Modeling Action-level Satisfaction for Search Task Satisfaction Prediction

Hongning Wang
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana IL, 61801 USA
wang296@illinois.edu

Yang Song, Ming-Wei Chang
Xiaodong He, Ahmed Hassan
Ryen W. White
Microsoft Research, Redmond, WA 98004 USA
{yangsong,minchang,xiaohe,hassanam,ryenw}
@microsoft.com

ABSTRACT

Search satisfaction is a property of a user's search process. Understanding it is critical for search providers to evaluate the performance and improve the effectiveness of search engines. Existing methods model search satisfaction holistically at the search-task level, ignoring important dependencies between action-level satisfaction and overall task satisfaction. We hypothesize that searchers' latent action-level satisfaction (i.e., whether they believe they were satisfied with the results of a query or click) influences their observed search behaviors and contributes to overall search satisfaction. We conjecture that by modeling search satisfaction at the action level, we can build more complete and more accurate predictors of search-task satisfaction. To do this, we develop a latent structural learning method, whereby rich structured features and dependency relations unique to search satisfaction prediction are explored. Using in-situ search satisfaction judgments provided by searchers, we show that there is significant value in modeling action-level satisfaction in search-task satisfaction prediction. In addition, experimental results on large-scale logs from Bing.com demonstrate clear benefit from using inferred action satisfaction labels for other applications such as document relevance estimation and query suggestion.

Categories and Subject Descriptors

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

Keywords

Search-task satisfaction modeling, action-level satisfaction

1. INTRODUCTION

Measuring search engine performance via behavioral indicators of search satisfaction has recently received considerable attention [1, 11, 12, 15]. In comparison with traditional relevance-based evaluations [4], such methods enable evaluation using real user populations, in naturalistic settings, and across a diverse set of information needs. It has been shown that users' search behaviors provide more accurate signals of search satisfaction than query-document relevance [12, 15].

The core problem in search-task satisfaction modeling is to understand whether users are satisfied with their search actions (i.e., whether they believe they were satisfied with the search results for a particular information need) when performing the task [1, 3, 9, 15]. Unfortunately, searchers' detailed action satisfaction labels are *unobservable* in search log data; and they are difficult to obtain at scale from the searchers or reliably from third-party assessors. As a result, most prior search satisfaction models do not directly consider user satisfaction at the *action level*, or elect to only approximate that with specific assumptions. For example, most of existing methods consider search-task as the unit, and extract holistic measures, such as total dwell time [11, 31] and search result clicks [12], to perform search satisfaction prediction. Other methods that consider action-level behaviors do not predict users' detailed satisfaction over those actions [1, 15, 16]. Instead they assume that all actions are satisfying in a satisfying task, and all actions are unsatisfying in an unsatisfying task. This masks the complex relationship between action-level satisfaction and overall search-task satisfaction: e.g., searchers can be ultimately satisfied by the search task, but most of their search actions might be quite unsatisfying [11]. Therefore, such a modeling assumption expropriates the model's ability to discriminate between different actions, i.e., satisfying vs. unsatisfying.

In this work, we hypothesize that users' perceived action-level satisfaction, even though unobservable in search logs, influences their observed search behaviors and contributes to overall search-task satisfaction. We conjecture that by modeling satisfaction at the individual action level, we can build more complete and more accurate predictors of search satisfaction. To achieve this, we consider the action-level user satisfaction as latent variables, and explicitly model their relationship to overall task satisfaction in a latent structural learning framework. By introducing the latent variables, expressive features and dependencies unique to the search satisfaction problem can be incorporated to depict searchers' complex behavioral patterns. Knowledge about users' in-task search behaviors, e.g., consistency between action-level and overall task satisfaction, is naturally modeled in the proposed learning framework to guide satisfaction modeling.

Our research contributions can be summarized as follows:

- Explicitly model latent action-level satisfaction as part of search-task satisfaction modeling;

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

http://dx.doi.org/10.1145/2600428.2609607.

- Perform extensive experimental analysis of the proposed method whereby several state-of-the-art search satisfaction models are compared and significant performance improvement on different data sets is achieved;
- Demonstrate clear utility of the inferred action-level satisfaction labels by improved performance in document relevance estimation and query suggestion.

2. RELATED WORK

Recent advances in retrieval evaluation have focused on modeling search behaviors and exploiting implicit feedback [2, 19]. Qualitative studies showed that users' search behaviors are good indicators of retrieval system performance [27] and search-task difficulty [3]. Smith and Kantor found that users adapted their search behaviors to the deliberately degraded retrieval systems, e.g., increase the rate of query entry and decrease the occurrence of repeated queries [27]. Aula et al. reported that when facing with difficult search tasks, users tended to use more diverse queries and more advanced operations, and spend longer time on the search result pages [3]. Such studies shed lights on the potential of evaluating search performance via searchers' behaviors.

Satisfaction has been studied extensively in a number of areas such as psychology [24] and commerce [26]. In the IR literature, search satisfaction is generally defined as the fulfillment of a user's information need [12, 15]. Fox et al. [12] used an instrumented browser to collect search activities and compared them against explicit user satisfaction judgments of full search sessions. They identified a strong association between users' search patterns and their explicit satisfaction ratings. Hassan et al. [15, 17] utilized a user's search action sequence to predict search satisfaction. Feild et al. [11] focused on the behavioral clues to detect search frustration, where various signals from query logs and physiological sensors were explored. In [21], Kim et al. introduced more sophisticated signals to calibrate click dwell time for better estimating click satisfaction.

Despite the wealth of research in this area, most prior studies regard search-tasks as the basic modeling unit, from which holistic measures, e.g, total dwell-time [31] and query-click ratio [11], are extracted for predicting search satisfaction. However, users' detailed *action-level* satisfaction was largely ignored in prior work, though it conveys important information about searchers' overall search satisfaction [3, 11]: searchers can be ultimately satisfied by the search task, but most of their search actions might be quite unsatisfying. Thus, methods that fail to consider satisfaction at the action-level may not be optimal for this prediction problem. To the best of our knowledge, Ageev et al.'s work in [1] was the first attempt to consider users' action-level satisfaction for search-task satisfaction prediction. In their work, a controlled lab experiment is performed to track users' search activities during predefined search tasks. They approximated users' action-level satisfaction by using manual relevance judgments, and they identified distinct search paths among the satisfying/unsatisfying actions in the satisfying versus unsatisfying search tasks. Their study confirms our claim that it is necessary to distinguish and model users' action-level satisfaction in search-task satisfaction prediction. As their solution, a CRF model was adopted to predict search-task satisfaction based on a set of behavior features. However, because they asserted that all action labels equaled the task label, discrimination between different search actions was not possible. Therefore, their method is still unable to distinguish action-level satisfaction, as we do in this paper.

3. PROBLEM DEFINITION

In this section, we formally define the problem of search-task satisfaction prediction. A search task is defined to be an atomic information need, which results in a series of search actions [20]. Various methods have been proposed to extract search tasks from users' search logs [5, 29], and we will assume such segmentation is given a priori in our problem.

Specifically, the input of a search-task satisfaction prediction problem is a sequence of user u's search actions in a particular search task t_u, in which the actions are chronologically ordered, i.e., $A^{t_u} = \{a_1^{t_u}, \ldots, a_n^{t_u}\}$. We adopt the action type definition in [15], and consider the following types of search actions in this work:

- Q, issue a query to a search engine;
- SERP, hit BACK button to return to the search result page or refresh the search result page;
- PAGN, go to the next page of search results;
- SR, click on a returned document in search result page;
- BR, click on a hyperlink in the current document (not in a search result page);
- RL, click on a related search suggestion result;
- SP, click on the spelling correction link.

Each action $a_i^{t_u}$ has an attribute $a_i^{t_u}.ref$ pointing to the previous action which leads to the current action.

The above action types cover most of the search actions a user typically performs during Web search. Additionally, to be consistent with our later description of the proposed method, we add two dummy actions into every search task, i.e., $a_0^{t_u}$ = START and $a_{n+1}^{t_u}$ = END, indicating the start and end of a search task respectively. In particular, we denote $\mathcal{Q} = \{\text{Q}, \text{SERP}, \text{PAGN}, \text{RL}, \text{SP}\}$ as query-related actions, and $\mathcal{C} = \{\text{SR}, \text{BR}\}$ as click-related actions.

The output of a search-task satisfaction prediction problem is an overall satisfaction label y^{t_u} indicating whether the user u has been satisfied in the search task t_u. In this work, we follow Aula et al.'s criterion [3] to define search-task satisfaction as,

Definition (Search-Task Satisfaction) Given a user u's search task t_u, search-task satisfaction is a binary label y^{t_u}: $y^{t_u} = 1$, if the user's information need has been met and thus resulting a satisfying search task; otherwise $y^{t_u} = 0$.

In literature, there are different terms, e.g., "search success" [1, 15] and "frustration" [11], and perspectives, e.g., subjective [3, 15] or objective [8], used for defining a similar concept. We want to emphasize that our definition characterizes search satisfaction from a user's *subjective* perspective: a search-task is considered as satisfying, if, and only if, the searcher is satisfied with the search results and believes that they has found the answer (but the answer could be factually incorrect).

As a result, the problem of search-task satisfaction prediction is to estimate a function $f(\cdot)$ from the given search action sequence A^{t_u} to a search-task satisfaction label y^{t_u}, such that the predicted satisfaction label agrees with users' belief on whether they have satisfied their information need.

Most of the previous approaches for search-task satisfaction prediction [1, 11, 12, 14, 15] fall into the above formalism. However, one important factor that has not yet been explicitly defined and explored in prior works is user u's satisfaction label $h_i^{t_u}$ related to a specific action $a_i^{t_u}$. Intuitively, $h_i^{t_u}$ characterizes the contribution of action $a_i^{t_u}$ towards user u's overall satisfaction of task t_u. Formally, we define a user's action-level satisfaction as,

Definition (Action-level Satisfaction) Action-level satisfaction $h_i^{t_u}$ is a binary outcome of a search action $a_i^{t_u}$ in task t_u, such that $h_i^{t_u} = 1$, if user u is satisfied with action $a_i^{t_u}$, e.g., found helpful information after clicking on a document; otherwise $h_i^{t_u} = 0$, e.g., a query action does not lead to any useful document.

It is worthwhile to note that despite defining $h_i^{t_u}$ as binary in the above definition, the potential label space for the variable $h_i^{t_u}$ is quite flexible, e.g., encoding it with multi-level ordinal labels to reflect users' complex information seeking behaviors (e.g., query refinement [3], exploring related information [30]). Our proposed method can be easily extended to the multi-label setting. In this work, we will follow this binary definition for simplicity and explicability.

4. METHOD

In this section, we describe the proposed latent structural model for search-task satisfaction prediction. We start with a real search task example to illustrate the necessity of modeling searchers' action-level satisfaction. Then we discuss our hypothesis about users' search behaviors with respect to action-level satisfaction. And based on it, rich structured features and dependency relations unique to search-task satisfaction modeling are devised. In the end, we discuss how to incorporate domain-knowledge to guide the proposed model in learning the latent structures effectively.

4.1 Motivating Example

Table 1 presents a real example of a satisfying search task extracted from Ageev et al.'s public search data set [1]. We applied several state-of-the-art search satisfaction models, including the Markov Model Likelihood (MML) method [15], logistic regression (LogiReg) model [11] and session-CRF model [1], and our proposed method on this case. In particular, the MML and LogiReg take a holistic view to directly predict the task-level satisfaction, while the session-CRF and our method consider action-level satisfaction in the task. Due to space limitations, we only showed the domain of clicked documents in the table. The action-level prediction results from session-CRF model (denoted as "CRF") and our method (denoted as "Ours") are illustrated in the last two columns of the table.

In this example, the searcher sought information on metals that can float on water. She rated this task as satisfying because she claimed the answer had been found after search. But it does not mean that she was satisfied with every action in the task. As we can observe, she first attempted three queries on Google, but was not satisfied with the search results: she kept reformulating the queries, spent a very short time on the clicked documents, and switched to Bing with the same query. After spending quite some time on Bing's search result page, she issued a very specific query to Google and reached the correct answer (the answer was verified by a human editor).

Models based on task-level implicit measures, i.e., MML and LogiReg, mistakenly predicted that the searcher was unsatisfied with the task: dwell times on the clicked documents were generally short, along with a number of query reformulations and search engine switches. Due to the restrictive assumption in Ageev et al.'s session-CRF model, i.e., all actions have to be satisfying in a satisfying task, it made a wrong prediction for this task as well, since most actions were unsatisfying. But once we consider the searcher's action-level satisfaction, as predicted in our method's output, we could reach the correct conclusion that the task is satisfying. From this example, we can clearly realize the importance of recognizing a user's fine-grained satisfaction at action level for search-task satisfaction prediction.

Table 1: Example of a satisfying search task. '+'/'-' indicates a predicted satisfying/unsatisfying action.

Search Actions	Engine	Time	CRF	Ours
Q: metals float on water	Google	10s	-	-
SR: wiki.answers.com		2s	-	-
BR: blog.sciseek.com		3s	-	-
Q: which metals float on water	Google	31s	-	-
Q: metals floating on water	Google	16s	-	-
SR: www.blurtit.com		5s	-	-
Q: metals floating on water	Bing	53s	-	-
Q: lithium sodium potassium float on water	Google	38s	-	+
SR: www.docbrown.info		15s	-	+

4.2 Hypothesis and the AcTS Model

As was discussed in our motivating example in Table 1, the action-level satisfaction labels H[1] convey informative clues about overall search-task satisfaction. If H is known, sophisticated features about users' perceived satisfaction of search activities can be extracted, e.g., examining if the task ends with a satisfying action or measuring the ratio of time spent on satisfying actions versus unsatisfying ones, for better predicting the overall task satisfaction label y. Unfortunately, H is hidden in search log data; and it is also quite challenging to be manually annotated at scale. This prevents previous works from directly utilizing such information for search-task satisfaction prediction.

To address this challenge, we devise a basic hypothesis about users' search behaviors:

Hypothesis. *The desire for satisfaction drives users' interaction with search engines and that the satisfaction attained during the search-task contributes to the overall satisfaction.*

This hypothesis makes two assumptions. First, users' overall search-task satisfaction depends on their satisfaction with the performed search actions, e.g., if all actions were satisfying, it is very likely that the user would end up with a satisfying search task. Second, users' search actions are mutually dependent via the latent action satisfaction labels. For example, if a query is unsatisfying, e.g., it is later reissued to another search engine [13], the result clicks in the first search engine's result page can hardly be satisfying.

We consider H as latent variables and realize our hypothesis about a user's search behaviors in a structured prediction model. We name the proposed method as Action-aware Task Satisfaction (AcTS) model, and describe the structural dependencies imposed in the AcTS model in Figure 1.

To formally encode the dependency assumptions in our hypothesis, we define a feature vector for the task satisfaction label y specified by the search action sequence A and corresponding hidden action satisfaction labels H as $\Phi(A, H, y)$. Based on this feature representation, AcTS predicts the search-task satisfaction at testing time by,

$$(\hat{y}, \hat{H}) = \underset{(y,H)\in\mathcal{Y}\times\mathcal{H}}{\arg\max}\ w^{\mathsf{T}}\Phi(A, H, y). \quad (1)$$

In Eq (1), $\mathcal{Y}$ and $\mathcal{H}$ represent the sets of all possible values of y and configurations of H respectively. w is the parameter vector in our AcTS model, and it reflects the relative importance of features in predicting search-task satisfaction.

[1]When no ambiguity is invoked, we will discard the user index u and task index t^u to simplify the notations.

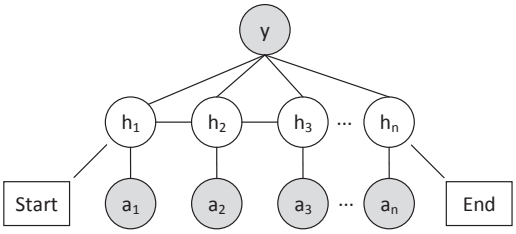


Figure 1: Structural dependency assumptions about a user's search behaviors postulated in AcTS model. Light circles represent latent variables and shadow circles represent observable variables. Lines indicate possible dependencies between the variables (the dependency between y and a is not shown to make the representation concise). In AcTS, a joint mapping of $f(A) \rightarrow H \times y$ is estimated.

In this paper, we refer to solving Eq (1) as the inference problem. In the solution of our inference problem, $\hat{y}$ becomes the output for the task-level satisfaction prediction and $\hat{H}$ is the inferred action-level satisfaction labels for the input search action sequence.

The inference problem of Eq (1) clearly distinguishes the proposed AcTS model from all the prior search satisfaction models. In order to make a prediction of the overall search satisfaction label y, we need to determine the latent action satisfaction labels H as well, which are mostly consistent with the observations in the input search actions A and support the predicted overall satisfaction label y in task t. Formally, we are estimating a joint mapping from input search action sequence A to task satisfaction label y and latent action satisfaction labels H, i.e., $f(A) \rightarrow H \times y$; while most prior works only estimate a binary mapping of $f(A) \rightarrow y$. Moreover, in the proposed AcTS model, a user's search actions A are no longer treated as independent, but instead, they are modeled as being correlated with each other via the latent action satisfaction labels H. Expressive features about a user's search behaviors can thus be designed, such as measuring the transition between a user's satisfying and unsatisfying search actions and examining whether a user is satisfied with all the query actions.

More importantly, the inferred action-level satisfaction labels H not only provide informative signals for determining overall search satisfaction, but also reveal the utility of those actions towards a user's information need. For example, based on the identified labels in H, we can easily recognize which clicked document is helpful in satisfying a user's information need, and which query leads to the helpful documents. The estimated utilities are beneficial for a variety of search applications, e.g., document relevance estimation and query suggestion. Nevertheless, such information is not available in any of the existing search satisfaction models.

In the following, we will discuss in detail about our design of the structured features $\Phi(A, H, y)$ in Section 4.3, and the use of domain knowledge for learning the optimal feature weights w in Section 4.4.

4.3 Structured Features

Previous work has developed a wide variety of behavioral features for search satisfaction prediction [1, 11, 12, 13, 31]. All of those features can be flexibly applied in our AcTS model. However, since most prior research only estimates a holistic mapping of $f(A) \rightarrow y$, their employed features (e.g., total dwell time [31] and number of result clicks [12]) cannot capture a user's action-level satisfaction. In this section, we focus on the newly developed structured features for AcTS, in which expressive signals about the dependency among search actions A, action-level satisfaction labels H and task-level satisfaction label y is explicitly explored via the latent variables. The devised features can be categorized into two classes: short-range features (specifying satisfaction label for a single action in task t) and long-range features (specifying satisfaction labels for a set of actions in task t).

• **Short-range features:** As shown in Figure 1, in our AcTS model, the features extracted from action a_i are directly used to predict the corresponding satisfaction label h_i and overall task satisfaction label y (i.e., $f(A) \rightarrow H \times y$). This is distinct from the features explored in most existing search satisfaction models, where the action-level observations are aggregated to determine the task-level satisfaction label y [11, 12, 13].

In a user's query-related actions, although not especially common, search engine switching (i.e., the voluntary transition between different search engines) usually indicates searcher frustration [13]. We encode this as $\phi_{switch}(y, A, h_i) = \delta(y, a_i, h_i)\delta(a_i.Engine \neq a_j.Engine)$, where a_j is the next query action following the current query action a_i and $\delta(\cdot)$ is the indicator function. Similarly, query reformulation also indicates the user is not satisfied with the search results of the current query [3]. We formalize this by measuring the similarity between two consecutive queries: $\phi_{reform}(y, A, h_i) = \delta(y, a_i, h_i)sim(a_i.Query, a_j.Query)$, where a_j is the query action following the current query action a_i, and $sim(X, Y)$ is the edit distance between query string X and Y. Besides, we also examine if the query is in a question form by $\phi_{question}(y, A, h_i)$ and calculate the proportion of stopwords in the query by $\phi_{stopword}(y, A, h_i)$ to estimate satisfaction for the query-related actions.

Among a user's click-related actions, the relevance quality of a clicked document to the given query can be an important criterion to measure user satisfaction [18]. Because we do not assume the availability of document content in our problem (it is usually unavailable in search log data), we can only measure relevance of the clicked documents according to their URL strings. In particular, we define $\phi_{rel}(y, A, h_i) = \delta(y, a_i, h_i)c(a_i.URL, a_k.Query)$, where $c(URL, Query)$ counts the number of query terms occurred in the URL string, and a_k is the query action that leads to the current click action a_i. In addition, the original rank position of the clicked URL in search-result page is also a good indicator of its relevance quality [18]. We encode it as $\phi_{pos}(y, a_i, h_i) = \delta(y, a_i, h_i)a_i.Pos$.

Besides, previous studies have demonstrated that a user's last search action is closely related to her search-task satisfaction [12]. We encode this as $\phi_{last}(y, h_n) = \delta(y = h_n)$, i.e., examine whether the satisfaction label of the user's last action agrees with her overall task satisfaction.

• **Long-range features:** We devise the first order transition feature $\phi_{trans}(y, h_i, h_{i+1}, a_i, a_{i+1})$ to capture a user's sequential search behaviors with respect to the latent action satisfaction labels. For example, in a satisfying search task, an unsatisfying query is more likely to be reformulated into a satisfying query rather than another unsatisfying one. In particular, we define $\phi_{trans}(y, h_i, h_{i+1}, a_i, a_{i+1}) = \delta(y = y', h_i = h', h_{i+1} = h'', a_i = a', a_{i+1} = a'')$, where (y', h', h'', a', a'') takes all the possible values for task satisfaction label, action satisfaction labels and action types. We should note that our transition features are different from those introduced in [14, 15, 17]: in those works, only the transitions between different action types are modeled, e.g., from Q to SR; while in our model, we distinguish search ac-

Table 2: Structured behaviorial features for search-task satisfaction modeling in AcTS.

Type	Feature	Description
Short-range	$\phi_{switch}(y, A, h_i)$	if the user switches search engine after this query action
	$\phi_{reform}(y, A, h_i)$	edit distance between two consecutive queries
	$\phi_{question}(y, a_i, h_i)$	if the query is a question
	$\phi_{stopword}(y, a_i, h_i)$	proportion of stopwords in query
	$\phi_{rel}(y, a_i, h_i)$	query term matching in URL string of a_i
	$\phi_{pos}(y, a_i, h_i)$	display position of the clicked URL
	$\phi_{last}(y, h_n)$	if T ends up with a satisfying search action
Long-range	$\phi_{trans}(y, h_i, h_{i+1}, a_i, a_{i+1})$	first order transition between actions with respect to satisfaction labels
	$\phi_{allQ}(y, H, A)$	if all the query-related actions in T are satisfying
	$\phi_{existQ}(y, H, A)$	if there exists a satisfying query-related action in T
	$\phi_{allC}(y, H, A)$	if all the click-related actions in T are satisfying
	$\phi_{existC}(y, H, A)$	if there exists a satisfying click-related action in T

tion transitions with respect to the latent action satisfaction labels, e.g., from satisfying Q to satisfying SR.

Beyond exploring the behaviorial patterns within adjacent search actions, a set of features are introduced to capture dependency at the whole task level by examing: I. if all the query-related actions are satisfying: $\phi_{allQ}(y, H, A) = \delta(\sum_{a_i \in \mathcal{Q}} h_i = \sum_{a_i \in \mathcal{Q}} 1)$; II. if there exists a satisfying query action: $\phi_{existQ}(y, H, A) = \delta(\sum_{a_i \in \mathcal{Q}} h_i > 0)$; III. if all the click-related actions are satisfying: $\phi_{allC}(y, H, A) = \delta(\sum_{a_i \in \mathcal{C}} h_i = \sum_{a_i \in \mathcal{C}} 1)$; and, IV. if there exists a satisfying click: $\phi_{existC}(y, H, A) = \delta(\sum_{a_i \in \mathcal{C}} h_i > 0)$.

We need to emphasize that the above long-range features can only be exploited by our AcTS model, since it explicitly models the users' action-level satisfaction across different actions in a search task. None of existing methods can utilize such information for search-task satisfaction prediction.

In addition to the above newly introduced structured features, we also included the action-level and task-level behavior features from [1] and [12] in our AcTS model, such as action dwell time and query-click ratio. The list of features[2] used in this work appears in Table 2.

4.4 Training AcTS with Weak Supervision

Because the ground-truth labels for a user's action-level satisfaction are unobservable in the search log data, we have no direct supervision to guide the model in learning about such latent structures. Fortunately, there is plenty of work in cognitive science and information science exploring users' search behaviors and strategies in performing a successful search task [3, 25, 30]. Such studies shed light on the insights of users' detailed in-task search behavior patterns. In this section, we propose the use of *structured loss functions* [7] to inject such domain knowledge as weak supervision for AcTS training (i.e., learning the weight vector w in Eq (1)).

To regularize the training of AcTS model with domain knowledge, we derive our learning algorithm for the AcTS model from the latent structural SVMs framework [7]. For a given set of search tasks with only task-level search satisfaction labels, i.e., $\{(A_m, y_m)\}_{m=1}^M$, AcTS model training can be formalized as the following optimization problem:

$$\min_{w,\xi} \ \frac{1}{2}||w||^2 + C\sum_{m=1}^{M} \xi_m^2 \qquad (2)$$
$$s.t.\ \forall m, \quad \max_{H \in \mathcal{H}} w^{\mathsf{T}}\Phi(A_m, H, y_m) \geq \max_{(\hat{y},\hat{H}) \in \mathcal{Y}\times\mathcal{H}} [w^{\mathsf{T}}\Phi(A_m, \hat{H}, \hat{y}) + \Delta(y_m, \hat{y}, \hat{H}, A)] - \xi_m.$$

[2]Details of features from [1, 12] are not listed in the table.

In Eq (2), $\Delta(y_m, \hat{y}, \hat{H}, A)$ measures the distance between the predicted labels $(\hat{y}, \hat{H})$ and the ground-truth (y_m, H_m^*), where H_m^* is the unobservable ground-truth of action-level satisfaction labels. $\{\xi_m\}_{m=1}^M$ is a set of slack variables to allow errors in the training data, and C controls the trade-off between empirical training loss and model complexity.

$\Delta(y_m, \hat{y}, \hat{H}, A_m)$ indicates the prediction error between $(\hat{y}, \hat{H})$ and (y_m, H_m^*); and thus it drives model learning. As H_m^* is unknown in the training data, we have no supervision to guide the AcTS model in learning about such latent structures. As our solution, weak supervision about users' search behaviors is injected via the design of $\Delta(y_m, \hat{y}, \hat{H}, A_m)$. Intuitively, we should increase $\Delta(y_m, \hat{y}, \hat{H}, A_m)$, i.e., penalize the prediction, when the inferred $\hat{H}$ contradicts our knowledge about a legitimate configuration of H. In this work, we define a set of structured loss functions $\sigma(\hat{y}, \hat{H}, A)$ to realize the knowledge about $\hat{H}$ in $\Delta(y_m, \hat{y}, \hat{H}, A_m)$ from different perspectives.

First, a good configuration of $\hat{H}$ has to be consistent with the predicted overall search-task satisfaction label $\hat{y}$. We measure this by:

$$\sigma_{sat}(\hat{y}, \hat{H}) = \begin{cases} 1 & \hat{y} = 1,\ \sum_{\hat{h}_i \in \hat{H}} \hat{h}_i = 0 \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

i.e., all the actions should not be unsatisfying in a satisfying task. And, vice versa,

$$\sigma_{dsat}(\hat{y}, \hat{H}) = \begin{cases} 1 & \hat{y} = 0,\ \sum_{\hat{h}_i \in \hat{H}} \hat{h}_i = \sum_{\hat{h}_i} 1 \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

Second, the configuration of $\hat{H}$ itself should be consistent. For example, an unsatisfying query cannot result in any satisfying search-result clicks [1], i.e.,

$$\sigma_{clk}(\hat{H}, A) = \begin{cases} 1 & \text{exist } a_i{=}Q, a_j{=}SR, a_j.\text{ref}{=}a_i, \hat{h}_i{<}\hat{h}_j \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

And when the user performs duplicated actions in the same task, e.g., submit the same query twice to the same search engine, their inferred satisfaction labels should be the same,

$$\sigma_{dup}(\hat{H}, A) = \begin{cases} 1 & \text{exist } a_i = a_j, \hat{h}_i \neq \hat{h}_j \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

The suggested query from a search engine's spelling correction, e.g., correcting the misspelt query "amazone" into its correct form "amazon," should not hurt user satisfaction,

$$\sigma_{sp}(\hat{H}, A) = \begin{cases} 1 & \text{exist } a_i{=}Q, a_j{=}SP, a_j.\text{ref}{=}a_i, \hat{h}_i{>}\hat{h}_j \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

Based on the above estimated distance between $\hat{H}$ and H^*_m, we can define the margin in Eq (2) as,

$$\Delta(y_m, \hat{y}, \hat{H}, A_m) = \delta(y_m = \hat{y}) + \sum_i \lambda_i \sigma_i(\hat{y}, \hat{H}, A_m). \quad (8)$$

where λ_i is a trade-off parameter between task-level 0/1 loss and action-level loss defined by the structured loss functions $\sigma(\hat{y}, \hat{H}, A)$ as described in Eq (3) to Eq (7).

The margin function defined above encodes the knowledge about a user's latent action-level satisfaction labels within a search task as weak supervision for latent structure learning [7]. It bridges the qualitative studies of users' search behaviors [3, 25, 30] and quantitative modeling approaches. We should note that the structured loss functions $\sigma(\hat{y}, \hat{H}, A)$ might be violated in a particular user's real search actions, and λ_i controls our confidence of such loss functions.

The optimization problem in Eq (2) can be efficiently solved by the iterative algorithm proposed in [7]. One thing we should note is that due to the long-range dependency introduced by the structured features proposed in Section 4.3, e.g., $\phi_{allQ}(y, H, A)$ and $\phi_{existQ}(y, H, A)$, the inference problems defined in Eq (1) and Eq (2) become computationally intractable. We address these inference problems via integer linear programming (ILP), and more details about this inference method can be found in [22].

5. EXPERIMENTS

In this section, we first quantitatively evaluate the effectiveness of the proposed AcTS model that models users' action-level satisfaction as latent variables, whereby several state-of-the-art search satisfaction models are compared over the in-situ task satisfaction labels from previous studies [1, 16]. Then we assess the quality of the inferred action-level satisfaction labels via their utilities in facilitating other information retrieval studies, where understanding users' detailed action-level satisfaction is important.

5.1 Data Sets

Hassan et al. [16] developed a toolbar plugin for the Internet Explorer browser to collect search activities and explicit search satisfaction ratings from the searchers. The authors explicitly asked the searchers to rate their search tasks immediately upon termination. This data set provides reliable first-hand annotation of search-task satisfaction. We refer to this data set as "toolbar data" in our experiments.

Ageev et al. [1] designed a game-like online contest for crowdsourcing search behavior studies. In their study, users were required to perform several predefined informational tasks via a Web search interface and submit the answers they found to the system. All users' search behaviors were logged and annotated by the authors. According to our search-task satisfaction definition described in Section 3, we treat the tasks in which the user has submitted an answer as satisfying (the answer might be incorrect with respect to the predefined information need). We refer to this data set as "contest data" in our experiments.

To investigate the utility of the proposed method in predicting search satisfaction in real-world search engine logs, we extracted large-scale query logs sampled from the Microsoft Bing Web search engine. In a four-month period, from December 2012 to March 2013, a subset of users were randomly selected. The search logs recorded their search activities, including the anonymized user ID, query string, timestamp, returned URL sets and the corresponding user clicks. These logs were segmented into search tasks by the method developed in [29]. This data set does not contain task-level nor action-level satisfaction labels. We refer to it as "search log data," and describe its usage in Section 5.3.

Table 3: Basic statistics of evaluation data sets.

Data set	# User	# Task	Action/Task	T^+:T^-
toolbar	153	7306	5.2(±6.6)	6.84:1
contest	156	1487	6.2(±5.9)	6.70:1
search log	2.4M	7.6M	7.1(±11.8)	-

Basic statistics of these data sets appear in Table 3.

5.2 Search-Task Satisfaction Prediction

To investigate the effectiveness of modeling users' action-level satisfaction as latent variables in AcTS model, we first quantitatively compare the performance of the proposed model with several state-of-the-art methods in predicting overall search-task satisfaction.

5.2.1 Baselines

Several methods have been proposed to predict search satisfaction based on users' search behaviors [1, 11, 14, 15]. We adopt several best-performing models from the previous works as our baseline methods.

Hassan et al. [15] proposed a Markov Model Likelihood (MML) method to predict search satisfaction. In MML, two sets of first order transition probabilities are estimated from the search action trails in satisfying and unsatisfying tasks. At testing time, MML calculates the likelihood ratio of an input search action sequence between the satisfying and unsatisfying models to determine the task satisfaction label. We followed the specification of MML in [14] to implement the model (they used the same set of action types as ours). Maximum *a Posteriori* estimator with Dirichlet priors is used to estimate the transition probabilities in MML. To model click dwell time in MML, we add two new action types, *SR_long* and *BR_long*, which represent the click actions (*SR* and *BR*) with dwell time longer than 30s.

Feild et al. [11] used a logistic regression model to predict search frustration, where features extracted from both query logs and physiological sensors are employed. We built a logistic regression model based on the features described in Section 4.3. The short-range features are aggregated in each task by action type, e.g., average the click position features $\phi_{pos}(y, a_i, h_i)$ over all SR actions in the same task. The long-range features, e.g., $\phi^t_{allQ}(y, H, A)$, are not included, since logistic regression cannot handle latent variables. We refer to this method as "LogiReg."

Ageev et al. proposed a session-CRF model [1] to predict search-task satisfaction. Although search actions were explicitly modeled, they asserted that action-level satisfaction labels equaled to the task-level label. Mathematically, this assumption makes their session-CRF degenerate to a logistic regression model. This obscures the complex dependency between task satisfaction and detailed action satisfactions in session-CRF. As a result, it cannot as effectively model the action-level user satisfaction as our model does. We adopted the same implementation of session-CRF as used in [1].

5.2.2 Effectiveness of the Latent Structure Model

As illustrated in Table 3, the distribution of task satisfaction labels in both toolbar and contest data are highly unbalanced: about 85% of the tasks are labeled as satisfying. In such an unbalanced data set, accuracy alone is inadequate to compare the performance of different methods. In our evaluation, we compute the f_1 scores for both satisfying (T^+-f_1) and unsatisfying tasks (T^--f_1). Following the metric used

Table 4: Search task success prediction performance on the toolbar data set.

	avg-f_1	T^+-f_1	T^--f_1	Accuracy
MML	0.707	0.897	0.518	0.830
LogiReg	0.740	0.918	0.563	0.861
session-CRF	0.728	0.910	0.545	0.850
AcTS	**0.761***	**0.938***	**0.584***	**0.893***
AcTS$_0$	0.739	0.924	0.554	0.868

$^*p$-value$<$0.05

Table 5: Search task success prediction performance on the contest data set.

	avg-f_1	T^+-f_1	T^--f_1	Accuracy
MML	0.658	0.901	0.414	0.831
LogiReg	0.682	0.930	0.435	0.875
session-CRF	0.685	0.921	0.449	0.862
AcTS	**0.701***	**0.934**	**0.469***	**0.882**
AcTS$_0$	0.687	0.925	0.449	0.868
labeled-AcTS	0.649	0.945	0.352	0.899

$^*p$-value$<$0.05

in [1], we also report the average f_1 between T^+-f_1 and T^--f_1. In order to avoid bias introduced by training/testing split, we performed five-fold cross-validation in each method by sampling tasks into different folds, and repeated it three times with different random seeds. As a result, we report the average performance of all methods from 15 different trials on the toolbar and contest data sets in Table 4 and Table 5. Paired two sample t-test is performed to validate the statistical significance of the improvement from the AcTS model against the best-performing baseline, LogiReg, under each of the performance metrics. In particular, we set the trade-off parameters λ_i to one in Eq (8) for AcTS model in all our experiments.

We can clearly observe the significant improvement from the proposed AcTS model over all baselines in both data sets. MML, which only models the sequential patterns in a user's search actions, performed the worst among all the methods. This indicates that a user's sequential search behaviors alone are insufficient to capture the overall search satisfaction. session-CRF behaved similarly as LogiReg. Although action-level labels are explicitly modeled in session-CRF, its restrictive assumption about the labels degrades the model's capability in distinguishing the action-level satisfaction labels, e.g., unsatisfying actions will not be allowed in a satisfying task in session-CRF. We accredit the encouraging performance improvement of the proposed AcTS model to its unique capability of modeling the action satisfaction labels as latent variables. By explicitly modeling a user's action-level satisfaction, AcTS can naturally include all the signals used in the baseline methods and explore richer structured information, as specified in our long-range features, which cannot be handled in any baseline method.

Beside exploring more expressive structured features for search-task satisfaction prediction, another unique advantage of modeling the action-level satisfaction labels as latent variables in AcTS is to incorporate domain-knowledge for model training via the structured loss functions. To investigate this aspect, we test a special setting of AcTS, in which we set the trade-off parameters λ_i to zero in Eq (9). As a result, we are training the AcTS model with only task-level supervision. We name this model as AcTS$_0$ and include its performance on both data sets in Table 4 and Table 5.

Without the structured loss functions, AcTS's performance dropped significantly: it performed similarly as the LogiReg and session-CRF baselines. The reason is that the task-level satisfaction label alone cannot guide the model in learning the latent structures of H. As a result, the inferred labels of H in AcTS$_0$ becomes arbitrary, and provides little help in predicting task satisfaction. This result confirms the need to explicitly model the *dependency* between action-level and task-level satisfaction in search satisfaction modeling.

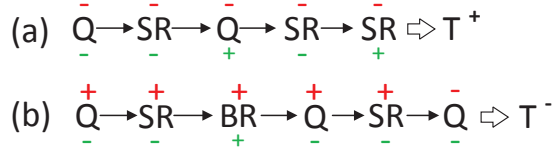


Figure 2: Case study of two manually annotated search sequences in the contest data set. Red labels on top of each action are the editor's annotations from [1], and green labels at the bottom are AcTS's predicted labels. T^+ and T^- indicate the task satisfaction labels provided by the users.

In addition, since action-level manual annotations are available in the contest data set, we can treat those labels as "ground-truth" action satisfaction labels, and train our AcTS model with a known structure. To incorporate these labels into AcTS training, we define a new margin for Eq (2) as,

$$\Delta(y, \hat{y}, H^*, \hat{H}) = \delta(y = \hat{y}) + \textstyle\sum_i \delta(h_i^* = \hat{h}_i) \qquad (9)$$

i.e., we are computing the Hamming distance between two labeled search sequences. We name this new model as labeled-AcTS, and list its performance in Table 5.

Surprisingly, the labeled-AcTS model did not outperform the original AcTS model with latent structures; and it performed significantly worse when predicting the unsatisfying tasks. To analyze the degraded performance, we examined the annotated search tasks in this data set and found many disagreements between the editor's judgements and searchers' actual behaviors. The discrepancy mainly stems from the inconsistent criteria between third-party editors and real users; and to understand it, we illustrate two typical inconsistent search sequences in Figure 2.

In Figure 2(a), all the clicked documents are judged to be irrelevant for this task. However the searcher still rated the task as satisfying. A reasonable explanation is that the searcher believed that she had found the correct answer, so was satisfied. While in Figure 2(b), according to the editor's judgment, the searcher has already issued several good queries and found the relevant documents for the question. However, the user rated it as unsatisfying in the end. The reason might be that the user did not notice the relevant passage(s) in the clicked documents, so was not satisfied with all of her search actions. The main reason for such disagreements is the annotation criterion devised in this data set [1]: Ageev et al. labeled a URL as a good URL if it contains the correct answer to the predefined question in the search task; and a query is judged to be good if it leads to a good URL in its search result page. Nevertheless, from a real searcher's perspective, because she may not have any knowledge about the questions beforehand, she cannot fully judge the helpfulness of search results in an objective way. As a result, the editor's objective judgments in this data set cannot precisely reflect a user's perceived satisfaction during search, which, however, is the goal of prediction in our task satisfaction prediction problem. Such discrepancy explains the degraded performance of labeled-AcTS: in Eq (9), we overly penalized the predictions in model training due to the inappropriate manual annotations.

Meanwhile, as shown in Figure 2, the inferred action labels from our AcTS model better aligned with the final task satisfaction labels in both cases: in Figure 2(a), the last query and last click action are predicted as satisfying; and in Figure 2(b), most of the actions are predicted as unsatisfying. Those inferred labels are more consistent with our above hypothetical analysis of users' search behaviors.

5.3 Action-Level Satisfaction Modeling

As an output of our structured inference problem defined in Eq (1), the inferred action-level satisfaction labels H have shown their ability in helping to predict the overall search-task satisfaction. Nevertheless, because the ground-truth labels of such output are not available in our evaluation data sets, we cannot directly evaluate the quality of the predicted action-level labels. In this section, we assess the quality of the inferred action satisfaction labels via their utilities in facilitating other information retrieval applications, where understanding users' action-level satisfaction is important.

5.3.1 Document Relevance Estimation

Accurately interpreting users' clickthroughs and extracting relevance signals for search engine optimization is an important topic in IR studies [2, 9]. The action-level satisfaction labels from AcTS can serve as a proxy to estimate the utility of clicked documents. In this section, we evaluate how the estimated document relevance can be used to improve the training of general learning-to-rank algorithms.

We chose LambdaMART [6] as our base learning-to-rank algorithm, and evaluate its ranking performance improvement by adding the features derived from AcTS model's output. A large set of manually annotated query-URL pairs are collected to create the evaluation data set. In this annotation set, each query-URL pair is labeled with a five-point relevance score, i.e., from 0 for "bad" to 4 for "perfect." And each pair is represented by a set of 398 standard ranking features, e.g., BM25, language model score and PageRank. We refer to this collection as the "annotation set."

In this experiment, we train an AcTS model based on all the search tasks in the toolbar data, and apply the learned model on the four-month search log data. We group the inferred satisfaction labels under each unique query-URL pair, and calculate the corresponding median, mean and standard deviation as the additional relevance features derived from AcTS. To reduce noise in this estimation, we ignore the query-URL pairs occurred less than five times in this corpus. In the end, we joined the query-URL pairs extracted from the search log data with those in the annotation set, and obtained 3,311 annotated queries and 128,120 query-URL pairs with additional features derived from AcTS.

The same feature generation strategy is applied to the session-CRF's output. However, the MML and LogiReg baselines are not directly applicable in this evaluation, since they cannot make predictions of individual search actions. To compare with them, we followed Hassan et al.'s method [16] to estimate document utility based on the predicted overall task satisfaction labels. In their method, the utility of a clicked document is assumed to be proportional to its dwell time. To distinguish document utilities between satisfying and unsatisfying tasks, they separated such scores into "utility" (for satisfying tasks) and "despair" (for unsatisfying tasks), which were used as two different relevance features. To make their relevance feature representation consistent with that from our AcTS model, i.e., one utility score per query-URL pair, we unified "utility" and "despair" by simply treating "despair" as negative "utility." As a result, we can apply the same aggregation strategy over all the query-URL pairs based on MML's and LogiReg's output to generate new relevance features from those methods.

Table 6: Ranking performance improvements of LambdaMART with additional document relevance features estimated from different methods.

%	P@1	MAP	NDCG@5	MRR
MML	+4.926	+3.482	+3.573	+2.650
LogiReg	+5.110	+3.352	+3.783	+2.776
session-CRF	+4.752	+3.402	+3.896	+2.616
SUM	+5.101	+3.405	+3.946	+2.807
AcTS†	**+5.366**	**+3.819**	**+4.278**	**+2.955**

†: p-value<0.01 in all the metrics.

In addition, we also include a session-based click model, i.e., Session Utility Model (SUM) [9], as a baseline in this experiment. SUM aims to extract the intrinsic relevance of documents to the given query from users' click behaviors in search sessions (tasks). However, it assumes all the search tasks are satisfying when modeling clicks. Thus, it is necessary to investigate if modeling search-task satisfaction is needed for estimating document relevance from user clicks.

In our experiment, we fixed the total number of trees in LambdaMART to 100, each of which has 15 nodes. The learning rate was set to 0.1. Five-fold cross-validation was used, where we used one fold of data for testing, one fold for validation and the reminder for training. We computed four standard IR evaluation metrics. By treating all the labels above "fair" as relevant, we calculated P@1, MAP and MRR. NDCG@5 was also computed based on the five-point relevance scale. The improvements of LambdaMART's ranking performance with different additional relevance features against the original features are listed in Table 6.

The new relevance features from AcTS significantly improved LambdaMART's performance against the original features under all the metrics (p-value <0.01). We examined the learned tree models in LambdaMART and found all the features generated by AcTS model are selected as important splitting factors (i.e., among the top 10 important features). The features from AcTS also significantly improved the MAP and NDCG@5 metrics (p-value<0.05) comparing to those from MML and SUM, which are the second best methods under these two metrics respectively. Since no baseline search satisfaction models can distinguish the fine-grained action-level satisfaction, their estimated relevance features are not as accurate as those from the inferred action-level labels of our AcTS model. Comparing to SUM, although it distinguishes the utility of different clicked documents, it does not consider overall task satisfaction when modeling user clicks, and thus the relevance features from AcTS led to better improved ranking performance. This result validates the need to distinguish overall task satisfaction in modeling clicks for document relevance estimation.

5.3.2 Query Reformulation Quality Estimation

Search tasks provide rich context for performing log-based query suggestion [10, 23]. Liao et al. [23] reported that the Log Likelihood Ratio (LLR) based query similarity metric achieved the best performance in their task-based query suggestion experiment. In this section, we investigate how the identified action-level user satisfaction labels can be used to further improve LLR in task-based query suggestion.

Given two queries q_a and q_b, assuming q_b is issued after q_a, LLR makes the null hypothesis H_0 as: $P(q_b|q_a) = p_0 = P(q_b|\neg q_a)$, i.e., q_a and q_b are independent; and the alterna-

tive hypothesis H_1 as: $P(q_b|q_a) = p_1 \neq p_2 = P(q_b|\neg q_a)$, i.e., q_a and q_b are dependent. Likelihood ratio test is used, in which the test statistic is defined as $-2\ln\frac{\max_{p_0} L(H_0)}{\max_{p_1,p_2} L(H_1)}$, to determine the dependency between q_a and q_b. If the value of test statistic is larger than a predefined threshold, the null hypothesis is rejected, i.e., q_a and q_b are determined to be dependent, and q_b will be selected as a suggestion for q_a.

In Liao et al.'s work, the probabilities of p_1, p_2 were estimated by the occurrences of consecutive queries in the same task, without considering the quality of query reformulations. For example, if a satisfying query q_a is frequently reformulated into an unsatisfying query q_b, even though they are strongly correlated according to LLR, we should not suggest q_b for q_a to users. To take the inferred action-level satisfaction into account, we weight the consecutive query pairs according to their inferred satisfaction labels by,

$$c(q_a, q_b) = \exp(h_b - h_a) \tag{10}$$

i.e., we emphasize the pair of queries in LLR calculation, where the follow-up query improved user satisfaction; and downgrade the reformulations that hurt user satisfaction. Based on this weighting scheme, the same LLR test statistics are computed for measuring correlation between queries.

The LLR test statistics for all consecutive query pairs in the identified search tasks are computed based on the first three-month search logs. The same threshold, 100 as used in [23], is applied to filter the suggestion candidates. The fourth-month search logs are used as the testing set to examine whether the suggested queries will be issued by users after the target query [28]. Such evaluation measures utility of the suggested queries in real usage context. In particular, the next three consecutive queries following the target query in the same search task are regarded as relevant in computing the evaluation metrics of P@3, MAP and MRR. To make the evaluation results comparable between the baseline (LLR without query weighting) and our method (LLR with query weighting), we only evaluated the overlapped target queries in both methods.

Beside this automatic evaluation method, we also collected a set of manual annotations to assess the quality of the suggested queries. We ordered all the target queries from the first three-month search logs according to their frequency, and treated the first third of queries as high frequency queries, the second third as medium, and the rest as low. 100 queries were randomly selected from each category. For each selected target query, the top five suggestions from both methods were selected and interleaved before being presented to the annotators, in order to reduce annotation bias. Six human annotators were recruited to label the suggestion results. They were instructed to judge if the suggestions are relevant to the given target query with binary labels. Annotators were separated into two groups, each of which was required to annotate 150 target queries selected evenly from the above three categories. The final relevance judgment was obtained by majority vote. We list the improvements of the LLR-based query suggestion performance from the new query weighting scheme on these two testing sets in Table 7.

As shown in the results, the new query weighting scheme greatly improved the original co-occurrence based query suggestion performance. In the log-based automatic evaluation, all the performance metrics are significantly improved. According to the manual judgments, the major improvement is derived from the low frequency queries: P@3 and MAP are improved by 14.8% and 15.1% accordingly (p-value<0.01). In Liao et al.'s reported result, their task-based LLR performed poorly on this category, which they attributed to the sparsity of query co-occurrence. Therefore, we can find clear benefit of distinguishing the quality of the reformulated queries when performing query suggestion for those low frequency queries. And the inferred action satisfaction labels from AcTS provide such a reliable quality estimator for further improving the query suggestion performance.

Table 7: Query suggestion performance improvements with query reformulation quality estimated by AcTS.

%	P@3	MAP	MRR
Query log†	+7.18	+8.60	+6.42
Annotation	+2.58	+6.79	-0.75

†: p-value<0.01 in all the metrics.

5.3.3 Analysis of Search Behavior Patterns

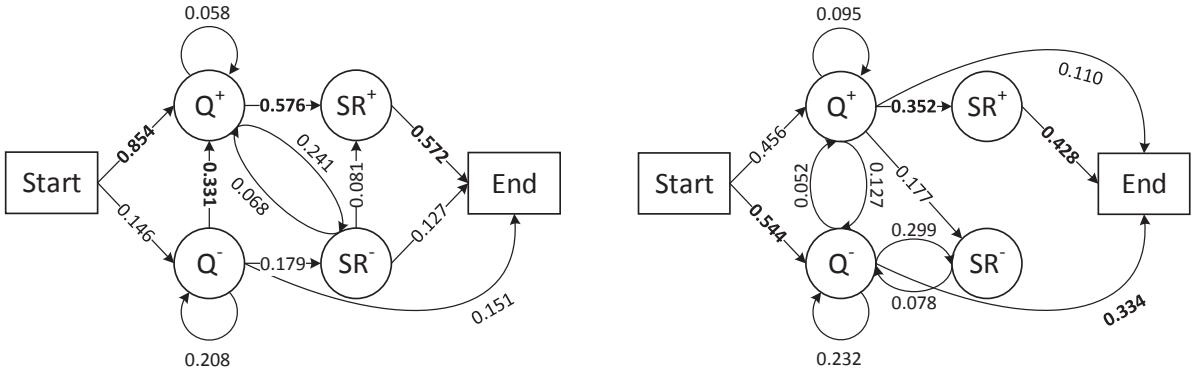
Analyzing users' search behavior patterns is important, since it helps us understand how people use search engines to solve search problems. Ageev et al. [1] analyzed search paths in different types of users and tasks, and identified distinct users' in-task behavioral patterns. In particular, they approximated the search paths based on the manually annotated search actions. However, such manual judgments are not generally available and expensive to acquire at scale. In contrast, our model is capable of performing such analysis of user search activities *without manual annotations*.

We performed this analysis on the toolbar data, where we do not have action-level annotations. The first-order transition probabilities between different action types with respect to the inferred action satisfaction labels are estimated. In Figure 3, we demonstrated two subgraphs of the identified search paths in satisfying and unsatisfying tasks. We ignored the edges with transition probability less than 0.05 and used bold font to highlight the outgoing edges with the highest transition probability from each node in the figure. Since we only showed a sub-graph from the original graph, the illustrated outgoing transition probabilities of some nodes may *not* sum up to one (e.g., $p(\text{SERP}^-|\text{SR}^-) = 0.558$ is not included in Figure 3(b)).

According to the search paths estimated from the inferred action satisfaction labels, users exhibit quite different behavior patterns in the satisfying and unsatisfying search tasks. In a satisfying task (as shown in Figure 3(a)), users usually start with a satisfying query ($p(\text{Q}^+|\text{START})$=0.854), which will very likely result in a satisfying click ($p(\text{SR}^+|\text{Q}^+)$=0.576); while in an unsatisfying task (as shown in Figure 3(b)), users are more likely to begin with an unsatisfying query ($p(\text{Q}^-|\text{START})$=0.544), and move to some unhelpful documents. An interesting search pattern we observed in the estimated search paths is that in a satisfying task, once users issue an unsatisfying query, they can quickly correct it and reformulate a satisfying one ($p(\text{Q}^+|\text{Q}^-)$=0.331); while in an unsatisfying task, users tend to get stuck in a sequence of unsatisfying queries ($p(\text{Q}^-|\text{Q}^-)$=0.232) and end up with a failed search task ($p(\text{END}|\text{Q}^-)$=0.334). These are examples of the types of insights that our automated method can yield, without having to apply expensive manual labeling.

6. CONCLUSIONS

In this work, we explicitly modeled searchers' satisfaction at the action level for search-task satisfaction prediction. A latent structural learning framework was developed to model the unobservable action-level satisfaction labels, which en-



(a) Search paths in satisfying tasks

(b) Search paths in unsatisfying tasks

Figure 3: Search paths estimated by the inferred action satisfaction labels from AcTS in toolbar data. Edges with transition probability less than 0.05 are discarded. Bold depicts highest outgoing transition probability.

abled us to explore rich structured features and dependency relations unique to search satisfaction modeling. Significant performance improvements in extensive experimental comparisons against several state-of-the-art search satisfaction models confirmed the value of modeling action-level satisfaction in search-task satisfaction prediction. Moreover, we demonstrated the clear benefit of the inferred action satisfaction labels in other search applications such as document relevance estimation and query suggestion.

As future work, we will investigate how to apply the developed framework for predicting search-task satisfaction in real time, action-by-action. If we can detect task failure early in the search process, search engines can adjust their ranking strategies or search support offered, before users abandon their searches. In addition, exploring the applications of action-level satisfaction labels in additional contexts is also an interesting direction to pursue.

7. REFERENCES

[1] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein. Find it if you can: a game for modeling different types of web search success using interaction data. In *SIGIR'11*, pages 345–354. ACM, 2011.

[2] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR'06*, pages 3–10. ACM, 2006.

[3] A. Aula, R. M. Khan, and Z. Guan. How does search behavior change as search becomes more difficult? In *CHI'10*, pages 35–44. ACM, 2010.

[4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*, volume 463. ACM Press New York, 1999.

[5] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *CIKM'08*, pages 609–618. ACM, 2008.

[6] C. Burges. From ranknet to lambdarank to lambdamart: An overview. *Microsoft Research Technical Report MSR-TR-2010-82*, 2010.

[7] M. Chang, D. Goldwasser, D. Roth, and V. Srikumar. Structured output learning with indirect supervision. In *ICML'10*, pages 199–206, 2010.

[8] H. T. Dang, D. Kelly, and J. J. Lin. Overview of the trec 2007 question answering track. In *TREC*, volume 7, 2007.

[9] G. Dupret and C. Liao. A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. In *WSDM'12*, pages 181–190. ACM, 2010.

[10] H. Feild and J. Allan. Task-aware query recommendation. In *SIGIR'13*, pages 83–92. ACM, 2013.

[11] H. A. Feild, J. Allan, and R. Jones. Predicting searcher frustration. In *SIGIR'10*, pages 34–41. ACM, 2010.

[12] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *TOIS*, 23(2):147–168, 2005.

[13] Q. Guo, R. W. White, Y. Zhang, B. Anderson, and S. T. Dumais. Why searchers switch: understanding and predicting engine switching rationales. In *SIGIR'11*, pages 335–344. ACM, 2011.

[14] A. Hassan. A semi-supervised approach to modeling web search satisfaction. In *SIGIR'12*, pages 275–284. ACM, 2012.

[15] A. Hassan, R. Jones, and K. L. Klinkner. Beyond dcg: User behavior as a predictor of a successful search. In *WSDM'10*, pages 221–230. ACM, 2010.

[16] A. Hassan, Y. Song, and L.-w. He. A task level metric for measuring web search satisfaction and its application on improving relevance estimation. In *CIKM'11*, pages 125–134. ACM, 2011.

[17] A. Hassan and R. W. White. Personalized models of search satisfaction. In *CIKM'13*, pages 2009–2018. ACM, 2013.

[18] S. B. Huffman and M. Hochster. How well does result relevance predict session satisfaction? In *SIGIR'07*, pages 567–574. ACM, 2007.

[19] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR'05*, pages 154–161. ACM, 2005.

[20] R. Jones and K. L. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *CIKM'08*, pages 699–708. ACM, 2008.

[21] Y. Kim, A. Hassan, R. W. White, and I. Zitouni. Modeling dwell time to predict click-level satisfaction. In *WSDM'14*, pages 193–202. ACM, 2014.

[22] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[23] Z. Liao, Y. Song, L.-w. He, and Y. Huang. Evaluating the effectiveness of search task trails. In *WWW'12*, pages 489–498. ACM, 2012.

[24] S. J. Lopez and C. R. Snyder. *The Oxford handbook of positive psychology*. Oxford University Press, 2011.

[25] R. Navarro-Prieto, M. Scaife, and Y. Rogers. Cognitive strategies in web searching. In *Human Factors & the Web*, pages 43–56, 1999.

[26] R. L. Oliver. *Satisfaction: A behavioral perspective on the consumer*. ME Sharpe, 2010.

[27] C. L. Smith and P. B. Kantor. User adaptation: good results from poor systems. In *SIGIR'08*, pages 147–154. ACM, 2008.

[28] Y. Song, D. Zhou, and L.-w. He. Query suggestion by constructing term-transition graphs. In *WSDM'12*, pages 353–362. ACM, 2012.

[29] H. Wang, Y. Song, M.-W. Chang, X. He, R. W. White, and W. Chu. Learning to extract cross-session search tasks. In *WWW'13*, pages 1353–1364. ACM, 2013.

[30] R. W. White and D. Morris. Investigating the querying and browsing behavior of advanced search engine users. In *SIGIR'07*, pages 255–262. ACM, 2007.

[31] Y. Xu and D. Mease. Evaluating web search using task completion time. In *SIGIR'09*, pages 676–677. ACM, 2009.