# Toward Whole-Session Relevance: Exploring Intrinsic Diversity in Web Search

Karthik Raman
Dept. of Computer Science
Cornell University
Ithaca, NY, USA
karthik@cs.cornell.edu

Paul N. Bennett
Microsoft Research
One Microsoft Way
Redmond, USA
pauben@microsoft.com

Kevyn Collins-Thompson
Microsoft Research
One Microsoft Way
Redmond, USA
kevynct@microsoft.com

## ABSTRACT

Current research on web search has focused on optimizing and evaluating single queries. However, a significant fraction of user queries are part of more complex tasks [20] which span multiple queries across one or more search sessions [26, 24]. An ideal search engine would not only retrieve relevant results for a user's particular query but also be able to identify when the user is engaged in a more complex task and aid the user in completing that task [29, 1]. Toward optimizing whole-session or task relevance, we characterize and address the problem of *intrinsic diversity* (ID) in retrieval [30], a type of complex task that requires multiple interactions with current search engines. Unlike existing work on extrinsic diversity [30] that deals with ambiguity in intent across multiple users, ID queries often have little ambiguity in intent but seek content covering a variety of aspects on a shared theme. In such scenarios, the underlying needs are typically exploratory, comparative, or breadth-oriented in nature. We identify and address three key problems for ID retrieval: identifying authentic examples of ID tasks from post-hoc analysis of behavioral signals in search logs; learning to identify initiator queries that mark the start of an ID search task; and given an initiator query, predicting which content to prefetch and rank.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models, search process*

## General Terms

Algorithms, Experimentation

## Keywords

Search session analysis, diversity, proactive search

## 1. INTRODUCTION

Information retrieval research has primarily focused on improving retrieval for a single query at a time. However, many complex tasks such as vacation planning, comparative shopping, literature surveys, *etc.* require multiple queries to complete the task [20].

| Initiator query | Successor queries |
|---|---|
| `snow leopards` | `snow leopard pics`<br>`where do snow leopards live`<br>`snow leopard lifespan`<br>`snow leopard population`<br>`snow leopards in captivity` |
| `remodeling ideas` | `cost of typical remodel`<br>`hardwood flooring`<br>`earthquake retrofit`<br>`paint colors`<br>`kitchen remodel` |

**Table 1: Examples of intrinsically diverse search tasks, showing the first (initiator) query and several successor queries from the same search session.**

Within the context of this work, we focus on one specific type of information seeking need that drives interaction with web search engines and often requires issuing multiple queries – namely intrinsically diverse tasks [30]. Table 1 gives examples of two intrinsically diverse tasks observed in a commercial web search engine. Intrinsic diversity, where diversity is a desired property of the retrieved set of results to satisfy the current user's immediate information need, is meant to indicate that diversity is intrinsic to the need itself; this is in contrast to techniques that provide diversity to cope with uncertainty in query intent (*e.g.*, `[jaguar]`).

Intrinsically diverse tasks typically are exploratory, comprehensive, survey-like, or comparative in nature. They may result from users seeking different opinions on a topic, exploring or discovering aspects of a topic, or trying to ascertain an overview of a topic [30]. While a single, comprehensive result on the topic may satisfy the need when available, several or many results may be required to provide the user with adequate information [30]. As seen in the examples, a user starting with `[snow leopards]` may be about to engage in an exploratory task covering many aspects of snow leopards including their lifespan, geographic dispersion, and appearance. Likewise when investigating remodeling ideas, a user may wish to explore a variety of aspects including cost, compliance with current codes, and common redecoration options. Note that the user may in fact discover these aspects to explore through the interaction process itself. Thus intrinsic diversity shares overlap with both exploratory and faceted search [9, 37]. However, unlike the more open-ended paradigm provided by exploratory search, we desire a solution that is shaped by the current user's information need


*SIGIR'13,* July 28–August 1, 2013, Dublin, Ireland.

and is able to discover and associate relevant aspects for a topic automatically in a data-driven fashion. For example, for the query `[snow leopards]`, our goal is to enable deeper user-driven exploration of that topic, by proactively searching for the relevant information that the user might want during the course of a session on that topic, thus reducing the time and effort involved in manual reformulations, aspect discovery, and so on.

To this end, we aim to design a system that addresses two key problems needed for ID retrieval: detecting the start of an ID task, and computing an optimal set of ID documents to return to the user given engagement on an ID task. For the former, the system must be capable of predicting when a user is likely to issue multiple queries to accomplish a task, based on seeing their first "initiator query". To do this, we first develop a set of heuristic rules to mine examples of authentic intrinsic diversity tasks from the query logs of a commercial search engine. The resulting tasks provide a source of weak supervision for training classification methods that can predict when a query is initiating an intrinsically diverse task. With these predictive models, we characterize how ID initiators differ from typical queries. We then present our approach to intrinsically diversifying for a query. In particular, rather than simply considering different intents of a query, we incorporate queries that give rise to related aspects of a topic by estimating the relevance relationship between the aspect and the original query. Given the intrinsically diverse sessions identified through log analysis, we demonstrate that our approach to intrinsic diversification is able to identify more of the relevant material found during a session given less user effort, and furthermore, the proposed approach outperforms a number of standard baselines.

## 2. RELATED WORK

The distinction between *extrinsic* and *intrinsic* diversity was first made by Radlinski *et al.* who coined these terms [30]. In contrast to extrinsically-oriented approaches, which diversify search results due to ambiguity in user intent, intrinsic diversification requires that results are both relevant to a single topical intent as well as diverse across aspects, rather than simply covering additional topical interpretations. Existing methods like maximal marginal relevance (MMR) do not satisfy these requirements well (*cf.* Sec. 5.3). Most diversification research has focused primarily on extrinsic diversity: this includes learning [39, 34] and non-learning approaches [6, 40, 7, 8]. Recent work [2], however, indicates real-world Web search tasks are commonly intrinsically diverse and require significant user effort. For example, considering average number of queries, total time, and prevalence of such sessions, common tasks include: discovering more information about a specific topic (6.8 queries, 13.5 min, 14% of sessions); comparing products or services (6.8 q, 24.8 m, 12%); finding facts about a person (6.9 q, 4.8 m, 3.5%); and learning how to perform a task (13 q, 8.5 m, 2.5%). Thus, improvements in retrieval quality that address intrinsically diverse needs have potential for broad impact.

Some previous TREC tracks, including the Interactive, Novelty and QA tracks, studied intrinsic diversity-like problems in which retrieval effectiveness was partly measured in terms of coverage of relevant aspects of queries, along with the interactive cost to a user of achieving good coverage. However, our task and data assumptions differ from these tracks. For example, the Interactive tracks focused more on coverage of fact- or website-oriented answers, while our definition of query aspect is broader and includes less-focused subtopics. In addition to optimizing rankings to allow efficient exploration of topics, we also predict queries that initiate intrinsically diverse tasks, and show how to mine candidates for ID tasks from large-scale search log data.

*Session*-based retrieval is a topic that has become increasingly popular. For example, Radlinski *et al.* [31] studied the benefit of using query chains in a learning-to-rank framework to improve ranking performance. Others have proposed different session-level evaluation metrics [17, 21]. Research in this area has been aided by the introduction of the Session track at TREC [22]; this has led to papers on session analysis and classification [27]. In particular, He *et al.* use a random walk on a query graph to find other related queries, which are then clustered and used as subtopics in their diversification system [16]. In our re-ranking approach, we also use related queries to diversify the results, but maintain coherence with the original query. Specifically, we identify a common type of information need that often leads to longer, more complex search sessions. However, in contrast to previous work, rather than using the session interactions up to the current point to improve retrieval for the current query, we use a query to improve retrieval for a user's *future* session and use sessions from query logs to evaluate the effectiveness of the proposed methods. While the TREC Session track evaluated the number of uncovered relevant examples for the final query, the emphasis was on the impact of session context up to the present query; in our case, we assume no previous context, but instead are able to characterize the need for intrinsic diversity based on the single query alone.

Session data has also been used to identify and focus on complex, multi-stage user search tasks that require multiple searches to obtain the necessary information [36, 24]. This has led to research on *task-based* retrieval [14, 15] where tasks are the unit of interest (as opposed to queries or sessions). *Trail-finding* research studies the influence of factors such as relevance, topic coverage, diversity and expertise [33, 38]. While these problems are certainly related to ours, *tasks* and *trails* tend to be more specialized and defined in terms of specific structures: *e.g.* tasks are characterized as a set or sequence of sub-tasks to be accomplished, while trails are defined in terms of specific paths of user behavior on the web graph. However, intrinsically diverse search sessions, *e.g.* as in Table 1, represent a broader, less structured category of search behavior. Similarly, our approach complements work on faceted search [23] and exploratory search [37] by providing a data-driven manner of discovering common facets dependent on the particular topic.

Query suggestions are a well-established component of web search results with a large research literature: common approaches include using query similarity (*e.g.*[42]) or query-log based learning approaches (*e.g.* [19]). Query suggestions can play an important role for intrinsically diverse needs, because they provide an accessible and efficient mechanism for directing users towards potentially multiple diverse sets of relevant documents. Therefore, query suggestion techniques that do not merely provide simple reformulation of the initial query, but correctly diversify across multiple facets of a topic may be particularly helpful for intrinsically diverse needs. Thus, recent research on diversifying query suggestions [28] has partly inspired our retrieval approach.

Our approach is also motivated by recent work on *interactive ranking*. Brandt *et al.* [5] propose the notion of dynamic

rankings, where users navigate a path through the search results, to maximize the likelihood of finding documents relevant to them. Our objective formulation closely relates to another recent work on two-level dynamic rankings [32], which studied the benefit of interaction for the problem of extrinsic diversity. Similarly, user interaction has been found to help in more structured and faceted search tasks [41, 13], in cases such as product search. However, while presenting interactive, dynamic rankings is one user experience that offers a way to surface the improved relevance to users, our techniques are more general: they may be used to present a summary of the topic to the user, recommend unexplored options, anticipate and then crowdsource queries to trade off latency and quality by prefetching, and more.

In contrast to previous work, we provide a way to not only identify complex search tasks that will require multiple queries but to *proactively* retrieve for future queries before the user has searched for them. Importantly, these future queries are neither simple reformulations nor completely unrelated, but are queries on the particular task that the user has started. Finally, we introduce diversification methods which, unlike previous methods, maintain coherence around the current theme while diversifying. Using these methods we demonstrate that we can improve retrieval relevance for a *task* by detecting an intrinsically diverse need and providing whole-session retrieval at that point.

# 3. INTRINSICALLY DIVERSE TASKS

An *intrinsically diverse task* is one in which the user requires information about *multiple, different aspects* of the *same topical* information need. In practice, a user most strongly demonstrates this interest by issuing multiple queries about different aspects of the same topic. We are particularly interested in identifying the common theme of an intrinsically diverse task *and* when a user initiated the task. We unify these into the concept of an *initiator query* where, given a set of queries on an intrinsically diverse task, the query among them that is most general and likely to have been the first among these set of queries is called the initiator query. If multiple such queries exist, then the first among them from the actual sequence (issued by the user) is considered the initiator. We give importance to the temporal sequence since the goal is to detect the initiation of the task and provide support for it as soon as possible.

While previous work has defined the concept of intrinsic diversity, there has been no further understanding of the problem or means to obtain data. We now identify and analyze authentic instances of intrinsically diverse search behavior, extracted from large-scale mining and analysis of query logs from a commercial search engine.

## 3.1 Mining intrinsically diverse sessions

Intuitively, intrinsically diverse (ID) tasks are topically coherent but cover many different aspects. To automatically identify ID tasks *in situ* where a user is attempting to accomplish the task, we seek to codify this intuition. Furthermore, rather than trying to cover all types of ID tasks, we focus on extracting with good precision and accuracy a set of tasks where each task is contained within a single search session. As a "session" we take the commonly used approach of demarcating session boundaries by 30 minutes of user inactivity [35]. Once identified, these mined instances could potentially be used to predict broader patterns of cross-session intrinsic diversity tasks [24, 1], but we restrict this study to mining and predicting the initiation of an ID task within a search session and performing whole-session retrieval at the point of detection.

To mine intrinsically diverse sessions from a post-hoc analysis of behavioral interactions signals with the search results, we developed a set of heuristics to detect when a session is topically coherent but covering many aspects. These can be summarized as finding sessions that are: (1) longer – the user must display evidence of exploring multiple aspects; (2) topically coherent – the identified aspects should be related to the same overall theme rather than disparate tasks or topics; (3) diverse over aspects – the queries should demonstrate a pattern beyond simple reformulation by showing diversity. Furthermore, since the user's interaction with the results will be used in lieu of a contextual relevance judgment for evaluation, we also desire that we have some "satisfied" or "long-click" results where we define a satisfied (SAT) click similar to other work as having a dwell of $\geq 30s$ or terminating the search session [11, 12].

Given these criteria, we propose a simple algorithm to collect intrinsically diverse user sessions. Our algorithm uses a series of filters, explained in more detail below. When we refer to "removing" queries, we mean they were treated as not having occurred for any subsequent analysis steps. For sessions, with the exception of those we "remove" from further analysis in Step 4, we label all other sessions as intrinsically diverse or *regular* (*i.e.*, not ID). We identify the *initiator* query as the first query that remains after all query removal steps, and likewise a *successor* query is any remaining query that follows the initiator in the session. More precisely, we use the following steps (in sequence) to filter sessions:

1. **Remove frequent queries:** Frequent queries – such as *facebook* or *walmart* – that are often interleaved with more complex tasks can obscure the more complex task the user is accomplishing. Therefore, we remove the top 100 queries by frequency as well as frequent misspellings related to these queries.
2. **Collapse duplicates:** We collapse any duplicate of a query issued later in the session as representing the same aspect but record all SAT clicks across the separate impressions.
3. **Only preserve manually entered queries:** To focus on user-driven exploration and search, we removed queries that were not manually entered, *e.g.* those obtained by clicking on a link such as by query suggestion or searches embedded on a page.
4. **Remove sessions with no SAT Document:** Since we would like to eventually measure the quality of rerankings for these session queries in a personal and contextual sense, we would like to ensure that there is at least one long-dwell click to treat as a relevance judgment. While this is not required for a session being an ID session, we simply require it for ease of evaluation. Thus, we removed sessions with no SAT clicks.
5. **Ensure topical coherence:** As ID sessions have a common topic, we removed any successor query that did not share at least one common top ten result with the initiator query. Note that this need not be the same result for every aspect. While this restricts the set of interaction patterns we identify, it enables us to be more precise, while ensuring semantic relatedness, and does not rely on the weakness of assuming one fixed static ontology.

6. **Ensure diversity in aspects:** Although we desire topical coherence across the queries, we do not want to identify simple reformulations or spelling corrections as aspects. Thus we restrict the syntactic similarity with the initiator query to avoid identifying trivial difference as substantially different aspects. To measure query similarity robust to spelling variations, we consistently use *cosine similarity with character trigrams* in this work. In particular, we remove queries where the similarity was more than 0.5.
7. **Remove long queries:** We observed a small fraction of sessions matching the above filters appear to consist of copy/paste homework questions on a common topic. While potentially interesting, we focus in this paper on completely user-generated aspects and introduce a constraint on query length, removing queries of length at least 50 characters.
8. **Threshold the number of distinct aspects:** Finally, to focus on diversity and complexity among the aspects, we threshold on the number of distinct successor queries. We identify a query as distinct when its maximum pairwise (trigram character cosine) similarity with any preceding query in the session is less than 0.6. Any session with less than three distinct aspects (including the initiator) are labeled as regular and those with three or more aspects are labeled as intrinsically diverse.

Putting everything together, we ran this algorithm on a sample of user sessions from the logs of a commercial search engine from the period April 1–May 31, 2012. We used log entries generated in the English-speaking United States locale to reduce variability caused by geographical or linguistic variation in search behavior. Starting with 51.2M sessions comprising 134M queries, applying all but the *SAT-click* filter, with the *Number of Distinct Aspects* threshold at two, led to more than 497K ID sessions with 7.0M queries. These ID tasks accounted for 1.0% of all search sessions in our sample, and 3.5% of sessions having 3 queries or more (14.4M sessions)[1]. Further applying the SAT-click filter reduced the number to 390K. Finally, focusing on the more complex sessions by setting the Number of Distinct Aspects filter to three, reduced this to 146K sessions.

Given that ID sessions require multiple queries, we hypothesize that ID sessions account for a disproportionately larger fraction of *time spent searching* by all users. To test this, we estimated the time a user spent in a session by the elapsed time from the first query to the last action (*i.e.,* query or click). Sessions with a single query and no clicks were assigned a constant duration of 5 seconds. Here, the time in session includes the whole session once an ID task was identified in that session. Our hypothesis was confirmed: while ID sessions with at least 2 distinct aspects represented 1.0% of all sessions, they accounted for 4.3% of total time spent searching, showing the significant role ID sessions play in overall search activity.

To assess the accuracy of our automatic labeling process, we sampled 150 sessions (75 each from the auto-labeled regular and intrinsic sets) of length at least 2 queries. We ignored single query sessions since those are dominated by regular intents and there may be a bias in labeling. Two assessors were given instructions similar to the description in the first paragraph of Section 3, examples of ID sessions such as those in Table 1, and all of the queries in the session and asked to label each session as regular or ID. The assessors had a 79% agreement with an inter-rater $\kappa$ agreement of 0.5875. Using each assessor as a gold-standard and taking the average, on sessions of length two or greater our extraction method has a precision of 73.9% and an accuracy of 73.7% (overall accuracy is higher because of single query sessions always being regular). Thus, with both good agreement and a moderate to strong accuracy and precision, the method provides a suitable source of noisy supervised labels. With enough data, we can hope to overcome the noise in the labels (as long as it is unbiased) with an appropriate learning algorithm [3].

[1]Because we do not focus on more complex ID information seeking, such as tasks that span multiple sessions, the true percentage associated with ID tasks is likely to be larger.

# 4. PREDICTING INTRINSICALLY DIVERSE TASK INITIATION

Given that we may want to alter retrieval depending on whether the user is seeking intrinsic diversity or not, we ask the question whether we can identify the initiator queries for intrinsically diverse tasks and treat this as a classification problem. In particular, while in Sec. 3 we used the behavioral signals of interaction between the initiator and successor queries of a session to automatically label queries with a (weak) supervised label, here we ask if we can predict what the label would be in the absence of those interaction signals – a necessary ability if we are to detect the user's need for intrinsic diversity in an operational setting. Ultimately our goal is to enable a search engine to customize the search results for intrinsic diversity only when appropriate, while providing at least the same level of relevance on tasks predicted to be regular. Recognizing that in most operative settings, it is likely important to invoke a specialized method of retrieval only when confident, we present a precision-recall tradeoff but focus on the high precision portion of the curve.

## 4.1 Experimental Setting

**Data:** We used a sample of initiator queries from the intrinsically diverse sessions described in Sec. 3.1 as our positive examples, and the first queries (after removing common queries as in Step 1 of Sec. 3.1) from *regular* sessions were used as negative examples. Note that since the label of a query, *e.g.* `[foo]`, comes from the session context, it is possible that `[foo]` occurs in both positive and negative contexts. In order to only train to predict queries that were clearly either ID or regular, we dropped such conflicting queries from the dataset; this only occurred 1 out of every 5K ID sessions. Also to weigh each task equally instead of by frequency, we sample by type: *i.e.,* we treat multiple occurrences of a query in the positive (resp. negative) set as a single occurrence. Finally, we downsample to obtain a 1:1 ratio from the positive and negative sets to create a balanced set. Unless otherwise mentioned, the dataset was sampled to contain 61K queries and split into an 80/5/15 proportion (50000 training, 3000 validation, 8000 test) with no class bias.

**Classification:** We used SVMs[18] with linear kernels, unless mentioned otherwise. We varied the regularization parameter ($C$) over the values: $\{10^{-4}, 2{\cdot}10^{-4}, 5{\cdot}10^{-4}, 10^{-3}, \ldots, 500, 10^{3}\}$. Model selection was done using the validation

| Feature Set | Examples | Cardinality | Coverage | Normalized? | Log? |
|---|---|---|---|---|---|
| Text | Unigram Counts | 44140 | 100% | **No** | **No** |
| Stats | # Words, # Characters, # Impressions, Click Count, Click Entropy | 10 | 81% | **Yes** | **Yes** |
| POS | Part-of-Speech Tag Counts | 37 | 100% | **No** | **No** |
| ODP | Five Most Probable ODP Class Scores from Top Two Levels | 219 | 25% | **Yes** | **Yes** |
| QLOG | Average Similarity with co-session queries, Average session length, Distribution of occurrences within session (start/middle/end) | 55 | 44% | **Yes** | **No** |

**Table 2: Features used for identification of initiator queries**

set by selecting the model with the best precision using the default margin score threshold (*i.e.*, 0).

**Features:** The features are broadly grouped into 5 classes as shown in Table 2. Apart from the text and POS tag features, all other features were normalized to zero mean, unit variance. Features with values spanning multiple orders of magnitude, such as the *number of impressions*, were first scaled down via the log function. Due to the large scale of our data, coverage of some features is limited. In particular, query classification was done similar to [4] by selecting the top 9.4M queries by frequency from a year's query logs previously in time and then using a click-based weighting on the content-classified documents receiving clicks[2]. Likewise **Stats** and **QLOG** features are built from four months' worth of query logs and have limited coverage as a result. The query logs chosen to build these features were from previous to April 2012 to ensure a fair experimental setting with no overlap with the data collection period of the intrinsically diverse or regular sessions. We found the coverage of these features to be roughly the same for both the positive and negative classes.

We also note that the cardinality of some feature sets will depend on the training set (*e.g.*,vocabulary size of *Text* grows with more training data); the values listed in Table 2 are for the default training set of 50,000 queries. Most of our experiments will use all of the 5 feature sets; the effect of using only a subset of the feature sets is explored in Sec. 4.3.

## 4.2 Can we predict ID task initiation?

To begin with, we would like to know the precision-recall tradeoff that we can achieve on this problem. Figure 1 shows the precision-recall curve for a linear SVM trained on 50K examples with all the features. The result is a curve with clear regions of high precision, indicating that the SVM is able to identify initiator queries in these regions quite accurately. Furthermore, performance is better than random (precision of 50% since classes are balanced) along the entire recall spectrum.

As Table 3 shows, we are able to achieve relatively high precision values at low recall values. For example, we can identify 20% of ID tasks with 80% precision.

## 4.3 Which features were most important?

We next investigate the effect of using different subsets of the features on performance. The results are shown in Figure 2 and Table 4. First, we note that **Stats, QLOG** and **ODP** feature sets help identify only a small fraction of the initiator queries but do so with high precision. On the other hand, the **Text** and **POS** feature sets, which have high coverage, provide some meaningful signal for all the queries, but cannot lead to high precision classification. We also find that a combination of features, such as the **Text** and **Stats** features, can help obtain higher precision as well as higher recall than either alone. In fact, such combinations perform almost as well as using all features, which is the best out of all feature combinations.

[2]For greater coverage this could be extended to a rank-weighted back-off as described in that paper.

| Recall @ Precision | | Precision @ Recall | |
|---|---|---|---|
| 5.9 | 90 | 84.9 | 10 |
| 9.8 | 85 | 79.3 | 20 |
| 18.3 | 80 | 75 | 30 |
| 30.3 | 75 | 72.8 | 40 |
| 49.0 | 70 | 69.4 | 50 |
| 61.4 | 65 | 65.4 | 60 |
| 78.8 | 60 | 62.4 | 70 |

**Table 3: Recall at different precision levels and vice-versa for predicting ID task initiation.**

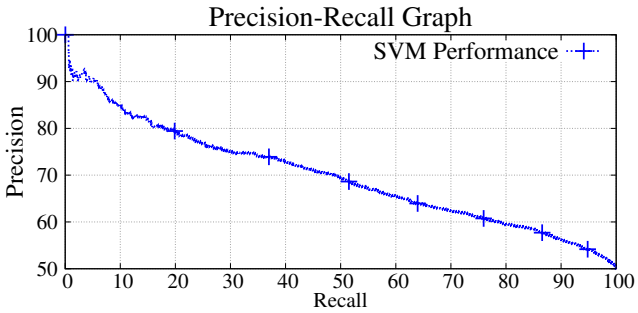


**Figure 1: P-R curve for predicting ID task initiation.**

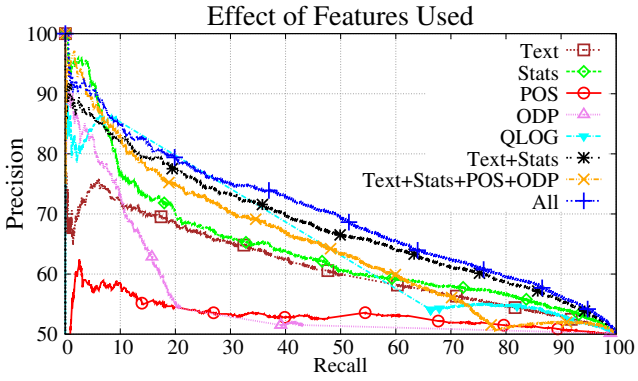


**Figure 2: Change in classification performance of initiator queries as feature sets are varied.**

## 4.4 Linguistic features of initiator queries

To further understand ID initiator queries, we identified the part-of-speech and text features most strongly associ-

| Feature Set | Rec@80%Prec | Prec@40%Rec |
|---|---|---|
| T | 0.1 | 62.6 |
| S | 9.2 | 63.7 |
| P | 0.0 | 52.8 |
| O | 5.6 | 51.6 |
| Q | 9.4 | 54.1 |
| TS | 13.6 | 69.7 |
| TSPO | 12.2 | 67.0 |
| TSPOQ | 18.3 | 72.8 |

**Table 4: Effect of feature set on precision & recall. T=Text, S=Stats, P=POS, O=ODP, Q=QLOG**

ated with them, by computing each feature's log-odds ratio (LOR)[3] compared to regular queries. Looking at the top-ranked features by LOR, we found that initiator queries are more likely to use question words (LOR=0.41); focus on proper nouns (0.40) such as places and people; use more 'filler' words (particles) found in natural language (0.27); and when they use general nouns, these tend to be plural (0.13) instead of singular (−0.052). Predominant text features indicated the importance of *list-like* nouns such as *forms, facts, types, ideas* (LOR=1.59, 1.45, 1.25, 0.92); verbs that are commonly used in questions such as *did* (1.34); and words indicating a broad need such as *information* and *manual* (1.64, 1.18). Strong negative features tend to encode exceptions – such as the most negative word *lyrics* (−2.25) used to find words to specific songs.

# 5. RE-RANKING FOR INTRINSIC DIVERSITY

While the previous section discusses the identification of queries that lead to ID tasks, in this section we discuss changes that can be made to the search results page to support queries for ID tasks. Specifically, we propose a re-ranking scheme that looks to satisfy not only the information need of the issued query, but also the future queries that the user is likely to issue later in the session on other aspects of the task. To the best of our knowledge, we are the first to address the problem of jointly satisfying the current query as well as future queries (unlike anticipatory search [25] which focuses solely on the latter).

We will use an interactive ranking-based paradigm here, using an approach related to the two-level rankings proposed in [32]. Given an issued query representing the start of an ID task, we consider rankings where each result can be **attributed** to some aspect of that task. We represent each aspect of the ID task by a related query of the issued query. One way this could be surfaced on a results page for a user is by placing the related query for an aspect adjacent to its corresponding search result. In such a setting, clicking on the related query could lead to results for that query being presented, thus enabling the user to explore documents for that aspect. This brings us to the question of how we find such a ranking.

## 5.1 Ranking via Submodular Optimization

We first describe precisely what we consider as an interactive ranking. In response to an initial query $q$, an interactive ranking $\mathbf{y} = (\mathbf{y}_D, \mathbf{y}_Q)$ comprises two parts: a ranking of documents $\mathbf{y}_D = d_1, d_2, \ldots$, which we refer to as the *primary* ranking; and a corresponding list of related queries $\mathbf{y}_Q = q_1, q_2, \ldots$, which represent the *aspects* associated with the documents of the primary ranking. The $i^{th}$ query in the list, $q_i$, represents the aspect associated with $d_i$. Structurally this can also be thought of as a ranked list of (document, related query) pairs $(d_i, q_i)$.

Given this structure, let us consider four conditions that comprise a good interactive ranking:

1. Since the documents in the primary ranking were displayed in response to the issued query $q$, they should be relevant to $q$.
2. As document $d_i$ is associated with the aspect represented by the related query $q_i$, document $d_i$ should be relevant to query $q_i$.
3. Aspects should be relevant to the ID task being initiated by the query $q$.
4. At the same time, the aspects should not be repetitive *i.e.,* there should be diversity in the aspects covered.

We now design a ranking objective function that satisfies these four conditions to jointly optimize the selection of documents and queries $(\mathbf{y}_D, \mathbf{y}_Q)$. Suppose we have an existing interactive ranking $\mathbf{y}^{(k-1)}$ that has $k-1$ (document, related query) pairs, and our goal is to construct a new ranking $\mathbf{y}^{(k)}$ by adding an optimal (document, related query) pair to $\mathbf{y}^{(k-1)}$ – an operation we denote by $\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)} \oplus (d_k, q_k)$.

Condition 1 above can be met by selecting $d_k$ such that $R(d_k|q)$ is large, where $R(d|q)$ denotes the probability of relevance of document $d$ given query $q$. Condition 2 can be met by selecting $d_k$ such that its relevance to the related query $q_k$, $R(d_k|q_k)$, is large. Conditions 3 and 4 imply a standard diversification tradeoff, but here we have that the aspects $q_k$ should be related to the initial query $q$ and diverse. If we use a similarity function between queries to estimate the relevance between queries, Condition 3 implies that the similarity function $Sim(q, q_k)$ between $q_k$ and $q$ should be large. Condition 4 requires that the diversity should be maximized between $q_k$ and all previous queries $\mathcal{Q} = q_1, \ldots, q_{k-1}$. Both Condition 3 and 4 can be jointly obtained by optimizing an MMR-like *diversity function* [6], $Div(q_k, \mathcal{Q})$, as described below. Intuitively, we would also like the change in the objective function on adding document-query pair $(d_k, q_k)$ to the ranking $\mathbf{y}$ to be no smaller than what we would gain if adding the pair to a larger ranking $\mathbf{y} \oplus \mathbf{y}'$: that is, the objective function should be *monotone* and *submodular*. Submodular objectives are desirable because they have the property that they can be optimized using a simple and efficient *greedy* algorithm which iteratively computes the next best $(d, q)$ pair to add to the ranking. Using the greedy algorithm ensures that the computed solution is at least $(1 - \frac{1}{e})$ times as good as the optimal.

We now consider the following objective satisfying the above conditions[4]:

$$\text{argmax}_{(d_1,q_1)\cdots(d_n,q_n)} \sum_{i=1}^{n} \gamma_i \cdot R(d_i|q) \cdot R(d_i|q_i) \cdot e^{\beta Div(q_i,\mathcal{Q})}$$

where $\mathcal{Q}$ is shorthand for the set of queries $Q = \{q_1, \ldots, q_n\}$,

[3]The LOR can be thought of as an approximation to the weight in a single-variable logistic regression.

[4]We omit the straightforward submodularity proof for space reasons.

and $Div(\cdot)$ is an MMR-like diversity function defined as

$$Div(q_i, \mathcal{Q}) = \lambda \cdot Sim(q_i, Snip(q)) - (1-\lambda) \max_{j<i} Sim(Snip(q_i), Snip(q_j)). \quad (1)$$

Here, $\lambda \in [0,1]$ and $\beta > 0$ are parameters, where $\lambda$ controls the tradeoff between related query aspect relevance and diversity while $\beta$ controls the rate at which returns diminish from additional coverage. Finally, $\gamma_i$ refers to the discount factor for position $i$: we use the common $\frac{1}{\log_2(i+1)}$ DCG discounting.

This objective can be interpreted as maximizing an expected utility (the exponential term) of covering related and diverse aspects where the expectation is over the maximum joint relevance of a document to both the initial query and the related query aspect. Furthermore, the joint probability is assumed to be conditionally independent to factor into the two relevance terms.

In this study, we define $Sim(x, y)$ as the cosine similarity between word-TF representations of $x$ and $y$, and $Snip(q_j)$ is the bag-of-words representation of caption text from the top-10 search results for $q_j$ using relevance score $R(d|q_j)$ alone. The MMR-like term appears within the exponent to ensure the objective is monotone.

Note that while the final objective optimizes for an interactive ranking, the primary ranking itself aims to present results from other aspects. We optimize this using the greedy algorithm presented in Algorithm 1, which we refer to as the **DynRR** method. In Alg. 1, the function $RelQ(q)$ denotes a function that returns related queries for query $q$, and $Top(\mathbf{y}_D)$ returns the top element in the ranking $\mathbf{y}_D$.

**Algorithm 1** Greedy-DynRR($\beta, \lambda, P(\cdot|\cdot), q$)

```
(y_D, y_Q) ← φ
for all q' ∈ RelQ(q) do
    Next(q') ← Document Ranking by R(·|q) · R(·|q').
for i = 1 → n do
    bestU ← −∞
    for all q' ∈ RelQ(q)/ y_Q do
        d' ← Top(Next(q')/ y_D)
        v ← R(d'|q) · R(d'|q') · e^{β·Div(q', y_Q)}
        if v > bestU then
            bestU ← v
            bestQ ← q'
            bestD ← d'
    (y_D, y_Q) ← (y_D, y_Q) ⊕ (bestD, bestQ)
return y
```

## 5.2 Evaluation Measures

As the problem of presenting results for both the current as well as future queries is a new one, we first discuss the evaluation methodology used. In particular, we use two kinds of evaluation metrics:

**Primary ranking metrics**: To compare against standard non-interactive methods of ranking, we simply evaluate the quality of the primary ranking, *i.e.,* completely ignore the related query suggestions attributed to documents. Since our goal is *whole-session relevance*, documents are considered *relevant* if and only if they are relevant to any query in the session. Given this notion of relevance, we compute the Precision, MAP, DCG and NDCG values.

| Dataset | # Train | # Test |
|---|---|---|
| MINED | 8888 | 2219 |
| MIXED | 4120 | 1027 |

**Table 5: Datasets used in re-ranking experiments**

**Interactive ranking metrics**: To evaluate the offline effectiveness and accuracy of the *predicted* future aspects (queries) and results (documents), we need to assume some model of human interaction. Consider the following search user model:

1. Users begin at the top of the ranking.
2. They click/expand the related query attributed to a document *if and only if* the document is relevant or the query is relevant. (We say a query is *relevant* if the top $k$ results of the query contain a (new) relevant document.)
3. On expanding the related query, the user views the top $k$ results for that related query, before returning to the original document ranking and continuing.
4. Users ignore previously seen documents, and click on all *new* relevant documents.

Under this user model, we can easily trace the ranking of documents that the user navigates and thus evaluate Precision@10 and DCG@10 for this ranking. We refer to these metrics as $PrecU_k$ and $DCGU_k$, and compare them with the primary Prec@10 and DCG@10 metrics.

We do not claim that this user model accurately captures all online users, nor that it is sophisticated. This is simply a well-motivated model for analyzing a rational user's actions, assuming the user is relatively accurate at predicting the relevance of an aspect based on either the top document or its related query. This in turn is intended to inform us about trends and relative differences we may see in online studies.

## 5.3 Experimental Setup

**DATA:** To evaluate the efficacy of the method, we used the data obtained from mining the search logs, as described in Section 3. We used two main datasets shown in Table 5. To analyze impact when most of the sessions are ID and more complex, the MINED dataset is obtained directly from the filtering algorithm by setting the threshold on the Number of Distinct Aspects to be 5. To determine the re-ranking impact when sessions may be a mixture of both ID and regular sessions, the MIXED dataset was obtained by predicting when a session was ID using the classifier from Sec. 4 over a mixture of the MINED dataset sessions and a random sample of regular sessions of the same size. More specifically, the combined sessions were split in a 45-10-45 split of training-validation and test sets. The trained classifier was used to classify the test set sessions as being ID or not, based on the initiator query. The sessions predicted as ID formed the *MIXED* dataset (prediction accuracy of 68.8% over the combined sessions); for those not predicted to be ID, we assume the standard ranking algorithm would be applied and thus relevance would be the same on those. The MIXED dataset is a reflection of an operational setting, where the query issued is used to predict if the resulting session will be an ID session or not, and the ones predicted to be ID are selected for re-ranking.

**Obtaining Probability of Relevance:** For our algorithm, we required the computation of the conditional relevance of a document given a query *i.e.,* $R(d|q)$. Thus, to en-

| Query | Length |
|---|---|
| Website | Log(PageRank) |
| Baseline Ranker | Reciprocal Rank (if in top 10) |
| URL | Length<br># of Query Terms Covered<br>Fraction of Query Covered<br>TF Cosine sim<br>LM Score(KLD)<br>Jaccard<br>Boolean AND Match<br>Boolean OR Match |
| Anchor (Weighted) | Same as URL |
| Anchor (Unweighted) | TF-Cosine Sim<br>KLD Score |

**Table 6: The 21 features used to train R(d|q).**

able easier reproducibility by others, we learned a model using **Boosted Regression Trees**, on a dataset labeled with the relevance-values for query-document pairs with 20,000 queries using graded relevance judgments (∼60 documents per query). The features used are given in Table 6. Features were all normalized to 0 mean, unit variance. To obtain the final model, we optimized for NDCG@5.

**Baselines:** As baselines we used the following methods:

- **RelDQ**: Ranking obtained by sorting as per $R(d|q)$.
- **Baseline**: A state-of-the-art commercial search engine ranker (also used to compute the rank feature mentioned earlier).

We also computed performance of other baselines, such as MMR and relevance-based methods such as BM-25 (using the weighted anchor text), but found them to perform far worse than RelDQ and Baseline and hence do not present the results for such other baselines.

**Related Queries:** To study the effect of the related queries, we used four different sources:

- **API**: We used the publicly available API of a commercial search engine (which returns 6-10 related queries)
- **Click-Graph**: Using co-click data, we obtained a set of $10-20$ related queries.
- **Co-Session Graph**: Using data of queries co-occurring in the same session, we obtained $10-20$ related queries.
- **Oracle**: As an approximate upper bound, we used the actual queries issued by the user during the session.

To ensure fairness, the graphs were constructed using data prior to April 2012. For most experiments, we only use the first 3 sources or only the second and third (which we distinguish by the suffix **C+S**).

**Settings:** The parameters for DynRR were set by optimizing for $DCGU_3$ on the training data[5]. All numbers reported here are for the test sets. We considered all SAT-clicked results in the session as relevant documents; since we compare *relative* to the baseline search engine, the assumption is that placing the SAT-clicked documents higher is better, rather than being an indication of absolute performance. Unless otherwise mentioned, the candidate document set for re-ranking comprises the union of the top 100 results (from the Baseline method) of the initiator query, and the top 10 results from each related query.

[5]We varied the $\lambda$ parameter from 0 to 1 in increments of 0.1, while the $\beta$ parameter was varied across the values $\{0.1, 0.3, 1, 3, 10\}$.

| PREC | Mined | Mixed | DCG | Mined | Mixed |
|---|---|---|---|---|---|
| $PrecU_1$ | 1.093 | 1.103 | $DCGU_1$ | 1.075 | 1.074 |
| $PrecU_2$ | 1.247 | 1.223 | $DCGU_2$ | 1.188 | 1.153 |
| $PrecU_3$ | 1.347 | 1.295 | $DCGU_3$ | 1.242 | 1.190 |
| $PrecU_5$ | 1.401 | 1.345 | $DCGU_5$ | 1.254 | 1.204 |

**Table 8: Interactive Performance of DynRR for different user models (as ratios compared to the Baseline Prec@10 and DCG@10)**

| Set | Comp. Metric | % Gains | | | % Losses | | |
|---|---|---|---|---|---|---|---|
| | | 0.2 | 0.5 | 1.0 | 0.2 | 0.5 | 1.0 |
| Mined | $DCGU_3$ | 34.4 | 13.0 | 1.6 | 9.9 | 2.7 | 0.1 |
| | DCG@10 | 19.6 | 5.2 | 0.3 | 12.7 | 3.8 | 0.3 |
| Mixed | $DCGU_3$ | 29.1 | 12.0 | 1.6 | 10.8 | 3.7 | 0.2 |
| | DCG@10 | 17.7 | 6.0 | 0.8 | 12.9 | 4.0 | 0.2 |

**Table 9: % of sessions for which the metric performance of DynRR differs from the Baseline DCG@10 by more than a certain threshold.**

## 5.4 Results

**Primary Evaluation:** We first study the re-ranking without any interactivity *i.e.,* using the *primary* ranking metrics to evaluate the quality of the top-level ranking. As seen in the results of Table 7, the re-ranking leads to improvements across the different metrics for both datasets. Thus, even without interactivity, the method is able to outperform the baselines in predicting future results of interest to the user, while also providing results for the current query. In particular, we found the DynRR method works best using the **C+S** related queries (which we return to later) with 9-11% gains over the baselines at position 10 across the various metrics with 3-5% relative gains. We also find that the method improves on the MIXED dataset supporting the question of whether the method can be robustly used in practical scenarios. Thus we improve an important segment of tasks while maintaining high levels of performance elsewhere; further improvements to the initiator classification model will improve the robustness further.

**Interactive Evaluation:** Next we evaluate the performance of the method while incorporating the interactivity. As seen in Table 8, the added interactivity leads to large increases in both the precision and DCG of the user paths navigated, across the different user models and datasets. In fact, we find 30-40% improvements in precision and 20-25% improvements in DCG, indicating that we are able to do a far better job in predicting future relevant results, and potentially, queries. These results also show that the method improvements are relatively robust to the user model.

**Robustness:** A key concern when comparing a new method against a baseline, is the robustness of the method. In particular, we are interested in the number of queries that are either improved or hurt, when switching from the Baseline method to the proposed re-ranking method. This is particularly crucial for the MIXED dataset, as we would want that the performance on non-ID sessions not be severely affected. Table 9 displays the % of examples for which the method either gains or loses above a certain threshold, compared to the Baseline method. We see that the number of gains far exceeds the number of losses, especially while comparing the interactive metric. We should also note that for

| Set | Method | Prec | | | MAP | | | DCG | | | NDCG | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | @1 | @3 | @10 | @1 | @3 | @10 | @1 | @3 | @10 | @1 | @3 | @10 |
| Mined | RelDQ | 1.00 | 0.94 | 0.97 | 1.00 | 0.97 | 0.98 | 1.00 | 0.97 | 0.99 | 1.00 | 0.97 | 0.99 |
| | DynRR | **1.06** | **1.03** | **1.02** | **1.06** | **1.05** | **1.04** | **1.06** | **1.04** | **1.04** | **1.06** | **1.05** | **1.05** |
| | DynRR C+S | **1.10** | **1.09** | **1.09** | **1.10** | **1.10** | **1.10** | **1.10** | **1.10** | **1.11** | **1.09** | **1.10** | **1.11** |
| Mixed | RelDQ | 1.00 | 0.94 | 0.99 | 1.00 | 0.98 | 0.98 | 1.00 | 0.96 | 0.98 | 1.00 | 0.97 | 0.98 |
| | DynRR | **1.03** | **1.02** | **1.04** | **1.03** | **1.04** | **1.03** | **1.03** | **1.03** | **1.03** | **1.03** | **1.03** | **1.05** |

**Table 7: Primary Performance of different methods(as a ratio compared to the Baseline)**

| RelQ | Prec | DCG | $PrecU_3$ | $DCGU_3$ | $SDCG$ |
|---|---|---|---|---|---|
| A | 0.905 | 0.880 | 1.082 | 0.997 | 1.174 |
| C | 1.015 | 1.014 | 1.333 | 1.214 | 1.488 |
| S | 1.051 | 1.074 | 1.248 | 1.198 | 1.384 |
| O | 1.476 | 1.397 | 2.211 | 1.827 | 2.500 |
| AS | 0.961 | 0.961 | 1.271 | 1.157 | 1.452 |
| AC | 0.986 | 1.013 | 1.244 | 1.176 | 1.408 |
| CS | 1.089 | 1.106 | 1.413 | 1.306 | 1.593 |
| ASC | 1.019 | 1.039 | 1.347 | 1.242 | 1.529 |
| ASCO | 1.179 | 1.144 | 1.580 | 1.386 | 1.802 |

**Table 10: Performance change on varying the related queries. All measures are @10 and reported as a ratio to the baseline values.**

| Task | Fleiss Kappa | % All agree | % 2 agree |
|---|---|---|---|
| IsTopicID? | .423 | 85.5 | 100 |
| AreQueriesID? | .452 | 67.1 | 100 |
| BestInitiatorQ | .694 | 55.3 | 98.7 |

**Table 11: Annotator agreement on TREC data.**

both datasets and both metrics, the DynRR method is statistically significantly better than the Baseline method, as measured by a binomial test at the 99.99% significance level.

**Effect of related query set:** Next we study the impact of the related queries on the method performance, using the MINED dataset. To do so, we considered different combinations of the four related query sources: API(**A**), ClickGraph(**C**), Co-Session(**S**) and Oracle(**O**). Table 10 shows the results. As we clearly see, the related query source can make a significant impact on both the primary ranking performance and the interactive performance. One thing which stands out is the extremely strong performance using the Oracle related queries, which suggest that any improvements we can make in the quality of the suggested related queries is likely to result in even better overall performance. On the other hand, we see that using the API related queries almost always hurts performance. In fact, simply using only the related queries from the click-graph and the co-session data leads to much better performance than that compared to using the API queries as well. Further analysis reveals that this is due to two reasons: (a) In many cases, the queries returned by the API are spelling corrections or reformulations, with no difference in aspect; (b) most importantly though, there is little to no diversity in the queries obtained from the API compared to those from the other sources.

## 5.5 TREC Session Data

We also ran experiments using the publicly available TREC 2011 Session data (which was constructed with ID topics in mind) using only publicly reproducible components. To do so, three assessors labeled the different sessions as *potentially* being intrinsically diverse or not, based: a) only on the queries issued; and b) on the narration and title of the session as well. We also asked them to label their opinion on the query best suited to be the initiator query, among the queries issued. Annotators were provided the definition of ID sessions as described at the start of Section 3. We found good agreement among the different annotators for all the different labeling tasks, as seen from Table 11. In fact, in 63 of the 76 total sessions all three annotators agreed the sessions were ID based on the narration, title, and queries. We used a 50-50 training-test split on all sets, with the training data used for selecting the parameters of the ranking methods. To obtain the conditional relevance $R(d|q)$, we trained a regularized linear regression model with features based on the scores of two standard ranking algorithms: BM25 and TFIDF. As labeled data we used the TREC Web data from 2010 and 2011, by converting the graded relevance scores for relevant and above from the $\{1, 2, 3\}$ scale to $\{\frac{1}{3}, 1, 1\}$. We used related queries from the Van Dang-Croft [10] method (**Q**) on the ClueWeb '09 anchor text, where the starting seed for the random walk would use the most similar anchor text to the query by TFIDF-weighted cosine if an exact match was not available. Our candidate document pool was set similar to the previous experiments. To evaluate, we again use the same metrics as before except using the TREC assessor relevance labels instead of clicks. We considered three different candidates for the initiator query: (a) Topic; (b) First query in the session; and (c) Labeled initiator query. As a baseline, we considered the method that ranked as per $R(d|q)$. For the DynRR method, we used the titles of the top 10 results of a query (as per the baseline), as the *snippet* of the query, since snippets were not made available. The results for the primary metric comparison are shown in Table 12. As we see from the table, the method improves in precision and DCG in most cases, with particularly large improvements when the title of the *topic* is used as the initiator query. This matches feedback the assessors gave us that the titles looked much more like the general queries issued by web users; in contrast, the TREC sessions would often start with a specific query before moving to a more general query. It could be that supplying the user with a well-formulated topic description before starting the search task influences

| Initiator | Method | Pr@1 | Pr@3 | DCG@1 | DCG@3 |
|---|---|---|---|---|---|
| Title | Baseline | 0.58 | 0.60 | 0.84 | 2.13 |
| Title | DynRR | $0.71^{\dagger}$ | 0.60 | $1.39^{\dagger}$ | 2.41 |
| First | Baseline | 0.53 | 0.47 | 0.94 | 1.94 |
| First | DynRR | 0.5 | 0.48 | 0.92 | 1.97 |
| Label | Baseline | 0.55 | 0.51 | 0.87 | 1.95 |
| Label | DynRR | 0.61 | 0.5 | 1.13 | 2.09 |

**Table 12: Absolute performance on TREC Session data. $^{\dagger}$ indicates significance at $p = 0.05$ by a paired one-tailed $t$-test.**

the user to search for a particular aspect, rather than issue a more general query as they might when no topic description is explicitly formulated.

# 6. CONCLUSIONS AND FUTURE WORK

In this paper, we studied *intrinsically diverse tasks* that typically require multiple user searches on different aspects of the same information need. As this is just the first step into this problem, this also opens many interesting future directions – such as iterative ways to combine the mining and query identification process or extending these techniques to other related problems like exploratory search. In this work, we motivated the problem using real-world data and presented an algorithm to mine data from search logs using behavioral interaction signals within a session. We then looked at the problem of identifying the queries that start these sessions, and treated it as a classification problem, along with an analysis of these queries. Finally, we presented an approach to alter the rankings presented to the user, so as to also provide them information on aspects of the task for which the user will search in the future. We validated our approach empirically using search log data, as well as TREC data, demonstrating significant improvement over competitive baselines in both cases.

# 7. REFERENCES


[1] E. Agichtein, R. White, S. Dumais, and P. Bennett. Search, Interrupted: Understanding and Predicting Search Task Continuation. In *SIGIR '12*, 2012.

[2] P. Bailey et al. User task understanding: a web search engine perspective. http://research.microsoft.com/apps/-pubs/default.aspx?id=180594, 2012.

[3] P. L. Bartlett, M. I. Jordan, and J. M. Mcauliffe. Large Margin Classifiers: Convex Loss, Low Noise, and Convergence Rates. In *NIPS 16*, 2004.

[4] P. N. Bennett, K. Svore, and S. Dumais. Classification-Enhanced Ranking. In *WWW '10*, 2010.

[5] C. Brandt, T. Joachims, Y. Yue, and J. Bank. Dynamic Ranked Retrieval. In *WSDM '11*, 2011.

[6] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98*, 1998.

[7] H. Chen and D. R. Karger. Less is more: Probabilistic models for retrieving fewer relevant documents. In *SIGIR '06*, 2006.

[8] C. L. Clarke et al. Novelty and diversity in information retrieval evaluation. In *SIGIR '08*, 2008.

[9] W. Dakka et al. Automatic Discovery of Useful Facet Terms. In *SIGIR 2006 Workshop on Faceted Search*, 2006.

[10] V. Dang, M. Bendersky, and W. B. Croft. Learning to rank query reformulations. In *SIGIR '10*, 2010.

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

[12] J. Gao, W. Yuan, X. Li, K. Deng, and J. Nie. Smoothing clickthrough data for web search ranking. In *SIGIR '09*, 2009.

[13] S. Gollapudi, S. Ieong, A. Ntoulas, and S. Paparizos. Efficient query rewrite for structured web queries. In *CIKM '11*, 2011.

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

[15] A. Hassan and R. W. White. Task tours: helping users tackle complex search tasks. In *CIKM '12*, 2012.

[16] J. He et al. CWI at TREC 2011: session, web, and medical. In *TREC '11*, 2012.

[17] K. Järvelin, S. L. Price, L. M. L. Delcambre, and M. L. Nielsen. Discounted cumulated gain based evaluation of multiple-query IR sessions. In *ECIR'08*, 2008.

[18] T. Joachims. Making large-scale support vector machine learning practical. In *Advances in kernel methods*, pages 169–184. MIT Press, 1999.

[19] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *TOIS*, 25(2), Apr. 2007.

[20] R. Jones and K. Klinkner. Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs. In *CIKM '08*, 2008.

[21] E. Kanoulas, B. Carterette, P. D. Clough, and M. Sanderson. Evaluating multi-query sessions. In *SIGIR '11*, 2011.

[22] E. Kanoulas et al. Overview of the TREC 2011 Session Track. In *TREC '11*, 2012.

[23] C. Kohlschutter, P.-A. Chirita, and W. Nejdl. Using link analysis to identify aspects in faceted web search. In *SIGIR '06*, 2006.

[24] A. Kotov, P. N. Bennett, R. W. White, S. T. Dumais, and J. Teevan. Modeling and analysis of cross-session search tasks. In *SIGIR '11*, 2011.

[25] D. J. Liebling, P. N. Bennett, and R. W. White. Anticipatory search: using context to initiate search. In *SIGIR '12*, 2012.

[26] J. Liu and N. Belkin. Personalizing information retrieval for multi-session tasks: The roles of task stage and task type. In *SIGIR '10*, 2010.

[27] J. Liu, M. J. Cole, C. Liu, R. Bierig, J. Gwizdka, N. J. Belkin, J. Zhang, and X. Zhang. Search behaviors in different task types. In *JCDL '10*, 2010.

[28] H. Ma, M. R. Lyu, and I. King. Diversifying query suggestion results. In *AAAI '10*, 2010.

[29] D. Morris, M. R. Morris, and G. Venolia. Searchbar: A search-centric web history for task resumption and information re-finding. In *CHI '08*, 2008.

[30] F. Radlinski, P. N. Bennett, B. Carterette, and T. Joachims. Redundancy, diversity and interdependent document relevance. *SIGIR Forum*, 43(2):46–52, Dec. 2009.

[31] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *KDD '05*, 2005.

[32] K. Raman, T. Joachims, and P. Shivaswamy. Structured learning of two-level dynamic rankings. In *CIKM '11*, 2011.

[33] A. Singla, R. White, and J. Huang. Studying trailfinding algorithms for enhanced web search. In *SIGIR '10*, 2010.

[34] A. Slivkins, F. Radlinski, and S. Gollapudi. Learning optimally diverse rankings over large document collections. In *ICML '10*, 2010.

[35] R. White and S. Drucker. Investigating behavioral variability in web search. In *WWW '07*, 2007.

[36] R. W. White, P. N. Bennett, and S. T. Dumais. Predicting short-term interests using activity-based search context. In *CIKM '10*, 2010.

[37] R. W. White, G. Marchionini, and G. Muresan. Editorial: Evaluating exploratory search systems. *Inf. Process. Manage.*, 44(2):433–436, Mar. 2008.

[38] X. Yuan and R. White. Building the trail best traveled: effects of domain knowledge on web search trailblazing. In *CHI '12*, 2012.

[39] Y. Yue and T. Joachims. Predicting diverse subsets using structural SVMs. In *ICML '08*, 2008.

[40] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR '03*, 2003.

[41] L. Zhang and Y. Zhang. Interactive retrieval based on faceted feedback. In *SIGIR '10*, 2010.

[42] Q. Zhao et al. Time-dependent semantic similarity measure of queries using historical click-through data. In *WWW '06*, 2006.