Measuring and Predicting Search Engine Users' Satisfaction

OVIDIU DAN and BRIAN D. DAVISON, Lehigh University

Search satisfaction is defined as the fulfillment of a user's information need. Characterizing and predicting the satisfaction of search engine users is vital for improving ranking models, increasing user retention rates, and growing market share. This article provides an overview of the research areas related to user satisfaction. First, we show that whenever users choose to defect from one search engine to another they do so mostly due to dissatisfaction with the search results. We also describe several search engine switching prediction methods, which could help search engines retain more users. Second, we discuss research on the difference between good and bad abandonment, which shows that in approximately 30% of all abandoned searches the users are in fact satisfied with the results. Third, we catalog techniques to determine queries and groups of queries that are underperforming in terms of user satisfaction. This can help improve search engines by developing specialized rankers for these query patterns. Fourth, we detail how task difficulty affects user behavior and how task difficulty can be predicted. Fifth, we characterize satisfaction and we compare major satisfaction prediction algorithms.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Measurement, Experimentation, Algorithms, Human Factors

Additional Key Words and Phrases: Abandonment, advanced users, browser plugin logs, browser toolbar logs, novice users, predicting satisfaction, query difficulty, query logs, query performance, satisfaction, search engine evaluation, search engine switching prediction, search sessions, search success, search tasks, task completion, task difficulty, user behavior models, user dissatisfaction, user frustration, user satisfaction, web search success

ACM Reference Format:
Ovidiu Dan and Brian D. Davison. 2016. Measuring and predicting search engine users' satisfaction. ACM Comput. Surv. 49, 1, Article 18 (July 2016), 35 pages.
DOI: http://dx.doi.org/10.1145/2893486

1. INTRODUCTION

Metrics such as mean average precision and discounted cumulative gain have been widely used to evaluate search engine performance. However, these measures require relevance judgments from human labelers. Such judgments are costly and have limited scale. Furthermore, collecting judgments directly from real users is even less practical if not impossible.

Mining user behavior directly from query log data is a promising line of research that can augment or replace manual judgments. Correlating explicit training data with implicit signals from logs can help build models to predict user satisfaction. These models can then be used by search engines to improve their services, increase user retention, and grow market share.

Authors' address: Computer Science & Engineering Department, Lehigh University, Packard Laboratory, 19 Memorial Drive West, Bethlehem PA, 18015; emails: {davison, ovd209}@cse.lehigh.edu.

DOI: http://dx.doi.org/10.1145/2893486

In this survey, we summarize research on characterizing and predicting the satisfaction of search engine users. Table I contains an overview of the research areas and references covered by this survey. We organize research on search engine users' satisfaction in the following five major areas:

- **Search engine switching** (Section 3): When users switch engines, they often do so because they are dissatisfied. In this section, we summarize literature on characterizing behavior signals that occur before users transition between engines. We also discuss methods of predicting search engine switches.
- **Good and bad abandonment** (Section 4): Traditionally, the information retrieval community has used query abandonment as a negative signal for relevance. More recent work has shown that, in fact, in around 30% of abandoned queries the users were actually satisfied as the answers or snippets on the page provided them with the information they were looking for.
- **Query difficulty and performance** (Section 5): Users have varied information needs, and search engines need to be general enough to cover most use cases. Unfortunately, this also means that, in some cases, users are left dissatisfied. Finding groups of such underperforming queries is required in order to train better ranking models.
- **Task Difficulty** (Section 6): Task Difficulty is a subjective assessment on the search effort needed to find information. This perceived difficulty can affect user behavior. Factors that affect difficulty include the search experience and domain knowledge of the user. Predicting difficulty can help search engines adapt their interface and ranking algorithms.
- **Predicting satisfaction** (Section 7): Obtaining explicit satisfaction ratings from users is challenging. Therefore, there is a need to approximate user satisfaction with models that use implicit behavior features to predict satisfaction at the query, search task, or session level.

Customer satisfaction has been an extensively discussed subject in the areas of consumer, marketing, and psychology research since the mid-1970s [Hennig-Thurau and Klee 1997]. In this context, satisfaction has been defined in numerous ways, including "the buyer's cognitive state of being adequately or inadequately rewarded for the sacrifices he has undergone" [Engel and Blackwell 1982] and "the consumer's response to the evaluation of the perceived discrepancy between prior expectations and the actual performance of the product as perceived after its consumption" [Tse and Wilton 1988].

In information retrieval, search satisfaction is defined as the fulfillment of a user's information need [Feild et al. 2010]. We can draw clear parallels from this definition to the ones used in marketing. In both cases, users have certain expectations of the performance of the product or service and their satisfaction is directly linked to how well the service actually meets these expectations.

The theory of *disconfirmation* has been used in marketing literature to frame work related to satisfaction. Disconfirmation arises from discrepancies between prior expectations and actual performance [Churchill Jr. and Surprenant 1982]. Expectations reflect the level of performance anticipated by a consumer before buying the product or service. Performance is the actual quality of the product as experienced by the consumer. Satisfaction and dissatisfaction are then given by the magnitude of the disconfirmation effect, that is, how much better or worse the experience of the user was than what she expected.

The cumulative effect of positive disconfirmation over a long period of time can lead to *loyalty*, which is a deeply rooted commitment to repurchase a product or reuse a service in the future [Oliver 2006]. Prior research in marketing has shown that a loyal customer may be worth up to 10 times as much as an average one [Anderson and Srinivasan 2003]. We can also see this effect in the context of research in search

Table I. Major Areas of Search Engine User Satisfaction Research

Search engine switching

[Mukhopadhyay et al. 2004] Competition between Internet search engines

[Juan and Chang 2005] An analysis of search engine switching behavior using click streams

[Heath and White 2008] Defection detection

[Laxman et al. 2008] Stream prediction using a generative model based on frequent episodes in event sequences

[White et al. 2008] Enhancing web search by promoting multiple search engine use

[White and Dumais 2009] Characterizing and predicting search engine switching behavior

[Feild et al. 2010] Predicting searcher frustration

[White et al. 2010] Modeling long-term search engine usage

[Guo et al. 2011] Why searchers switch

[Savenkov et al. 2013] Search engine switching detection based on user personal preferences & behavior patterns

Good and bad abandonment

[Li et al. 2009] Good abandonment in mobile and PC internet search

[Stamou and Efthimiadis 2010] Interpreting user inactivity on search results

[Chilton and Teevan 2011] Addressing people's information needs directly in a web search result page

[Huang et al. 2011] No clicks, no problem

[Chuklin and Serdyukov 2012a] Good abandonments in factoid queries

[Chuklin and Serdyukov 2012b] Potential good abandonment prediction

[Diriye et al. 2012] Leaving so soon?

[Song et al. 2014] Context-Aware Web Search Abandonment Prediction

Query difficulty and performance

[Cronen-Townsend et al. 2002] Predicting query performance

[He and Ounis 2004] Inferring query performance using pre-retrieval predictors.

[Carmel et al. 2006] What makes a query difficult?

[Hauff et al. 2008] Improved query difficulty prediction for the web

[Guo et al. 2010] Predicting query performance using query, result, and user interaction features

[Dan et al. 2012] Mining for insights in the search engine query stream

[Kim et al. 2013] Playing by the rules

[Hassan et al. 2013] Toward self-correcting search engines

Task difficulty

[Kim 2006] Task difficulty as a predictor and indicator of web searching interaction

[White and Morris 2007] Investigating the querying and browsing behavior of advanced search engine users

[Aula et al. 2010] How does search behavior change as search becomes more difficult?

[Liu et al. 2010a] Can search systems detect users' task difficulty?

[Liu et al. 2010b] Predicting task difficulty for different task types

[Liu et al. 2012a] Task difficulty and domain knowledge effects on information search behaviors

[Liu et al. 2012b] Exploring and predicting search task difficulty

[Hassan et al. 2014] Struggling or exploring? Disambiguating long search sessions

[Liu et al. 2014] Predicting Search Task Difficulty at Different Search Stages

[Arguello 2014] Predicting Search Task Difficulty

Predicting satisfaction

[Fox et al. 2005] Evaluating implicit measures to improve web search

[Huffman and Hochster 2007] How well does result relevance predict session satisfaction?

[Al-Maskari et al. 2007] The relationship between IR effectiveness measures and user satisfaction

[Hassan et al. 2010] Beyond DCG

[Ageev et al. 2011] Find it if you can

[Hassan et al. 2011] A task level metric for measuring web search satisfaction and its application on improving relevance estimation

[Hassan 2012] A semi-supervised approach to modeling web search satisfaction

[Hassan and White 2013] Personalized models of search satisfaction

[Hassan et al. 2013] Beyond clicks

[Kim et al. 2014] Modeling dwell time to predict click-level satisfaction

[Wang et al. 2014] Modeling Action-level Satisfaction for Search Task Satisfaction Prediction

[Jiang et al. 2015] Understanding and Predicting Graded Search Satisfaction

engines. For instance, both White et al. [2010] and Hu et al. [2011] find a positive correlation between search success and the rate of search engine re-use. In other words, the amount of positive and negative experiences in time has an effect on the behavior of users. Bolton reaches a similar conclusion in marketing research. He finds that customers who have a higher prior cumulative satisfaction have longer relationships with the organization [Bolton 1998].

Consumer research has found that *inertia* can suppress the impact of satisfaction to some degree, as 40% to 60% of shoppers visit the same store out of habit rather than satisfaction [Beatty and Smith 1987]. Interestingly, even though switching cost is close to zero when it comes to search engines [Mukhopadhyay et al. 2004], we still find the same effect when modeling long-term search engine usage. White et al. [2010] find that users of one of the studied search engines had a negative correlation between usage and satisfaction. They suggest users might use that particular engine because of factors beyond satisfaction, such as loyalty inertia.

Jones et al. [1995] study the relationship between satisfaction and loyalty in industries with different regulatory and competitive environments, such as telephone companies, hospitals, and computer manufacturers. They stress the importance of keeping customers completely satisfied in highly competitive industries. As an example, they use Xerox Corporation, which discovered that *merely* satisfied customers were 6 times less likely to buy products again when compared to *totally* satisfied customers. Search engines are similarly a highly competitive industry, especially in international markets.

The same study classifies users into four different classes depending on their satisfaction, loyalty, and behavior: *loyalists*, *mercenaries*, *defectors*, and *hostages* [Jones et al. 1995]. Although research in search satisfaction uses different names for categories, large-scale and long-term analysis on the behavior of search engine users has found that they conform exactly to the first three categories. White and colleagues [2010] have found three behavioral patterns: *no switch* users, corresponding to *loyalists*, are customers who never switch from their primary search engine; *switch* users, which correspond to the *defectors* category, permanently switch from one engine to another; and *oscillating* users, which are the same as *mercenaries*, frequently change between engines. The fourth category, *hostages*, cannot be found among search engine users as there is intense competition between search engines and there is no monopoly in search, at least internationally.

In summary, research in search satisfaction often mirrors findings in the much older field of customer satisfaction. Similarly to large manufacturing or services companies, search engines need to model and track the satisfaction of search engines in order to remain competitive and gain market share. In the next sections we present an overview of the current state of the art in characterizing and predicting user satisfaction in the context of search engines.

2. DEFINITIONS

Definition 2.1 (*Abandonment*). Abandonment occurs when a search engine user does not click on any results shown by the search engine. Refer to Section 4 for alternate definitions of *abandonment*.

Definition 2.2 (*CG*). An acronym for *cumulative gain*. It is a measure of ranking quality obtained through human judgments that sums the gain (relevance) of each result, regardless of its position in the ranking [Järvelin and Kekäläinen 2002].

Definition 2.3 (*DCG*). An acronym for *discounted cumulative gain*. It is a measure of ranking quality obtained through human judgments. Unlike CG, gain is accumulated

from the top of the result list to the bottom with the gain of each result discounted at lower ranks. See Järvelin and Kekäläinen [2002] for additional details and a formal definition.

Definition 2.4 (*DSAT*). An abbreviation for *dissatisfaction*, often used as a class when predicting if users were satisfied (*SAT*) or dissatisfied (*DSAT*) with the results of a web search.

Definition 2.5 (*F-Measure, F_1 score, $F_{0.5}$ Measure*). F-Measure (also known as F_1 score) is the weighted harmonic mean of precision and recall:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}. \tag{1}$$

$F_{0.5}$ Measure is a variant of F-Measure that puts twice as much emphasis on precision than recall. For more details, please refer to the information retrieval textbook by Manning et al. [2008].

Definition 2.6 (*Good Abandonment*). An instance of a query has good abandonment if the user's information need was successfully addressed by the search results page, with no need to click on a result or refine the query [Li et al. 2009]. Refer to Section 4 for different types of *good abandonment*.

Definition 2.7 (*Language model*). Language models are probability distributions over sequences of words. They can be used in many fields, such as speech recognition, document similarity, and machine translation.

Definition 2.8 (*MAP*). An abbreviation that stands for *mean average precision*. In information retrieval, its value is the mean of the average precision scores for each query in a query set.

Definition 2.9 (*NDCG*). An acronym for *normalized discounted cumulative gain*. It is the discounted cumulative gain measure after normalization to range from 0 to 1. See Järvelin and Kekäläinen [2002] for additional detail and formal definition.

Definition 2.10 (*ODP*). An acronym for the *Open Directory Project* located at http://www.dmoz.org/, which is a directory of websites organized in categories. The website is maintained by volunteers.

Definition 2.11 (*Overfitting*). In machine learning and statistical pattern classification, *overfitting* refers to the situation in which a trained model is so complex that it can classify training data very well but is unable to perform well on new (never before seen) patterns [Duda et al. 2000].

Definition 2.12 (*Precision*). In the context of information retrieval, precision is the fraction of the returned results that are relevant to the information need [Manning et al. 2008]:

$$Precision = \frac{\#Relevant \bigcap \#Retrieved}{\#Retrieved}. \tag{2}$$

Definition 2.13 (*Recall*). In the context of information retrieval, recall is the fraction of the relevant documents in the collection that were returned by the system [Manning et al. 2008]:

$$Recall = \frac{\#Relevant \bigcap \#Retrieved}{\#Relevant}. \tag{3}$$

Definition 2.14 (*SAT*). An abbreviation for *satisfaction*, often used as a class when predicting if users were satisfied (*SAT*) or dissatisfied (*DSAT*) with the results of a web search.

Definition 2.15 (*Search Engine Switching*). Search engine switching is the voluntary transition of users between search engines [Guo et al. 2011].

Definition 2.16 (*Search Frustration*). *Frustration* during a search session can be defined as the impediment of search progress [Feild et al. 2010]. When users are frustrated, they often give up the search or they switch to another engine.

Definition 2.17 (*Search Goal*). A *search goal* is an atomic information need, resulting in one or more related search queries issued to accomplish a single discrete task [Hassan et al. 2010; Guo et al. 2011].

Definition 2.18 (*Search Satisfaction*). In the context of information retrieval, Feild et al. [2010] define *satisfaction* as the fulfillment of a user's information need. In other words, a search session has ended in user satisfaction if the information need of the user has been successfully addressed by the results [Huffman and Hochster 2007].

Definition 2.19 (*Search Session*). A *search session* is a sequence of user activities that begins with a query, includes subsequent queries and URL visits, and ends with a period of inactivity [Guo et al. 2011]. A 30-minute inactivity threshold is widely used to demarcate the end of a session in several related articles [Juan and Chang 2005; White and Drucker 2007; Heath and White 2008; White et al. 2008; Guo et al. 2011; Hu et al. 2011; Hassan and White 2013].

Definition 2.20 (*Search Trail*). A *search trail* is an ordered sequence of actions performed by the user during a search goal [Hassan 2012]. Search trails originate with a search and proceed until a point of termination, where it is assumed that the user has completed his or her information-seeking activity. White and Drucker [2007] define several termination activities that can be used to determine the endpoints of search trails, including returning to the browser home page, checking web e-mail, typing a URL in the address bar, visiting a bookmarked page, a timeout of 30 minutes, or closing the browser window.

Definition 2.21 (*SERP*). An acronym for *search engine results page*. It refers to the page that displays search results when a user issues a query to a search engine.

Definition 2.22 (*TREC*). *TREC*, the Text REtrieval Conference, is a series of meetings that encourage research in information retrieval by providing large test collections to evaluate different retrieval systems. The workshops are organized by the National Institute of Standards and Technology and the Intelligence Advanced Research Projects Activity.

3. SEARCH ENGINE SWITCHING

Search engine switching is the voluntary transition of users between search engines. From the perspective of a user, both the cost of using a search engine and the barrier of switching to another engine is typically close to zero [Mukhopadhyay et al. 2004; White et al. 2010]. However, users transitioning to competing services can have a major economic cost to search engines [Heath and White 2008; White and Dumais 2009]. Consequently, search engines have an incentive to track and reduce the number of switches. Research in this areas focuses on both characterizing and predicting search engine switches.

Engine switches can be classified in three categories depending on when and how often users switch search engines [White et al. 2008]:

- **Within session switching**: Users switch between multiple search engines within the same session. Most literature in this area focuses on this type of switching.
- **Between session switching**: Users pick a particular search engine to use for the entirety of a new session if they feel their particular information need is best served by that search engine due to certain features or verticals. Such users might use different search engines for different sessions.
- **Long-term switching**: Finally, some users switch search engines and never return to the original one. White et al. [2010] further break down this type of behavior by describing users who oscillate between two or more search engines over a long period of time.

Comparing the query issued immediately after a switch to the one before it in the same session can yield the following classification [Guo et al. 2011]:

- **Same Query**: The pre and post-switch queries are identical. As discussed below, in this case the user is often dissatisfied with the results found with the previous search engine. Large-scale log analysis by White and Dumais on 1.1 billion search sessions [2009] has shown that 12.6% of switches exhibited the same pre-switch and post-switch query. A study with a much smaller scope of only 562 instances has shown that the queries are the same in 32% of the cases [Guo et al. 2011].
- **Related Queries**: The queries are not identical but they do share at least one non-stop-word in common. The analysis by Guo et al. [2011] has shown that approximately 50% of query pairs have at least one non-stop-word in common. The *same query* category is a subset of this category.
- **Different Queries**: The two queries do not share any non-stop-word terms. In work by Guo et al. [2011], this accounts for 50% of the pairs.

Users can initiate switching engines in several ways. White and Dumais [2009] have classified 58.6 million switching instances collected from the logs generated by an opt-in browser toolbar into three action categories:

- **Browser**: The query is issued directly in a browser search box or toolbar by first selecting the new search provider if needed. This transition action was the most popular one, accounting for 69.2% of all instances.
- **Navigate**: The user first visits the home page of the new search engine by inserting the URLs in the browser address bar and then issues the query. This case covered 18.3% instances.
- **Query-to-Navigate**: The user performs a search on the old search engine with the name of the new search engine, then visits the new search engine's homepage. This action is the least popular, accounts for only 12.5% of switching instances.

3.1. Reasons for Switching

Users choose their primary search engine based on factors such as reputation, familiarity, effectiveness, and interface usability [Fallows 2005].Users often also switch between two or more search engines due to a variety of reasons, including:

- **Dissatisfaction with search results** [Heath and White 2008; White et al. 2008; White and Dumais 2009; Guo et al. 2011]
- **Bad interface** [White et al. 2008]
- **Curiosity, Verification** [White and Dumais 2009]
- **Coverage** [White and Dumais 2009; Guo et al. 2011]

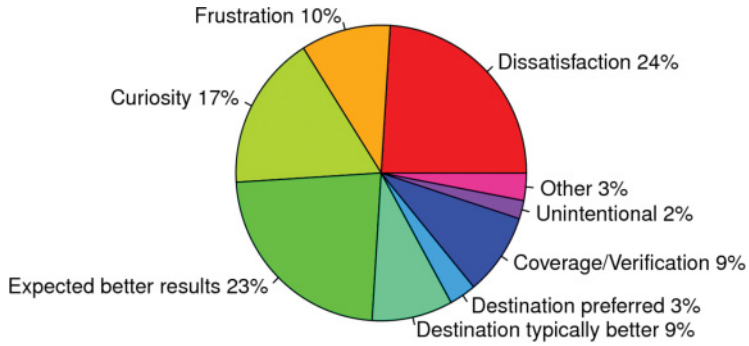

Fig. 1. Reasons given for search engine switching [White and Dumais 2009].

- **Destination preferred or typically better** [White and Dumais 2009; Guo et al. 2011]
- **Advertising campaign or word of mouth** [White et al. 2008]
- **Better for task type** [Guo et al. 2011]
- **By accident** [White et al. 2008; White and Dumais 2009; Guo et al. 2011]

A survey of 488 Microsoft employees by White and Dumais [2009] addressed the main reasons why users switch search engines. The results of the survey are reproduced in Figure 1. The findings show that dissatisfaction with the results (dissatisfaction, frustration, expected better results) is the most important reason for switching, with 57%, followed by a desire to verify the original search engine or augment information with secondary sources (coverage, verification, curiosity) at 26%.

Guo and colleagues [2011] have carried out a similar survey at the switching instance level using a browser plugin that displayed a questionnaire whenever users switched search engines. A breakdown of the 562 assessments shows that dissatisfaction accounts for 26% of switches, followed surprisingly by 22% of switches labeled as unintentional. A more detailed analysis on the type of pre- and post-switch queries suggests the unintentional switches are high only when the pairs of queries are related or differ completely. When the queries are the same, the unintentional rate is only 5%, which suggests the other instances could be different search tasks. Other reasons for switching included transitioning to their usual engine, at 20%, and picking an engine because it is better at a particular task type, at 11%.

An interesting subset of user behavior related to switching is *frustration*. While White and Dumais [2009] classify all instances of frustration as dissatisfaction, Feild and colleagues [2010] take a more nuanced approach. They state that users might become frustrated even if ultimately they succeed in their search task. Their breakdown of user-reported task success shows that users were successful in 62% of the tasks, but in one third of these cases they experience some degree of frustration.

3.2. Benefits of Switching

In the previous section we have looked at the reasons why users decide to switch. We will now investigate if users get any benefit once they do switch. Mukhopadhyay et al. [2004] point out that since search engines use different algorithms and return different search results for the same query, there is always a degree of "residual demand" for smaller or lesser quality engines, even if there is a high-quality dominant engine in the market.

The main benefit of switching search engines is higher-quality results. In order to confirm and quantify this benefit, White and colleagues [2008] use NDCG and click-through rates as measurements of quality. Within the 4,921 queries that were issued

Table II. Switching Prevalence in Large-Scale Log Analysis

Paper	[Heath and White 2008]	[White et al. 2008]	[White and Dumais 2009]	[Savenkov et al. 2013]
Users	"Hundreds of thousands"	5 million	14.2 million	N/A
Sessions	N/A	N/A	1.1 billion	8.6 million
Time	3 months	5 months	6 months	1 month
Source	Browser plugin	Browser plugin	Browser plugin	N/A
Users Switch	**50% of users** switched at least once a month	**36.4% of users** switched at least once across all 5 months	**72.6% of users** switched at least once across all 6 months	**N/A**
Sessions Switch	**8% of sessions** contained at least one switch	**6.8% of sessions** contained at least one switch	**4% of sessions** contained at least one switch	**16.5% of sessions** contained at least one switch

at least 5 times to each engine, for around 52.3% of them the results would be more accurate if they would be issued on a different search engine. Similarly, switching would improve the clickthrough rates by 54.5%.

An analysis carried out on a separate large-scale dataset covering 6 months of browser toolbar logs reinforces the hypothesis that switching search engines can be beneficial to search quality [White and Dumais 2009]. Using only clickthrough as a proxy for quality, the authors suggest that around half of the switches were successful because they were immediately followed by a search engine result click. Furthermore, users immediately switched back to the original search engine in only 20% of all switches and around 6% of switches lead to the use of a third engine. This again strengthens the case for transitioning between engines.

3.3. Switching Prevalence

While developing an economic model of search engine choice, Mukhopadhyay and colleagues [2004] hypothesized that users may often sample more than one search engine during a single session. Several follow-up large-scale studies summarized in Table II have confirmed that this is often the case. The studies show that 36.4% to 72.6% of users switched engines at least once in certain time spans, and between 4% and 16.5% of all sessions contained at least one switch. This suggests that switches are relatively commonplace and could have an adverse effect on the market share of search engines if users choose to defect permanently.

It is unclear why the statistics presented in Table II vary wildly among articles, but we can entertain several possible explanations. First, the literature is often vague as to what constitutes an engine switch during a session, so there could be differences in the switching detection methodology between studies. While White and Dumais [2009] characterize three separate types of engine switches, as described earlier in this section, no other work goes into such detail. It is possible that the large percentage of users that have switched engines at least once in half a year is due to the more comprehensive switch detection employed by the authors. Second, authors often filter data in different ways before computing statistics. For instance, Heath and White [2008] mention that they remove all users with fewer than five search sessions per month, while White and colleagues [2008] remove users with five or fewer queries across the entire dataset. However, White et al. [2008] also state in a footnote that if they vary the thresholds between 1 and 10 queries, then the proportion of users ranges from 26.7% up to 54%, confirming our hypothesis that data filtering can play a large role in the variety of results we have found. Third, data collection can also influence the results depending

Table III. Methods and Results for the Traditional Switching Prediction Task

Paper	Method	Sample size	Results
[Heath and White 2008]	Encode session actions as characters. Predict based on running count of substrings in session.	Train on half a month of browser plugin logs, test on the other half.	**Precision greater than 85% only at 5% recall.**. Poor precision and recall curve.
[Laxman et al. 2008]	Encode session actions as characters. Predict using a generative model based on mixtures of episode-generating Hidden Markov Models.	Train on half a month of browser plugin logs, test on the other half.	**Precision greater than 95%** at recalls 75%–80% only for sessions longer than 16 events. Recall drops to ∼22% for sessions of minimum length 4.
[White and Dumais 2009]	Encode session actions as characters. Cast as classification problem using logistic regression. Query, session, and user level features.	Test of 100,000 randomly chosen sessions, test on 10,000. Extracted from 6 months of browser plugin logs.	**Precision 23% at 10% recall** for sessions with three of more queries.
[Savenkov et al. 2013]	Personalized switch prediction.	27 days of sampled Yandex query logs	**AUC score of 0.845**

on the browser used, the demographics of the user, the geographical location of users, and search engine preference of the entire population.

3.4. Predicting Switching

Preempting user switches could help search engines improve user retention and consequently increase search engine revenue [Heath and White 2008]. Since often the full browsing log of users is not available, predicting switches based on incomplete and one-sided query logs can also provide valuable insight on the performance of a search engine [Savenkov et al. 2013]. Once a search engine has detected that a user is likely to switch, it could take actions such as offering a new search interface or providing a completely different search experience, such as an instant messaging conversation with a domain expert [Heath and White 2008].

While conducting preliminary analysis, several authors have observed actions that correlate with engine switches. Heath and White [2008] have found that users are less likely to click on non-algo links in the SERP before switching and that users who have recently switched are more likely to switch again. A study by White and Dumais [2009] finds that switches are more likely to occur in longer sessions, when users issue the same query multiple times, when they do not click the results, and when they visit individual SERPs for only a short time. Furthermore, Guo et al. [2011] note that longer queries, longer time between queries, smaller number of SAT clicks, and more bounces all indicate a higher chance of switching.

Based on these observations the articles propose several search engine switch prediction methods. Table III summarizes the methods and results for the traditional switching prediction task, which aims to predict whether a user will switch to another search engine during a given session. The results show that this task is still an open problem as most evaluations yield high precision only at low recall and vice versa.

Heath and White [2008] were the first to encode user actions as alphabet letters in the context of search engine switching. Previously, this concept was introduced in the context of information retrieval by Fox et al. [2005]. Later, encoding behavior as characters for switch prediction was also used by Laxman and colleagues [2008], as well as White and Dumais [2009]. Refer to the citation graph in Figure 2 for a visual representation of the relationship between the articles discussed here. An arrow from one article to another signifies that the first article cites the second article. If an arrow

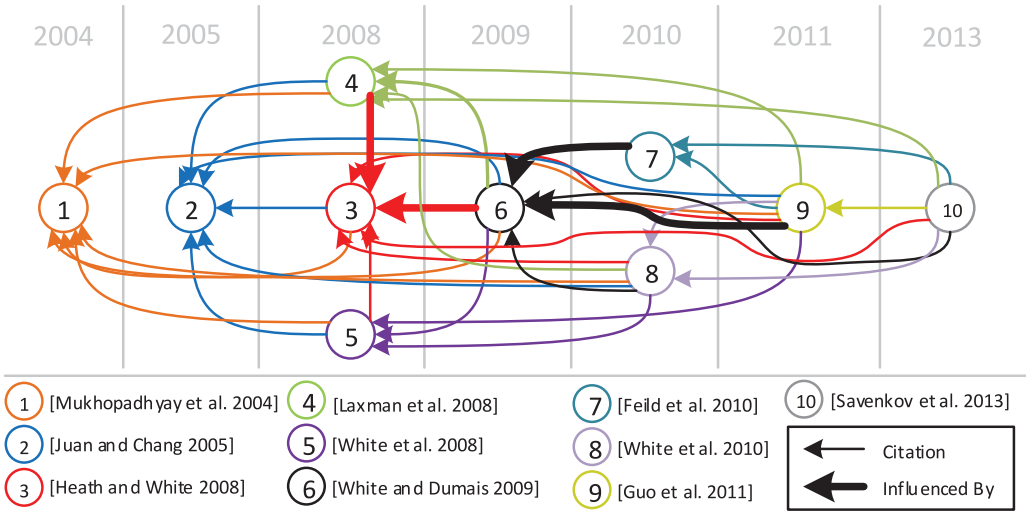

Fig. 2. Citation graph among Search engine switching articles.

is slim, then it signifies a simple citation, while if the arrow is thick the citation signifies that the first article heavily builds on methods described in the second article and/or that it borrows datasets from the second article.

The work by Savenkov et al. [2013] is particularly interesting because it describes personalized models of switch prediction. To achieve this, the authors build two types of personalized models. First, they create a personalized version of the Markov model used by Hassan and colleagues [2011] for predicting satisfaction and adapt it to the switching task. Since user level data are sparse, they smooth the personalized models using the global one by training a classifier where the Markov models are treated as features. Second, they propose a personalized machine-learning approach that uses logistic regression and gradient boosting trees for classification.

There are several task variations that differ from traditional engine-switching prediction. The subtask of predicting searcher frustration is addressed by Feild et al. [2010]. While, as described previously, not all user frustration leads to engine switches, it is nonetheless a closely related problem. To solve the problem, the authors build seven models that use logistic regression. Throughout the models, they make heavy use of features from previous work in engine switches [White and Dumais 2009] and search success prediction [Hassan et al. 2010]. The best results are obtained by a model that uses features from White and Dumais [2009], as well as a model that additionally also uses the time features from Hassan and colleagues [2010]. The highest accuracy, $F_{0.5}$, and MAP scores at 0.75, 0.80, and 0.87, respectively, are achieved by the former model, while the latter model achieves the highest precision at 0.85.

Another variation of engine switching is presented by White et al. [2008], who, instead of predicting when a user is about to switch, determine for which queries the user would benefit from switching to another engine. Prediction is carried out using features extracted from the SERP pages of all candidate search engines. The goal is to suggest switching to users only when the relevance of results as computed using NDCG is higher on the alternate search engine. The proposed approach can attain high precision at low recall levels: around 0.8 precision at recall 0.05. The authors also describe results when only the features of the current search engine are available. As expected, in this case the precision is even lower at about 0.62–0.68, when recall stays the same.

Modeling long-term search engine usage is one more task related to switch detection. White and colleagues [2010] study 6 months of query logs collected through a browser toolbar and find that users can fall into three groups: users who never switch, users who oscillate often between search engines, and users who have permanently switched from one engine to another during the studied period. Oscillating users issue a significantly larger number of queries and are less satisfied than the other two groups. Users of one of the studied engines have a negative correlation between usage and satisfaction, which suggests they might be using the search engine due to factors beyond satisfaction, such as brand loyalty. The authors implement classifiers that predict if users will switch. Even using just 1 week of training data, the classifiers have better than random results. Using 10 weeks of training data, the accuracy increases above 55%.

Finally, Guo et al. [2011] address the problem of classifying the reasons why users switched. They focus on predicting among the following classes: *DSAT*, *Coverage*, and *Other*. *Coverage*, while not defined clearly, seems to refer to user-perceived insufficient topic diversity. They create both separate binary classifiers for each class, as well as a single multi-class version. The performance of the binary classifiers is high only for the *DSAT* class, which has an $F_{0.5}$ score of 85.69, while the baseline that uses class distribution has an $F_{0.5}$ score of 72.4. The same measures for the *Coverage* class are 47.84 and 27.12, respectively, while for the *Other* class they are 29.01 and 17.40. The classifiers include Post-switch features, which are usually not available to a single search engine. For the *DSAT* class, the $F_{0.5}$ measure reduces to 81.12 when using pre-switch features only.

A summary of the classes of features used for engine-switching tasks is shown in Table IV. Rows list the features and columns contain check marks whenever an article uses the corresponding feature. Looking at the check marks along rows shows how popular each feature is in solving the classification task. Session, query, and engagement features are the most widely used. In terms of the predictive power of features, White et al. [2008] note that SERP features contribute the most to determining which search engine can provide the best relevance for a given query, and White and Dumais [2009] find that session features, followed by query and user features, obtain the best performance for predicting search engine switches.

4. GOOD AND BAD ABANDONMENT

The information retrieval community has traditionally considered clicks on search results to signify a positive implicit feedback of relevance. There is ample work that supports this hypothesis. For instance, Joachims and others [2005] have found that there is significant agreement between implicit user clicks and explicit relevance judgments. Fox et al. [2005] have asked 179 participants to use an instrumented browser to solve search tasks and have found that 68% of clicks users were at least partially satisfied with the results. Conversely, the lack of clicks has been used as a negative signal for relevance. Some authors have found that unsuccessful search goals are 10 times more likely to end with abandonment than to end with any other end state [Hassan et al. 2010]. These findings have lead other authors to use clickthrough-based features to solve a variety of tasks, such as learning ranking functions, predicting satisfaction, and predicting query difficulty. Figure 3 presents a citation graph among the articles discussed in this section.

Abandonment has been defined in different ways in the literature. All definitions generally agree on the fact that abandonment is a lack of clicks for a particular query instance, but there are differences on the definitions of what a clicks is and how abandonment is detected. Li and colleagues [2009] consider a query instance as abandonment only if the user does not click on any result or issues any query for a 24-hour period. However, other authors define a query to be abandoned even if it is immediately

Table IV. Classes of Features Used for Tasks Related to Engine Switching

Feature Class and Top Features in class	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
Past actions in current session: Actions encoded in alphabet, then n-grams counted or actions used in Markov model.	✓	✓						✓
SERP Features			✓	✓				
Algo title: # characters, words			✓					
Algo snippet: # characters, words			✓					
Algo URL: domain tld, path features			✓					
How much does text in algos match query			✓					
Number and position of results				✓				
Number and position of other page blocks				✓				
Engagement / Behavior Features				✓			✓	✓
Clicks and click positions				✓			✓	✓
Breakdown per type of click (SAT clicks, bounces, etc.)							✓	✓
Query Features			✓	✓		✓	✓	
Length: # words, characters, stop words			✓	✓			✓	
Has advanced operators				✓				
Has spell correction				✓				
Session/Task Features				✓	✓		✓	✓
Time: between queries, clicks, dwell, and session/task total				✓	✓			✓
Session/task length in queries, URLs				✓	✓		✓	✓
Ratio of queries with clicks				✓				✓
Reformulations							✓	
Requeries								✓
User Features				✓				✓
Length: # sessions, queries / session, time / session URLs / session, words / query				✓				✓
Engine preference				✓				
Switch probabilities based on past history								✓
Engine Features				✓		✓		
Engine name				✓				
Query stats per engine						✓		
External Sensors: mouse tracking, char sensors, camera					✓			

[1] [Heath and White 2008] **[2]** [Laxman et al. 2008] **[3]** [White et al. 2008] **[4]** [White and Dumais 2009] **[5]** [Feild et al. 2010] **[6]** [White et al. 2010] **[7]** [Guo et al. 2011] **[8]** [Savenkov et al. 2013]

followed by another search [Stamou and Efthimiadis 2010; Huang et al. 2011; Chuklin and Serdyukov 2012b]. Diriye et al. [2012] further state that clicks that lead to other SERP pages should also be treated as abandonment. The article, which provides the most structured definition, also defines several conditions which can be used to detect abandonment, including manual requeries, closing the browser tab, manual URL entry, and timeouts.

Abandonment can be split into two types, good and bad, depending on whether the SERP itself has satisfied the user. While previously all abandonment was considered bad, Li et al. [2009] introduced the concept of good abandonment, which is defined as a query for which the user's information need was successfully addressed by the results themselves, with no need to click on a result or refine the query. Take, for instance, the query *weather in seattle*. Many search engines show the weather forecast in an "answer," which is a block of page meant to directly display the desired information to a user, without requiring them to click on any results [Chilton and Teevan 2011]. There are also cases where the text snippets in the traditional search results can also provide enough information as to fulfill the user's information need [Stamou and Efthimiadis 2010].

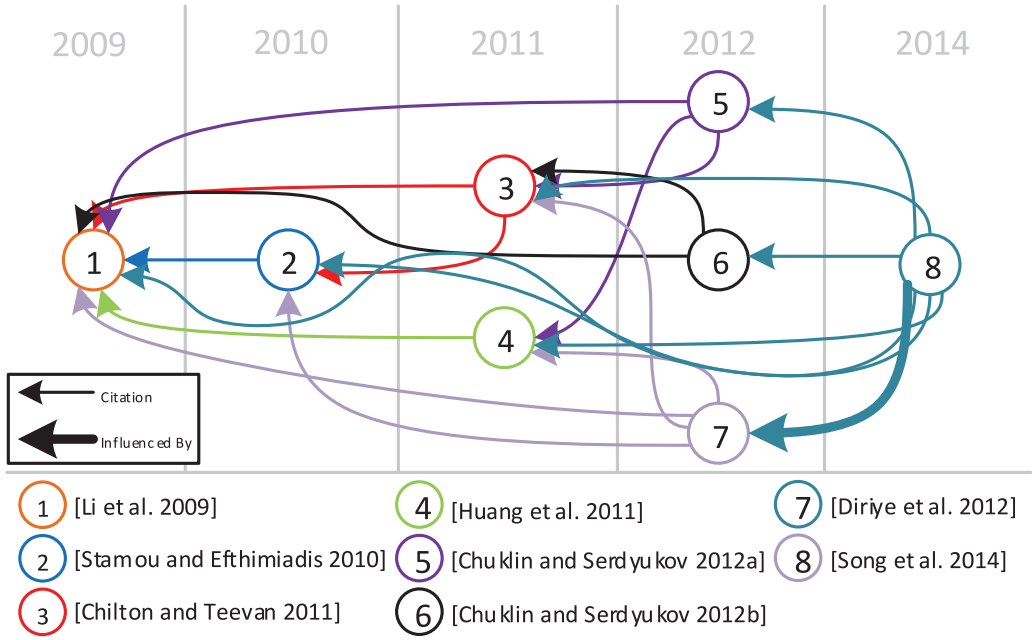

Fig. 3. Citation graph among good and bad abandonment articles.

4.1. Characterizing Good and Bad Abandonment

Potential good abandonment is another term first coined by Li and colleagues [2009]. It represents an upper bound of all queries where human labelers felt the query could theoretically be answered by reading the information on the SERP. The labelers made this determination solely on the query. *Likely good abandonment* is a subset of potential good abandonment queries for which the labelers, on studying the results shown to the user, have concluded that the information on the page does indeed contain enough information to answer the query. They also consider this to be an upper bound because not all users might pay attention to the snippets on the page as judges do and might miss relevant information. The authors consider the difference between potential and likely good abandonment queries to be *headroom* where search engines can improve.

Table V summarizes reported occurrences of good abandonment among the datasets studied in literature. Depending on the way data were collected and labeled, we split good abandonment into *potential*, *likely*, and *self-reported*. Here *self-reported* means that, instead of relying on human labelers, the authors were able to ask the original users who performed the query if the query was satisfied by the results themselves. Generally, the consensus places the prevalence of good abandonment at about 30% to 40% of all abandoned query instances, which covers a significant amount of search engine traffic. For example, Diriye et al. [2012] find that 22% of 39,606 SERP visits were abandoned, which would place good abandonment at 7% of all traffic.

The studies listed in the table also cover a few interesting findings about user behavior and abandonment. Li and colleagues [2009] report that 70% of potential good abandonment mobile queries were successfully answered by the search engine versus 56% for PC searches. They explain this difference by mobile searchers having more focused and less complex information needs.

Figure 4 shows another breakdown of abandonment data [Diriye et al. 2012]. The X-axis lists end-state actions that cause the abandonment, while the bars across the

Table V. Good Abandonment in Abandonment Queries

	Data Sample	Good abandonment
[1]	Google logs. 400 abandoned mobile and desktop queries for Japan and US markets each, and 1,000 abandoned mobile and desktop queries for China each.	**Potential**: *Yes* and *Maybe* potential good abandonment was **32.3% to 54.8%** for mobile searches, and **19% to 31.8%** for desktop searches, depending on the market. **Likely**: *Yes* and *Maybe* likely abandonment was **17.3% to 41%** for mobile and **11.2% to 19.8%** for desktop, depending on the market.
[2]	Study on 6 users. Browser plugin and questionnaire. 705 queries with clickthrough, 261 without.	**Self-reported**: Out of 145 queries where users specifically expected snippets alone to answer the query and where there was no click, 75% of them actually did contain the correct answer. **Self-reported**: Out of 87 queries where users originally intended to click results, in 49.4% of cases there was no need to click as information was on SERP.
[3]	859 abandonment queries from Microsoft users labeled authors	**Likely**: **21%** of queries contained the answer in the snippet content.
[4]	1,245 abandonment queries from Yandex; labeled by authors.	**Potential**: **34%** of queries labeled as *Good abandonment*, **16%** as *Maybe*, **49%** as *Bad*
[5]	Retrospective survey of 186 Microsoft employees.	**Self-reported**: In **31%** of instances users abandoned past queries because they were satisfied.
[5]	Paper also analyzed 1,799 abandonment instances collected with browser plugin distributed to Microsoft employees.	**Self-reported**: In **38%** of instances users abandoned past queries because they were satisfied.
[6]	Same data set as the one used by Diriye et al. [2012], but made use of all available data, yielding 7,419 labeled abandonment instances from 928 participants.	**Self-reported**: Breakdown of abandoned queries was 3,104 SATs (**42%**), 2,524 DSATs, 501 Interrupted or Unimportant, and 1,077 Other.

[1] [Li et al. 2009] **[2]** [Stamou and Efthimiadis 2010] **[3]** [Huang et al. 2011] **[4]** [Chuklin and Serdyukov 2012b] **[5]** [Diriye et al. 2012] **[6]** [Song et al. 2014]

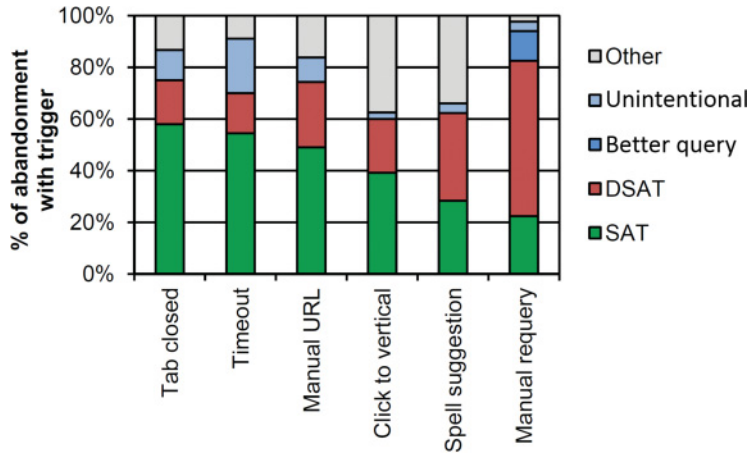

Fig. 4. Abandonment reasons broken down by trigger conditions [Diriye et al. 2012].

Y-axis break down the types of abandonment. The data show that when users close the browser tab, allow a timeout of 30 minutes without any click, or insert a new URL in the address bar, the actions correlate more with satisfaction, while when they manually requery the action correlates more with dissatisfaction.

Song et al. [2014] also analyze user behavior at the query and session level on 7,419 abandonment instances labeled by users. At the query level they find that query length is shorter on average for good abandonment instances when compared to bad abandonment, where the time to the next query is shorter for bad abandonment queries. At the session level, they determine that good abandonment queries are more likely to be the last query in the session, bad abandonment queries are more likely to be followed by reformulations, session length is shorter for good abandonment versus bad abandonment, and that two queries with bad abandonment in the same session are more likely to be reformulations of the same intent.

4.2. Predicting Abandonment

Preliminary studies on good abandonment have shown that query intent plays an important role in abandonment, since answers are only displayed for certain intents. Li and colleagues [2009] find that queries seeking local information or short answers are the top classes that lead to good abandonment. Stamou and Efthimiadis [2010] break down query intent for SERP pages with and without clicks. Their data similarly reveal that when the query intent is to get a quick answer like currency, search for local businesses, or translate words, the clickthrough rate is low. Conversely, informational, transactional, and navigational queries have much higher clickthrough rates.

Statistics generated on mouse cursor movement are able to distinguish between good and bad abandonment to some extent. Huang et al. [2011] find that cursor trail length, movement time, and cursor speed are all lower on average for good abandonment than for bad abandonment. However, mouse cursor movement information is usually not available on a large scale to search engines.

The quality of the snippets can be an indicator of good abandonment. Some results reveal that the relevance of snippets and the clickthrough rate are inversely correlated [Chuklin and Serdyukov 2012a]. That is, better snippets lead to lower clickthrough rates and higher good abandonment. Furthermore, users who find certain answers valuable often issue them repeatedly. Answers like weather or finance that users repeat often can lead to good abandonment [Chilton and Teevan 2011].

Building on previous user behavior research, several articles frame the problem of predicting abandonment as a classification problem [Chuklin and Serdyukov 2012b; Diriye et al. 2012; Song et al. 2014]. While Chuklin and Serdyukov [2012b] attempt to predict among three classes, *good*, *bad*, and *maybe*, Diriye et al. [2012] aim to predict four categories, namely *SAT*, *DSAT*, *Unintentional*, and *Other*. *Unintentional* means that the user closed the tab by mistake or was interrupted by somebody else, or they lost network connectivity. The *Other* category includes reasons such as the user finding the answer on his or her own, without a search engine. Finally, Song and colleagues [2014] simplify the categories by only considering *SAT* and *DSAT* (good and bad abandonment) instances. The dataset used by Chuklin and Serdyukov [2012b] contains 1,245 manually labeled abandoned queries, the set used by Diriye et al. [2012] contains 1,799 instances, and the data used by Song and colleagues [2014] contains 7,419 instances.

The articles solve slightly different tasks. Chuklin and Serdyukov [2012b] are interested in predicting *potential good abandonment* using features derived only from the queries. Diriye et al. [2012] make use of more features derived from the session and abandoned SERP itself to predict the reasons for abandonment. Song and colleagues [2014] also make use of an extensive set of features but only predict if the abandonment

Table VI. Classes of Features Used for Predicting Good and Bad Abandonment

Feature Class and Top Features in class	[1]	[2]	[3]	[4]
Query Features		✓	✓	✓
Query Length			✓	✓
Position of query in session				✓
Query is URL				✓
Similarity to previous query or queries			✓	✓
IDF or Query Popularity		✓	✓	
Query Language		✓		
Query Intent		✓		
Search Result Features			✓	✓
Number of features (answers, ads, spelling, algo, etc.)			✓	✓
Similarity of query to search results			✓	
Ranking score of search results			✓	
Session Features			✓	✓
Session entry point				✓
Browser type				✓
Session length in queries			✓	✓
Session length in dwell time				✓
Number and position of abandoned queries			✓	✓
Engagement/Behavior Features		✓	✓	✓
Search result clicks		✓	✓	✓
Search result click positions		✓	✓	✓
Query dwell time			✓	✓
Mouse cursor features	✓		✓	
Mouse cursor features	✓		✓	

[1] [Huang et al. 2011] **[2]** [Chuklin and Serdyukov 2012b] **[3]** [Diriye et al. 2012] **[4]** [Song et al. 2014]

is good or bad. Table VI summarizes the features used in these three articles. The table also references work by Huang et al. [2011], which does not discuss a prediction model but shows that there are behavioral differences in terms of mouse cursor movements between queries with good and bad abandonment.

Chuklin and Serdyukov [2012b] report results generated using a Support Vector Machine classifier. The *good*, *bad*, and *maybe* classes reach F-Measures 0.55, 0.71, and 0.38, respectively. Combining the *good* and *maybe* categories and optimizing for precision can yield a classifier with precision 1.0 and recall 0.15.

Diriye et al. [2012] experimented with a variety of classification algorithms and have settled on multiple additive regression trees. They used both L_1 and L_2 loss models to avoid overfitting. The use of L_1 selects effective features, and L_2 penalizes extreme feature weights. Instead of reporting F-Measure, the authors emphasize precision by computing $F_{0.5}$. For *SAT*, *DSAT*, *Unintentional*, and *Other*, the performance is 0.63, 0.71, 0.04, and 0.45, respectively. Removing the mouse cursor features (which are often not available) does not degrade the results considerably. Also, ignoring the training instances that fall under the smaller *Unintentional* and *Other* categories yields a binary classifier with an $F_{0.5}$ score of 0.78. This result is significantly better than the author's baseline of classifying instances as *SAT* when an answer is present that has a score of 0.61.

Finally, Song and colleagues [2014] propose a more advanced method of modeling and predicting types of abandonment by using a structured learning framework, which takes into account the dependencies among the abandonment labels within each session. The authors extend a Linear Structural SVM to a linear chain Hidden Markov Model by incorporating not only the label transition probability but also the features that can depend on any arbitrary pairs of labels. The article compares the proposed method to two baselines: a Boosted decision tree classifier and a structured

SVM framework with a reduced set of features. The proposed method outperforms both baselines significantly, reaching an accuracy of 87%, compared to 71% and 82%, respectively. The article also proposes two new ranking features based on the abandonment prediction model. Training a ranker by combining 400 existing features with the two new proposed features yield an improvement on the NDCG score of 2 percentage points.

5. QUERY DIFFICULTY AND PERFORMANCE

Query performance prediction aims to determine whether a query will have a high average precision given retrieval from a particular document collection [Hauff et al. 2008]. The ability to predict query performance and to find underperforming queries is crucial for improving search engines. Since users have varied information needs, the ranking models used by search engines need to be general enough to cover most use cases. This is evident when considering that roughly 50% of all queries are issued a single time [Hauff et al. 2008]. However, optimizing for a broad range of queries invariably leads to corner cases where users are left dissatisfied for some categories of queries. Finding such queries would allow us to obtain better training data for general rankers and it would enable creating more specialized rankers for certain categories of underperforming queries.

The task of predicting query performance is often evaluated by correlating difficulty scores with retrieval precision, which is defined as the intersection of relevance documents and retrieved documents, over the number of retrieved documents.

Query performance can be predicted using either pre-retrieval [He and Ounis 2004] or post-retrieval predictors [Cronen-Townsend et al. 2002; Hauff et al. 2008; Guo et al. 2010; Dan et al. 2012; Hassan et al. 2013]. Articles by Carmel et al. [2006] and by Kim and colleagues [2013] propose variations of their methods for both types of predictors.

- **Pre-retrieval predictors** are features that can be computed offline before retrieving the ranked list of results. The main advantage of these features is that they can be computed relatively quickly using statistics from query logs or the collection of documents. The disadvantage is that the predictions might not be as accurate.
- **Post-retrieval predictors** are more complex and computationally demanding and can only be computed once the search engine has returned the ranked list of results. Since in this case more features are available, predicting query performance might yield more accurate results.

We classify articles based on the types of features they employ for finding underperforming queries. The first category, covered in Section 5.1, makes heavy use of language models to determine the distance between queries and documents. The second category, discussed in Section 5.2, uses more diverse features based on user behavior. Most research in this second category aims to group queries and assign them human readable labels of the reasons they are underperforming. Please refer to the citation graph in Figure 5 that visualizes the relationship between the articles and to Table VII, which contains a summary of features.

5.1. Predicting Query Performance using Language Models

Clarity score, which is introduced by Cronen-Townsend et al. [2002] and extended by Hauff and colleagues [2008], aims to determine the degree of ambiguity in user queries with respect to a collection of documents. The intuition behind this metric is that a query that returns highly coherent results about a single topic will have better performance than a query that returns a mix of articles about different topics with low coherence. A query language model is computed on all the documents in the collection that match at least one query term. The clarity score score is then produced by the Kullback-Leibler divergence between the query language model and a language

Table VII. Classes of Features Used for Determining Query Difficulty

Feature Class and Top Features in class	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
Distance Features	✓	✓	✓	✓				
Between relevant (retrieved) documents and document collection	✓	✓	✓	✓				
Between queries and document collection			✓					
Among relevant documents			✓					
Between queries and relevant documents			✓					
SERP Features					✓	✓		✓
Vertical (web, images, news, etc)						✓		✓
Number and position of results					✓	✓		
Number and position of other page blocks					✓	✓		✓
Has definitive - known best result, mainly navigational queries					✓			
Algo relevance					✓			
Algo target page topic								
Engagement/Behavior Features					✓	✓	✓	
Number of submitted queries							✓	
Clicks and click positions					✓	✓	✓	
Breakdown per type of click (SAT clicks, dwell, bounces, etc.)					✓	✓		
SERP dwell time					✓	✓		
Quick back - how often users return quickly to SERP after click						✓		
PSkip - how often users skip first results and click on lower algos						✓		
Abandonment rate					✓			
Pagination rate					✓			
Engine switches					✓			
Document Collection Features		✓		✓				
Term level statistics like *idf*		✓		✓				
Session and Task Features								
Overall Topic (intent) of session								
Length of session								
Average engagement across session								
Query Features		✓			✓	✓	✓	✓
Length: # words, terms		✓			✓	✓		
Historical usage or frequency					✓	✓		
Named Entities Extraction							✓	
Wikipedia categories corresponding to queries							✓	
Query Intent classifiers (including ODP categories)						✓	✓	✓
Has spell correction or alteration					✓			✓
Stopwords, question marks, other symbols								
Query similarity, term differences								
User Features						✓		✓
Geographical location (including market, language)						✓		✓
Other								✓
Engine name								✓
Bookmarks								
Mouse or scroll								

[1] [Cronen-Townsend et al. 2002] **[2]** [He and Ounis 2004] **[3]** [Carmel et al. 2006] **[4]** [Hauff et al. 2008] **[5]** [Guo et al. 2010] **[6]** [Dan et al. 2012] **[7]** [Kim et al. 2013] **[8]** [Hassan et al. 2013]

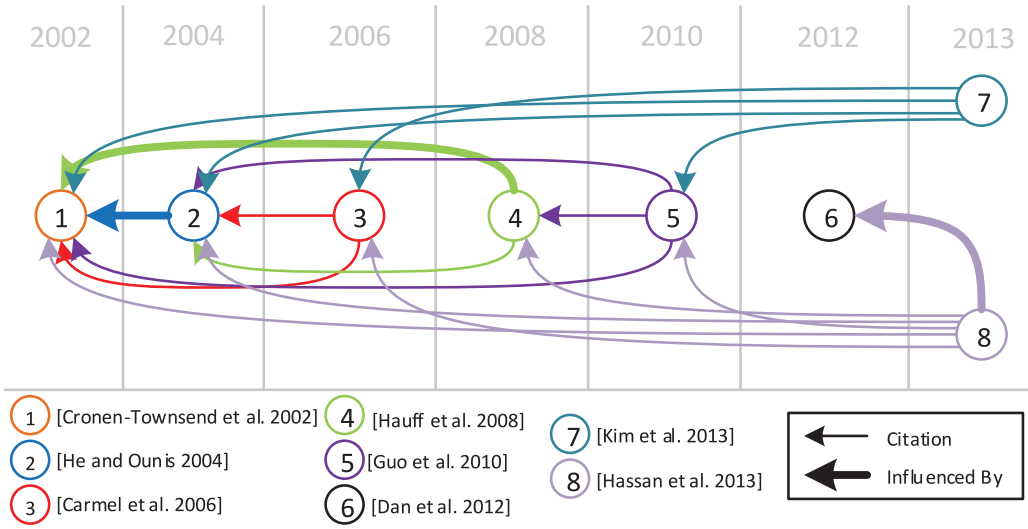

Fig. 5. Citation graph among Query Difficulty and Performance articles.

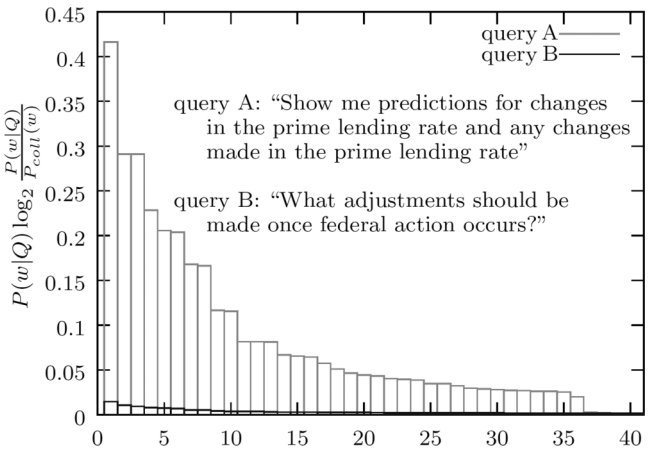

Fig. 6. Contribution of top 40 words to clarity score for two queries [Cronen-Townsend et al. 2002].

model computed on the document collection as a whole. Results by Cronen-Townsend et al. [2002] show that there is a strong correlation between clarity score and average precision in several TREC test collections.

Figure 6 shows the scores of the top 40 words that contribute to the clarity score of two queries. Examining the graph we can see that query A will have a much higher clarity score than query B. The final score can be obtained by summing the values over all words for each query individually. In this example, the word "bank" has the most contribution for query A. The high score is due to the term having much higher probability in the query model than in the collection model.

Hauff et al. [2008] propose two improvements to *Clarity score*. First, they propose a method to determine the sample size for the documents used to compute the query language model. While Cronen-Townsend and colleagues [2002] use a threshold of 500 documents, here Hauff and colleagues demonstrate that varying the threshold can

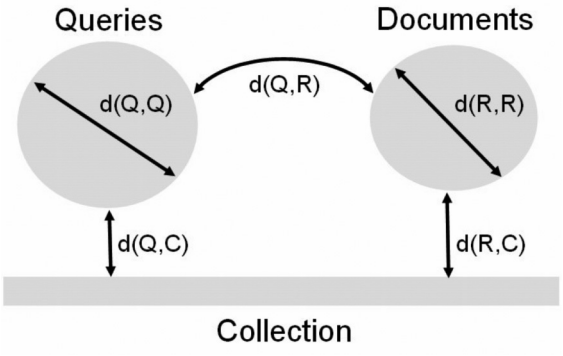

Fig. 7. A general model for difficulty. d(...) represents a distance function, Q, is the query, R is the set of relevant documents, and C is the collection of all documents indexed by the search engine [Carmel et al. 2006].

have a big impact on the prediction performance for certain TREC topics. Instead, they propose using only documents that contain all query terms. This effectively sets the threshold automatically for each query. The obvious disadvantage is that for some queries the model will need to be computed on a large number of documents.

A second improvement for Clarity score described by Hauff et al. [2008] uses expectation maximization to learn a separate weight for each of the terms in the set of retrieved documents. This method reduces noise from terms that are frequent in the collection. Experimental results show that the improved clarity score compares favorably to the original method in terms of correlation to mean average precision. For instance, the improved version has a Kendall tau correlation of 0.44 on TREC 401-450 compared to the original version, which has a correlation of 0.3.

Instead of relying on post-retrieval features, such as the ranked list of documents sorted by relevance scores, He and Ounis [2004] propose using pre-retrieval predictors. The four features they propose are query length, the inverse document frequency of the query terms, a simplified version of clarity score, and query scope, which is given by the ratio of the number of documents containing at least one query term and the total number of documents in the collection. They find that the best-performing predictor in terms of correlation to average precision is the simplified version of clarity. This version makes use of only the document collection model and simple features derived from the query itself, such as query length and query word occurrence count.

Carmel and colleagues [2006] further formalize the usage of language models to determine query difficulty. They propose a general model for topic or query difficulty, which depends on the relationship among three components: the query, the relevant documents, and the entire collection, as can be seen in Figure 7. They use Jensen-Shannon divergence as a measure of distance. The distance between the retrieved documents and the collection, distance between the query and the collection, and the distance between the retrieved documents have the highest Pearson correlation to average precision, in this order, at 0.32, 0.167, and 0.150, respectively. Interestingly, the distance between the query and the retrieved documents had almost no correlation. Combining the distance features using an SVM classifier yields a Pearson correlation of 0.362 on a dataset with 100 training topics.

5.2. Predicting Underperforming Queries Using User Behavior

So far we have discussed work that focuses on features based on language models to predict query difficulty. More recent work in this area has made use of more varied

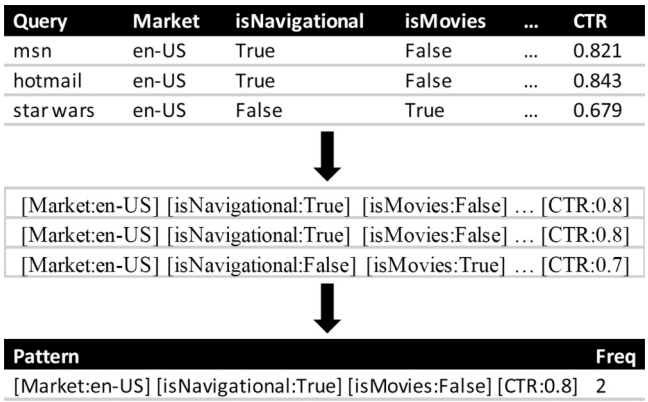

Fig. 8. Example of steps taken to discretize features and mine for frequent patterns [Dan et al. 2012].

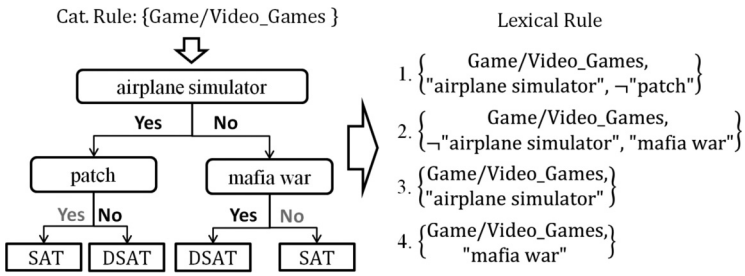

Fig. 9. Example of decision tree-based lexical rule generation [Kim et al. 2013].

predictors based on queries, results, and user behavior. One such example is work by Guo et al. [2010], which uses features derived from queries, algo results, and user engagement (behavior) to train a regression model using Multiple Additive Regression Trees. They find that the top two features by predictive power are average click position on results and average number of clicks. Their model achieves a Pearson's correlation of 0.699 with DCG on 2,834 queries.

Predicting difficulty at a query level might be insufficient, as individual underperforming queries cannot be used directly to improve ranking models [Hassan et al. 2013]. One line of work solves this obstacle by finding groups of queries with similar characteristics, making it possible to train specialized rankers that improve the group as a whole. Dan et al. [2012] introduce the technique of using association rule mining to find underperforming queries. They start from features aggregated at a query level, such as query intent, query length, and clickthrough rate. They then discretize these features and run a distributed and parallel version of the FP-Growth algorithm. Finally, they filter the patterns down to cases where the consequent (right-hand side) represents dissatisfaction, such as low clickthrough rate. Figure 8 illustrate these steps.

Kim and colleagues [2013] adopt a similar method by running the same algorithm on granular query intent categories extracted from ODP and Wikipedia, as well as behavior features. In addition, they generate lexical rules by enriching the extracted rules with keywords. They achieve this by adopting a two-tiered approach where the keywords are mined among the queries that match the initial rules. In order to combine the two types of rules, they then use a decision tree classifier, which has the advantage of also supporting negation. Figure 9 demonstrates this process. The authors use the semi-supervised method described by Hassan et al. [2012] to assign SAT and DSAT labels

to 1.5 million search sessions collected using a browser toolbar. The query difficulty algorithm compares favorably to several strong baselines [Cronen-Townsend et al. 2002; Guo et al. 2010].

Finally, Hassan and colleagues [2013] bring the difficulty prediction task to its logical conclusion by using the extracted rules to improve ranking models. Before extracting the rules they label a dataset with *SAT* and *DSAT* by training an engine switch classifier with features inspired by Guo et al. [2011]. After extracting rules from 100,000 *DSAT* instances, they train a ranker on one of the groups that covers 2% of these instances. They compare the ranker to a general one trained on all the queries and to a specialized one trained on queries with low clickthrough rate. The ranker trained on the group shows improvements in NDCG@1 of 1.52% and 0.51%, respectively.

6. TASK DIFFICULTY

Search task difficulty represents a users' assessment of the amount of effort required to complete a search task [Arguello 2014]. Since this assessment is subjective, the perceived difficulty of the search task can be influenced by multiple factors, including the familiarity of users with search engines [White and Morris 2007], users' domain knowledge [Liu et al. 2012a], and the intrinsic complexity of the task [Aula et al. 2010].

An example of a task that was deemed difficult by study participants was to find an establishment in San Francisco that is known as the oldest seafood restaurant in town [Kim 2006]. A few reasons given by participants for marking a task as difficult include the inability to determine which terms to use in queries, trouble accessing websites, and problems navigating within websites.

We divide research on task difficulty in two major groups. First, we discuss the effect that task difficulty has on user behavior. Second, we describe research on how search experience and domain knowledge change users' perspective of task difficulty. Third, we present work in predicting task difficulty.

6.1. Task Difficulty and User Behavior

Several articles have found a strong correlation between task difficulty and user behavior. Multiple studies have found that task completion time is longer for more difficult tasks [Kim 2006; Aula et al. 2010; Liu et al. 2010a, 2010b, 2012a, 2012b; Arguello 2014] and the number of issued queries [Kim 2006; Liu et al. 2010a, 2010b, 2012a, 2012b, 2014; Arguello 2014] and that viewed documents [Kim 2006; Liu et al. 2010a, 2010b; Arguello 2014] is higher for difficult tasks, the overall SERP dwell time is longer for difficult tasks [Liu et al. 2010b, 2012a, 2012b], and that users bookmarked more documents for difficult tasks when compared to easy tasks [Kim 2006; Liu et al. 2012b; Arguello 2014]. A less common finding shows that difficult tasks had longer first document dwell time [Liu et al. 2010a, 2012b].

However, there are studies that contradict some of these popular findings. Liu et al. [2012a] find that overall users visited significantly fewer content pages in difficult task sessions than in easy task sessions. They also find that task difficulty did not have any effect on the number of SERPs and issued queries visited in each session, which again contradicts other research. Furthermore, while Liu and colleagues [2010a] find that difficult tasks had longer overall document dwell time, other articles disagree. In fact, another work by Liu et al. [2012b] does not find any significant differences in document dwell time. Both Liu et al. [2014] and Arguello [2014] find that document dwell time is actually shorter in difficult tasks. On possible explanation for these discrepancies is that many of these studies use small datasets. For instance, an article by Liu et al. [2010a] uses 48 students and 8 tasks, an article by Liu and colleagues [2012a] has 37 participants and 148 sessions, and another article by Liu et al. [2012b] covers 38 participants.

6.2. Search Experience and Domain Knowledge

The perception of difficulty depends on the search experience and the domain knowledge of the user. Next we will briefly discuss research that views task success and difficulty from the perspective of the user. We start by discussing advanced search users and advanced search operators. Advanced search engine users are users who use search operators such as "+", "−", and "site:" in their queries. White and Morris [2007] set out to explore the use of such query operators in more detail to determine if they correlate with search success. They analyze 13 weeks of browser toolbar logs and determine that 20.1% of 188,405 users have used search operators at least once. However, overall, only 1.12% of all issued queries contained operators. The authors conjecture that since advanced search operators are difficult to find, userswho make use of them are a distinct class of searchers with common behaviors. Their hypothesis is confirmed by data analysis, which shows significant differences between the behaviors of novice and advanced users. More specifically, advanced users:

- Are consistently more successful as measured by average relevance scores.
- Query less frequently in a session but submit more queries per day.
- Compose longer queries.
- Click further down the result list but are less likely to click on a result.
- Repeat queries more often but revisit pages in the trail less often.
- Spend less time traversing each search trail and less time viewing each document.
- Branch less often and follow search trails with fewer steps.
- Use a more directed searching style (direct path) than non-advanced users.

Aula and colleagues [2010] describe behavior differences between users undergoing successful and unsuccessful tasks, respectively. They run two experiments, a lab study with 23 participants and an online study with 179 participants. Each volunteer is assigned 22.3 tasks on average from a pool of 100 search goals with varying levels of difficulty. Interestingly, the authors note that advanced operators' usage increases when users have more difficulty with their search task. This finding does not contradict the conclusion of White and Morris [2007], who state that successful users use more advanced operators, because Aula et al. [2010] purposely provide users with more difficult tasks to see how they adapt to them. Below we summarize further findings.

- Participants often resorted to asking direct questions in natural language after several unsuccessful attempts with keywords.
- In successful tasks users started with a more general query and made the query more specific (longer) with each refinement.
- Searchers spend more time (11s on average) on the results page for more difficult tasks, compared to 8s overall. The time becomes significantly longer for tasks where the users give up entirely.
- The number of queries decreased steadily as the task success rate increased.
- Harder tasks tended to have longer queries.
- The use of advanced query operators was significantly higher in unsuccessful tasks.
- In easier tasks, users formulate their longest query towards the end of the session.
- In more difficult tasks, the longest query tends to occur in the middle of the task, suggesting that users switch to other strategies that have shorter queries.
- When they had difficulties, users spent a lot of their time on the results page
- When faced with difficult tasks, searches spent a larger portion of their total task time on the SERP.

Finally, we summarize findings by Liu et al. [2012a], who investigate how domain knowledge changes the perception of task difficulty. Their study finds that when

Table VIII. Task Difficulty Features

Feature Class and Top Features in class	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Task									
Time spent on task / Task completion time	✓	✓	✓	✓		✓	✓	✓	
Task or session topic							✓		
Queries									
# queries or SERPs in task	✓	✓	✓	✓	✓	✓	✓	✓	✓
# queries with advanced operators		✓							
# queries with question		✓							
Query length in characters or tokens		✓				✓	✓		✓
Types of words in queries						✓			✓
Query similarity / term differences in task or session							✓		
Historical query usage of frequency							✓		✓
SERP									
# search result pages viewed	✓		✓	✓	✓	✓		✓	
# pages saved (bookmarked)	✓			✓		✓			✓
Dwell time on SERP or result pages		✓	✓	✓	✓	✓	✓	✓	
Rank of clicked or saved search results						✓			
Algo target page topic							✓		
Clicks and click positions							✓	✓	
Breakdown per type of click (SAT click, bounces, etc.)							✓		
Abandonment rate							✓		✓
Mouse or scroll									✓

[1] [Kim 2006] **[2]** [Aula et al. 2010] **[3]** [Liu et al. 2010a] **[4]** [Liu et al. 2010b] **[5]** [Liu et al. 2012a] **[6]** [Liu et al. 2012b] **[7]** [Hassan et al. 2014] **[8]** [Liu et al. 2014] **[9]** [Arguello 2014]

searching for difficult tasks, users who have more domain knowledge have lower dwell time on content pages and higher dwell time on SERPs when compared to users with lower levels of domain knowledge. The authors conjecture that users with lower domain knowledge were trying to find the exact terms from the search tasks as they appeared in the text of the clicked documents, so their dwell time on clicked documents was longer.

6.3. Task Difficulty Prediction

Predicting task difficulty can help search engines adapt their ranking algorithm, the interface, and even intervene during the session to help users. Earlier work by Liu et al. [2010a] proposed predicting task difficulty using simple thresholds such as the number of viewed documents but did not present any evaluation. A more developed method based on this work proposes using logistic regression models with a limited number of features, such as unique SERPs, number of queries, and dwell time [Liu et al. 2010b]. On a dataset containing 288 search sessions the overall prediction accuracy was 77.1%. Later, the work by Liu et al. is further improved by building four models with different features depending on the level of aggregation (after the first query in the session, for the whole session, etc.) [Liu et al. 2012b]. The models yield an accuracy between 75% and 79%. However, the trivial baseline of predicting that all tasks are difficult also achieves a very high accuracy of 71%. Table VIII summarizes the features used in articles that study predicting task difficulty and user behavior in regards to task difficulty. Figure 10 shows a citation graph on the subset of articles on task difficulty prediction.

More recent articles further expand on the task by predicting task difficulty at different stages of the session. Arguello [2014] aimed to predict task difficulty after the first query in the session and after the whole session. He asked 30 crowdsourcing participants to classify 20 tasks each, for a total of 600 search sessions. Each participant

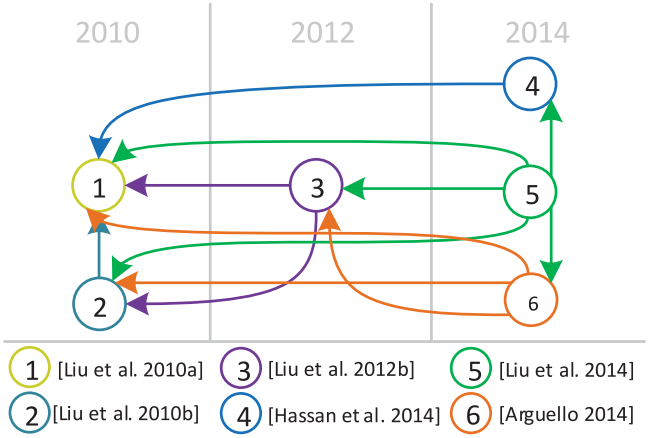

Fig. 10. Citation graph among Task Difficulty prediction articles.

answered pre-task and post-task questionnaires. Some of the more unique features used for the prediction model include mouse and scroll features. Using a logistic regression model, he achieved an average precision of 56.3% when only considering the first query and 61.8% for the whole session. He finds that bookmark features were the most predictive in the first-query model and that time dwell features were the most predictive in the whole session model.

While Arguello built prediction models only after the first and last queries in a session, Liu and colleagues [2014] also built a model for predicting in the middle of the session. They asked 32 journalism students to complete four types of search tasks and self-rate their difficulty, yielding 99 search sessions. The authors then used the captured user behavior to train decision trees for the three search points in the session using recursive partitioning. Evaluation resulted in F-Measures of 62.6%, 55.8%, and 55.9% for the first query, middle point, and end of session models, respectively. Surprisingly, the first query model yielded better results than the other two, which differs from the findings by Arguello [2014]. The first query model makes use of only dwell time and first query interval features.

Finally, an interesting task related to task difficulty prediction is presented by Hassan et al. [2014], who attempt to distinguish between search sessions where users are struggling and sessions where they are exploring. The compiled data set covers 3,000 topically coherent sub-sessions extracted from the logs of the Bing search engine, which were manually labeled to be *struggling* or *exploring*. One distinguishing group of features used for prediction is Topic features, which are based on the topic of clicked URLs. The topic is derived from Open Directory Project categories. The rest of the major features are summarized in Table VIII. Tenfold cross validation on a classification model built using Multiple Additive Regression Trees yields an accuracy of 81.67% and F-Measures of 83.68% and 79.17% from the *Exploring* and *Struggling* categories, respectively. The baselines are based on subsets of features and data and predictably they have worse results.

7. PREDICTING SATISFACTION

Search satisfaction is defined as the fulfillment of a user's information need [Feild et al. 2010]. Although satisfaction is a personal emotion and thus subjective, it is not practical to explicitly request feedback from users [Hassan and White 2013]. Therefore, modeling user satisfaction using implicit behavior instead is vital for search engines.

Most of the literature in this area is concerned with finding implicit behavior signals that act as a proxy of user satisfaction.

7.1. Training and Evaluation Data

Data used to evaluate and train satisfaction models need to be provided by humans. Hassan and White [2013] point out that obtaining this type of training data is a major challenge, since authors cannot request labels from the original users. Thus, often datasets are constructed using human labelers that are asked to re-enact queries or even complete sessions extracted from query logs and to provide their own opinion of satisfaction [Huffman and Hochster 2007; Hassan et al. 2010, 2013; Kim et al. 2014]. However, the approach of re-enacting user sessions might not provide a true representation of user satisfaction as judges are only guessing the original user's intent [Hassan et al. 2011].

Whenever possible, satisfaction ratings and behavior information are collected from small cohorts of users internal to the company or university performing the research [Fox et al. 2005; Hassan et al. 2011; Hassan 2012; Jiang et al. 2015]. In these cases, users are asked to install browser plugins that track their search activity over a longer period of time. Participants are asked to use search engines as they normally would but also rate their satisfaction with the results at the end of each search task. These types of data pose their own challenges, as they can be biased. In several cases, the demographics were skewed by using technically savvy participants.

Ageev et al. [2011] take a more active approach in training data collection by deciding what participants should search for. First, instead of using queries extracted from search logs, they select tasks from community question websites such as Yahoo! Answers. Second, they employ a unique way of motivating participants by posing the search tasks as games. Third, they offer extra monetary compensation to persuade participants to persist through difficult search tasks. The result is an increased rate of task completion and a reduction in low-quality data.

The most scalable training data collection method is presented by Hassan and White [2013], who use engine switches to find instances of dissatisfaction. As previous research has shown that only 60% of switches are due to dissatisfaction, they make use of a classifier previously described by Guo et al. [2011], which has an F-score of 78.99 for the *DSAT* class. For more details on search engine switching, please refer to Section 3 of this survey. A similarly scalable data collection method is presented by Kim and colleagues [2014], where authors mine query logs and assume that a click followed by query reformulation is a proxy for dissatisfaction, while other click instances signify satisfaction.

7.2. Prediction Methods

Early efforts in predicting satisfaction aim to determine how well relevance metrics correlate to user ratings at the result and session levels. Al-Maskari and colleagues [2007] ask labelers to judge results from 104 Google queries in total on a range of common IR measures such as CG, DCG, and NDCG. The authors find that CG and precision correlate better than NDCG with user satisfaction. Huffman and Hochster [2007] required participants to re-enact 200 sessions each starting with real user queries extracted from Google logs. The data shows a surprisingly strong relationship between the relevance of the first query in the session and session-level satisfaction.

Fox and colleagues [2005] published the most influential early work in both characterizing and predicting satisfaction. The authors constructed Bayesian models using the feature classes described in Table IX. The model yielded a session-level accuracy of 74% for the *SAT* class and 62% for the *DSAT* class compared to a baseline of 56% for *SAT*. They then built decision trees separately for each node of the Bayesian networks.

Table IX. Classes of Features Used for Tasks Related to Predicting Satisfaction

Feature Class and Top Features in class	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
SERP Features	✓	✓	✓		✓				✓	✓	✓	✓
Algo position	✓											
Algo URL: domain features					✓				✓		✓	
How much does text in algos match query									✓			
Number other page blocks									✓			
Time spent on SERP page (dwell)	✓									✓	✓	✓
Page scrolled features	✓										✓	
Page stats: page size, number of images, # of scripts	✓									✓	✓	
Other page actions: added to favorites, printed	✓										✓	
Exit type: new query, closed tab, URL entry, timeout	✓										✓	
Algo relevance: CG, DCG, NDCG, etc.		✓	✓									
Clicked page features (topic, readability, HTML tags)										✓		
Engagement / Behavior Features	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Clicks, SAT/DSAT clicks, time to click, click positions	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
Abandoned searches (no clicks)												✓
Clicked URL domain features					✓						✓	
Engine switches											✓	
Query Features					✓				✓	✓	✓	
Length: # words, characters, stop words					✓				✓		✓	✓
Has advanced operators					✓						✓	
Historical stats: frequency, clicks, etc.									✓			
Has spell correction									✓			
Query topic or intent										✓		
Session/Trail/Task Features	✓			✓	✓	✓	✓	✓	✓		✓	✓
Time: between queries, dwell, and session/task total									✓			✓
Session/task length in queries	✓				✓				✓		✓	
# queries with clicks on certain blocks on the page	✓								✓		✓	
Reformulations or query similarity								✓	✓		✓	✓
Previous actions in session or trail	✓			✓	✓	✓	✓	✓			✓	
Time between actions and time per action	✓			✓	✓	✓	✓	✓			✓	
Ends in SAT/DSAT action											✓	
User Features	✓				✓						✓	
Length: # sessions, queries / session, time / session URLs / session, words / query					✓						✓	
Session exit type: timeout, URL entry, closed browser, etc.	✓										✓	
Engine Features					✓						✓	
Engine name					✓						✓	

[1] [Fox et al. 2005] **[2]** [Huffman and Hochster 2007] **[3]** [Al-Maskari et al. 2007] **[4]** [Hassan et al. 2010] **[5]** [Ageev et al. 2011] **[6]** [Hassan et al. 2011] **[7]** [Hassan 2012] **[8]** [Hassan and White 2013] **[9]** [Hassan et al. 2013] **[10]** [Kim et al. 2014] **[11]** [Wang et al. 2014] **[12]** [Jiang et al. 2015]

Action		Page (basic)		Page (advanced)	
q	Query	R	SERP	A	SERP (short)
p	Pagination	P	Non-SERP	D	SERP (medium)
s	Click result			E	SERP (long)
c	Click other			F	Non-SERP (short)
b	Back one page			G	Non-SERP (medium)
j	Back many pages			H	Non-SERP (long)
n	Navigate to page				

Fig. 11. Example of alphabet used to encode user actions [White and Dumais 2009].

An often-cited finding derived from the rules of the decision trees is that clicks followed by a dwell time of more than 30s are more likely to lead to a *SAT* rating than clicks where users return more quickly to the SERP.

Another important concept introduced by Fox et al. is encoding user behavior as alphabet characters and mining for commonly occurring sequences. Figure 11 shows an example of such an alphabet. In this example, the sequence *qRnP* would mean that the user issued a query, clicked on the SERP results, and then navigated to a non-SERP page using the browser address bar. This methodology has also been adapted for solving other kinds of tasks, such as engine switch prediction [Heath and White 2008], as described in Section 3.4.

Hassan and collaborators build on the foundation laid out by Fox and colleagues [2005] in a series of articles as shown in the citation graph from Figure 12. Hassan et al. [2010] use a similar behavior alphabet to build two Markov models, one for *SAT* and the other for *DSAT*. Given a new search goal, they estimate the likelihood of the pattern being generated from the two models. Finally, they compare the likelihood of the goal under the two models. They further extend these models by taking into account the time between each user action. Later, the authors use an approach based on a generating model and the expectation maximization algorithm in a semi-supervised setting [Hassan 2012]. They show that, using a combination of labeled and unlabeled data, they can outperform previous methods. Methods and results for all articles discussed in this section are summarized in Table X.

Another work that directly improves the model created by Hassan et al. [2010] is presented by Ageev and colleagues [2011]. The Markov Model approach is augmented with additional behavior features using Conditional Random Fields. The features are derived from other tasks such as detecting user frustration [Feild et al. 2010] and investigating the behavior of advanced search engine users [White and Morris 2007]. In a later article, Hassan et al. [2012] compare this CRF method to the one based on EM and find them to have very similar performance.

Personalized satisfaction prediction models can outperform a baseline using global features only. Hassan and White [2013] demonstrate differences between users by plotting *SAT* vs *DSAT* labels for abandonment, query refinement, and dwell time data at an individual user level. They build a user dissatisfaction classifier using query, session, and SERP features and train it for individual users and for cohorts of users based on expertise, interests, and engine preference. Results for all cohorts outperform the global baseline and are overall competitive when compared to previous methods in the literature.

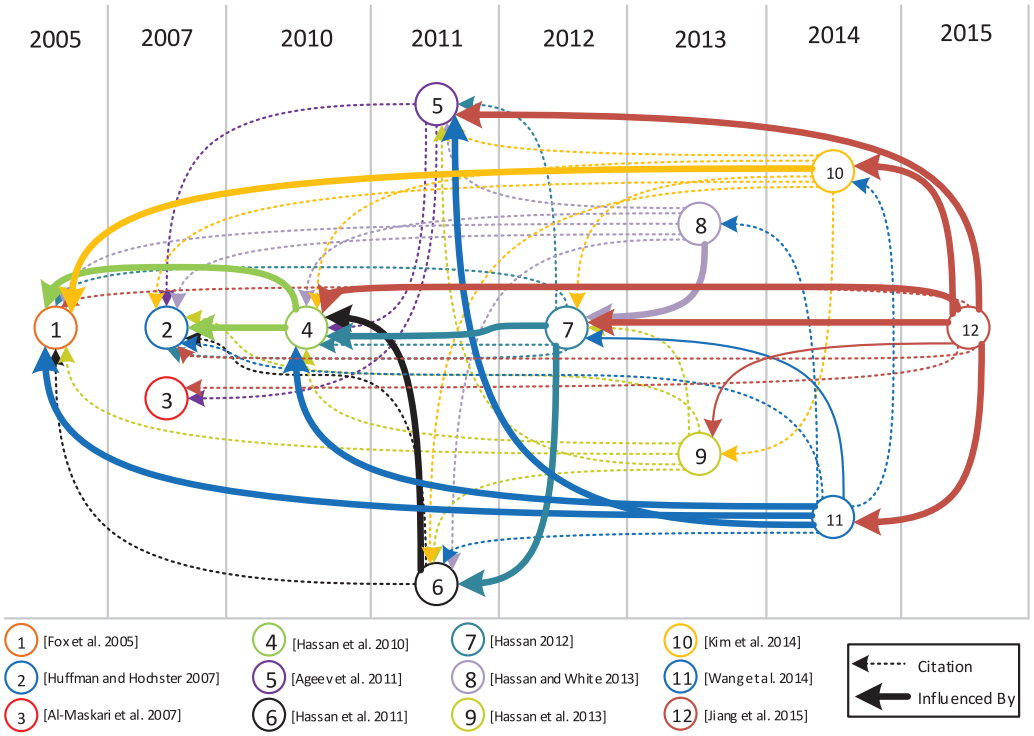

Fig. 12. Citation graph among articles discussing predicting satisfaction.

Query reformulation can also be used as a strong signal of user dissatisfaction. Given pairs of queries, Hassan et al. [2013] experiment with determining user satisfaction with the first query by determining its overlap with the second query. Their best-performing model also makes use of click dwell information. Authors claim that this type of determination can be made for a majority of queries, since 67% of queries in a large commercial search engine dataset had a next query.

Kim and colleagues [2014] devote an entire article to further exploring the relationship between satisfaction and dwell times. They show that the dwell time threshold used for determining if the click was satisfied or dissatisfied varies depending on the reading level (complexity) of the clicked target page. Previous research has set this threshold to be a fixed of 30s [Fox et al. 2005]. Similarly, the article shows that the threshold differs depending on the topic of the page. For example, technical websites have higher dwell time thresholds than shopping websites. The authors then propose a satisfaction prediction model at the click level that takes into account factors such as the topic of the target page, its length, its readability level, and the topic of the query. They train a classifier using Multiple Additive Regression and show that it outperforms baselines such as a classifier with a subset of the features, query clarity [Cronen-Townsend et al. 2002], inverse collection term frequency [He and Ounis 2006], and query term length.

In Wang et al. [2014], the authors predict satisfaction at the task level by considering all the actions that make up the task, such as issuing queries, hitting the back button to return to the SERP, and clicking on related searches or spelling. They propose the Action-aware Task Satisfaction model (AcTS), which treats the individual actions as latent variables. The model is able to predict satisfaction at both the action and

Table X. Methods and Results for Predicting Search Success

Paper and data	Method	Results
[Fox et al. 2005] 2,560 sessions and 3,659 page visits from 146 participants over a span of 6 weeks.	Built two Bayesian network models for predicting satisfaction at the query and session levels, respectively. Decision trees can be used to summarize any node of the models.	**Query-Level accuracy**: 70% for SAT, 47% for PSAT (potential satisfaction), and 52% for DSAT **Session-Level accuracy**: 74% for SAT, 57% for PSAT (potential satisfaction), and 62% for DSAT
[Huffman and Hochster 2007] 200 queries from Google, labelers determined ranking and satisfaction.	Computed Pearson correlation between DCG relevance and satisfaction ratings from labelers.	Pearson correlation between session-level satisfaction and relevance of first query in the session is 0.727.
[Al-Maskari et al. 2007] 26 users search four queries each from a pool of 104.	Computed Pearson correlation among CG, DCG, NDCG, prediction, accuracy, coverage, and satisfaction.	CG and precision correlate better than NDCG with user satisfaction.
[Hassan et al. 2010] Judges labeled 2,712 goals over 5,000 queries from 1,000 user sessions.	Sequences of actions in search goals encoded using Markov models. Models also use action transition times. Gradient Boosted Decision Trees as classifier.	Markov Model with Time where clicks on top or bottom of algo list is modeled as different actions: Precision, Recall, and Accuracy were 83.6, 94.4, and 88.7, respectively.
[Ageev et al. 2011] Pool of 40 search tasks, 159 Mechanical Turk participants, 1,487 search sessions.	Conditional Random Fields used to extend Markov Model proposed by Hassan et al. [2010] by adding additional search behavior features.	The CRF model achieves a 4% to 26% increase in accuracy over the original model proposed by Hassan and colleagues [2010] for varying definitions of success.
[Hassan et al. 2011] Browser toolbar, 115 employees and interns, 6 weeks, 12,000 search goals, 30,000 page visits.	Same models as proposed by Hassan et al. [2010], applied on more data. Sequences of actions in search goals encoded using Markov models.	Search Task level accuracy: at recall 0.7 the precision of the *SAT* and *DSAT* classes is 0.79 and 0.93, respectively.
[Hassan 2012] Labeled data from browser toolbar: 10,000 unique searches, self-reported satisfaction.	Generative model for user behavior. Allows using both labeled and unlabeled data together. Uses EM on unlabeled data to find MAP estimates of parameters.	Performance of supervised model has accuracy 0.785. When percentage of labels is withheld the unsupervised models have better performance than the supervised ones.
[Hassan and White 2013] Five weeks of browser toolbar logs, train on 23 days for training, 1 week for testing.	Logistic regression classifiers. Query, Session, and SERP features. Build classifier on all data, use as feature in personal classifiers.	Accuracy 73% to 83.06% and F-Measure 76.55% to 85.41% depending on cohort used for personalization.
[Hassan et al. 2013] Data sampled randomly from 1 week of engine query logs. Labeled 6,000 SAT and 6,000 DSAT query pairs. Pairs have same user, same session.	Use clicks and query reformulation features like Levenshtein and number of keywords in common between query pairs to train classifier. Type of classifier not specified.	Authors compare 10 methods for predicting satisfaction. Best-performing method can reach an F-Measure of 79.43% and 87.27% for *SAT* and *DSAT*, respectively.
[Kim et al. 2014] Labeled data: 3,204 SAT / DSAT click instances balanced 50/50 sampled from Bing logs; Pseudo-labeled data: 104,00 click instances where reformulation is DSAT, balanced 50/50.	Click level satisfaction prediction. MART classifier with query and click target page features, including query and page topic, page readability, page length. Model also uses dynamic SAT and DSAT dwell time distributions.	Tenfold cross validation on the human labeled dataset. The best method that includes the dynamic dwell time features achieves 0.8 SAT F1, 0.81 DSAT F1, and 0.81 accuracy.

(Continued)

Table X. Continued

Paper and data	Method	Results
[Wang et al. 2014] Obtained datasets from Ageev et al. [2011] and from Hassan and colleagues [2011].	AcTS model treats all actions within the task as latent variables. Query and task level features. Also uses features developed by Fox et al. [2005] and by Ageev et al. [2011].	Task level satisfaction. 0.76 average F1 and 0.89 accuracy for toolbar data set. 0.7 average F1 and 0.88 accuracy for contest dataset.
[Jiang et al. 2015] 476 sessions sampled from Bing logs. Each query and overall session given satisfaction ratings by judges.	Poisson regression model predicts satisfaction on a continuous interval. Feature categories include search behavior and effort, action transition.	Tenfold cross validation. NRMSE of 0.16 and correlation with satisfaction of 0.43. Results statistically significant using a Welch t-test.

task levels, although only the task-level prediction is evaluated thoroughly. The model makes use of multiple types of features, including short-range structured features at the action level, such as edit distance between consecutive queries, the SERP position of the clicked URL, and engine switch, as well as long-range features, such as the first-order transitions between actions with respect to satisfaction labels. The article also implements the action-level and task-level features previously covered by Fox et al. [2005] and by Ageev and colleagues [2011]. The model also includes structured loss functions, which capture domain knowledge as weak supervision and contain rules such as *"all the actions should not be unsatisfying in a satisfying task."* The proposed method is compared against three strong baselines: the Markov Model Likelihood method described by Hassan et al. [2010], the logistic regression (LogiReg) model using features extracted from query logs and physical sensors proposed by Feild et al. [2010], and the session-CRF model proposed by Ageev and colleagues [2011]. The AcTS model surpasses all baselines at the task level, with an average F_1 score of 0.76 and accuracy of 0.89 on a dataset that was first used by Hassan et al. [2011].

Finally, Jiang et al. [2015] go beyond binary labels and attempt to predict satisfaction on a continuous interval. The methodology they adopt is similar to earlier research in satisfaction prediction, where relatively simple features are used to train a regression model. The features include behavior markers such as click and query dwell time, satisfied vs dissatisfied clicks, number of queries without clicks, rank of the clicks, query length, and query similarity. Poisson regression yielded the best results with NRMSE of 0.16 and correlation with satisfaction of 0.43.

8. CONCLUSIONS

Measuring and anticipating user satisfaction using implicit behavior signals is crucial for search engines in order to maintain and grow their market share. In this survey, we have presented the state of the art for the major areas of search satisfaction. We began by discussing the parallels between customer research and search engine users' satisfaction in Section 1. We have found that there are several concepts and findings in search satisfaction that mirror that of the older research branch of customer satisfaction.

After defining several important terms related to search behavior in Section 2, we defined and characterized search engine switching in Section 3. Here we have found that users have various reasons for transitioning to other search engines, including dissatisfaction with search results, bad interfaces, curiosity, advertising campaign, preferences for certain verticals, or even switching by accident. We then summarized the benefits users can get by switching search engines, and we presented statistics

on how often they actually switch. We also outlined several techniques for predicting switching.

In Section 4 we described the difference between good and bad abandonment. By aggregating several large scale studies on abandonment, we found that around 30% of all abandoned queries are actually caused by good abandonment; that is, while users did not click on the SERP at all, they were in fact satisfied as the search results snippets themselves contained the information they were looking for.

In order to improve their ranking models, search engines need to find queries with low user satisfaction. Section 5 covered articles on predicting query difficulty and performance. This task can be carried out with either pre-retrieval or post-retrieval predictors. While pre-retrieval predictors can be computed more easily since they do not require the ranked list of results, post-retrieval predictors can yield more accurate results. Two important techniques in this area are language modeling, which enables determining the ambiguity of queries with respect to a document collection, and association rule mining, which enables finding groups of underperforming queries.

Next, in Section 6, we described research on the relationship between perceived task difficulty and user behavior, on the effect of users' search experience and domain knowledge on difficulty, and on predicting task difficulty to improve search engines. When dealing with difficult tasks, common changes in user behavior include longer task completion times, an increase in the number of issued queries, an increase in the number of viewed document, and a longer overall SERP dwell time.

Finally, we discussed predicting satisfaction at the query, search task, and session level in Section 7. Early efforts in this area aimed to determine how well satisfaction correlated with relevance metrics. An important concept used by several articles is encoding different user behaviors as alphabet characters. Satisfaction ratings along with the transitions between these actions were used to build prediction models using Bayesian models, Markov models, and classification.

REFERENCES

Mikhail Ageev, Qi Guo, Dmitry Lagun, and Eugene Agichtein. 2011. Find it if you can. In *SIGIR'11*. ACM Press, New York, NY, 345.

Azzah Al-Maskari, Mark Sanderson, and Paul Clough. 2007. The relationship between IR effectiveness measures and user satisfaction. In *SIGIR'07*. ACM Press, New York, NY, 773.

Rolph E. Anderson and Srini S. Srinivasan. 2003. E-satisfaction and e-loyalty: A contingency framework. *Psychol. Market.* 20, 2 (Feb. 2003), 123–138.

Jaime Arguello. 2014. Predicting search task difficulty. In *Advances in Information Retrieval*. Springer, Berlin, 88–99.

Anne Aula, Rehan M. Khan, and Zhiwei Guan. 2010. How does search behavior change as search becomes more difficult? In *CHI'10*. ACM Press, New York, NY, 35.

Sharon E. Beatty and Scott M. Smith. 1987. External search effort: An investigation across several product categories. *J. Consum. Res.* (1987), 83–95.

Ruth N. Bolton. 1998. A dynamic model of the duration of the customer's relationship with a continuous service provider: The role of satisfaction. *Market. Sci.* 17, 1 (1998), 45–65.

David Carmel, Elad Yom-Tov, Adam Darlow, and Dan Pelleg. 2006. What makes a query difficult? In *SIGIR'06*. ACM Press, New York, NY, 390.

Lydia B. Chilton and Jaime Teevan. 2011. Addressing people's information needs directly in a web search result page. In *WWW'11*. ACM Press, New York, NY, 27.

Aleksandr Chuklin and Pavel Serdyukov. 2012a. Good abandonments in factoid queries. In *WWW'12 Companion*. ACM Press, New York, NY, 483.

Aleksandr Chuklin and Pavel Serdyukov. 2012b. Potential good abandonment prediction. In *WWW'12 Companion*. ACM Press, New York, NY, 485.

G. A. Churchill Jr. and C. Surprenant. 1982. Churchill, Gilbert A., Jr, an investigation into the determinants of customer satisfaction, 19:4 (1982:Nov.) p.491. *J. Market. Res.* 4, 19 (1982), 491–504.

Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *SIGIR'02*. ACM Press, New York, NY, 299.

Ovidiu Dan, Pavel Dmitriev, and Ryen W. White. 2012. Mining for insights in the search engine query stream. In *WWW'12 Companion*. ACM Press, New York, NY, 489.

Abdigani Diriye, Ryen White, Georg Buscher, and Susan Dumais. 2012. Leaving so soon? In *CIKM'12*. ACM Press, New York, NY, 1025.

Richard O. Duda, Peter E. Hart, and David G. Stork. 2000. *Pattern Classification*. John Wiley & Sons, New York, NY.

J. F. Engel and R. D. Blackwell. 1982. *Consumer Behavior*. Dryden Press.

Deborah Fallows. *Search Engine Users*. Technical Report. Pew Internet and American Life Project. http://www.pewinternet.org/files/old-media/Files/Reports/2005/PIP_Searche ngine_users.pdf.pdf.

Henry A. Feild, James Allan, and Rosie Jones. 2010. Predicting searcher frustration. In *SIGIR'10*. ACM Press, New York, NY, 34.

Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. 2005. Evaluating implicit measures to improve web search. *ACM Trans. Inform. Syst.* 23, 2 (April 2005), 147–168.

Qi Guo, Ryen W. White, Susan T. Dumais, Jue Wang, and Blake Anderson. 2010. Predicting query performance using query, result, and user interaction features. In *Adaptivity, Personalization and Fusion of Heterogeneous Information*. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 198–201.

Qi Guo, Ryen W. White, Yunqiao Zhang, Blake Anderson, and Susan T. Dumais. 2011. Why searchers switch. In *SIGIR'11*. ACM Press, New York, NY, 335.

Ahmed Hassan. 2012. A semi-supervised approach to modeling web search satisfaction. In *SIGIR'12*. ACM Press, New York, NY, 275.

Ahmed Hassan, Rosie Jones, and Kristina Lisa Klinkner. 2010. Beyond DCG. In *WSDM'10*. ACM Press, New York, NY, 221.

Ahmed Hassan, Xiaolin Shi, Nick Craswell, and Bill Ramsey. 2013. Beyond clicks. In *CIKM'13*. ACM Press, New York, NY, 2019–2028.

Ahmed Hassan, Yang Song, and Li-wei He. 2011. A task level metric for measuring web search satisfaction and its application on improving relevance estimation. In *CIKM'11*. ACM Press, New York, NY, 125.

Ahmed Hassan and Ryen W. White. 2013. Personalized models of search satisfaction. In *CIKM'13*. ACM Press, New York, NY, 2009–2018.

Ahmed Hassan, Ryen W. White, Susan T. Dumais, and Yi-Min Wang. 2014. Struggling or exploring? Disambiguating long search sessions. In *WSDM'14*. ACM Press, New York, NY, 53–62.

Ahmed Hassan, Ryen W. White, and Yi-Min Wang. 2013. Toward self-correcting search engines. In *SIGIR'13*. ACM Press, New York, NY, 263.

Claudia Hauff, Vanessa Murdock, and Ricardo Baeza-Yates. 2008. Improved query difficulty prediction for the web. In *CIKM'08*. ACM Press, New York, NY, 439.

Ben He and Iadh Ounis. 2004. Inferring query performance using pre-retrieval predictors. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 3246. 43–54.

Ben He and Iadh Ounis. 2006. Query performance prediction. *Inform. Syst.* 31, 7 (Nov. 2006), 585–594.

Allison P. Heath and Ryen W. White. 2008. Defection detection. In *WWW'08*. ACM Press, New York, NY, 1173.

Thorsten Hennig-Thurau and Alexander Klee. 1997. The impact of customer satisfaction and relationship quality on customer retention: A critical reassessment and model development. *Psychol. Market.* 14 (1997), 737–764.

Victor Hu, Maria Stone, Jan Pedersen, and Ryen W. White. 2011. Effects of search success on search engine re-use. In *CIKM'11*. ACM Press, New York, NY, 1841.

Jeff Huang, Ryen W. White, and Susan Dumais. 2011. No clicks, no problem. In *CHI'11*. ACM Press, New York, NY, 1225.

Scott B. Huffman and Michael Hochster. 2007. How well does result relevance predict session satisfaction? In *SIGIR'07*. ACM Press, New York, NY, 567.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inform. Syst. (TOIS)* 20, 4 (2002), 422–446.

Jiepu Jiang, Ahmed Hassan Awadallah, Xiaolin Shi, and Ryen W. White. 2015. Understanding and predicting graded search satisfaction. In *WSDM'15*. ACM, New York, NY, 57–66.

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR'05*. ACM Press, New York, NY, 154.

Thomas O. Jones, W. Earl Sasser, and others. 1995. Why satisfied customers defect. *Harvard Bus. Rev.* 73, 6 (1995), 88.

Yun-Fang Juan and Chi-Chao Chang. 2005. An analysis of search engine switching behavior using click streams. In *Special Interest Tracks and Posters - WWW'05*. ACM Press, New York, NY, 1050.

Jeonghyun Kim. 2006. Task difficulty as a predictor and indicator of web searching interaction. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems - CHI EA'06*. ACM Press, New York, NY, 959.

Youngho Kim, Ahmed Hassan, Ryen W. White, and Yi-Min Wang. 2013. Playing by the rules. In *WSDM'13*. ACM Press, New York, NY, 133.

Youngho Kim, Ahmed Hassan, Ryen W. White, and Imed Zitouni. 2014. Modeling dwell time to predict click-level satisfaction. In *WSDM'14*. ACM, New York, NY, 193–202.

Srivatsan Laxman, Vikram Tankasali, and Ryen W. White. 2008. Stream prediction using a generative model based on frequent episodes in event sequences. In *KDD'08*. ACM Press, New York, NY, 453.

Jane Li, Scott Huffman, and Akihito Tokuda. 2009. Good abandonment in mobile and PC internet search. In *SIGIR'09*. 43.

Chang Liu, Jingjing Liu, and Nicholas J. Belkin. 2014. Predicting search task difficulty at different search stages. In *CIKM'14*. ACM Press, New York, NY, 569–578.

Chang Liu, Jingjing Liu, Michael Cole, Nicholas J. Belkin, and Xiangmin Zhang. 2012a. Task difficulty and domain knowledge effects on information search behaviors. *Proc. Am. Soc. Inform. Sci. Technol.* 49, 1 (Jan. 2012), 1–10.

Jingjing Liu, Chang Liu, Michael Cole, Nicholas J. Belkin, and Xiangmin Zhang. 2012b. Exploring and predicting search task difficulty. In *CIKM'12*. ACM Press, New York, NY, 1313.

Jingjing Liu, Chang Liu, Jacek Gwizdka, and Nicholas J. Belkin. 2010a. Can search systems detect users' task difficulty? In *SIGIR'10*. ACM Press, New York, NY, 845.

Jingjing Liu, Chang Liu, Jacek Gwizdka, and Nicholas J. Belkin. 2010b. Predicting task difficulty for different task types. *Proc. Am. Soc. Inform. Sci. Technol.* 47, 1 (Nov. 2010), 1–10.

Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, and others. 2008. *Intro. Inform. Retriev.* Vol. 1. Cambridge University Press.

T. Mukhopadhyay, U. Rajan, and R. Telang. 2004. Competition between internet search engines. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004*. IEEE, Los Alamitos, CA, 10 pp.

R. L. Oliver. 2006. Customer satisfaction research. In *The Handbook of Marketing Research*. 1–40.

Denis Savenkov, Dmitry Lagun, and Qiaoling Liu. 2013. Search engine switching detection based on user personal preferences and behavior patterns. In *SIGIR'13*. ACM Press, New York, NY, 33.

Yang Song, Xiaolin Shi, Ryen White, and Ahmed Hassan Awadallah. 2014. Context-aware web search abandonment prediction. In *SIGIR'14*. ACM, New York, NY, 93–102.

Sofia Stamou and Efthimis N. Efthimiadis. 2010. Interpreting user inactivity on search results. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 5993 LNCS. 100–113.

David K. Tse and Peter C. Wilton. 1988. Models of consumer satisfaction formation: An extension. *J. Market. Rese.arch* 25, 2 (1988), pp. 204–212.

Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ahmed Hassan, and Ryen White. 2014. Modeling action-level satisfaction for search task satisfaction prediction. In *SIGIR'14*. ACM, New York, NY.

Ryen W. White and Steven M. Drucker. 2007. Investigating behavioral variability in web search. In *WWW'07*. ACM Press, New York, NY, 21.

Ryen W. White and Susan T. Dumais. 2009. Characterizing and predicting search engine switching behavior. In *CIKM'09*. ACM Press, New York, NY, 87.

Ryen W. White, Ashish Kapoor, and Susan T. Dumais. 2010. Modeling long-term search engine usage. In *User Modeling, Adaptation, and Personalization*. Springer, Berlin, 28–39.

Ryen W. White and Dan Morris. 2007. Investigating the querying and browsing behavior of advanced search engine users. In *SIGIR'07*. ACM Press, New York, NY, 255.

Ryen W. White, Matthew Richardson, Mikhail Bilenko, and Allison P. Heath. 2008. Enhancing web search by promoting multiple search engine use. In *SIGIR'08*. ACM Press, New York, NY, 43.

Received May 2015; revised February 2016; accepted February 2016