Beyond DCG: User Behavior as a Predictor of a Successful Search

Ahmed Hassan[*]
Department of EECS
U. Michigan Ann Arbor
Ann Arbor, MI
hassanam@umich.edu

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

Kristina Lisa Klinkner[†]
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213
klinkner@cmu.edu

ABSTRACT

Web search engines are traditionally evaluated in terms of the relevance of web pages to individual queries. However, relevance of web pages does not tell the complete picture, since an individual query may represent only a piece of the user's information need and users may have different information needs underlying the same queries. We address the problem of predicting user search goal success by modeling user behavior. We show empirically that user behavior alone can give an accurate picture of the success of the user's web search goals, without considering the relevance of the documents displayed. In fact, our experiments show that models using user behavior are *more* predictive of goal success than those using document relevance. We build novel sequence models incorporating time distributions for this task and our experiments show that the sequence and time distribution models are more accurate than static models based on user behavior, or predictions based on document relevance.

Categories and Subject Descriptors

H.3.5 [**Information Search and Retrieval**]: Online Information Services—*Web-based services*

General Terms

Algorithms,Experimentation,Measurement

Keywords

search engine evaluation, user satisfaction, user behavior models, query log analysis, search sessions

1. INTRODUCTION

Web search engines are traditionally evaluated in terms of the relevance of web pages to individual queries. However, users modify and reformulate their queries [22, 11] and can have complex information needs, so an individual query may represent only a piece of the user's information need or goal. In previous work it has been shown that relevance of web pages to queries is correlated with user search success [10]. However, relevance of web pages does not tell the complete picture, since queries can be ambiguous in many ways, and users can have different information needs underlying the same queries.

Consider two users searching for "free clip art" to embellish a document. They each see the same set of web search results for this query, which have the same fixed relevance or DCG [12]. The first user finds the kind of clip art she is looking for on the first page of results, clicks on it, copies it into her document and her search goal is successful. However, the second user has something else in mind. After seeing the results, he reformulates his query to "easter religious clip art", reformulates it again and ultimately abandons his search. This search goal is unsuccessful, despite starting with the same query, with the same DCG.

In this work, we look at ways to predict whether a user's particular search goal is successful. We consider features which take into account the entire pattern of user search behavior, including query, click and dwell-time as well as number of reformulations. It is notable that once trained, our techniques require no editorial judgments, and thus can be automated to add to standard metrics for search engine evaluation.

Our contributions include (1) a method of evaluating search engines that takes into account the entire pattern of the user experience during the multiple queries and clicks that span a search (2) a fully automated evaluation technique that does not require multiple impressions of each query-url pair (3) a model of usage behavior which integrates time in a natural way, which is highly predictive of user search goal success (4) empirical validation that this type of model is *more* predictive of user search goal success than editorial relevance judgments on the results for the first query in the session.

Our problem definition, data, and the labeling process we are trying to match are described in Section 2. Section 3 discusses the related work. Section 4 describes how user behavior is used as a predictor of a successful search. Experiments and comparison against other methods are presented in Section 5. In Section 6 we discuss results and give examples of typical successful and unsuccessful search patterns, as well as comparison of transition probabilities in successful and unsuccessful searches.

[*], [†]Work conducted while interning at Yahoo! Inc

WSDM'10, February 4–6, 2010, New York City, New York, USA.
Copyright 2010 ACM 978-1-60558-889-6/10/02 ...$10.00.

Time	Query	# Clicks	Avg. Dwell Time
t_1	sea bass in oven	1	Short
t_2	baked sea bass	1	Short
t_3	baked sea bass recipe	6	Long

Table 1: Example of a Successful Goal

Time	Query	# Clicks	Avg. Dwell Time
t_1	gauage mod for rfactor	0	NA
t_2	gauges for rfactor	1	Short
t_3	new gauges for rfactor	0	NA
t_4	gauges mod for rf	0	NA
t_5	new tacks for rfactor	1	Short
t_6	rfactor gauge plugin	0	NA

Table 2: Example of an Unsuccessful Goal

2. USER SEARCH GOAL SUCCESS

We consider a user's search sequence of one or more queries with the same atomic information need to be a goal, as defined by Jones and Klinkner [14]. When we examine the different actions performed by the user during the goal, we can distinguish *successful* and *unsuccessful* goals. Examples of a successful and an unsuccessful goal are shown in Table 1, and Table 2 respectively.

2.1 Data

Our data consists of a random sample of 1000 user sessions from the Yahoo! search engine engine during a week in April 2007. Each user session was a three days long, and included all queries, search result page impressions and clicks on all results on the search result page from that user in the timeframe. The three day time period was arbitrary, but deemed long enough to capture extended search patterns for some users. The editorial team were then instructed to examine each session and "re-enact" the user's experience.[1]

2.2 Editorial Guidelines

Definition 1. *A* search goal *is an atomic information need, resulting in one or more queries.*

A goal can be thought of as a group of related queries to accomplish a single discrete task. The queries need not be contiguous, but may be interleaved with queries from other goals (e.g., a user who is both looking for work-related information and information for the evening's entertainment).

The editors identified the goal of each query and labeled each query with a goal number. The data contained a total of 2712 distinct goals over approximately 5000 queries.

Success of a goal was judged on a five point scale: definitely successful, probably successful, unsure, probably unsuccessful, and definitely unsuccessful. The editors used information about landing page content and how well it matched query terms as well as the actual sequence of queries in a goal (e.g. whether the user seemed to be increasing in specification) as well as the click patterns on search results and suggestions such as spelling and related searches. Editors had a training period in which labels were discussed to ensure consistency.

Re-enacting the user's search in this way and attempting to estimate their success in finding their desired information is not without bias. The benefit to the editors of judging success at the session level is that they have a great deal of *context* to help them determine ambiguous cases. In the case of success judged for a single query, or pageview, the editor doesn't have this context at all, so interpreting the intent of the user becomes much more difficult.

For each transition from one-query to the next within a goal, the editors labeled that transition as a generalization, specialization, parallel move or same-meaning. For one query per user-session, the editors also labeled documents with relevance judgements on a five-point scale with the values (Perfect, Excellent, Good, Fair, Bad).

2.3 Problem Definition

Assume we have a stream of queries being submitted by users to a search engine. In response to each query, a Search Results Page (SERP) is displayed to the user. The search results page presents web search results, sponsored search results and other results like spelling suggestions and related searches. The user may then click on 0 or more results and then either end the session or submit another query.

Each user session may consist of one or more goals, where a goal is defined as an atomic information need that may result in one or more queries. So given a search goal, our objective is to predict whether that goal ended up being successful or not. In order to convert the editorial judgments to a binary classification task, we treated "definitely successful" and "probably successful" goals as the positive class, and all other goals as the negative class.

In this work we assume that the partitioning of sequences of queries and their associated clicks into goals has been carried out a pre-processing step. We use the editorially assigned goal labels for this. We could also use automatic goal boundary identification following the methods proposed in Jones and Klinkner [14] which report accuracy of around 92%. By using oracle editorial goal labels we can examine goal success prediction decoupled from the goal identification task.

3. RELATED WORK

Here we describe work on evaluating the quality of search results on a per-query basis, at the level of tasks and sessions, modeling user web usage patterns, and finally work on identifying session and goal boundaries.

3.1 Estimating Query Level Relevance

State of the art measurement of query result-set relevance for web search uses relevance metrics such as discounted cumulative gain (DCG) [12]. DCG can be calculated based on manual judgments of the relevance of documents in the search result list to individual queries, or estimated using models derived from user click behavior (eg. [4, 1, 5]. Query document-relevance judgments allow the data to be reused, and lend themselves to use as training data for ranking. The problem with this approach is that query-document relevance does not always mean user satisfaction. An individual query may represent only a piece of a user's information need. In addition, it has been shown that users can satisfy their information needs even with a poorly performing

[1] The user was designated with an anonymous identifier, and the annotation was done in accordance with Yahoo!'s privacy policy, with no information used to map the query stream to a particular user.

search engine, by exerting extra effort with query formulation and reformulation [23].

Piwowarski et al. [18] describe a model that uses Bayesian Networks to predict the relevance of a document in absence of document content models. Unlike previous work on query-url relevance, which require tens or hundreds of instances of a query-url pair to learn a relevance score, Piwowarski et al's work does allow the prediction of relevance even for queries issued only once or documents viewed only once. This modeling of user behavior patterns and times is close to our approach, but is modeled in an unsupervised fashion, and used to predict document relevance instead of user search goal success. As we will show in Section 5.2, even editorially labeled query url relevance is not as good a predictor of goal success as user behavior.

3.2 Task and Session Level Evaluation

Session Discounted Cumulative Gain (sDCG) [13] applies a discount to relevant results found in response to queries later in the user's search session. This takes into account the multiple queries that can be part of a search goal, but still requires manual relevance judgments. Our approach of predicting goal success using user actions both takes into account the entire process of the search goal, and once trained can also be fully automated to evaluate a search engine without requiring manual relevance judgments.

Huffman and Hochster [10] address a similar task to ours. The purpose of their study is to look into the correlation between user satisfaction and simple relevance metrics. They report a strong correlation between the two attributes and construct a model to predict user satisfaction using the relevance of the first three results and the query type. We will show in Section 5.2 that our trained Markov model of user behavior outperforms their model using editorial judgments. Xu and Mease [24] show that total task completion time is correlated with user satisfaction for difficult tasks, and that variation in time across users is greater than within users. We model the total time span of a task in our experiments and see that for our user tasks, which are randomly selected from real online user activity and span a variety of difficulty levels, our Markov model incorporating detail of transitions and time provides significant improvements. Downey et al [6] propose models of the sequence of actions using Bayesian networks, and include both actions and time. Our work differs in that we represent time as a probability distribution, and use the models to predict user search goal success. Fox et al [7] is the work most similar to ours. They attempt to predict user-annotated levels of satisfaction using a variety of implicit measures based on search behavior. They propose *gene patterns* to summarize the sequences of user behavior. Our work generalizes these patterns by representing sequences in a Markov model, allowing representation of transition probabilities, as well as time distributions over the transitions.

Radlinski et al. [20] look at predicting relative search engine performance using metrics including abandonment rate, reformulation rate and time to first click, and find that these metrics do not perform as well as interleaving on their small search engine dataset. The metrics they consider aggregate over all users of a search engine and do not consider the individual search goal. In Section 5.1 we consider analogous features including number of queries in a goal and time to first click as part of a list of static features, and show that our Markov model of user actions out-performs these at predicting user goal success. Our fine-grained prediction of goal success allows us both to evaluate search engine performance at a finer-grained level, for individual or groups of users or starting queries, as well as allowing us to compare pairs of search engine in terms of goal success rate and measures of user effort such as number of queries per successful goal.

Jung et al. [15] show that considering the last click of a session may be the most important piece of information in relating user clicks to document relevance. We consider this feature as one of our predictors in section 5.1 and again show that it is not as good as the Markov model in predicting goal success. The dwell time of the last click of a session is represented as part of our time-based Markov model described in Section 4.5.

3.3 Modeling User Search Behavior

Boldi et al. use a query-flow graph, a graph representation of query reformulations in query logs [2]. They use this model for finding logical session boundaries and query recommendation. Agichtein et al. show that incorporating user behavior data can significantly improve ordering of top results in real web search setting [1]. Borges and Levene model the user navigational patterns as a probabilistic grammar [3]. They use an N-gram model where the next page visited by the user is affected by the last N pages browsed. Sadagopan and Li find atypical user sessions by detecting outliers using Mahanalobis distance in the user session space [21]. These papers concentrate on using the query logs as a data source for learning about the world or to improve the search engine, whereas we focus on evaluating the success of the user search goal itself.

3.4 Identifying Search Task Boundaries

The problem of classifying the boundaries of the user search tasks within sessions in web search logs has been widely addressed before. This task is an important preprocessing step for out work as we need to find the boundaries between goals before we can predict whether they are successful or not. Early models used time and word and character overlap (eg. [17, 19]), but were on small data sets or did not compare to ground-truth. Jones and Klinkner address the problem of classifying the boundaries of the goals and missions [14], and a similar problem has been addressed by Boldi et al [2]. For the experiments in this paper we use editorially labeled goal boundaries, but we could substitute the automatic methods described in these papers which have been reported at around 92%.

4. APPROACH

In this section, we describe a model which, given a search goal, predicts whether it is successful or not. To do this we extract patterns describing the user behavior for each search goal. We then use those patterns to build two Markov Models, describing user behavior in case of successful, and unsuccessful, search goals. Given a new search goal, we extract the corresponding pattern and estimate the likelihood of this pattern being generated from the two models. We then compare the likelihood of the test goal under the two models to decide if it is a successful or unsuccessful goal. This model does not take time between user actions into consideration. In Section 4.5, we describe a method for combining time with user behavior to better predict goal success.

4.1 Goals as a Sequence of Actions

Each goal consists of a set of queries and zero or more clicks on search results for each query. A user search goal can be also represented by an ordered sequence of user actions along with the time between those actions.

Given a set of actions $a_1..a_n$, a goal can be defined as:

$$G = \langle START, \langle a_1, t_1 \rangle, \ldots, \langle a_n, t_n \rangle, END \rangle$$

where $START$, and END are the start and end states respectively. $a_1, \ldots, a_n \in \mathbb{A} = \{Q, SR, AD, RL, SP, SC, OTH\}$ is the possible set of user actions. $t_1, \ldots, t_n \in \mathbb{N}$ is the time between actions.

The following types of actions could appear in a goal:

- START: the user starts a new goal (manually labeled in our data)
- A query (Q)
- A click of any of the following types:
  - Algorithmic Search Click (SR).
  - Sponsored Search Click (AD).
  - Related Search Click (RL).
  - Spelling Suggestion Click (SP).
  - Shortcut Click (SC).
  - Any Other Click (OTH), such as a click on one of the tabs.
- END: the user ends the search goal (manually labeled in our data)

Most of the action types are self explanatory except for the related search and the shortcut clciks. A related search click is a click on a query similar to the user's query. The search engine lists similar queries when other people have done searches similar to the user's search. A shortcut is a quick way to get to the information the user wants. It automatically appears when it is relevant to the user's search. Some example of shortcuts are images, videos, news,...etc.

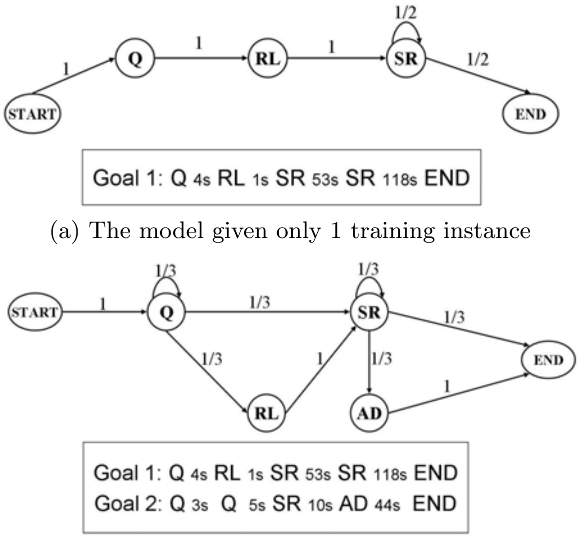
Incorporating actions like related search and shortcut clicks could be very useful for assessing the utility of such important search engine features by looking at how their usage correlates with the overall goal success.

Consider the following example: A user enters the query "guess", then 4 seconds later he clicks on the related search suggestion "guess watches", after one more second, the user clicks the first search results, after another 53 seconds, the user clicks on the third result and after 118 seconds, the goal ends. This user goal can be represented by the following sequence of actions: Q $_{4s}$ RL $_{1s}$ SR $_{53s}$ SR $_{118s}$ END

4.2 Model Language Variations

Some actions can also be associated with a number. For example, Q could be replaced by $\{Q\} \times \mathbb{N}$ to distinguish the first query from the second query and so on. This can be defined at several levels of granularity. For example, we may decide to only distinguish the first query from the rest. Similarly, we can do the same by replacing SR with $\{SR\} \times \mathbb{N}$ to distinguish clicks on different positions. The number could represent the result position, the number of the page at which it appeared or any other custom definition.

Given a higher-order model, the example given above would be represented by the following sequence of actions, which distinguishes the click on the first-ranked search result from the click on the third-ranked one: Q $_{4s}$ RL $_{1s}$ SR_1 $_{53s}$ SR_3 $_{118s}$ END


1/2
START
1
Q
1
RL
1
SR
1/2
END
Goal 1: Q 4s RL 1s SR 53s SR 118s END
1/3
1/3
START
1
Q
1/3
SR
1/3
END
1/3
1
1/3
1
RL
AD
Goal 1: Q 4s RL 1s SR 53s SR 118s END
Goal 2: Q 3s Q 5s SR 10s AD 44s END

(a) The model given only 1 training instance

(b) The model given only 2 training instances

Figure 1: Sequences of actions could represent a path in a graph

4.3 Building the Model

Each sequence of actions represents a chain or a path in a graph. The sequence of actions from the previous example can be represented as a path in a graph as shown in Figure 1(a). As we have more sequences representing more goals, the graph could evolve as shown in Figure 1(b).

This graph could be defined as $G = (V, E, w)$ where:

- $V = \{Q, SR, AD, RL, SP, SC, OTH\}$ is the set of possible user actions during the goal.
- $E \subseteq V \times V$ is the set of possible transitions between any two actions.
- $w : E \to [0..1]$ is a weighting function that assigns to every pair of states (s_i, s_j) a weight $w(s_i, s_j)$ representing the probability that we have a transition from state s_i to state s_j.

This graph simply represents a Markovian model of the user behavior during goals. The state space of the Markov model is the set of actions and the transition probability between any two states s_i, and s_j is estimated using Maximum Likelihood estimation as follows:

$$Pr(s_i, s_j) = \frac{N_{s_i,s_j}}{N_{s_i}}$$

where N_{s_i,s_j} is the number of times we saw a transition from state s_i to state s_j, and N_{s_i} is the total number of times we saw state s_i in the training data.

4.4 Predicting Goal Success

We split our training data into two splits; the first containing all successful goals and the second containing all unsuccessful goals. Given the methodology described in the previous section, we build two Markov models. The first model M_s characterizes the user behavior in successful goals, and the second model M_f characterizes the user behavior in unsuccessful goals.

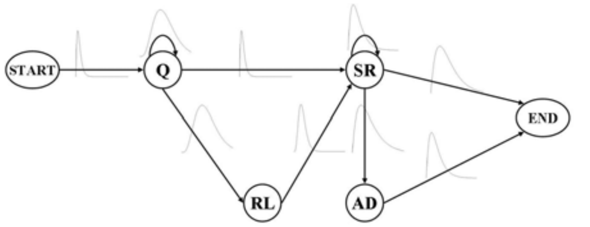


Figure 2: Time distributions are estimated for each transition

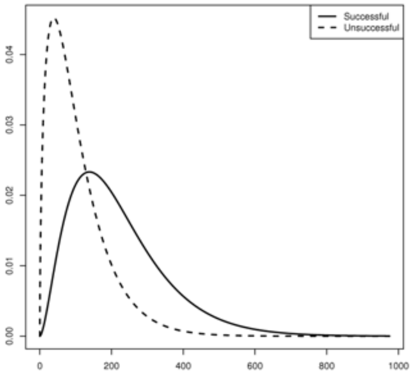


Figure 3: Time distributions of $SR \rightarrow Q$ transitions for successful and unsuccessful search goals.

Given a new user goal, we can use the two models to estimate the log likelihood that this action sequence was generated from both models. Given a model M, and sequence of actions $S = (S_1, S_2, \ldots, S_n)$, the probability of this action sequence being generated from M is:

$$Pr_M(S) = \prod_{i=2}^{n} Pr(S_i|S_1, \ldots, S_{i-1}) = \prod_{i=2}^{n} W(S_{i-1}, S_i)$$

where n is the number of actions in the sequence, and W is the probability transition function.

The log likelihood is then defined as:

$$LL_M(S) = \sum_{i=2}^{n} W(S_{i-1}, S_i)$$

and goal success is defined as:

$$Pred(S) = \begin{cases} 1 & \text{if } \frac{LL_{M_s}(S)}{LL_{M_f}(S)} > \tau, \\ 0 & \text{otherwise.} \end{cases}$$

where S is the goal's sequence of actions, $LL_{M_s}(S)$ is the log likelihood of the goal given the success model, $LL_{M_f}(S)$ is the log likelihood of the goal given the failure model and τ is a threshold that is usually set to 1.

4.5 Adding Time to the Model

So far our model does not take transition times into consideration. Time between actions is a very important predictor of success. For example, it is widely believed that long dwell time of clicks is an important predictor of success. It has also been shown that the time to first click is correlated with search success [20].

We assume that there is a distinctive distribution that governs the amount of time the user spends at each transition. The distribution governs how much time the user spends at state S_i before moving to state S_j for each possible transition $S_i \rightarrow S_j$. The distribution at each transition is estimated from the training data. We collect all transition times for all goals from the training set for each transition and use them to estimate the time distribution for that transition as shown in Figure 2.

The first step is selecting the parametric form of the time distributions. The gamma distribution is a rich two parameter family of continuous distributions. It has a scale parameter θ, and a shape parameter k. If k is an integer, the distribution represents the sum of k independent exponentially distributed random variables [9]. The gamma distribution is frequently used as a probability model for waiting times [9].

The probability density function of the gamma distribution can be expressed as:

$$f(x; k; \theta) = x^{k-1} \frac{e^{-x/\theta}}{\theta^k \Gamma(k)} \text{ for } x, k, \theta > 0 \qquad (1)$$

Given N independent and identically distributed observations $(x_1, \ldots, x_N)$ for the transition times between two states S_i, and S_j,, the likelihood function is:

$$L(k, \theta) = \prod_{i=1}^{N} f(x_i; k, \theta) \qquad (2)$$

Substituting 1 in 2, and finding the maximum with respect to θ, we get:

$$\hat{\theta} = \frac{1}{kN} \sum_{i=1}^{N} x_i$$

Finding the maximum with respect to k, we get:

$$\ln(k) - \psi(k) \approx \frac{1}{k} \left(\frac{1}{2} + \frac{1}{12k + 2} \right) \hat{\theta} = \frac{1}{kN} \sum_{i=1}^{N} x_i$$

which can be solved numerically.

We again split our training data into two splits; the first containing all successful goals and the second containing all unsuccessful goals. We then estimate the gamma distributions parameters for every transition once in each model. Given a new goal, we estimate the likelihood that the transition times have been generated from the success model and the likelihood that they have been generated from the failure model. The ratio of the two likelihoods can then be used as feature, along with the the likelihood ratio from the sequence models, to predict success. Our hypothesis is that some transitions will have different time distributions for success and failure models. Hence the ratio of the likelihood of the transition times could be used as a predictor of success. Figure 3 compares the estimated time distributions for the transition time between a search result click and a query submission. We see from the figure that users tend to spend more time on search results in successful goals which agrees with previous research that shows that long dwell time on a click is an indicator of a good click. We contrast this with the short times users tend to spend on search results in the case of unsuccessful goals where they quickly go back and rewrite the query.

5. EXPERIMENTS

Our evaluation data consisted of 2712 goals obtained from a commercial search engine's query log. Human editors were instructed to classify goals as either successful or not as described in Section 2.2. We used Gradient Boosted Decision Trees (GBDT) as a classifier [8]. We used 10 fold cross validation for all tests. We evaluate our results in terms of *Precision*, *Recall*, *F-measure*, and *Accuracy*. Statistical significance was tested using a 2-tailed paired t-test.

We compare our Markov model method to several other methods of goal success prediction. The first baseline poses the problem as a classic machine learning problem where a set of static features are used to predict success (Section 5.1). The second baseline uses query-url relevance (DCG), similar to [10], to predict goal success (Section 5.2). We show that the Markov model out-performs both of these baselines.

5.1 Static Features Based on User Search Behavior

Our first baseline poses the problem as a classic machine learning problem where we come up with a set of features and train a classifier using them. We tested a number of features, and those which performed best are described here:

Features
Number of queries during goal
Number of clicks during goal
Number of clicks on sponsored results during goal
Number of clicks on next page during goal
Number of clicks on spell suggestion during goal
Number of clicks on also try during goal
Number of clicks on shortcut during goal
Maximum time between clicks during goal
Minimum time between clicks during goal
Average time between clicks during goal
Time span of goal
Average time to first click during goal
Average dwell time

An important feature that we consider is the dwell time. Dwell time of a click is the amount of time between the click and the next action (query, click, or end). We calculate the dwell times for all clicks during goal and use the maximum, minimum, and average dwell times as features to predict success.

Figure 4 compares the precision-recall curves for the static features classifier and the proposed Markov model likelihood based method (which we will refer to simply as "MML"). Table 3 shows the precision, recall, f-measure, and accuracy for the static features classifier, the dwell time classifier and the proposed method. Thresholds were set by the gradient boosted decision tree classifier. We notice that the dwell time features are doing well compared to the static features. The performance we get by using them is comparable to that of using all static features. We also notice that the Markov model method significantly outperforms the static features and the dwell time classifiers. All measures are considerably improved. The accuracy improved by more than 6 points or 9% of the static feature classifer accuracy.

The Markov model action sequence model has several advantages over the static features classifier. Both models try to describe the behavior of users during search. However, the static features classifier uses aggregated features describing the user behavior collectively. While, this is simple and easy to compute, it ignores a lot of the details inherent in the user behavior. On the other hand, the Markov model action sequence approach gives a more accurate picture of the user behavior. For example, the static features classifier uses the number of clicks of different types per goal to predict success. While this could be helpful, it ignores the characteristics of each click and the transitions to/from the clicks. For example, a search click followed by a query rewrite might be quite different from a search click followed by another search click.

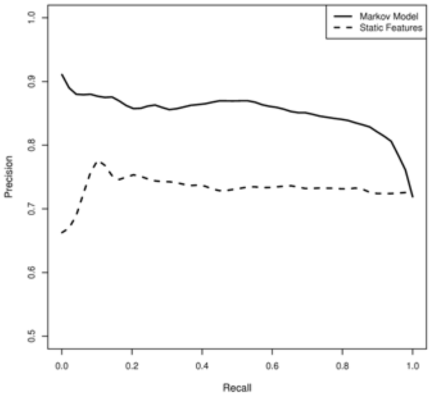


Figure 4: Precision-Recall Curves for Markov Model Likelihood (MML) and static features classifiers.

	Precision	Recall	F1	Accuracy
Static Features	78.0	89.5	83.3	74.1
Dwell Time	76.1	93.3	83.8	73.2
Markov Model (MML)	83.5	91.8	87.5	80.4
MML+Static	81.7	93.3	86.5	79.9
MML+Click Pos	84.2	92.2	88.0	81.5
MML+Time	84.2	93.4	88.6	**82.1**
MML+Click Pos+Time	83.6	94.4	88.7	**82.2**

Table 3: Precision, Recall, F1, and Accuracy for Static Features, Markov Model, and Markov Model + Time Classifiers. Each set of results separated by horizontal bars has statistically significantly higher accuracy than the set above it.

We also tried to use the the score we get from the MML as a feature, adding it to the static features. The performance of that classifier is shown in Table 3. We see that we did not get any improvement when we added the static feature to the MML feature. We believe that modeling the user behavior captures all the information captured by the static features and more.

5.2 Relevance based Prediction

Huffman and Hochster [10] show that there is a reasonably strong correlation between user satisfaction and relevance metrics. In this section, we compare our approach to a method based on [10]. They use the relevance of the first query in a goal to predict search success. They use a simple position-weighted mean which comes from the discounted cumulative gain (DCG) [12] family of measures. Given relevance judgments on a five-point scale, they scale them to lie between 0 and 1 and define their aggregate measure as:

$$Relevance = \frac{R_{pos1} + R_{pos2}/2 + R_{pos3}/3}{1 + \frac{1}{2} + \frac{1}{3}} \qquad (3)$$

where R_{pos1},R_{pos2}, and R_{pos3} are the relevance judgments between 0 and 1 for the first three results.

We implemented DCG in this fashion in order to compare to their results. Other more standard implementations of DCG use a log weighting on rank, relevance weightings which weight "Perfect" and "Excellent" documents more highly, and can incorporate results to arbitrary ranks, though DCG at ranks 1, 3 5 and 10 are all commonly used. We also implemented DCG in a more standard form as described in [12]:

$$DCG_p = rel_1 + \sum_{i=2} p \frac{rel_i}{\log i}$$

where rel_i is the relevance of result at position i on a five-point scale.

We compare the performance of our method to the relevance (DCG) based classifier using 10 fold cross-validation on a subset of the data for which we have query-url relevance judgments by human editors - which had 607 goals. Figure 5 compares the precision-recall curves for the relevance (DCG) based classifier and the action sequence Markov model likelihood based method. Table 4 shows the precision, recall, f-measure, and accuracy for the MML method and the two relevance classifiers, based on Equation 3 and DCG [12]. It also shows the same measures for a classifier combining the MML method and DCG. We notice that the two relevance based metrics perform pretty much the same. The different is not statistically significant. We also notice that the Markov model method outperforms the relevance based method at all operating points on the precision-recall curve except for the case of very low recall. Precision is improved by almost 6 points, however recall decreased by a single point. The accuracy improved by by more than 5 points or 6.5% of the relevance based classifer accuracy. When we combined the MML method with DCG we did not see any statistically significant improvement.

Although relevance is a good predictor of success, it does not tell the complete picture, since an individual query may represent only a piece of the user's information need and users may have different information needs underlying the same queries. On the other hand, directly modeling user behavior is a better predictor of success because it is based on personal assessment of the utility of the returned results from the user prospective.

We had a closer look at the goals where the relevance based classifier failed, and the Markov model approach succeeded at accurately predicting whether the goal was successful or not. In the first example, the user entered the query "furniture auctions". The relevance judgments of the first three results are "Excellent", "Good", and "Good" respectively, for "furniture auctions" for people wanting to buy furniture. From the relevance point of view, the goal seemed to be successful. However, the user was actually intending to *sell* furniture in auctions; rewriting the query to "sell to furniture auction" to reflect this intent and the user ended up failing to find what he wanted. On the other hand, the behavior of the user conformed to the behavior characterized by the user failure model and the ratio of the likelihood of his actions sequence being generated from the success and failure models clearly indicated that he was not successful.

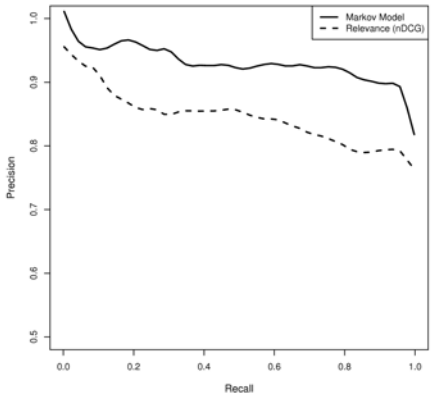


Figure 5: Precision-Recall Curves for Markov Model Likelihood (MML) and Relevance (DCG) based Prediction.

	Precision	Recall	F1	Accuracy
Relevance (Eqn. 3)	84.2	93.3	88.4	80.2
DCG	84.7	91.0	87.6	79.1
Markov Model	89.8	92.3	91.1	**85.2**
Markov Model + DCG	88.7	94.1	91.3	**85.4**

Table 4: Precision, Recall, F1, and Accuracy for Relevance(Eqn. 3), DCG, Markov Model and Markov Model + DCG classifiers, cross-validated on the subset of 607 goals for which we have query-url relevance judgments.

A second example illustrates a case where the relevance of the first 3 results is not quite indicative of the search success. In this example, the user entered the query "homes for sale by owner in Pembroke Pines". The relevance judgments of the first three results are "Bad", "Bad", and "Fair" respectively which results in the goal being labeled as unsuccessful by the relevance based model. However, this prediction is incorrect because the user ended up finding what he was looking in the result at position 11. If you increase the number of results included in the relevance metric to cover result number 10, it will not have a significant effect on the overall metric. On the other hand, the goal was correctly predicted as successful by the behavior model as the user behavior greatly conformed to the behavior characterized by the success model.

5.3 Adding Time

The experiments we have described so far only use the sequence of user actions without taking time into consideration. We fit a gamma time distribution for each transition for both the successful and the unsuccessful models as described in Section 4.5.

Figure 6 compares the precision-recall curves for the Markov model likelihood based method with and without time. Table 3 shows the precision, recall, f-measure, and accuracy for the two methods. We notice that adding time improves precision, recall, and accuracy. The gain in accuracy we get from adding time to our model is around 2%. We see that using time alone is surprisingly good (Figure 6). It is

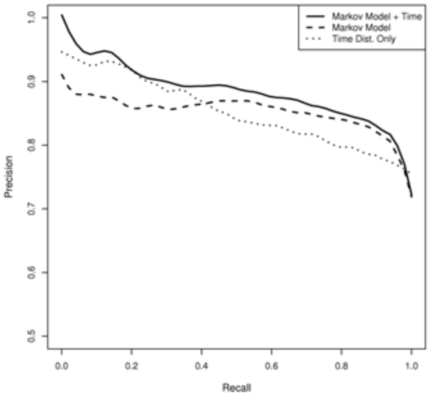


Figure 6: Precision-Recall Curves for Markov Model Likelihood (MML) and Markov Model Likelihood with Time.

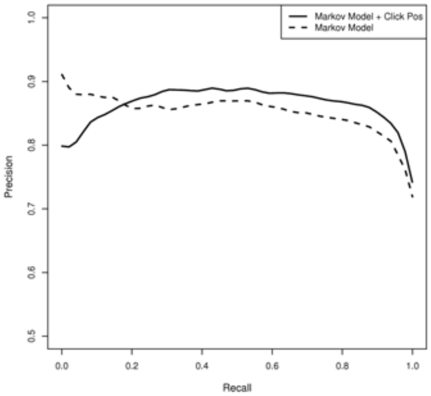


Figure 7: Precision-Recall Curves for Markov Model Likelihood (MML) and Markov Model Likelihood with click position included in the model language.

much better than using the static features baseline described above. We also see that using time alone is even better than the action sequence MML model in the low recall condition. However, the Markov action sequence model is better for moderate and high values of recall. Combining both models gives us a classifier that is better than both models.

5.4 Other Experiments

In this section we describe several other experiments that we performed. Some of those experiments yielded negative results, some yielded positive results and the rest left the performance intact.

In the first set of experiments, we varied the Markov model language. First we replaced the single query state Q with several other states to distinguish the first query from the second query and so on. The difference in performance due to this change was not statistically significant. Another hypothesis is that the different types of queries transitions might have different meanings for distinguishing successful from unsuccessful goals. Hence, we distinguished the starting query from a query rewrite where the user rewrites the query to generalize, specialize, use a parallel move or same-meaning (using the editorial labeling described in Section 2.2), for a total of five query states. Again, the difference in performance due to this change was not statistically significant. We also tried to replace the the single search result click state with several other states to distinguish clicks at different positions. We tried different variations of this modification. We tried to add the click position or the page position for different page sizes. We get the best performance when we model the rank of the clicked results in increments of 5. For example, a click on one of the first 5 results will be mapped to the state SR_{1-5}. A click on one of the results at positions from 6 to 10 will be mapped to the state SR_{6-10} and so on. Figure 7 compares the precision-recall curves for the proposed Markov model likelihood based method with and without search result positions. We get a slight improvement in performance due to this modification. We believe this happens because the position of the clicked result is a proxy for result relevance, and user behavior could change after a search result click according to whether the result was relevant or not.

First order Markov models assume that the next state is only dependent on the present state. To validate this appropriateness of this assumption on our data, we trained higher order Markov models on the data and compared the performance to first order models. We trained second order and third order Markov models and the performance ended up being degraded

In general there is a trade-off between the complexity of the model and the amount of data available for training. For the amount of training data we have right now, it seems that first order models are the best fit. We even tried to train a second order model with back off to first order but the difference in performance was statistically insignificant. It would be very interesting to increase the size of the training dataset and observe the effect of this increase on the performance of the higher order models.

We also tried to add query and query-url related features to the model. We used 6 months of query log data and calculated the number of impressions for each query. We also calculated, for each query-url pair, the number of times the url received the first and last click, the first but not last click, the last but not first click, or neither the first nor the last click. We added those features to the Markov model likelihood ratio feature described earlier but we did not see any statistically significant improvement.

6. DISCUSSION

It is interesting to observe the transition probabilities learned in the Markov models for successful and unsuccessful goals. In Table 5 we see the odds ratio of transition probabilities from query to other actions in the successful goals compared to unsuccessful goals. We see that in successful goals users are twice as likely to click on shortcuts, and nearly twice as likely to click on a search result. Users are more likely to click on spelling suggestions and to issue a new query without clicking on any search results in unsuccessful goals. Unsuccessful goals are ten times as likely to end with no clicks of any kind, that is with a transition from the query to the end-state, which is also called an *abandoned query.* Abandonment alone is not a perfect predictor of document

Action following query	Odds-ratio
SC	2.0
SR	1.8
RL	1.2
SP	0.9
Q	0.5
OTH	0.3
END	0.1

Table 5: Odds-ratio of transitions from query to other actions in successful goals, compared to unsuccessful goals.

Action leading to end	Odds-ratio
SR	1.5
SC	1.2
OTH	1.0
RL	0.7
Q	0.1

Table 6: Odds-ratio of transitions to end from other actions in successful goals, compared to unsuccessful goals.

relevance or goal success, since many queries can be satisfied with document snippets [16], but it is useful information as part of a model over the entire search goal.

In Table 6 we see the odds-ratio of transition probabilities to the end state in the success model compared to the failure model. Successful goals are much more likely to transition from search result clicks to the end state, while unsuccessful goals are much more likely to transition from a query or a related searches to the end state.

We can gain additional insights by looking at likely sequences from the Markov model. Table 7 shows some of the most probable paths through the Markov model for successful goals. We see that a single query followed by one, two or three clicks is very likely to be successful. A reformulated query followed by two clicks is also very likely to be successful. In Table 8 we see highly probable paths through the Markov model for unsuccessful goals. A query or a reformulated query with no subsequent clicks is very likely to be unsuccessful.

One important question is how much data is required to train models of user search goal success. We constructed a learning curve, shown in Figure 8, by fixing the test set size at one tenth of the data, and varying the training set size. We carried out ten-fold cross validation as with our previous experiments. We see that adding more data continues to increase the accuracy, and that accuracy is quite sensitive to the training data. This suggests that adding more data to this model could lead to even better results.

Our Markov model parameters are learned from our data. As we can see from the strong cross-validation performance, it generalizes well across diverse users and search goals. However, we tested this with only a single search engine interface. Different interface design and distribution of result types could lead to different transition probabilities and time distributions, which may require retraining of the model. A question for further investigation is which elements are invariant to such changes and which are sensitive to it.

The classification of goal success we have described here is part of a bigger picture of measuring and improving user satisfaction with search. When we can successfully automate the prediction of goal success, we can couple that with measures of user effort to come up with an overall picture of user satisfaction.

Highly probable successful paths
Q SR END
Q SR SR END
Q SR SR SR END
Q SR SR SR SR END
Q AD END
Q SC END
Q SR Q SR SR END

Table 7: Some of the highly probable successful paths.

Highly probable unsuccessful paths
Q END
Q Q END
Q OTH END
Q SR Q END
Q Q Q END
Q RL END
Q Q SR Q SR Q END

Table 8: Some of the highly probable unsuccessful paths.

7. CONCLUSIONS

We have shown that training a supervised Markov model of user behavior including the sequence of all queries and clicks in a user search goal as well as the times between actions allows us to predict the user's success at that goal. This model is more accurate than predictions based on query-url relevance as quantified in DCG. A search-goal-success based evaluation naturally incorporates query ambiguity and diverse user intents underlying the same query strings, which are conflated by traditional DCG measurements. Thus automated evaluation of search engines with search goal success may be the path to improving search engines beyond a plateau of optimizing relevance for the "generic user". Coupling goal success measurements with measures of user effort (for example query formulation effort and measures of time spent reading) will give us a complete picture of user web search satisfaction.

8. ACKNOWLEDGMENTS

Thanks to Tony Thrall and Isabella Barbier for designing the data sampling and support of this work.

9. REFERENCES

[1] E. Agichtein, E. Brill, and S. T. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR 2006: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, New York, NY, USA, 2006. ACM.

[2] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *Proceeding of the 17th ACM conference on Information and knowledge management (CIKM 2008)*, pages 609–618, 2008.

[3] J. Borges and M. Levene. Data mining of user navigation patterns. In *WEBKDD*, pages 92–111, 1999.

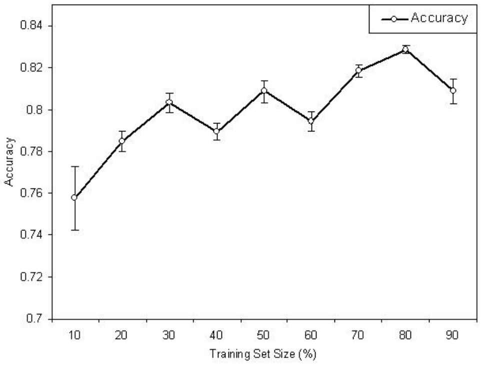


Figure 8: Accuracy Learning Curve for Markov Model Likelihood (MML).

[4] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In *Proceedings of Twenty-First Annual Conference on Neural Information Processing Systems (NIPS 2007)*, 2007.

[5] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In J. Quemada, G. León, Y. S. Maarek, and W. Nejdl, editors, *WWW*, pages 1–10. ACM, 2009.

[6] D. Downey, S. Dumais, and E. Horvitz. Models of searching and browsing: Languages, studies, and applications. *Journal of the American Society for Information Science and Technology (JASIST)*, 58(6):862–871, 2007.

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

[8] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2001.

[9] R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics*. Macmillan, New York, 4th edition edition, 1978.

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

[11] B. J. Jansen, M. Zhang, and A. Spink. Patterns and transitions of query reformulation during web searching. *IJWIS*, 3(4):328–340, 2007.

[12] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[13] K. Järvelin, S. L. Price, L. M. L. Delcambre, and M. L. Nielsen. Discounted cumulated gain based evaluation of multiple-query IR sessions. In C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. W. White, editors, *ECIR*, volume 4956 of *Lecture Notes in Computer Science*, pages 4–15. Springer, 2008.

[14] R. Jones and K. L. Klinkner. Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs. In *Proceedings of ACM 17th Conference on Information and Knowledge Management (CIKM 2008)*, 2008.

[15] S. Jung, J. L. Herlocker, and J. Webster. Click data as implicit relevance feedback in web search. *Information Processing and Management (IPM)*, 43(3):791–807, 2007.

[16] J. Li, S. Huffman, and A. Tokuda. Good abandonment in mobile and pc internet search. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, New York, NY, USA, 2009. ACM.

[17] S. Ozmutlu. Automatic new topic identification using multiple linear regression. *Information Processing and Management*, 42(4):934–950, 2006.

[18] B. Piwowarski, G. Dupret, and R. Jones. Mining user web search activity with layered bayesian networks or how to capture a click in its context. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, 2009.

[19] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In R. Grossman, R. Bayardo, and K. P. Bennett, editors, *KDD*, pages 239–248. ACM, 2005.

[20] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K.-S. Choi, and A. Chowdhury, editors, *CIKM*, pages 43–52. ACM, 2008.

[21] N. Sadagopan and J. Li. Characterizing typical and atypical user sessions in clickstreams. In *Proceedings of the Seventeenth International Conference on the World-Wide Web (WWW08)*, 2008.

[22] C. Silverstein, M. R. Henzinger, H. Marais, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[23] A. Turpin and F. Scholer. User performance versus precision measures for simple search tasks. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–18, New York, NY, USA, 2006. ACM.

[24] Y. Xu and D. Mease. Evaluating web search using task completion time. In J. Allan, J. A. Aslam, M. Sanderson, C. Zhai, and J. Zobel, editors, *SIGIR*, pages 676–677. ACM, 2009.