction in Information Retrieval

While frustration prediction has not been directly studied in the field of IR, *searcher satisfaction* has. Satisfaction in search can have different meanings [1, 8, 9]. We define searcher satisfaction as the fulfillment of 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, but still have been quite frustrated in the process [3].

In previous work, satisfaction has been examined at the task or session level[1] [1, 8, 9, 10]. These satisfaction models only cover user satisfaction *after* a task has been completed, not *while* a task is in progress. As such, satisfaction models are useful for retrospective analysis and improvement, but not as a real-time predictor. In contrast, a frustration model that is defined throughout a search, these real-time solutions are available.

In web search study, Fox et al. [8] 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 also analyzed browsing patterns and found some more indicative of satisfaction than others, such as entering a query, clicking on one result, and then ending the task. Clicking four or more results was more indicative of dissatisfaction.

Huffman and Hochster [10] found a relatively strong correlation with session satisfaction using a linear model encompassing the relevance of the first three results returned for the first query in a search task, whether the information need was navigational, and the number of events in the session. In a similar study of search task success, Hassan et al. [9] used a Markov model of search action sequences to predict success at the end of a task. The model outperformed a method using the DCG of the first query's result set, suggesting that general relevance is not sufficient to model satisfaction, but a model of the interactions derivable from a query log is better.

2.3 Frustration in Tutoring Systems

While we have not found any discussion of *predicting* frustration in the IR literature, we did find studies that model frustration in the ITS literature. Cooper et al. [4] 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.

Cooper et al. found that across the three experiments they conducted, the mental state camera was the best standalone 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. In another study using the same sensors, but different features, Kapoor, Burleson, and Picard [11] 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%.

2.4 User Modeling in Information Retrieval

In this section, we summarize several models used in IR prediction tasks that rely, at least in part, on query log data [6, 9, 10, 15]. We are specifically interested in the types of model used (e.g., linear regression) and the key features.

Huffman and Hochster [10] used a regression model using the relevance of the top three results returned for the first query, the type of information need, and the number of actions in the session to predict session satisfaction. Hassan et al. [9] used a Markov model to predict task success and found that sequences of actions, as well as the time between the actions, were good predictors.

Downey et al. [6] created a Bayesian dependency network model using sequences of browsing actions parameterized by a long list of user, session, query, result click, non-search action, and temporal features to predict the next action in a search sequence. The model predicts the next user action given the previous n actions. They found that using an action history with more than just the immediately preceding action was not helpful, and in fact hurt performance.

White and Dumais [15] explored search engine switching. Their goal was "not to optimize the model but rather to determine the predictive value of the query/session/user feature classes for the switch prediction challenge." They used a logistic regression model that encompassed query, session,

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

and user level features. They found that using all three feature classes outperformed all other combination of feature classes and did much better than the baseline for most recall levels.

3. TASK AND EVALUATION

In this section, we outline the details of the frustration modeling task and describe how we handle evaluation of the task.

3.1 Task

Our goal is to predict whether a user is frustrated at the end of each query interaction during a session. We define a query interaction as all interactions between a user and the the browser pertaining to a specific query up until either another query is entered or the session ends. We will refer to these as *searches*. The session consists of one or more searches directed at fulfilling a specific information need or task. We will refer to these as *tasks*. At the end of a search, we ask, "Is the user frustrated at this point of the task?" To make the prediction, we can derive features from the search just completed or from all the searches conducted in the task so far. We refer to these feature sets as search and task features, respectively. In addition, features can be derived from a user's other tasks, which we call user features.

In this paper, we consider frustration prediction as a binary task. However, multi-class prediction may also be useful, using either regression or a multi-class machine learning method. We also focus on general frustration, but predicting types of frustration may also be useful, e.g., predicting the fifteen types of frustration outlined by Xie and Cool [16].

3.2 Evaluation

In this section, we describe the metrics that we use to evaluate frustration models. Our ultimate goal is to use frustration models to decide when to intervene with the user. Since many interaction methods with which we would like to intervene are not typically used because of their undesirable, frustration-causing attributes (i.e., interaction and latency), we are interested in minimizing our false-positives (non-frustrated searchers that our models say are frustrated), potentially at the cost of recall. For that reason, our predominate evaluation metric is a macro-average (across users) F-score with $\beta = 0.5$, which gives increased weight to precision over recall. We also use 11-point interpolated average precision to compare models across users, regardless of score threshold. This metric tells us how well, on average, a model can rank instances of frustration by user.

Comparing across users rather than with a micro approach avoids one frustrated searcher in the test data skewing the results. Un-weighted macro-averaging treats all users equally. A desirable model is one that performs well across all users, not just on one specific user. In Section 6 we report macro accuracy, precision, $F_{\beta=0.5}$, and mean average precision (MAP). To be clear, MAP is uninterpolated, in contrast to 11-point interpolated average precision.

We use an approximation of Fisher's randomization test to obtain a double sided p-value for significance. Using 100,000 trials for every model comparison, the error at $\alpha = 0.05$ is ± 0.001 (2% error) [13].

4. USER STUDY

1.	What is the average temperature in [Dallas, SD/Albany, GA-/Springfield, IL] for winter? Summer?
2.	Name three bridges that collapsed in the USA since 2007.
3.	In what year did the USA experience its worst drought? What was the average precipitation in the country that year?
4.	How many pixels must be dead on a MacBook before Apple will replace the laptop? Assume the laptop is still under warranty.
5.	Is the band [Snow Patrol/Greenday/State Radio/Goo Goo Dolls/Counting Crows] coming to Amherst, MA within the next year? If not, when and where will they be playing closest?
7.	What was the best selling television (brand & model) of 2008?
8.	Find the hours of the PetsMart nearest [Wichita, KS/Thorndale, TX/Nitro, WV].
9.	How much did the Dow Jones Industrial Average increase/decrease at the end of yesterday?
10.	Find three coffee shops with WI-FI in [Staunton, VA/Canton, OH/Metairie, LA].
11.	Where is the nearest Chipotle restaurant with respect to [Manchester, MD/Brownsville, Oregon/Morey, CO]?
12.	What's the helpline phone number for Verizon Wireless in MA?
13.	Name four places to get a car inspection for a normal passenger car in [Hanover, PA/Collinwood, TN/Salem, NC].

Table 1: The information seeking tasks given to users in the user study. Variations are included in brackets.

In October 2009, we conducted a user study with thirty participants from the University of Massachusetts Amherst. The mean age of participants was 26. Most participants were computer science or engineering graduates, others were from English, kinesiology, physics, chemical engineering, and operation management. Two participants were undergraduates. All but three users reported a 5 (the highest) on a five-point search experience scale; two reported a 3, and one a 4. Seven participants were female and twenty-three were male.

Each participant was asked to complete seven[2] tasks from a pool of twelve (several with multiple versions) and to spend no more than seven minutes on each, though this was not strictly enforced. The order of the tasks was determined by four 12×12 latin squares, which removed order effects from the study. Users were given tasks one at a time, so they were unaware of the tasks later in the order. The tasks were designed to be difficult to solve with a search engine since the answer was not easily found on a single page. The complete list of tasks is shown in Table 1.

The study relied on a modified version of the Lemur Query Log Toolbar[3] plugin for Firefox.[4] At the beginning of a task, participants had to click on a 'Start Task' button. This would prompt them with the task and a brief questionnaire about how well they understood the task and the degree to which they felt they knew the answer. They were asked to use any of four search engines: Bing, Google, Yahoo!, or Ask.com and were allowed to switch at any time. Links to these appeared on the toolbar and their order was randomized at the start of each task. Users were allowed to use tabs within Firefox.

For every query entered, users were prompted to describe their expectations for the query. Each time they navigated away from a non-search page, they were asked the degree (on a five-point scale) to which the page satisfied the task,

[2]Two participants completed eight tasks, but it took longer than expected, so seven tasks were used from then on.
[3]http://www.lemurproject.org/querylogtoolbar/
[4]http://www.mozilla.com/en-US/firefox/firefox.html

Query Frustration	None				Extreme
Feedback value:	1	2	3	4	5
Frequency:	235	128	68	25	7
Percentage:	51%	28%	15%	5%	1%

Table 2: Distribution of user-reported frustration for searches.

Task Success	Bad	Fair&Good	Excellent	Perfect
Feedback value:	1	2&3	4	5
Frequency:	14	66	48	83
Percentage:	7%	31%	23%	39%

Table 3: Distribution of user-reported task success. An error in the logging software caused the 'bad' and 'fair' levels to be conflated.

with an option to evaluate later. At the end of a search (determined by the user entering a new query or clicking 'End Task'), users were asked what the search actually provided relative to their expectations, how well the search satisfied their task (on a five point scale), how frustrated they were with the task so far (on a five point scale), and, if they indicated at least slight frustration (2–5 on the five-point scale), we asked them to describe their frustration.

When users finished the task by clicking 'End Task', they were asked to evaluate, on a five point scale, how successful the session was, what their most useful query was, how they would suggest a search engine be changed to better address the task, and what other resources they would have sought to respond to the task.

A total of 211 tasks were completed (one participant completed one fewer task because of computer problems), feedback was provided for 463 queries, and 711 pages were visited. On the frustration feedback scale, 1 means *not frustrated at all* and 5 is *extremely frustrated*. In Table 2 we see that users reported frustration for around half of their queries. In a preliminary analysis of the reasons participants gave for being frustrated, the most common sources of frustration were: (1) off-topic results, (2) more effort than expected, (3) results that were too general, (4) uncorroborated answers, and (5) seemingly non-existent answers.

4.1 Success and Frustration

We find that users become frustrated even when they succeed at their information seeking task. Table 3 shows the breakdown of user-reported task success. The majority of users reported their task to be satisfied at the 'excellent' or 'perfect' levels. Table 4 shows that while not finding the information can be frustrating, even when the information is found, users can get frustrated. Users were successful in 62% of all tasks, but experience some degree of frustration in over a third of those successful tasks. This evidence supports the exploration of frustration modeling and differentiates it from task success or satisfaction prediction.

4.2 Individual Variation

Since we measure self-reported frustration, the results may depend on the individual's temperament as well as introspection. In Figure 1 we see that individuals from the test set do indeed vary in their self-reported frustration. The training set shows a similar trend. One phlegmatic individual did not report any frustration for any task. In our experimental section we will conduct leave-one-user-out cross-validation to concentrate on the aspects of frustration that generalize across users.

	Frustration	No Frustration	Total
Success	46	85	131
Failure	72	8	80
Total	118	93	211

Table 4: The number of tasks for which users were highly successful (levels 4–5) or not versus whether or not the task had any searchers for which the user was at least somewhat frustration.

5. MODELING SEARCHER FRUSTRATION

In this section, we describe the models we use to predict frustration. We consider a number of features that have been used in previous studies, both in the IR and ITS fields. The first set of features include those derived from a client-side query log, while the second set includes those from three physical sensors.

5.1 Query Log Features

The query log used in this study is client-side. Interactions between the user and the Web were recorded by means of a Firefox plugin, adapted from the Lemur Query Log Toolbar. The toolbar captures data including page focuses, click events, navigation events such as the back and forward buttons, copy and paste actions, page scrolling, and mouse movements, among others. Every event includes a timestamp.

Given the section of the log that corresponds to a particular task, we can derive search and task features (Section 3.1). The search features include search duration, pages visited, length of query, and max page scroll, and others. The task features include a summarization of the searches for the current task seen up through the end of the most recent search. They include aggregates of the search features, such as task duration, queries entered, average search duration, total pages visited, average pages visited per search, etc. Due to space constraints, we have not included a full listing of the forty-seven features. However, they are very similar to features used in previous query log analyses [8, 15].

5.2 Sensor Features

We used three physical sensors in our study: a mental state camera, a pressure sensitive mouse, and a pressure sensitive chair. These are three of the four sensors used by Cooper et al. [4]; we use the same features. The camera software provides confidence values for six mental states: agreeing, disagreeing, unsure, interested, thinking, and confident. The mouse consists of six pressure sensors—two on top and two on either side. Following Cooper et al. [4], we calculate the following feature with the values:

$$mouse = \frac{\sum_{i=1}^{6} MS_i}{1023}, \quad (1)$$

where MS represents the six mouse sensors and the denominator is the maximum pressure reading provided by any one sensor. This feature has a range from 0 to 6. Finally, the chair has three pressure sensors on the back and three on the

seat. We derive three aggregate features: net seat change, net back change, and leaning forward [4]:

$$netSeatChange(t) = \left|\sum_{i=1}^{3} SS_i[t-1] - SS_i[t]\right|, \quad (2)$$

$$netBackChange(t) = \left|\sum_{i=1}^{3} BS_i[t-1] - BS_i[t]\right|, \quad (3)$$

$$sitForward(t) = \begin{cases} 0 & \text{if } \bigvee_{i=1}^{3} BS_i > 200, \\ 1 & \text{if } \bigwedge_{i=1}^{3} 200 \geq BS_i > -1, \\ NA & \text{otherwise,} \end{cases} \quad (4)$$

where SS corresponds to the three seat sensors, BS the three back sensors, and t is the time step at which the feature is being computed. These were found to be useful features by both Cooper et al. [4] and D'Mello et al. [5].

To derive features, we find the minimum, maximum, mean, and standard deviation for each reading over some time frame. Previous studies used window sizes of 150 seconds preceding the aspect being predicted [4, 11]. In our setting, we used three appropriate time frames: aggregating the features from the beginning of the task, from the beginning of the search, and thirty seconds preceding the end of the search where we are predicting frustration. The first two are equivalent to the query log task and search features, respectively. In addition, we decided to use two versions of the each window: one that ignored any segments of time where a user was responding to a feedback prompt and a version that used those time segments. See Section 4 for details about the feedback prompts.

In total, this yields (6 camera readings + 1 mouse reading + 3 chair readings) × {min | max | mean | stddev} × {task | search | 30-seconds} × {prompts | no-prompts} = 240 features.

5.3 Models

We consider two baselines for this study: (1) always predicting users are frustrated and (2) predicting they are frustrated only when they have abandoned their query (i.e., they did not click on anything). We believe the latter is a reasonable approximation of frustration.

We construct six additional models using logistic regression. All features were normalized per user prior to training. One model uses all of the features from both the sensors and the query logs and is referred to as QL+Sensors. Three of the models are based on sequential forward feature selection on just the query log features, just the sensor features, and all the features. We name these SFS-QL, SFS-Sensors, and SFS-QL+Sensors, respectively. The sequential forward selection process starts with an empty feature set, considering all of the features under consideration as 'unused.' On each iteration of the algorithm, the unused feature that performs best in combination with the current pool of 'used' features is moved from the 'unused' to the 'used' pool. The algorithm stops when no improvement in performance is made. The 'used' features are the final selection.

We optimized our feature selection for $F_{\beta=0.5}$ using macro precision and recall at any logistic regression score threshold. For example, if a subset of features achieved a macro F-score of 0.6 with a score threshold of 0.5 and another subset achieved an F-score of 0.7 at a score threshold of 0.4, the latter would be selected and the corresponding threshold noted. The features selected for each model are listed in Table 5. For the query log features, *task_MaxQrCharLen* is the maximum length (in characters) of any query seen so far in a task; *task_Duration* is the time, in seconds, of the task; *task_QryPropUnq* specifies the number of unique queries seen so far in a task; *task_AvgPgMaxScroll* is the mean average max scroll per page per query in the task; *search_RsltsVisitedPrev* is the number of results visited during a search that were visited previously in the task.

SFS-QL	
task_MaxQryCharLen,	task_Duration,
task_QryPropUnq,	search_RsltsVisitedPrev,
task_AvgPgMaxScroll	
SFS-Sensors	
task_inclPrmpts_thinkingConf-stddev,	
wind30s_inclPrmpts_sitForward-max,	
search_inclPrmpts_disagreeConf-stddev,	
search_noPrmpts_agreeConf-mean,	
task_noPrmpts_netSeatChange-mean	
SFS-QL+Sensors	
task_Duration,	task_QryPropUnq,
wind30s_noPrmpts_concentratingConf-stddev,	
search_noPrmpts_netBackChange-min,	
search_noPrmpts_concentratingConf-min,	
wind30s_noPrmpts_unsureConf-mean,	
search_inclPrmpts_unsureConf-min	
W&D	
search_QryCharLen,	search_AvgTokenLen,
task_Duration,	task_ActionCount,
task_AvgURLCount	

Table 5: The models derived from subsets of the query log and sensor feature sets. The meaning of the sensor feature names are self-evident based on Section 5.2.

We create a seventh model (the fifth to use logistic regression) based on the features that White and Dumais [15] found were most important for predicting search engine switching, which we refer to as W&D. The features in this model (Table 5) are: the most recent query's length in characters (*search_QryCharLen*), the average token length of the most recent query (*search_AvgTokenLen*), the duration of the task in seconds (*task_Duration*), the number of events in the task (*task_ActionCount*), and the average URL count per task for the current user (*task_AvgURLCount*).

The eighth model we explore is the Markov Model Likelihood (MML) used by Hassan et al. [9] to predict task success. We used the version that incorporates the time between events by using gamma distributions.

Given a list of event sequences from a training set, the MML builds two models: one for sequences that end in frustration and one for those that do not. Given a test event sequence, the log probabilities of the transitions are estimated and summed, once in each of the frustrated and non-frustrated transition models. The frustrated sum is divided by the non-frustrated sum, yielding a ratio between 0 and ∞, with scores closer to 0 meaning the sequence is more consistent with frustration, 1 being indifferent, and scores greater than 1 meaning the sequence is more consistent with non-frustration. We then use the following variation of Platt smoothing to transform the score into one more consistent with those output by our logistic regression scores:

$$MML(x) = \frac{1.0}{1 + e^{\alpha(-x)+\beta}}, \quad (5)$$

where x is the ratio and α and β are the smoothing parameters, set to 4 as determined by our personal judgment on the range of ratios output in the training and development phases.

Event	Description
Q	Enter query.
RF	Focus on a search results page.
RC	Click on a link on a results page.
S	Scroll.
OF	Focus on a non-results list page.
OC	Click on a link on a non-results list page.

Table 6: The event types used in conjunction with the Markov Likelihood Model.

The events we used for the MML model are listed in Table 6. The MML uses task-level event sequences—each instance consists of the sequence of events starting from the beginning of the task up until the point where frustration is being predicted. Duplicate events were ignored and the time between events was recorded in seconds. We refer to this model as MML+time in the rest of this paper.

On the training/development data (Section 7), the W&D model performed very well, so we decided to add the MML-time as an additional feature, creating the ninth model W&D+MML-time. We felt that the sequence information captured by the MML model would benefit the static features used by the W&D model.

6. RESULTS AND DISCUSSION

We randomly selected twenty of the thirty participants' data for training and development. In the training / development set, we put each user's data into its own fold, giving us a total of twenty folds. This avoids using a particular user's data for both training and testing for cross validation experiments.We used twenty-fold cross validation select features and tune the score threshold at which $F_{\beta=0.5}$, precision, and MAP are computed.

For the results presented here, we re-trained our models (maintaining the features and score thresholds) on all twenty users in the training/development set and tested on the remaining ten users. The macro-averages were calculated across users.

The training set contained 323 queries (51% of which were frustrated) for which there was feedback across 136 tasks for twenty users. The test set contained 137 query-feedback instances (45% of which were frustrated) across seventy-one tasks for ten users. One query from the training set and two from the testing set were removed due to logging errors that prevented the queries from being properly processed. We should note that during the study, two participants were accidentally given the same ordering of tasks and both users were randomly selected for testing. While this does increase the chances of ordering bias, we believe the effect is small due to the similar performance of the models on the training and testing set. Figure 1 shows the number of total and frustrated instances per user in the test set.

Table 7 shows the results of the experiments. Accuracy is measured across all users. However, the other three metrics are only measured for nine of the users, as user '25' never indicated frustration, causing the metrics to be undefined. The baseline model that assumes the user is frustrated if they abandon their query is undefined for three of the metrics. For precision and F, this is because of undefined values for certain users. For MAP, both no-clicks and always-frustrated are undefined since it involves ranking scores, and both baselines have only binary outputs.

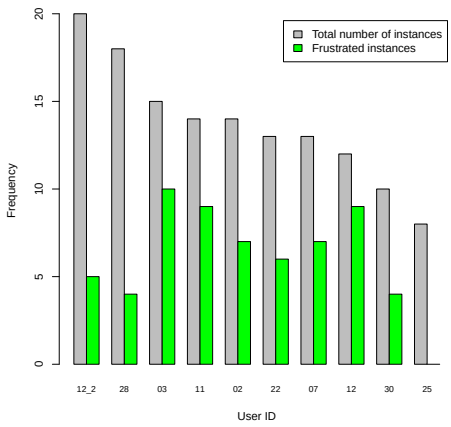

Figure 1: The total number of feedback instances and frustrated instances per user, ordered by total instances.

	Accuracy	Precision	$F_{\beta=0.5}$	MAP
W&D	0.75	0.81	0.80	0.87
QL+Sens	0.54	0.50	0.49	0.59
SFS-QL+Sens	0.69	0.74	0.72	0.85
SFS-QL	0.69	0.74	0.73	0.80
SFS-Sens	0.45	0.52	0.57	0.66
MML-time	0.66	0.61	0.62	0.68
W&D+MML	0.74	0.78	0.78	0.85
No clicks	0.57	—	—	—
Always frustrated	0.44	0.49	0.55	—

Table 7: Macro-level results for the models on the test set. Accuracy is over all ten users. The other three metrics do not include user '25', as the user had no frustrated instances. Accuracy, precision, and F were all computed using the threshold determined for each model in the development phase.

The metrics show that the relatively simple W&D model outperforms the rest for every metric. Not all differences are significant, however. As there is no concise way to illustrate significance for all pairs of systems for each metric, we will describe the most critical differences for $F_{\beta=0.5}$ and MAP. The three top performing models with respect to $F_{\beta=0.5}$—W&D, W&D+MML-time, and SFS-QL—are statistically different from the Always frustrated baseline and QL+Sens. W&D is statistically different from every other system except W&D+MML-time and SFS-QL, though the p-value for the difference between W&D and SFS-QL is just above α at 0.056. W&D is statistically different from all other models except W&D+MML-time for the MAP metric.

Figure 2 shows the 11-point interpolated average precision across users for each model. Using all features outperforms the baseline, but is much worse then selecting only a subset of the features (SFS-QL+Sensors). This graph suggests that the W&D+MML-time model outperforms using only W&D, which conflicts with the metrics presented in Table 7, but is understandable given the differences between the two systems are not statistically significant.

To understand some of the differences among the models in terms of MAP, Figure 3 shows average precision broken down by user. Some models have difficulty with particular

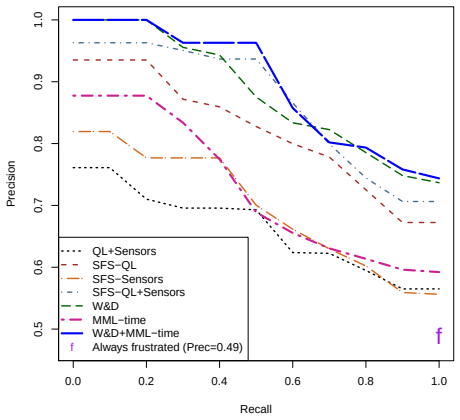

Figure 2: Macro 11-point interpolated average precision across the test users for each model.

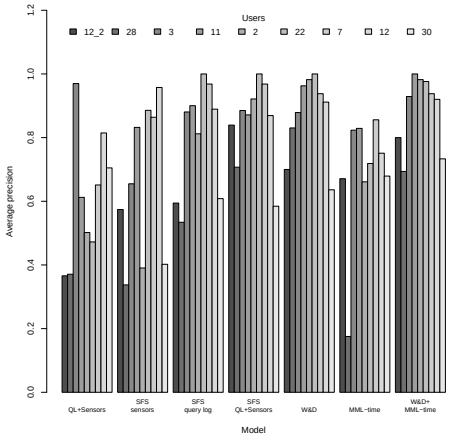

Figure 3: The average precision per use by model. Users are ordered by their total number of instances (i.e., queries for which they gave feedback).

users, such as user '28'—most models perform poorly with this user, except the W&D model. The SFS-QL+Sensors model performs best for user '12_2', though the combination model W&D+MML-time is close behind. The figure also shows us that adding sequence information to static query log features helps some users (e.g., user '30'), but hurts others (e.g, user '28'). This suggests that personalization could be useful in determining how much influence each feature type should have for a particular user.

As we mentioned in Section 3.2, macro $F_{\beta=0.5}$ is the metric we are most interested in. While we calculated this value using the score threshold for each model selected during development as shown in Table 7, observing how the score threshold affects the F-score can help us understand how stable each model is between the development and test sets. Figure 4 shows $F_{\beta=0.5}$ as the score threshold ranges from 0 to 1.0 for select models. We selected two models we believe are rather robust and two that are not. The W&D and W&D+MML-time models reach their optimum on the test set at nearly the same score threshold as in the development set. In contrast, the MML-time model peaks at a lower threshold on the test set. While the maximum for SFS-Sensors on the test set is close to its score threshold, there is a substantial decrease in performance just past its 0.5 cutoff. This suggests that the trained model is sensitive to new data.

6.1 Discussion

Many practical observations can be gleaned from the results, as well as a number of interesting questions raised. In this section we will discuss a few of each.

First, the results suggest that a few relatively simple query log features can reliably predict frustration. This is useful in developing a search system that predicts frustration. However, the information necessary to extract the most useful features is client-side (i.e., actions off of a search results page need to be recorded), which means the user would likely need to download a browser plug-in.

The features we found most useful for detecting frustration are the same as those White and Dumais [15] found most useful for detecting when a user will switch search engines. This suggests that this feature set may have a broader scope of predictive power for related tasks, such as task satisfaction and query abandonment.

One of the surprises of this research for us was the performance of the sensors. Given the results of Cooper et al. [4], we expected the sensors would strongly correlate with user-reported frustration and that we would struggle to find a set of query log features to even come close to the performance of the sensors. In fact, the opposite was true. There may be several reasons for this. First, the study occurred in an open room and up to five participants were active in the study at a given time. The presence of other participants may have affected how an individual maintained their composure. Arguments could be made that this is or is not a realistic situation; it probably varies by where people search (e.g., in an open office space vs. a cubicle vs. a living room). Another possibility is that the way the feedback was gathered upset the natural reactions of the participants. However, we hope that including a set of features that ignored sensor readings during the time intervals when prompts were shown would have removed such bias.

Another surprise was the performance of the MML-time model. We thought that the sequence data would have been more helpful than it was. One reason for its performance may be the event language. We used a simple, high-level set of events. This is in contrast to Hassan et al. [9], who used events such as the type of link clicked on a search results page. Adding more advance features may be more useful. However, there is another problem with sequences on our data set: data sparsity. While our data set is sufficient for static feature classification, there is likely an insufficient number of unique sequences to build a reliable model. A Web-scale data set, such as those used in other studies [6, 9], would be more useful in combination with this model. The trade off is that user-reported frustration is not included with Web-scale search logs.

Finally, we expected the sequential forward selection method to provide a better approximation to the optimal

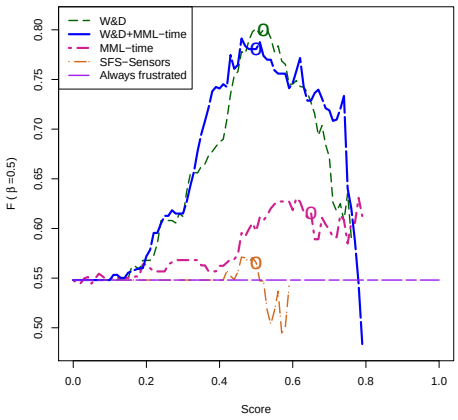

Figure 4: $F_{\beta=0.5}$ using macro precision and recall for select models over the ten test users. The circles denote the threshold that was chosen in development for each model.

feature sets. The W&D model is a subset of the query log feature set, and thus could vary well have been chosen. However, SFS is a simple greedy algorithm and falls victim to the same problems as all other greedy algorithms. Future work should explore more advanced selection techniques to find a better approximation for this task.

7. SUMMARY AND FUTURE WORK

In this paper we used features derived from a client-side query log and three physical sensors to predict user-reported frustration during Web search. We compared several models based on those used in both the information retrieval [8, 9, 15] and the intelligent tutoring systems [4, 5, 11] literature. We found that using a few simple query log features performed best.

In addition, the toolbar along with all of the data collected during the study are being made publicly available.[5] Much more data was collected than was used in this paper, such as user-reported page relevance, query satisfaction, and task satisfaction. All pages viewed were downloaded, including search results pages. Mouse movements were also collected, from which a useful feature could be derived for tasks such as frustration prediction, among others. The data set serves as a means for others to compare against the results of this paper, as well as provide insight into ways to build on the study design used.

One direction of future work is building a system that detects searcher frustration in real time. However, while the models we explored predict frustration well, they depend on client-side information. Another direction of future work is finding a set of useful features from the less-rich server-side information. This would allow a system to be built that does not depend on plugins or other client-side instrumentation. Finally, we intend to conduct a study that explores what types of interventions are appropriate for addressing searcher frustration and when to use them.

[5] http://ciir.cs.umass.edu/~hfeild/downloads/frustrationUserStudy/

8. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor. We would like to thank Beverly Woolf, David Cooper, and Winslow Burleson for loaning the sensors and logging software and Yahoo! for access to the modeling software.

References

[1] 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.

[2] D. Bilal and J. Kirby. Differences and similarities in information seeking: children and adults as Web users. *Information Processing and Management*, 38(5):649–670, 2002.

[3] 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.

[4] 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.

[5] S. D'Mello, R. Picard, and A. Graesser. Toward an affect-sensitive AutoTutor. *IEEE Intelligent Systems*, pages 53–61, 2007.

[6] D. Downey, S. Dumais, and E. Horvitz. Models of searching and browsing: languages, studies, and applications. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2740–2747. Morgan Kaufmann Publishers Inc., 2007.

[7] A. Druin, E. Foss, L. Hatley, E. Golub, M. L. Guha, J. Fails, and H. Hutchinson. How children search the internet with keyword interfaces. In *IDC '09: Proceedings of the 8th International Conference on Interaction Design and Children*, pages 89–96, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-395-2. doi: http://doi.acm.org/10.1145/1551788.1551804.

[8] 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.

[9] A. Hassan, R. Jones, and K. L. Klinkner. Beyond DCG: User behavior as a predictor of a successful search. 2010.

[10] 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.

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

[12] D. J. Lawrie. *Language models for hierarchical summarization*. PhD thesis, 2003. Director-Croft, W. Bruce.

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

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

[15] R. W. White and S. T. Dumais. Characterizing and predicting search engine switching behavior. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 87–96, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. doi: http://doi.acm.org/10.1145/1645953.1645967.

[16] I. Xie and C. Cool. Understanding help seeking within the context of searching digital libraries. *J. Am. Soc. Inf. Sci. Technol.*, 60(3):477–494, 2009. ISSN 1532-2882. doi: http://dx.doi.org/10.1002/asi.v60:3.