Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs

Rosie Jones
Yahoo! Research
3333 Empire Ave
Burbank, CA 91504
jonesr@yahoo-inc.com

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

ABSTRACT

Most analysis of web search relevance and performance takes a single query as the unit of search engine interaction. When studies attempt to group queries together by task or session, a timeout is typically used to identify the boundary. However, users query search engines in order to accomplish tasks at a variety of granularities, issuing multiple queries as they attempt to accomplish tasks. In this work we study real sessions manually labeled into hierarchical tasks, and show that timeouts, whatever their length, are of limited utility in identifying task boundaries, achieving a maximum precision of only 70%. We report on properties of this search task hierarchy, as seen in a random sample of user interactions from a major web search engine's log, annotated by human editors, learning that 17% of tasks are interleaved, and 20% are hierarchically organized. No previous work has analyzed or addressed automatic identification of interleaved and hierarchically organized search tasks. We propose and evaluate a method for the automated segmentation of users' query streams into hierarchical units. Our classifiers can improve on timeout segmentation, as well as other previously published approaches, bringing the accuracy up to 92% for identifying fine-grained task boundaries, and 89-97% for identifying pairs of queries from the same task when tasks are interleaved hierarchically. This is the first work to identify, measure and automatically segment sequences of user queries into their hierarchical structure. The ability to perform this kind of segmentation paves the way for evaluating search engines in terms of user task completion.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Query formulation

---

[*]This work was conducted while this author was at Yahoo! Inc

General Terms

Algorithms, Experimentation, Measurement

Keywords

query log segmentation, query session, query session boundary detection, search goal

1. INTRODUCTION

Web search engines attempt to satisfy users' information needs by ranking web pages with respect to queries. But the reality of web search is that it is often a process of querying, learning, and reformulating. A series of interactions between user and search engine can be necessary to satisfy a single information need [18].

To understand the way users accomplish tasks and subtasks using multiple search queries, we exhaustively annotated 3-day long query sequences for 312 web searchers. We limited the duration to three days to allow complete annotation of every query sequence, with an extremely thorough approach. These spans of time allowed us to identify tasks which result in queries placed over multiple days, as well as multiple tasks which may occur over several days. We manually annotated these query sequences with tasks and subtasks (which we will define as *missions* and *goals*), finding that many tasks contained subtasks, and many different tasks and subtasks were interleaved. While previous work has examined the way users interleave tasks [9], no previous work has examined the way tasks contain subtasks.

If we are able to accurately identify sets of queries with the same (or related) information-seeking intent, then we will be in a better position to evaluate the performance of a web search engine from the user's point of view. For example, standard metrics of user involvement with a search engine or portal emphasize *visits* or *time spent* [1]. However, each page view can constitute small pieces of the same information need and each visit could encompass some larger task. If we could instead quantify the number of information needs or tasks which a user addresses via a website, we would have a clearer picture of the importance of the site to that user. In particular, we could evaluate user effort in terms of queries issued or time spent on a task, as the user attempts to satisfy an information need or fulfill a more complex objective.

To this end, we built classifiers to identify task and subtasks boundaries, as well as pairs of queries which correspond to the same task, despite being interleaved with queries from other tasks.

Our contributions include (1) analysis of typical timeouts used to divide query streams into sessions, and demonstration that they are less than optimal for this task (2) hierarchical analysis of user search tasks into short-term goals and longer-term missions (3) a detailed study of the frequency and patterns of real user queries forming extended and interleaved tasks which can be analyzed as missions and goals in this hierarchy (4) a comparison of previously published feature sets on our data and tasks, and (5) a list of features going beyond timeouts and previously published feature sets that can be used effectively to identify goal and mission boundaries, and pairs of non-adjacent queries belonging to the same goal or mission.

In Section 2, we discuss related work both in defining "sessions" and automated segmentation of query logs into tasks and sessions. In Section 3 we provide our definitions, detail on the manual annotation of our data, statistics on the missions and goals we find, and show that time-based thresholds are of limited accuracy in identifying task boundaries. In Section 4 we introduce the supervised classification which we perform to improve task identification, as well as the features and methods we use. In Section 5 we show that a model combining feature types can identify goals and missions with extremely high accuracy, even when they are interleaved. We also discuss performance of the individual features on the classification tasks. Lastly, in Section 6, we discuss conclusions and future directions for the work.

2. RELATED WORK

In library search systems, "sessions" were easy to identify: users logged in for a single task then logged out again, so login IDs could be used. Thus historically a session was simultaneously (1) a set of queries to satisfy a single information need (2) a series of successive queries, and (3) a short period of contiguous time spent querying and examining results. On the internet, however, we seldom have users logging in and out on a task-by-task basis. In addition, identifiers such as IP addresses and cookies may be shared by multiple users, as in the case of a shared computer. Thus the term session has necessarily been split between these various meanings, sometimes used for one, sometimes for another.

For web search, there have been a number of conflicting attempts to segment and define sessions, which don't directly address the idea of user information needs, but do rely on a notion of similar context, topic, or temporal characteristics. Many of these use the idea of a "timeout" cutoff between queries. A timeout is the time between two successive activities, and it is used as a session boundary when it exceeds a certain threshold. Often sessions are identified using a 30-minute timeout, apparently following Catledge and Pitkow's 1994 work, which claimed to find a 25.5 minute timeout based on user experiments[4]. We will show in Section 3.3 that this threshold is no better than random for identifying boundaries between user search tasks.

Other time cutoffs have been proposed, from 5 to 120 minutes [11][17][6][2]. Montgomery and Faloutsos [11] tried several cutoff values, but found that the choice of cutoff did not matter. Additionally, a variety of mean session lengths (number of queries in a session) have been found, most ranging between 2-3 queries per session[17][8]. Mean session durations (amount of time a session lasts) of 5 and 12 minutes have been reported [6][8]. In Section 3.3 we look at all of these timeout thresholds applied to real search engine data,

and find that no time threshold is effective at identifying task boundaries.

Jansen et al. [8] defined a session as "a series of interactions by the user toward addressing a single information need", and found experimentally that sessions were better identified by query content – a single word in common between queries – than by temporal cutoffs. Spink et al. by contrast [19] discuss topic switching and multitasking in two and three query sessions, implicitly defining a session as a sequence of queries from a web-searcher, independent of the information need.

A few researchers have worked on automatically detecting session boundaries. He et al. [6] tried to detect topic shifts in user query streams, devising an algorithm that checks for deleted or added terms in queries thereby specifying a few categories of user behavior, which are also applied in Ozmutlu and Cavdur [12]. Ozmutlu et al. [13] [14] used the same categories and algorithm but a different classifier. All of these rely solely on the feature of words in common between queries, and rewrite classes such as *generalization* and *specification*. There was no manual labeling of ground truth, and none of them addressed the interleaved, nested nature of these topic shifts. Both cases rely solely on the feature of words in common between queries, and used continuous values to predict rewrite classes such as generalization and specification, defined in terms of word insertion and deletion, and they did no manual labeling of ground truth. None of them addressed the interleaved, nested nature of these topic shifts.

A separate body of work models the formal syntax of users' interactions with the search engine, rather than making distinctions regarding what they seek. An entire list of these approaches to date may be found in Downey et al. [5], two of which deserve more detail.

Lau and Horvitz [9] manually assigned queries into a few classes, including one to account for interleaving, labeling each query in the context of the previous query. They examined the relationship between the inter-query time interval and these classes in order to better use timeouts to predict topic transitions in a probabilistic Bayesian network.

Radlinski and Joachims [15] identified sequences of queries on the same topic. The authors manually grouped sequences of queries into topics that had no hierarchical or interleaved structure. They built a classifier using features both based on shared words in the queries themselves, and shared words in documents retrieved in response to the queries, but judged a 30-minute timeout to be accurate at over 90%, and thus used only that feature to group their chains. These results were based on a library search engine, however, as we will show in Section 3.3 their results with timeouts do not apply to a general purpose search engine, where users often interleave topics, and time is not a good indicator of task completion. In Section 5.1 we will demonstrate the performance of their word and time features on our tasks, and show that this is the best of previous methods for our tasks.

We will show in Section 5 that time combined with some of the features proposed by previous authors can be used to classify queries into their hierarchical structure: thus demonstrating for the first time that we can identify user tasks and their sub-tasks, even in the face of task interleaving. We will also show that by adding novel query-log features we can perform even better on the original tasks, as well as our novel tasks.

3. SEARCH GOALS AND MISSIONS

In this section we define search goals and missions, then describe the way we manually annotated them. We perform an exploratory analysis and demonstrate that time-outs are a poor way of identifying boundaries related to user tasks.

DEFINITION 1. *A search session is all user activity within a fixed time window.*

A session, for us, is just a slice of user time. Other definitions (which conflict among themselves) involve an absence of periods of inactivity, or imply a single information need on the part of the user [4], [8], [2]; ours does not, since we will be more specific with the terms *goals* and *missions*, defined below, and use inactivity as a predictor, rather than as a definition.

DEFINITION 2. *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).

DEFINITION 3. *A search mission is a related set of information needs, resulting in one or more goals.*

A mission is then an extended information need (such as finding a variety of interesting hiking spots, for a searcher interested in hiking). In Figure 1, we see the resulting hierarchy: sessions containing missions, which contain goals. A goal may have multiple queries, and a mission may have multiple goals. Temporally, these can all be interleaved.

3.1 Annotation procedure

We sampled 3 day-sessions for 312 users who submitted a query to the Yahoo! Search engine during a week of mid-2007. The sampling was stratified over days of the week, so we did not have particular days of the week overrepresented. The three day time period was deemed long enough to capture extended search patterns for some users, exceeding typical 30-minute timeouts, and allowing for goals and missions to extend over multiple days. While it is the case that some missions or goals may start or end before or after the 3 day window of queries, this truncation should occur randomly, and thus introduce no systematic bias in the subject of the mission or goal. Since none of our features depend on the position of the query in the mission or goal, our model should not be affected. However, our density estimates for mission and goal lengths will necessarily be truncated at 3 days.

A group of annotators were instructed to exhaustively examine each session and "re-enact" the user's experience.[1] The annotators inspected the entire search results page for each query, including URLs, page titles, relevant snippets, and features such as spelling and other query suggestions. They were also shown clicks to aid them in their judgments. They were asked to then use their assessment of the user's
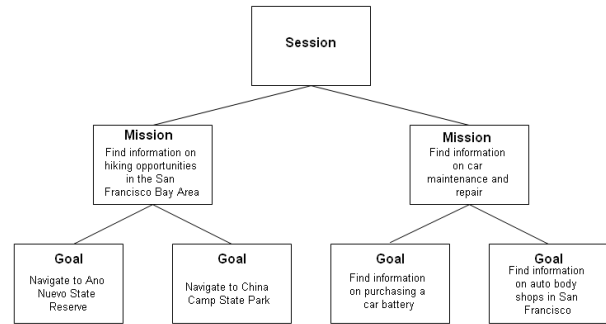
Figure 1: Sample hierarchy of user missions and goals, which correspond to some of the goals in the sample session of Table 1.

objectives to label each session so that every query belonged to a goal and every goal to a mission. Each unique goal and mission was given an ID number and a description reflecting the user's objective. Each query was labeled with a goal and mission ID number. A guideline for queries to belong to the same goal was that they have the same criteria for "success", in terms of satisfying the user's information need.

This is somewhat similar in nature to the editorial task in [7], though the annotators did not re-submit queries, they examined the query logs as input by the users themselves. It differs somewhat from manual annotation methods in [8] and [15] and [13] in that the criteria for grouping queries is, not only hierarchical in nature, but tied to the users' intents, as opposed to just clustered in terms of "relatedness" to an unspecified degree.

A strength of this approach is that the data is recorded without any intervention, and as such can complement laboratory and field studies. Additionally, we are able to look at a large number of users at one time, over an extended period of time. While it is clearly possible that editors may be biased in their interpretation of a user's missions and goals, preliminary results in [7] seem to indicate that this method can achieve reliable results. One of the future directions for this work involves obtaining measures of inter-rater reliability for the editorial work, as well as studying editorial bias against user self-reporting.

Our aggregate data consists of 312 user sessions, with 1820 missions, 2922 goals and 8226 queries. In Table 1 we see a sample sequence of user queries, annotated with goals and missions. Then timestamps are also given - note that many queries from the same goal are separated by several minutes, while a goal-change (goal 3 to goal 4) takes place within three seconds.

3.2 Patterns of Goals and Missions

In Table 2 we see a summary of goal and mission lengths, in terms of number of queries, and elapsed time from first to last query in the goal or mission[2]. Median goal and mission length of two queries is in line with the findings of [8] for session lengths based on segmentation by query topic. Figure 2

---

[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.

[2]Note that duration is calculated as $t(q_n) - t(q_1)$, where t is time, q_n is the last query in the mission or goal and q_1 is the first. Thus our measure of duration includes query times only, and not clicks or other actions. This is similar to the duration used in [8], but not in [4]. It also means that many durations are zero.

QUERY and TIMESTAMP	GOAL #	MISSION #	DESCRIPTION
hiking; san francisco Tue Apr 17 23:43:17 2007 (4m 17s)	1	1	MISSION 1: Find info on hiking opportunities in and around San Francisco
hiking; san francisco bay area Tue Apr 17 23:47:34 2007 (4m 59s)	1	1	GOAL 1: Find info on hiking trails in San Francisco and the Bay Area
ano nuevo state reserve Tue Apr 17 23:52:33 2007 (7m 54s)	2	1	GOAL 2: Navigate to Ano Nuevo State Reserve and ↓nd out about distances
ano nuevo state reserve; miles Wed Apr 18 00:00:27 2007 (3m 34s)	2	1	
nature trails; san francisco Wed Apr 18 00:04:01 2007 (16m 15s)	1	1	
lobos creek trail Wed Apr 18 00:20:16 2007 (0m 3s)	3	1	GOAL 3: Navigate to Lobos Creek Trail
china camp state park; san rafael Wed Apr 18 00:20:19 2007 (2m 35s)	4	1	GOAL 4: Navigate to China Camp, San Rafael and ↓nd out about distances
china camp; miles Wed Apr 18 00:22:54 2007 (20m 2s)	4	1	
hike; san francisco Wed Apr 18 00:42:56 2007 (3m 19s)	1	1	
fort funston Wed Apr 18 00:46:15 2007 (1h 51m 26s)	5	1	GOAL 5: Navigate to Fort Funston
			MISSION 2: Find info on car maintenance and repair
brake pads Wed Apr 18 03:36:47 2007 (16m 36s)	6	2	GOAL 6: Find info on brake pads
auto repair Wed Apr 18 03:53:23 2007 (8m 0s)	7	2	GOAL 7: Find info on an auto body shop in San Francisco
auto body shop Wed Apr 18 04:01:23 2007 (3m 31s)	7	2	
batteries Wed Apr 18 04:04:54 2007 (0m 29s)	8	2	
car batteries Wed Apr 18 04:05:23 2007 (2m 8s)	8	2	GOAL 8: Find info on purchasing a car battery
auto body shop; san francisco Wed Apr 18 04:07:31 2007 (3m 33s)	7	2	
buy car battery online free shipping Wed Apr 18 04:11:04 2007	8	2	

Table 1: Sample of a sequence of user queries annotated with goals and missions. Horizontal lines mark changes in goal, and double horizontal lines mark changes in mission. The description for a goal or mission is input around the same line as the first query which belongs to that goal or mission. In this example, the goals are interleaved, but the missions are not. However, in general the missions may be interleaved as well; in our data 17% of missions were interleaved.

	Goals	Missions
Num queries		
min	1	1
max	52	233
median	2	2
Duration		
min	0 mins	0 mins
max	71 hours	71 hours
median	0.42 secs	38 secs

Table 2: Summary statistics about missions and goals. The distributions of number of queries and task duration can be seen in Figures 2 and 3.

shows the density of number of queries per goal and mission. Density plots show Gaussian kernel density estimates, with bandwidths chosen by Silverman's rule [16].

63% of goals are under one minute, but 15% spanned 30-minute periods of inactivity. This means that a 30 minute time-out will break up 15% of goals. The density of goal and mission durations are in Fig 3.

Most goals and missions have few queries, though a few have many queries. Some missions lasted the entire 3 day session, and those users appeared to make related or repeated queries an average of every few hours, in some cases looking at baby names, or checking up on favorite television stars. Recall that missions and goals may be interleaved, so these long durations do not necessarily entail continuous engagement at some task. 16% of goals are revisited or interleaved with other goals, and 17% of missions are revisited or interleaved with other missions. Of the interleaved missions, 41% contained multiple goals, whiled 59% contained a single goal (which was itself interleaved with a goal from another mission). It is not surprising that users would repeat infor-

mation needs; Teevan et al [20] found that 40% of queries are "re-finding" queries. In addition, we find that 20% of missions contain multiple goals. An example mission containing multiple goals consisted of wedding planning queries, for wedding gowns, invitations, and wedding planning lists. We also see the evolution of users' shopping intent over the course of a mission, with a query for "bridal dresses" on one day, and another query in the same mission the following day, containing "bridal dresses" and the name of a bridal dress store. The user is moving from learning about general options to bridal dresses, to looking for a particular store to buy the dress.

Thus any task-segmenting approach which does not consider the hierarchical and interleaved nature of search tasks will break up tasks which belong together.

In preliminary experiments we found many queries repeated immediately after one another, representing either the user re-issuing the query, hitting the 'next' button, refreshing the page, or an automatic resubmission on the part of the browser. Removing the repeated queries decreased the total number of queries from 8226 to 6043, thus just over a quarter of all queries were repeats of the previous query.

3.3 Analysis of Session Timeouts

Most previous work has used temporal features, commonly a "timeout": an elapsed time of 30 minutes between queries which signifies that the user has discontinued searching. However, on our data time does not appear to be an especially good predictor, particularly of goal boundaries. Precision for different values of inter-query time-lag are shown in Figure 4.

In Table 3 we see that a 30-minute threshold on inter-query interval is more accurate than the baseline (always guess there is no task boundary between each sequential
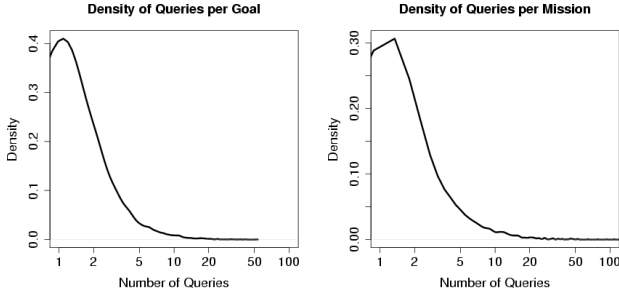
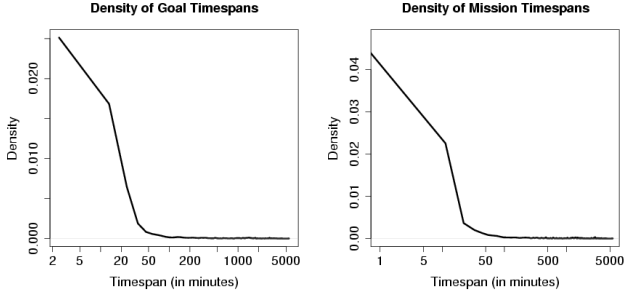**Figure 2: Density for the number of queries per goal, and number of queries per mission, on a log plot.**

Figure 3: Density for the time span of goals and missions, in minutes, on a log plot.
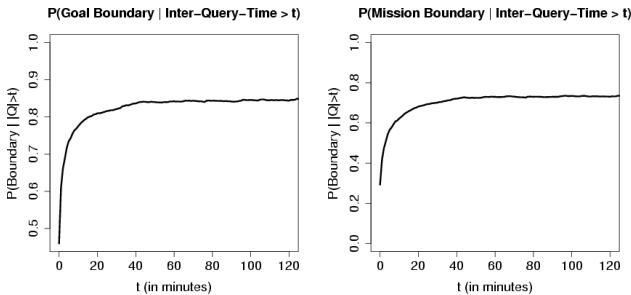

Figure 4: As we increase the inter-query interval threshold, the precision at identifying mission and goal boundaries increases, however, we do not see precision much above 70% for identifying missions and above 80% for identifying session boundaries.

pair of queries). Training a threshold (thirteen minutes) does not improve the accuracy for mission boundary identification, but the learned threshold of just under 5 minutes improves goal boundary identification from 67% accuracy at a 30 minute threshold, to 71%. All differences are statistically significant. Our conclusions agree with those of [11] that multiple threshold choices give similar accuracy.

Clearly, while using session timeouts can achieve task breaking with accuracy better than assuming there are no breaks, in general there is no ideal choice of threshold, and using time the precision is capped at 70-80%, depending on whether we seek goal or mission boundaries. The 30-minute standard receives no support from our results. In the next

Time threshold	Goals	Missions
	Boundary	Boundary
Baseline	54.2%	70.9%
5 minute	71.2%	75.6%
30 minute	66.5%	78.6%
60 minute	64.2%	77.6%
120 minute	62.0%	76.1%
Trained time	71.2%	78.6%

Table 3: For data which includes repeat queries: In-sample accuracy at predicting goal and mission boundaries, as well as same-goal/mission, using inter-query thresholds alone. Trained times for goal and mission boundaries were *5 mins* and *13 minutes*, respectively.

section we will show that we can greatly improve on task segmentation by considering multiple predictors, going beyond inter-query time to include properties of the queries themselves.

4. AUTOMATIC DETECTION OF GOALS AND MISSIONS

In this section we describe our formulation of automatic detection of search goals and missions as a supervised machine learning task. We then describe the features we use in our experiments for identifying goals and missions, and the classifiers we use to learn to recognise them. In particular, we introduce a way of identifying when queries belong to the same goal or mission, despite being interleaved, something not addressed in any previous work.

4.1 Formulation for Supervised Learning

If goals and missions are not interleaved, as has been assumed by previous work, it suffices to find a boundary between one task and the next. To do this we can look at each sequential pair of queries and ask whether this pair straddles a boundary. Thus we look at *task boundary detection*. This task has been addressed by previous approaches for identifying goals, so we can examine how well previous approaches work on our data. Previous work has not considered this task for higher-level missions, but we can also consider the efficacy of their features on this novel problem.

In order to address interleaved missions and goals, we must consider all possible pairs of queries, and consider whether the pair of queries come from the same task. Correctly performing this task will allow hierarchichally organized interleaved goals and missions to be correctly identified. We call this *same-task* identification. No previous work has addressed this problem.

4.1.1 Task Boundary Detection

Each pair of sequential queries from a user is a possible boundary between goals. Thus we seek to take each such pair and decide whether the pair crosses a boundary between goals, i.e., whether the two queries come from different goals. This is the task traditionally addressed using timeouts. Formally we consider the task:

$$\{\langle q_i, q_j \rangle : (t(q_i) < t(q_j)) \bigwedge (\nexists q_k : t(q_i) < t(q_k) < t(q_j))\} \to \{0, 1\}$$

where $t(q_i)$ is the time query q_i was issued by the user.

Of our original 8226 queries including repeats, some begin and end user sessions, so we trivially ignore these boundaries and wind up with 7914 pairs of sequential queries. Of these 3622 were goal boundaries, so we should guess a position is a boundary 45.8% of the time, and a non-boundary otherwise. Thus our baseline is 54.2% accuracy.

As with goal boundaries, any sequential pair of queries can mark the transition from one mission to another. (Note that a mission boundary is always a goal boundary.) Our baseline is 71% as 5608 of 7914 pairs of sequential queries are not mission boundaries.

4.1.2   Same-Task Identification

Since goals can be inter-leaved, any pair of queries from the same user could be from the same goal. We seek to learn a classifier to take a pair of queries and map it to a 1 if they are from the same goal, and a 0 of they are from different goals. We consider all pairs of queries q_i, q_j such that q_i was issued before q_j:

$$\{\langle q_i, q_j \rangle : t(q_i) < t(q_j)\} \rightarrow \{0, 1\}$$

where $t(q_i)$ is the time query q_i was issued by the user.

Since we are considering all pairs of queries, not just sequential ones, there are many more instances to consider. We see 305,946 pairs of queries, of which 278,152 or 91% are not for the same-goal so our baseline accuracy is 91%. Using the same definitions for same-mission our baseline is 67.5% since 67.5% of 305,946 query pairs were from different missions. Any pair of queries corresponding to the same goal will also correspond to the same mission.

Solving these two problems, *same-goal* and *same-mission* for arbitrary pairs of queries from a user's query stream will allow us to identify their complete set of tasks, even those that are nested and hierarchical.

4.2   Features for Identifying Goals and Missions

We experimented with many features from the following four general types: temporal, edit-distance, query log and web search. Below we give details of those that contributed to classification efficacy. Temporal features have been commonly used in previous work ([11][17][6][2]); edit distance has been used in several previous works ([6],[14]) and web search features have also been used in previous work [15]. No previous work has used query log session features, and we will show that combinations of these four types of features provide superior performance on the boundary detection problem, and superior performance on the previously untackled same-task problem.

4.2.1   Temporal Features

While we showed in Section 3.3 that timeouts alone are poor predictors of task boundaries, they may be helpful in conjunction with other features. Temporal features we experimented with are:

- inter-query time threshold as a binary feature (5 mins, 30 mins, 60 mins, 120 mins)
- time_diff: inter-query time in seconds: we may be able to learn good thresholds of inactivity for identifying goal and mission boundaries.
- sequential-queries: binary feature which is positive if the queries are sequential in time, with no intervening

Feature	Description
lev	normalized Levenshtein edit distance
edlevGT2	1 if lev > 2, 0 otherwise
char_pov	num. characters in common starting from the left
char_suf	num. characters in common starting from the right
word_pov	num. words in common starting from the left
word_suf	num. words in common starting from the right
commonw	num. words in common
wordr	Jaccard distance between sets of words

Table 4: Word and Character Edit Features used for predicting goal and mission boundaries and cooccurrence on query pairs.

Query Pair	log (LLR)
uofa → university of arizona	8.4
wedding strapless gowns → strapless wedding gowns	7.9
large daisy → flower	7.3

Table 5: Pairs of queries which occur in user query sequences much more than would be expected by chance, along with the log-likelihood ratio score.

queries from the same user. We expect this feature to be useful for the same-goal and same-mission tasks.

4.2.2   Word and Character Edit Features

Sequences of queries which have many words and/or characters in common tend to be related via a query reformulation, for example word insertion or deletion [15, 18]. In addition, related queries from the same goal or mission may have some words in common. Character-edit distance can capture spelling variants and common stems, while word-level features capture common words.

Specific features are shown in Table 4.

4.2.3   Query Log Sequence Features

Sometimes goals and missions may contain pairs of queries which are semantically related, but which do not share any terms. For example, "new york hairdresser" and "tribeca salon" may be from the same goal: looking for a hair-stylist in New York City. To try to capture these semantic relationships, we use a separate data source[3] to identify pairs of queries, $\langle q_i, q_p \rangle$, which occur together much more than chance, which we test using the log-likelihood ratio score [10] (LLR).

Because we test millions of pairs of queries, we can have false positives of coincidentally cooccurring query pairs. A threshold level for the LLR was chosen in order to control false positives, including adjustment for multiple testing, correcting a standard chi-square cutoff for 95% significance. Sample query pairs which pass this threshold are shown in Table 5. After thresholding on the LLR, we use a number of features related to the frequencies and probabilities of seeing the query pair together. Specific features which used rewrite probabilities for the query pair $\langle q_1, q_2 \rangle$ are shown in Table 6.

- llr: the result of a statistical test (LLR [10]) indicating that the pair of queries occur in sequence more than could be expected by chance

---

[3]Two weeks of pairs of sequential queries to the search engine, in 2006.

Feature	Description
llr	LLR of cooccuring query pair
peos_q2	prob q_2 is a user's last query of the day
pq12	$p(q_1 \to q_2)/max_{q_j} p(q_1 \to q_j)$
entropy_X_q1	$\sum_i p(q_1\|q_i)log_2(p(q_1\|q_i))$
entropy_q1_X	$\sum_i p(q_i\|q_1)log_2(p(q_i\|q_1))$
nsubst_X_q1	$count(X : \exists p(X \to q_1)$
nsubst_X_q2	$count(X : \exists p(X \to q_2)$
nsubst_q2_X	$count(X : \exists p(q_2 \to X)$
seen_in_logs_qp	1 if $LLR(q_1, q_2) >$ threshold
p_change	$\sum(p_1 \to p_j) : p_j \neq p_1$

Table 6: Query Log Features used to help identify goal and mission boundaries $p(q_1 \to q_2)$ **is the probability** q_1 **is reformulated as** q_2 **in a large query log.**

- peos_q2: the probability that q_2 is a user's last query of the day, based on aggregating queries that are the last before midnight

- pq12: the normalized probability that q_1 is rewritten as q_2 aggregated over many user sessions, $p(q_1 \to q_2)/max_{q_j}p(q_1 \to q_j)$

- entropy_X_q1: the entropy of rewrite probabilities from queries which can be rewritten as q_1 (after LLR filtering), $\sum_i p(q_1|q_i)log_2(p(q_1|q_i))$

- entropy_q1_X: analagously $\sum_i p(q_i|q_1)log_2(p(q_i|q_1))$

- nsubst_X_q1: the number of different queries that have been seen in the logs rewritten as q_1 (after LLR thresholding) $count(X : \exists p(X \to q_1)$

- nsubst_X_q2: $count(X : \exists p(X \to q_2)$

- nsubst_q2_X: $count(X : \exists p(q_2 \to X)$

- seen_in_logs_qp: 1 if $LLR(q_1, q_2) >$ threshold

- p_change: $\sum(p_1 \to p_j) : p_j \neq p_1$

4.2.4   Web Search Features

We include features which depend on the documents retreived by the search engine for the queries. Similarity between a query pair is measured by commonalities among the terms or characteristics of those documents.

- *Prisma*: cosine distance between vectors derived from the first 50 search results for the query terms. In this case, the terms are limited due to a dictionary, in a method developed in Anick [3][2]. It is similar in nature to the best performing feature in Radlinski and Joachim's classifier [15].

4.3   Classifier

We use a logistic regression model, with 10-fold cross-validation. In order to better handle feature selection for large sets of correlated features, we also tried LASSO, which includes regularization, and CART decision trees, however, we achieved similar results in both cases. When performing feature selection for small subsets, we used an exhaustive search of linear models with Akaike Information Criterion (AIC) to select the best features (also known as all subsets regression).

Feature	Goals		Missions	
	Boundary	Same	Boundary	Same
Baseline	54.2%	90.9%	70.9%	67.5%
lev	89.0%	95.3%	84.1%	77.9%
wordr	86.9%	95.1%	83.9%	78.6%
commonw	82.9%	91.0%	83.9%	79.7%
Time interval	62.5%	90.9%	73.8%	67.6%

Table 7: For data which includes repeat queries: normalized Levenshtein distance dominates other features for most tasks. Thanks to the large number of examples in our tests, all differences are statistically significant.

Features	Goals		Missions	
	Boundary	Same	Boundary	Same
Baseline	63.1%	94.8%	59.9%	70.5%
30 minute	57.2%	90.9%	73.8%	74.4%
Trained time	69.5%	92.6%	75.8%	74.4%
commonw	80.7%	94.9%	79.3%	78.9%
commonw+prisma+time	84%		82.1%	

Table 8: Prediction accuracy based on features proposed in previous work. Trained time thresholds for boundaries were *1.5 mins* for goals and *6 mins* for missions. For identifying same-goal *17.2 mins* and same-mission *47 mins*. The best performing previously published feature combination is commonw+prisma+time. All differences are statistically significant.

5.   RESULTS

Used alone, Levenshtein character edit distance did well in 3 of 4 tasks, especially when repeat queries were included (Table 7). However, a quarter of all sequential query pairs consisted of identical queries, belonging to the same goal or mission. Thus much of our data was easy to classify with Levenshtein distance (a distance of zero).[4] In this section we address the harder problem of classifying those query pairs which are not repetitions. Our baselines, with repeated queries removed, are 63% for goal boundary identification, 60% for mission boundary, 95% for same-goal identification, and 71% for same-mission.

5.1   Baselines from Previous Work

In this section we give results using features proposed in previous work, applied to our data. To compare to approaches using time, we use both a thirty-minute threshold, as well as time thresholds learned using cross-validation. To compare with word-insertion and deletion approaches ([6],[14]), we use the word-edit feature *commonw*, and to compare to word-edit, web-result and time features we use *commonw+prisma+time* [15]. In all cases we trained a logistic regression function on training data, and used ten-fold cross-validation for testing.

We see in Table 8 that on web search data, Radlinski and Joachims' result does not hold, that a thirty-minute threshold obtains similar results to a combination of common words and web-search result similarity [15]. We see that our re-implementation of their classification features, using commonw+prisma+time, is the best-performing previously published approach.

---

[4]Additionally, these identical query pairs do not have well-defined query log features, since repeated queries were removed before query log feature computation.

Features	Mission Boundary	
	Boundary	+repeats
Baseline	59.9%	70.9%
commonw+prisma+time	82.1%	
lev, wordr, peos_q1		
prisma, time_diff, peos_q2		
n_subst_X_q2 , seen_in_logs_qp	84.4%	90.8%

Features	Goal Boundary	
	Boundary	+repeats
Baseline	63.1%	54.2%
commonw+prisma+time	84%	
lev, prisma, wordr,		
word_suf, char_suf,char_pov		
word_pov, n_subst_X_q1	87.3%	93.0%

Table 9: **Accuracy at predicting mission and goal boundaries, using the most predictive models, as judged by exhaustive search of logistic regression models with AIC as selection criterion. commonw+prisma+time is the best performing previously published feature combination for the goal boundary task. All differences are statistically significant.**

5.2 Best Classifier Results

We are able to build highly accurate classifiers for goal and mission boundaries as well as identifying pairs of queries from the same goal or mission. When we combine all four types of features we achieve best results, as shown in Tables 9 and 10.

In these tables we summarize features and performance for learned models in the no-repeat cases, as well as folding in the repeated-queries under the assumption that a query repeat is a non-boundary (which is correct in all cases). We see that our best model exceeds the best previously published feature combination on our data for the task boundary problem, as well as providing strong results for the new problem of identifying query pairs as being from the same task, even when interleaved and hierarchically organized.

We see that after removing repeated queries, we clearly gain from a combination of features in all four groups: edit-based, query-log, web-search and time. The models were selected via exhaustive search of all models with 8 predictors, using AIC as the selection criterion. While these models contain small subsets of the original features, the accuracy was comparable to that obtained with all features. We restricted this exhaustive search to 8 features as when we used subsets larger than 8 features, the feature selection began to become computationally extremely expensive. When we fold back in the instances of repeated queries, we can compare our results to those in Table 7 where we trained with the repeated queries. While the best accuracy for goal boundaries with timeouts was 62.5%, this combined model gives an accuracy of 93.0%, improving on edit distance features alone, which gave an accuracy of 89%. For mission boundaries, the combined model greatly outperforms the time-interval and edit-distance models, lifting from 67.6% and 79.7% to 90.8%.

5.3 Contributions of Feature Classes

We showed above that by using a combination of four types of features, we could outperform time-based segmentation (which we showed achieved accuracies of around 70%,

Features	Same Mission	
	Same	+repeats
Baseline	70.5%	67.5%
commonw, word_suf, entropy_X_q1		
nsubst_q2_X, pq12, char_pov		
111r12, time_diff	88.36%	88.8%

Features	Same Goal	
	Same	+repeats
Baseline	94.8%	90.9%
edlevGT2, wordr, char_suf		
nsubst_q2_X, time_diff, sequential,		
prisma, entropy_q1_X	97.09%	97.2%

Table 10: **Accuracy at predicting same-mission/goal, using the most predictive models, as judged by exhaustive search of logistic regression models with AIC as selection criterion. All differences are statistically significant.**

Edit Distance	Goals		Missions	
	Boundary	Same	Boundary	Same
Baseline	63.1%	94.8%	59.9%	70.5%
lev	85.0%	95.2%	78.2%	77.0%
wordr	83.9%	95.3%	79.2%	77.9%
commonw	80.7%	94.9%	79.3%	78.9%

Table 11: **Prediction accuracy for features based on edit distance between queries alone. All differences are statistically significant.**

Edit Distance	Goals		Missions	
	Boundary	Same	Boundary	Same
Baseline	63.1%	94.8%	59.9%	70.5%
lev+time	85.0%	95.8%	78.3%	76.8%
commonw+time	81.5%	95.3%	79.3%	78.9%
wordr+time	84.2%	95.9%	79.3%	77.0%

Table 12: **Prediction accuracy based on edit distance between queries as well as inter-query interval. The latter does not appear to help. All differences are statistically significant.**

compared to our accuracies of around 90%). We also showed that by using 8 features from the four types edit-distance, web-search, query log and time, we could improve over the best previously published combination of features (commonw+prisma+time). In this section we examine the contribution of each of the feature types separately.

5.3.1 Edit Distance

Even after removing repeated queries, Levenshtein character edit distance is the best edit-based feature for identifying goal boundaries (Table 11), and words-in-common (commonw) is the best edit-based feature for identifying mission boundaries as well as same-mission. Ratio of words in common (Jaccard distance) performs best for same-goal. In general, this group of features performs well for all four tasks. Adding time to character edit distance does not help in identifying goal or mission boundaries (Table 12). This is similar to most previous work on identifying task boundaries ([6],[14]).

Query Log Feature	Goals		Missions	
	Boundary	Same	Boundary	Same
Baseline	63.1%	94.8%	59.9%	70.5%
pEOSq2	63.1%	94.8%	66.4%	70.5%
p_change	67.1%	94.8%	59.9%	70.5%

Table 13: Prediction accuracy using best-performing query session cooccurrence features. These improve over baseline for boundary identification, but not the same-goal/mission task. All differences are statistically significant.

Web Search Feature	Goals		Missions	
	Boundary	Same	Boundary	Same
Baseline	63.10%	94.75%	59.87%	70.51%
Prisma Score	78.5%	94.8%	77.1%	73.0%

Table 14: Prediction accuracy with Prisma vector similarity, the best-performing web search feature we tried.

5.3.2 Query Log Features

End-of-session features help with mission boundary identification (Table 13), and probability of being rewritten does slightly better at detecting goal boundaries, but other query log features in isolation do not improve over the baseline. It's possible that the probability of the first query being rewritten may indicate that the second query is more likely to be a reformulation of the first, and thus part of the same goal. One can imagine that "last query of the day" might be useful as an indicator for the last query in a task, since often people will finish up a task before turning in for the day. In our external data-source, days were truncated at midnight. We may be able to improve the suitability of query log features for this task by considering different ways of breaking the data using other markers.

5.3.3 Web Search Features

Of the web search features we tested, the Prisma score outperformed all others (Table 14). However, unlike those other features, as well as the query log features, it had not been pre-computed for the more common queries, to save computational cost. This makes it slower to obtain, yet gives it superior coverage.

6. CONCLUSIONS AND FUTURE WORK

We have shown that a diverse set of syntactic, temporal, query log and web search features in combination can predict goal and mission boundaries well. (When used independently, word and character based features perform best). Our classifiers achieve at least 89% accuracy in all four tasks, and over 91% in all but one task, matching within the same goal. Additionally, we've shown that the task of matching queries within the same interleaved goal or mission is harder than identifying boundaries. This may indicate that the best approach to clustering queries within the same goal or mission may build on first identifying the boundaries, then matching subsequent queries to existing segments. It may also be effective to use multi-task machine learning to join the tasks of identifying mission and goal boundaries together.

The utility of adopting a hierarchical model for the grouping of user queries will allow us to more easily model what type of task the user may be doing when querying, e.g. is the user performing a series of searches with information needs which are the same, or are the information needs only peripherally related? This may help us determine when the user is performing a more complicated task, vs. a simpler task. Including the interleaving in the model allows us to more accurately measure the length of time or number of queries a user needs to complete tasks. If we ignore the fact that a more involved task may be interrupted with other needs for information, we lose the ability to model these more involved tasks.

Our work sets the stage for evaluating search engines, not on a per-query basis, but on the basis of user tasks. In future work we will combine task segmentation with prediction of user satisfaction, which opens up the possibility of truly understanding how web search engines are satisfying their users.

8. REFERENCES

[1] Comscore announces new "visits" metric for measuring user engagement, 2007. http://www.comscore.com/press/release.asp?press=1246.

[2] P. Anick. Using terminological feedback for web search refinement - a log-based study. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 88–95, 2003.

[3] P. G. Anick. *Automatic Construction of Faceted Terminological Feedback for Context-Based Information Retrieval*. PhD thesis, Brandeis University, 1999.

[4] L. Catledge and J. Pitkow. Characterizing browsing strategies in the world-wide web. In *Proceedings of the Third International World-Wide Web Conference on Technology, tools and applications*, volume 27, 1995.

[5] 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.

[6] D. He, A. Goker, and D. J. Harper. Combining evidence for automatic web session identification. *Information Processing and Management*, 38:727–742, 2002.

[7] 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 (SIGIR 2007)*, pages 567–574, 2007.

[8] B. J. Jansen, A. Spink, C. Blakely, and S. Koshman. Defining a session on web search engines. *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2000.

[9] T. Lau and E. Horvitz. Patterns of search: Analyzing and modeling web query refinement. In A. Press, editor, *Proceedings of the Seventh International Conference on User Modeling*, 1999.

[10] C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[11] A. Montgomery and C. Faloutsos. Identifying web browsing trends and patterns. *IEEE Computer*, 34(7):94–95, 2007.

[12] H. C. Ozmutlu and F. Cavdur. Application of automatic topic identification on excite web search engine data logs. *Information Processing and Management*, 41(5):1243–1262, 2005.

[13] H. C. Ozmutlu, F. Cavdur, A. Spink, and S. Ozmutlu. Investigating the performance of automatic new topic identification across multiple datasets. In *Proceedings 69th Annual Meeting of the American Society for Information Science and Technology (ASIST) 43*, Austin (US), 2006.

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

[15] 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.

[16] B. W. Silverman. *Density Estimation*. Chapman and Hall, London.

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

[18] A. Spink, B. J. Jansen, and H. C. Ozmultu. Use of query reformulation and relevance feedback by Excite users. *Internet Research: Electronic Networking Applications and Policy*, 10(4):317–328, 2000.

[19] A. Spink, M. Park, B. J. Jansen, and J. Pedersen. Multitasking during web search sessions. *Inf. Process. Manage.*, 42(1):264–275, 2006.

[20] J. Teevan, E. Adar, R. Jones, and M. Potts. History repeats itself: Repeat queries in Yahoo's logs. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 703–704, 2006.