Search Result Diversity for Informational Queries

Michael J. Welch*
Yahoo! Inc.
Sunnyvale, CA 94089
mjwelch@yahoo-inc.com

Junghoo Cho
UCLA Computer Science Dept
Los Angeles, CA 90095
cho@cs.ucla.edu

Christopher Olston
Yahoo! Research
Santa Clara, CA 95054
olston@yahoo-inc.com

ABSTRACT

Ambiguous queries constitute a significant fraction of search instances and pose real challenges to web search engines. With current approaches the top results for these queries tend to be homogeneous, making it difficult for users interested in less popular aspects to find relevant documents. While existing research in search diversification offers several solutions for introducing variety into the results, the majority of such work is predicated, implicitly or otherwise, on the assumption that a single relevant document will fulfill a user's information need, making them inadequate for many *informational* queries. In this paper we present a search-diversification algorithm particularly suitable for informational queries by explicitly modeling that the user may need more than one page to satisfy their need. This modeling enables our algorithm to make a well-informed tradeoff between a user's desire for *multiple* relevant documents, probabilistic information about an average user's interest in the subtopics of a multifaceted query, and uncertainty in classifying documents into those subtopics. We evaluate the effectiveness of our algorithm against commercial search engine results and other modern ranking strategies, demonstrating notable improvement in multiple document scenarios.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation, Performance

Keywords

search diversity, expected hits, informational queries

1. INTRODUCTION

Web search engines typically display a linear list of results for a user query, ranked by numerous factors such as relevance to the search terms and overall popularity. Search queries, however, are often underspecified, ambiguous, or multifaceted. The query "virus" could refer to, for example, a computer virus or a biological virus, and it is nearly impossible to know which meaning the user intended. With an ambiguous query, a few interpretations often dominate the top results, leaving less popular aspects uncovered. Users interested in less prevalent meanings encounter difficulty finding relevant documents.

Ambiguous queries constitute a significant fraction of query instances [19, 20], and we must find suitable ways to cope with them. Studies on search diversification aim to address this problem by introducing a diverse set of pages into search results. Common to a majority of prior research, however, is the "single relevant document assumption." In fact some proposed approaches are provably optimal for various retrieval metrics under the assumption a user requires only *one* relevant document from their intended subtopic. We argue, however, that this assumption is an over simplification. Many users will not be satisfied with only one relevant document, particularly for *informational* queries, and a search diversification strategy must properly account for them.

In this paper we concentrate on the problem of diversifying search results for informational queries. Improving the results for informational queries will significantly improve the search experience for many users because they tend to spend a disproportionate amount of time on informational queries. That is, navigational queries result in short interactions because the user already has a particular website in mind and simply uses the search engine as a pseudo-bookmark to locate the URL. For informational queries, however, the exact documents of interest are not known in advance. Users typically inspect the results for an informational query more in-depth, carefully exploring many pages in the result set [11]. Optimizing these queries will reduce the burden placed on the user by helping them find a sufficient number of relevant documents more quickly.

The distinct search behavior for informational queries dictates the following modeling requirements: (1) Users often need more than one document to satisfy their information need, so the diversification model should properly account for users who need multiple relevant documents. (2) Ambiguous queries often have several potential subtopics. While a user tends to have one particular subtopic in mind, that subtopic is not known by the search engine. (3) The content of each document also tends to focus on only one of the possible subtopics, but the search engine lacks explicit topic classification for the majority of documents.

In this paper we present a model that accounts for the above requirements for informational queries and define a

*Work completed while author was a graduate student in the UCLA Computer Science Department.

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

measure of user satisfaction with respect to that model. We then present an algorithm which introduces diversity into search results for informational queries such that we maximize the number of users who are able to find a sufficient number of documents related to their intended subtopic. Experiments show that our algorithm increases the expected performance of the top 10 results by 130% compared to a commercial search engine and 51% over a state-of-the-art diversification algorithm [1] in certain cases, while still performing well on traditional metrics designed under single relevant document assumptions.

The remainder of this paper is organized as follows. In Section 2 we discuss related work in search diversification. We present an overview of our model and assumptions and define our goal metric in Section 3. In Section 4 we analyze two simplifications of those assumptions and explore the complete formulation of our algorithm. In Section 5 we describe a related algorithm in more detail and highlight the differences between our approaches. Section 6 describes potential methods for approximating the probability distributions necessary for our algorithm. We present our experimental results in Section 7 and discuss conclusions and areas for further research in Section 8.

2. RELATED WORK

Search diversification has been studied in several contexts with many different approaches. Early techniques focused on the content of documents already selected, traditionally weighing between measures of query relevance and relative novelty of new documents. These methods tend to produce diversity as a side effect of novelty and make no use of explicit knowledge of potential subtopics or user intent. Carbonell and Goldstein's work on Maximal Marginal Relevance (MMR) [4] is a classic example of such a strategy, which can be employed to re-rank documents and promote diversity.

Zhai, Cohen, and Lafferty [24] propose a framework which models dependent relevance and describe a generic greedy approach to ranking documents for subtopic retrieval. Their ranking strategy is based on a tradeoff between selecting documents of high value and minimizing cost, where documents which include relevant, previously uncovered information have higher value, and those that are irrelevant or repeat already seen information have a larger cost. With the goal of optimizing a ranking for their Subtopic recall (S-recall) and Subtopic precision (S-precision) metrics, they implicitly assume that a single document relevant to a category is sufficient for a user.

Chen and Karger [5] use Bayesian retrieval models and condition selection of subsequent documents by making assumptions about the relevance of the previously retrieved documents. While their approach is capable of selecting anywhere between $0 < k \leq n$ relevant documents, they focus primarily on optimizing single document ($k = 1$) and perfect precision ($k = n$) scenarios. Their model does not explicitly consider user intent or document categorizations, and it is unclear how their technique can best be applied to interleave documents from multiple subtopics into a single ranking when single document assumptions are removed.

Wang and Zhu introduce an approach to diversification based on economic portfolio theory [23]. Their model considers a "risk" tradeoff between the expected relevance of a set of documents and correlation between them, modeled as the mean and variance. They demonstrate the algorithm is capable of a wide range of "risk preferences", though it is unclear how to choose the proper parameters to maximize their algorithm's performance under our proposed model.

Pretschner and Gauch [15] present early work in modeling user profiles as weighted nodes in an explicit taxonomy, and explore methods for employing those taxonomies in search personalization for ambiguous queries. Their work shows modest gains in relevance are possible with re-ranking and filtering based on those profiles. Liu et al. [12] study the use of general and per-user profiles constructed from category hierarchies for disambiguation of user queries.

Agrawal et al. introduce a model similar to ours in [1], where their objective is to maximize the probability an average user finds *at least one* useful result. Under assumptions of probabilistic query intent and document categorization, they present a proof showing the selection of documents which optimize against that criteria is NP-hard, and offer an approximation algorithm with a bounded error from the optimal solution under certain assumptions. They also show their algorithm is optimal when all documents belong to a single category. Their algorithm does, however, contain potential weaknesses, which we explore in more depth in Section 5.

Researchers have also considered meaningful ways to evaluate the performance of search diversification and subtopic retrieval algorithms. Classic ranked retrieval metrics such as NDCG, MRR, and MAP have been augmented [6, 1] to take user intent into account. Metrics such as search length (SL) [7] and k-call [5], and their aggregated forms, are well suited to evaluate diversification of search systems under single document assumptions. The %no metric [22] measures the ability of a system to retrieve at least one relevant result in the top ten. Other metrics, such as Subtopic recall and Subtopic precision [24], explicitly measure the subtopic coverage of a result set or the efficiency at which an algorithm represents the relevant subtopics.

We use several of these existing metrics to evaluate the performance of our algorithm under single document scenarios. We also define the *expected hits* metric to evaluate diversification algorithms under the more general assumption that a user may require multiple documents. We will detail our metric in the following sections.

3. DIVERSIFICATION MODEL OVERVIEW

Given an ambiguous query, our goal is to select the set of documents which will satisfy the majority of users. Commercial search engines frequently return homogeneous document sets for such queries, which is sub-optimal in most cases. We therefore study ambiguous queries as a search diversification problem, with the goal of introducing diversity by identifying the relevant subtopics for an ambiguous query and using the probability of user interest in each of those subtopics to produce a document ranking which increases the likelihood an average user finds sufficient relevant documents.

We concentrate on informational queries, where users often require more than one relevant document. Our model takes probabilistic information about query intent, relevance of documents to the possible query subtopics, as well as the number of pertinent documents a user requires into consideration, and assumes these factors to be independent. By considering query intent likelihood, we are able to identify which subtopics are most important to the users. Document categorization probabilities help estimate how likely a doc-

ument is to satisfy a particular subtopic. Estimating how many relevant documents a user will require enables us to weigh the expected benefits of providing additional documents from already represented subtopics against exploring less covered subtopics. The assumption that users often require *multiple* documents relevant to their intended subtopic breaks with traditional work in search diversification.

Each of the necessary distributions are discussed further in the subsequent sections, followed by the definition of our goal metric.

3.1 Relevant Document Requirements

For informational queries it is important to consider how many relevant documents a user will visit. For example, if most users want to see 10 relevant documents, diversifying the results in the top 10 may actually lower the satisfaction for many users. The number of documents j a user requires to satisfy her need, however, is often relatively small. Showing a user more than j relevant documents is generally unnecessary. We model j as a distribution over the number of relevant documents a user requires: user U is expected to require j documents related to their subtopic of interest with probability $\Pr(J = j|U)$, for $j > 0$.

3.2 User Intent

User intent represents the likelihood an average user is interested in a particular facet of an ambiguous query. The user intent probability distribution is important for determining the relative importance of each subtopic. In our model, a user issues a search query for an ambiguous topic T which has m subtopics $T_1, T_2, \cdots, T_m$. For a given user U who queries for topic T, we consider a distribution over subtopics of interest to U: U is interested in subtopic T_i with probability $\Pr(T_i|U)$.

3.3 Document Categorization

Web search engines perform quite well at retrieving documents relevant to query terms. To select a diverse set of documents for an ambiguous query, however, first requires determining which subtopic(s) each document belongs to. Automatic classification is a difficult problem, and manual classification of documents is infeasible on a web scale. Accurate document categorization is also important, as it tells us, probabilistically, which subtopic(s) a particular document satisfies. We model document categorization as a probability distribution. For a document d which is relevant to topic T, we assume a distribution over the subtopics: d is relevant to T_i with probability $\Pr(T_i|d)$.

3.4 Objectives

Given the probability distributions $\Pr(J|U)$, $\Pr(T_i|U)$, and $\Pr(T_i|D)$, the absence of any additional contextualizing information, and a choice of any n documents to display, our task is to select documents such that we maximize the likelihood of user satisfaction.

To be clear about our objective, we must define "user satisfaction". The simplest satisfaction measurement could be binary: a user either does or does not find as many documents as they desired from their intended subtopic. While it is possible to define a goal function and optimize for such criteria, this model does not seem to adequately reflect the real world. If a user wants five relevant documents, but only finds four to click on, they are likely still partially satisfied. We therefore define our objective in terms of *hits*, where a *hit* constitutes a click on a document which satisfies the subtopic the user is interested in. We then achieve our goal of optimal user satisfaction by maximizing the expected number of *hits* for the average user.

Consider a simplified example where a user issues the query *virus*. Assume they are interested in biological viruses, and 3 of the returned documents R are about biological viruses. Using the required-documents distribution $\Pr(J|U)$ we can calculate how many documents the user is expected to click on. If the user is interested in one, two, or three relevant documents, they are expected to click on as many. If they are interested in more than three documents, they can only click on the three that are displayed. We thus compute the expected number of hits as:

$$E(R) = 1 \cdot \Pr(J = 1|U) + 2 \cdot \Pr(J = 2|U) + 3 \cdot \sum_{j=3}^{|R|} \Pr(J = j|U)$$

The above example shows how, given $\Pr(J|U)$, we can compute the expected number of hits for a set of documents when user intent and document categorizations are known. In reality these are not known values, but rather probability distributions. In the next section we will show how these distributions factor in to the model and present our algorithm for selecting a set of results R such that we maximize the expected number of hits.

4. DIVERSIFICATION MODEL

Our general approach is to successively select documents, at each step choosing the document which adds the maximum additional expected hits. If our goal were to return at least one relevant result, this document would most likely come from a subtopic not yet covered. In our model, however, this is not always the case, as we may benefit more users by returning additional documents from a popular subtopic.

To determine how to best select documents, we must examine the effects of the probability distributions discussed in Section 3 on the expected number of hits. We begin by analyzing two simplified cases of those distributions. First, we will assume perfect knowledge of user intent. Second, we will assume perfect document classification.

4.1 Perfect Knowledge of User Intent

The first case we examine is when we know exactly which subtopic T_i a user is interested in but document classification is probabilistic. To calculate the expected number of hits for a set of documents when we know the user intent, we must consider how many documents j the user requires, and how many of the documents presented are relevant, denoted as k. A user will click on at most j documents, so returning more than j is unnecessary. Likewise, a user will see at most k relevant documents, and thus can click on no more than k.

We compute the expected number of hits for a set of n documents R as:

$$E(R) = \sum_{j=1}^{n} \Pr(J = j|U) \sum_{k=1}^{n} \Pr(K_i = k|R) \min(j, k) \quad (1)$$

In Equation 1, K_i is defined as the event that k documents in R belong to T_i. To compute this probability, we begin by

defining the probability that no documents from R satisfy T_i as:

$$\Pr(K_i = 0|R) = \prod_{r=1}^{n}(1 - \Pr(T_i|d_r))$$

In the general case where a user requires k relevant documents, we can expand this equation to:

$$\Pr(K_i = k|R) = \Pr(T_i|d_1)\Pr(K_i = k-1|R \setminus \{d_1\}) + (1 - \Pr(T_i|d_1))\Pr(K_i = k|R \setminus \{d_1\})$$

From Equation 1, $\Pr(J|U)$ is independent of which subtopic the user is interested in, and thus only $\Pr(K_i|R)$ will be affected by the choice of documents. Since $\Pr(K_i = k|R)$ is the only term in the equation dependent on the selected documents, and the user is only interested in subtopic T_i, we can maximize the the expected number of hits $E(R)$ by selecting the documents with the highest $\Pr(T_i|D)$ values, that is, by maximizing $\Pr(K_i = n|R)$. Under these conditions, our strategy for selecting documents is similar to the greedy approach for optimizing k-call presented by Chen and Karger [5], using $\Pr(T_i|D)$ to select the documents most likely related to T_i.

4.2 Perfect Document Classification

We next make the assumption that each document is classified into a single subtopic category, but user intent is unknown. In terms of the probability distributions described in Section 3, perfect classification means D is divided into non-overlapping subsets $D_1, D_2, \cdots D_m$ such that for each subtopic T_i, $\Pr(T_i|d \in D_i) = 1$ and $\forall_{j \neq i} \Pr(T_j|d \in D_i) = 0$.

In this case, we study how to combine user intent and relevant document requirement distributions to best allocate documents from subtopics and maximize user satisfaction. We start by again defining the number of documents selected from subtopic T_i as K_i and enforce the condition that for the m subtopics of T, $\sum_{i=1}^{m} K_i = n$. As in the previous case, a user will click on up to j documents from subtopic T_i, and can click on at most K_i documents if $K_i < j$.

We calculate the expected number of hits for an average user with the following equation:

$$E(R) = \sum_{j=1}^{n}\sum_{i=1}^{m}\Pr(T_i|U)\Pr(J = j|U)\min(j, K_i) \qquad (2)$$

4.2.1 Solving For K

When we know exactly which subtopic each document belongs to, our main task becomes deciding how many documents from each subtopic should be included in the results. That is, we need to pick the set of $\{K_i\}$ values which will maximize the expected number of hits.

Given all the possible values for $\{K_i\}$, we can calculate the expected hits of each and choose an optimal solution. With n documents to choose from m subtopics, the number of combinations of $\{K_i\}$ values which satisfy the requirement $\sum_{i=1}^{m} K_i = n$ is $\binom{n+m-1}{n}$, making it infeasible to consider all possible combinations for a query. We can greatly reduce the search space, however, as many combinations are clearly not optimal. Allocating all documents to the least probable subtopic, for example, will not result in the maximum number of hits. Intuitively, an optimal solution should contain at least as many documents from the most probable subtopic as a less popular one. We formalize this notion with the following Proposition:

PROPOSITION 4.1. *Without loss of generality, label the subtopics of topic T as $T_1, T_2, \cdots, T_m$ such that $\Pr(T_1|U) \geq \Pr(T_2|U) \geq \cdots \geq \Pr(T_m|U)$. Then an optimal solution to Equation 2 satisfies the following properties:*

- $\sum_{j=1}^{n}\Pr(J = j|U) = 1$
- $\sum_{i=1}^{m} K_i = n$
- $K_1 \geq K_2 \geq \cdots \geq K_m$

PROOF. Assume an initial set of K values $\{K_1, K_2, \cdots, K_m\}$ such that $\sum_{i=1}^{m} K_i = n$ and $K_x < K_y$ for some $x < y$, with expected number of hits $E(K)$ as defined in Equation 2. Then we can construct a set $\hat{K} = \{K_1, \cdots, K_x+1, \cdots, K_y-1, \cdots K_m\}$ with expected hits:

$$\begin{aligned} E(\hat{K}) &= E(K) \\ &\quad + (\Pr(T_x|U)\sum_{j=K_x+1}^{n}\Pr(J = j|U)) \\ &\quad - (\Pr(T_y|U)\sum_{j=K_y}^{n}\Pr(J = j|U)) \\ &\geq E(K) + (\Pr(T_x|U) - \Pr(T_y|U))\sum_{j=K_y}^{n}\Pr(J = j|U) \\ &\geq E(K) \end{aligned}$$

□

4.2.2 Document Selection

In practice it is not necessary to enumerate and test all possible $\{K_i\}$ values, as we can optimize for Equation 2 directly. We select the documents to return using an algorithm which factors in both $\Pr(T_i|U)$ and $\Pr(J|U)$ while adhering to Proposition 4.1, and update K_i after each selection accordingly.

Algorithm *KnownClassification*
(∗ Rank documents to maximize Equation 2 ∗)
1. $R \leftarrow \emptyset$
2. $D \leftarrow$All relevant documents
3. $K_1 = K_2 ... = K_m = 0$
4. **while** $|R| < n$
5. $\quad i \leftarrow$ARGMAX($\Pr(T_i|U)\Pr(J > K_i|U)$)
6. $\quad K_i \leftarrow K_i + 1$
7. $\quad R \leftarrow R \cup$ NextDocument(D_i, R)

To choose each successive document, *KnownClassification* takes a greedy approach. The algorithm first determines which subtopic will provide the maximum marginal benefit to the average user. The marginal utility of a subtopic is the expected increase in hits produced by adding another document from it, and is the product of the user interest in the subtopic $\Pr(T_i|U)$ and the probability that users will want another document from that subtopic $\Pr(J > K_i|U)$. Once the next subtopic is chosen, a search engine may select the next document to return from $D_i \setminus R$ using its standard ranking functions.

4.3 Complete Model

We now eliminate the simplifying assumptions and discuss how to compute the expected hits when neither document classifications nor user intents are perfectly known. With

user intent uncertain, we need to calculate the expected hits probabilistically over all of the possible subtopics instead of only a single, known T_i from Equation 1. From Equation 2 we can no longer say the user will click on $\min(j, K_i)$ documents, as we have no guarantees on the number of documents which actually satisfy subtopic T_i. Instead, we expect the user to click on $\min(j, k)$ documents, based on the probability that k relevant documents are available to the user.

Combining the two simplified equations and making use of all three probability distributions, the equation for expected number of hits becomes:

$$E(R) = \sum_{j=1}^{n} \sum_{i=1}^{m} \Pr(T_i|U) \Pr(J = j|U) \sum_{k=1}^{n} \Pr(K_i = k|R) \min(j, k) \quad (3)$$

Algorithm *Diversity-IQ*
(∗ Rank documents to maximize Equation 3 ∗)
1. $R \leftarrow \emptyset$
2. $D \leftarrow$All relevant documents
3. **while** $|R| < n$
4. $d \leftarrow$ARGMAX($\Delta E(d|R, D)$)
5. $R \leftarrow R \cup \{d\}$
6. $D \leftarrow D \setminus \{d\}$

Diversity-IQ outlines how to select the set of documents R such that we maximize the expected number of hits for an ambiguous informational query. We adopt the approach of *KnownClassification*, selecting each successive document by determining which will maximize the increase in expected hits given the documents already returned.

The ΔE computation for a document is dependent on several factors, including its subtopic scores, the user interest in those subtopics, and the conditional probabilities of how many documents from each subtopic are already included in R. Using a dynamic programming algorithm, we can update the $\Pr(K_i|R)$ values once per iteration. Thus we have an overall computational complexity of $O(|R| \cdot |D| \cdot m)$ for choosing each successive document, or $O(n^2 \cdot |D| \cdot m)$ for re-ranking the top n documents.

5. COMPARISON WITH IA-SELECT

In this section we briefly go over the work on search diversification by Agrawal et al. [1] to better understand when prior work may perform sub-optimally, and how our approach may overcome such scenarios.

5.1 Overview of IA-Select

Agrawal et al. investigate the problem of ambiguous queries with the overall objective of maximizing the probability that an average user finds *at least one* relevant document in the top n search results. Their model assumes an explicit taxonomy of subtopics is available, and both documents and queries may fall into multiple subtopics. Queries belong to a set of subtopics with a known probability distribution, which effectively represents the user intent for a given query $\Pr(T_i|U)$. Likewise, documents belong to a set of subtopics, and the relevance to each subtopic is measured probabilistically, much like $\Pr(T_i|d)$.

Given this model and set of distributions, they formulate the *Diversify* function, which measures the probability that a set of n documents satisfies an "average" user for an ambiguous query. The objective to select the set of documents which maximizes this probability is proven to be NP-Hard, and the authors propose the *IA-Select* algorithm as an approximation, which is shown to produce an optimal solution to *Diversify* when every document belongs to a single subtopic.

Key to their algorithm is the notion of a conditional probability of subtopics, $U(T_i|R)$, which measures the probability that the user is still interested in subtopic T_i given the documents already chosen in R. The conditional probability of each subtopic is initialized to the user intent probability $\Pr(T_i|U)$. The algorithm successively selects documents which have the highest marginal utility, computed for each document as the sum, over each subtopic, of the subtopic's conditional probability and the document's score for that subtopic:

$$g(d|R) = \sum_{i=1}^{m} \Pr(T_i|d) U(T_i|R)$$

After a document d is selected, the conditional probability of each subtopic is updated to reflect the inclusion of d in R using Bayes' theorem:

$$\forall i : U(T_i|R) = (1 - \Pr(T_i|d)) U(T_i|R \setminus \{d\}) \quad (4)$$

5.2 Observed Limitations of IA-Select

In our experiments with *IA-Select*, we observed the algorithm often selects one document from each subtopic, in the order of subtopic popularity, and then degenerates into random document selection. We believe this behavior is suboptimal. Even if every subtopic is represented once in the results, an average user is more likely to want to see additional documents from more popular subtopics if there is room.

From our investigation, we find that this behavior is due to the following limitation. When deciding which document to select next, *IA-Select* uses the conditional probability $U(T_i|R)$, which measures the likelihood that the user is still interested in subtopic T_i given the documents already selected in set R. *IA-Select* assumes that the user is no longer very interested in subtopic T_i once at least one document believed to satisfy T_i is present in R, meaning $U(T_i|R)$ becomes very small.

When *IA-Select* is used with any document classification function which assigns subtopic scores approaching 1.0, the Bayesian update step in Equation 4 is problematic. To illustrate the issue more clearly, consider an extreme case where a document is classified to "perfectly" belong to any subtopic ($\Pr(T_i|d) = 1$). In that case, the subtopic will have its conditional probability set to zero. That is, if even one document from each subtopic has such a score, every conditional utility value will be set to zero, and the algorithm is reduced to random selection. Note that this behavior is not limited to the extreme case when $\Pr(T_i|d) = 1$. As long as the $\Pr(T_i|d)$ values are sufficiently high, all conditional subtopic probabilities will quickly become very small, and the algorithm exhibits similar behavior.

Zero-utility is particularly problematic if we consider that selecting multiple documents from a subtopic may be beneficial, which may be the case even in simple situations such as a query having fewer subtopics than document "slots" to fill ($m < n$). We avoid the zero-utility problem by computing the marginal benefit of a subtopic in terms of the probability that a user wants additional documents from it, which depends on $\Pr(J|U)$ and $\Pr(K_i|R)$. As long as $\Pr(J|U) > 0$, each subtopic will always have non-zero utility.

To illustrate the issue more clearly and accentuate how our algorithm avoids the zero-utility problem, we will walk through a simple example. In this example we use binary classification scores for clarity and to underscore the potential problems with *IA-Select* only. As we will see later in our experimental section, *IA-Select* exhibits similar behavior, to a lesser degree, even under widely used probabilistic classifiers which may assign any value between 0 and 1.

5.3 Descriptive Example

Assume two subtopics T_1 and T_2, with two documents classified into each subtopic. Our example will use the following probabilities and subtopic scores:

- $\Pr(T_1|U) = 0.7$ and $\Pr(T_2|U) = 0.3$.
- $\Pr(J|U) = (0.6, 0.3, 0.1)$.
- $D = \{d_1 = d_2 = (1.0, 0.0), d_3 = d_4 = (0.0, 1.0)\}$
- $n = 3$.

5.3.1 *Diversity-IQ*

To choose the first document, *Diversity-IQ* computes the marginal utility of each document:

$$\Delta E(d_1|\emptyset) = \Delta E(d_2|\emptyset) = 0.7 \sum_{j=1}^{3} \Pr(J = j|U) = 0.7$$
$$\Delta E(d_3|\emptyset) = \Delta E(d_4|\emptyset) = 0.3 \sum_{j=1}^{3} \Pr(J = j|U) = 0.3$$

The first document selected is chosen arbitrarily between $\{d_1, d_2\}$. To choose the second document, we compute the marginal utility of each remaining document:

$$\Delta E(d_2|\{d_1\}) = 0.7 \sum_{j=2}^{3} \Pr(J = j|U) = 0.28$$
$$\Delta E(d_3|\{d_1\}) = \Delta E(d_4|\{d_1\}) = 0.3$$

As d_3 and d_4 both provide the same increase in expected hits, we again choose arbitrarily between them. Thus we have $R = \{d_1, d_3\}$ after the first two iterations. To choose the third document, we again compute the marginal utility of the remaining documents:

$$\Delta E(d_2|\{d_1, d_3\}) = 0.7 \sum_{j=2}^{3} \Pr(J = j|U) = 0.28$$
$$\Delta E(d_4|\{d_1, d_3\}) = 0.3 \sum_{j=2}^{3} \Pr(J = j|U) = 0.12$$

Since d_2 has a higher marginal utility than d_4, it is added to the result set, for a final ranking $R = \{d_1, d_3, d_2\}$ with expected hits $E(R) = 1.28$.

5.3.2 *IA-Select*

For *IA-Select*, we initialize the utility of each subtopic to the user intent probabilities and compute the marginal utility of each document:

$$g(d_1|\emptyset) = g(d_2|\emptyset) = \sum_{i=1}^{2} \Pr(T_i|d)U(T_i|R) = 0.7$$
$$g(d_3|\emptyset) = g(d_4|\emptyset) = \sum_{i=1}^{2} \Pr(T_i|d)U(T_i|R) = 0.3$$

After choosing arbitrarily between $\{d_1, d_2\}$, we update the conditional probability of the subtopics:

$$U(T_1|\{d_1\}) = (1 - \Pr(T_1|d_1))U(T_1|\emptyset) = 0.0$$
$$U(T_2|\{d_1\}) = (1 - \Pr(T_2|d_1))U(T_2|\emptyset) = 0.3$$

We recompute the marginal utility of each document:

$$g(d_2|\{d_1\}) = \sum_{i=1}^{2} \Pr(T_i|d)U(T_i|R) = 0.0$$
$$g(d_3|\{d_1\}) = g(d_4|\{d_1\}) = \sum_{i=1}^{2} \Pr(T_i|d)U(T_i|R) = 0.3$$

Again, we choose arbitrarily between $\{d_3, d_4\}$ and update the conditional probability for each subtopic:

$$U(T_1|\{d_1, d_3\}) = (1 - \Pr(T_1|d_3))U(T_1|\{d_1\}) = 0.0$$
$$U(T_2|\{d_1, d_3\}) = (1 - \Pr(T_2|d_3))U(T_2|\{d_1\}) = 0.0$$

At this point, we still need to select a third document ($n = 3$), but the conditional utility of each subtopic is zero, meaning the marginal utility of every document will be zero. Intuitively, we would expect the more probable subtopic to be a better choice for the "average" user in this situation, but *IA-Select* will randomly choose between $\{d_2, d_4\}$. Note that there is a substantial difference in the expected hits depending on which we choose: d_2 will increase the expected hits by 0.28, while d_4 by only 0.12.

5.4 Discussion

One may contend that the outlined issues can easily be patched by smoothing or enforcing a limit on the maximum score assigned to any particular subtopic. In our evaluations we will also show that, while placing such arbitrary limits on subtopic scores can improve *IA-Select's* performance on the expected hits metric to a certain degree, it still exhibits similar behavior, and the improvements come at the cost of degraded performance on other metrics.

It is also worth noting that Equation 3 is, in fact, a generalization of the *Diversify* goal function. That is, if we make the assumption that all users require exactly one document (setting $\Pr(J = 1|U) = 1$), *Diversity-IQ* will yield the same ranking as *IA-Select*.

6. DISTRIBUTION MEASUREMENTS

Our algorithm requires three distributions which describe (1) the number of relevant documents a user is expected to require, (2) the probability of user intent in each subtopic, and (3) the probability a document satisfies each subtopic. Given the broad range of possible queries and the number of documents on the web, automatic methods for approximating these distributions are necessary for a real world deployment. In this section we suggest possible techniques to approximate them using data sources available to web search engines. In Section 7 we specify which data sources are used for each experiment.

6.1 Measuring Document Requirements

Knowing the number of relevant documents a user is expected to require is necessary to determine how much diversity we can introduce in the results without harming the hit-rate for popular subtopics. One method to approximate this distribution is using click-through data from query logs. Figure 1 shows the number of click-throughs for each query session with at least one click from a locally collected query log [16]. We observe that other publicly available query logs show a similar distribution. In our log, users clicked on an average of 1.52 results for queries with at least one click-through. Other studies of web search logs report an average of 3.18 clicks-per-query [13] when empty sessions are removed.

6.2 Measuring User Intent

The user intent distribution measures the probability that an average user is interested in a particular subtopic for a given query. We have shown in Section 4.2.1 that if subtopic

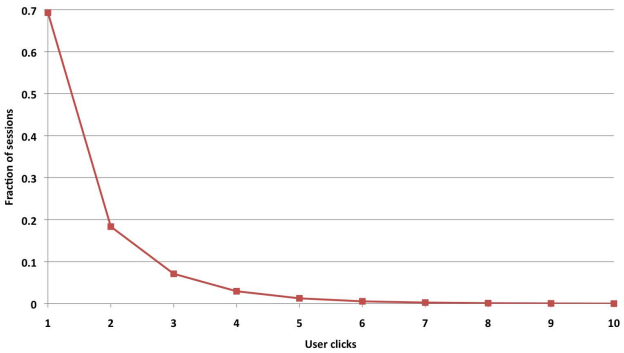


Figure 1: Clicks per query

T_i is *more likely* than subtopic T_j, then showing the user at least as many documents from T_i as T_j is a necessary condition for optimizing the expected number of hits. Therefore an accurate estimate of user interest in each subtopic is important. Possible sources for this information include:

- The frequency of popular query refinements for ambiguous queries [17].
- The click-through history for documents returned by an ambiguous query.
- Frequency of subtopic queries, which can be measured using information about search volume and trends.[1][2]

6.3 Measuring Document Categorization

The document categorization distribution tells us which of the m subtopics a particular document belongs to. Unsupervised document classification techniques often require an estimate for m and a sufficiently large collection of relevant documents. We look at these issues next.

6.3.1 Subtopic Estimation

We investigated two sources for discovering the subtopics for a given ambiguous query: (1) WordNet [8], and (2) Wikipedia.[3] WordNet is a popular lexical database which includes term relationships. Unfortunately, data for queries such as movies, song titles, and proper nouns are sparse in WordNet. The second source we examined is Wikipedia. Among the millions of articles on Wikipedia are over 60,000 disambiguation pages for the English language. These pages list several possible meanings of term, covering a wider range of entity types.

Other approaches for identifying subtopics include mining popular query refinements [17] and using categories from the Open Directory Project.[4]

6.3.2 Document Classification

Radlinski and Dumais show that homogeneity in the top results generally hinders the effectiveness of personalization [17]. Without a sufficient number of documents from each subtopic, unsupervised classification techniques will be unable to generate meaningful topics. In our experiments we therefore opt for a metasearch strategy. We form a collection of documents for an ambiguous query by issuing each relevant Wikipedia subtopic page title as a search query. We merge the top 200 results from each subtopic query to form a single document set.

Given a set of m subtopics T and collection of documents D, a document classification function $C(d, T)$ assigns a normalized probability score $\Pr(T_i|d)$ for each $T_i \in T$, such that $\sum_{i=1}^{m} \Pr(T_i|d) = 1$. We consider two such classification functions in our evaluations: (1) *query-based* classification, and (2) Latent Dirichlet Allocation (LDA) [2].

Query-based classification uses knowledge of which subtopic queries returned each document. For each of the subtopic queries that returned a document d in its top 200 results, we compute a score $S_i(d) = cos(d^c, W(T_i))$, where d^c and $W(T_i)$ are vector space model representations of the text from the search result snippet and the Wikipedia page for subtopic T_i, respectively, and *cos* is the cosine similarity function. We normalize these scores to assign $\Pr(T_i|d) = \frac{S_i(d)}{\sum_{j=1}^{m} S_j(d)}$.

The second classification method we consider is Latent Dirichlet Allocation (LDA). LDA requires a set of documents, a number of topics m, and two hyperparameters α and β which control smoothing of Dirichlet priors for topics and words. In typical applications, $0 < \alpha < 1$ and β is set to 0.1 or 0.01 [21]. We construct an LDA topic model for each document set and assign $\Pr(T_i|d)$ from the resulting θ distribution. The LDA topics are aligned with the known subtopics in a greedy fashion using knowledge of which subtopic queries retrieved the document.

7. EVALUATION

We conducted several experiments to assess the overall effectiveness of *Diversity-IQ*. The evaluations include an analysis of our objective of maximizing the expected number of hits, as well as comparisons using established subtopic retrieval metrics. For each metric, we compare our *Diversity-IQ* algorithm against the state-of-the-art *IA-Select* algorithm presented by Agrawal et al. [1] as well as the original ranking returned by a commercial web search engine (SE).

7.1 Query Set

One of the difficulties in evaluating a system designed to introduce diversity is the lack of standard testing data. Evaluating diversification requires a set of ambiguous queries, and until recently, no benchmark query sets or relevance judgements exist explicitly for the task of diversification research. TREC[5] added a diversity task to the Web track beginning in 2009. The data includes 50 queries, each with a set of selected subtopic aspects. Unfortunately, this dataset and evaluation criteria were designed assuming a single relevant document is sufficient, and so it is difficult to adapt them to our setting.

Although techniques exist to identify ambiguous queries [20] which would be beneficial in a real world deployment, we're primarily concerned with evaluating the performance benefits of our algorithm, and thus we opted for a simpler approach to form a testing set. We generated a set of ambiguous queries using a small search log with a few hundred thousand entries collected from our local network. A query is marked as ambiguous if a Wikipedia disambiguation page exists for the terms. We randomly selected 50 queries from this candidate set for our evaluations.

[1] http://www.google.com/insights/search/
[2] http://www.bing.com/xrank/
[3] http://www.wikipedia.org/
[4] http://www.dmoz.org/
[5] http://trec.nist.gov/

7.2 Probability Distributions

Figure 1 shows the click-through distribution for *all* queries, but studies indicate that navigational queries account for anywhere from 10-25% of web searches [3, 18] and typically result in a single click [11]. Removing these queries from the query log would produce a more accurate distribution for our algorithm, but automatically classifying queries as informational or navigational is a difficult task. To avoid unfairly penalizing our algorithm with a click-distribution containing navigational queries, we approximate the distribution of how many relevant documents a user will require using the geometric series $\Pr(J = j|U) = 2^{-j}$, which represents an average of 2 clicks-per-query, displays an exponential decay characteristic like Figure 1, and has the property $\lim_{n\to\infty} \sum_{j=1}^{n} \Pr(J = j|U) = 1$, which conforms with the conditions of Proposition 4.1.

To measure the user intent distribution $\Pr(T_i|U)$ for our experiments, we conducted a survey using Amazon's Mechanical Turk[6]. For each query we asked 10 survey participants to select all of the subtopics they associate with the query from the given choices. To keep the task manageable, we limited our study to queries with at most 20 subtopics, with an average of 8.5 subtopics per query. To ensure all subtopics were considered, those which received no votes were assigned a non-zero value of 0.01.

Unless otherwise noted, the document-subtopic probability scores $\Pr(T_i|d)$ were assigned using the GibbsLDA++[14] implementation of LDA (see Section 6.3.2). Parameters were set at $\alpha = 0.2$ and $\beta = 0.1$, based on values found to work well for text collections [9].

7.3 Expected Hits

We analyzed *Diversity-IQ* with our overall goal metric of maximizing the expected number of hits. For each test query we compute the expected number of hits for each ranking strategy over increasing values of n using Equation 3. A majority of users do not look beyond the first result page [10], making efficiency of the top documents particularly important. We limit our evaluation to the top 10 results, as commercial search engines typically show 10 results per page.

Figure 2 shows the mean expected hits computed over the test query set for the top 10 results with the three ranking approaches. Figure 2(a) assigns $\Pr(T_i|D)$ using the subquery-based classification method, while Figure 2(b) uses LDA to assign subtopic scores. In both cases, the expected hits from the top document is comparable, as providing at least one document from the most probable subtopic is generally the initial strategy taken by both our algorithm and *IA-Select*. After the top few results, however, our algorithm may find additional benefits from providing additional documents from popular subtopics, and thus our algorithm tends to increase the expected hits more rapidly.

We measured the runtime performance of each algorithm on a 2.6GHz Intel Core 2 Duo CPU with 4 GB memory running Mac OS X 10.6. Implementations were written in Python. To select the top 10 results, *Diversity-IQ* required an average of 28.8ms, while *IA-Select* averaged 28.5ms.

7.3.1 Classification Score Range

We briefly look at how the range of average classification scores can effect the expected hits for both algorithms. For each query we performed LDA classification with varying values of α and β. We identify the subtopic with the highest individual score for each document and compute the average of these scores over all documents. Figure 3 plots this average "top" subtopic score against the corresponding expected hits for each algorithm. The figure shows that *Diversity-IQ* outperforms *IA-Select* on expected hits regardless of the classification scores, and *IA-Select* suffers a significant drop in performance as potential subtopic scores approach 0.7.

[6]https://www.mturk.com/mturk/welcome

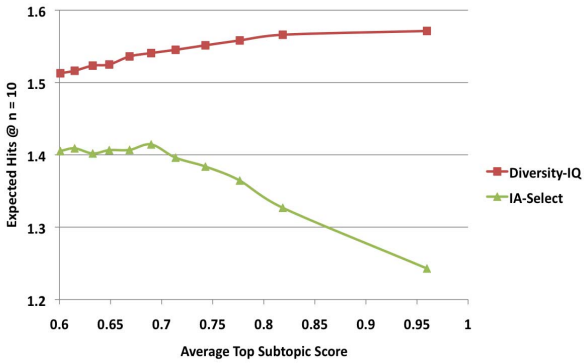


Figure 3: Effect of subtopic scores

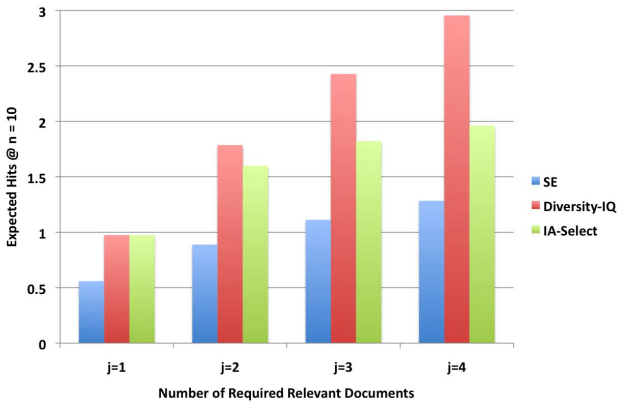


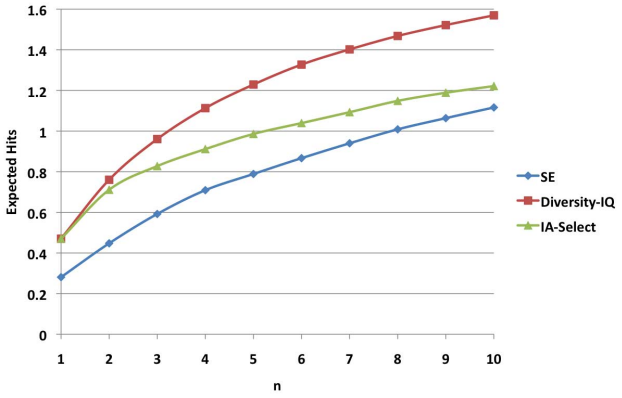
Figure 4: Varying number of required documents

7.3.2 Requiring Multiple Documents

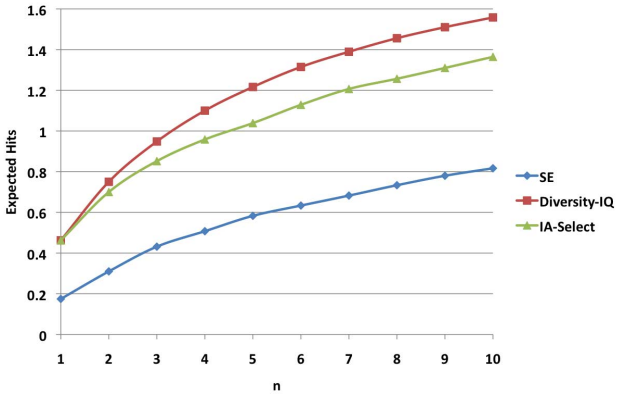
We next study the effects of $\Pr(J|U)$ on expected hits, and in particular performance as the number of relevant documents a user is expected to require increases. Figure 4 shows the expected hits for $n = 10$ as we vary the number of documents users are expected to require from 1 to 4. As we can see, for users who require only one relevant document ($j = 1$), our algorithms have equal performance. In all cases where users want more than one document, however, *Diversity-IQ* outperforms *IA-Select*. As expected, we can see that our algorithm's relative performance improves as users are expected to require additional documents.

7.4 Single Document Metrics

Having demonstrated the performance advantages of our algorithm with respect to the more general model, we turn our attention to metrics based on returning *at least one* relevant document. As our algorithm may find it beneficial to return multiple documents from popular subtopics before any documents from unpopular subtopics, we expect an algorithm focused on returning at least one relevant document, such as *IA-Select*, to outperform ours on these met-



(a) Expected Hits (Query-based)



(b) Expected Hits (LDA)

Figure 2: Expected Hits

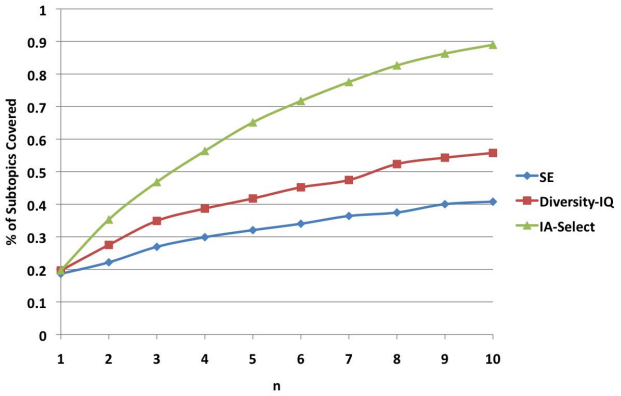


Figure 5: Subtopic recall

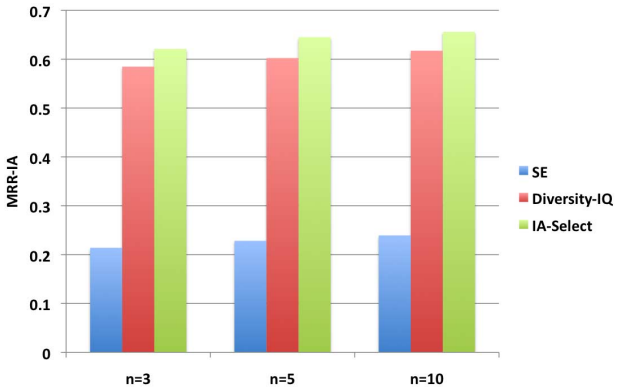


Figure 6: MRR-IA @ n

rics. Nonetheless, we feel it is important to compare the algorithms on a level playing field and quantify the differences in these scenarios as well.

7.4.1 *Subtopic Recall*

Subtopic recall (S-recall) at rank N was defined by Zhai, Cohen, and Lafferty [24] as the percentage of relevant subtopics covered by the top N documents. Assuming all users want one relevant document and a uniform user intent probability distribution, S-recall serves as an indication of expected user satisfaction with the top N. S-recall requires *binary* relevance judgements: a document either does or does not satisfy a particular subtopic. To compute the S-recall for our evaluation we consider a document as satisfying a subtopic if its subtopic score is above a certain threshold, which we set at $\Pr(T_i|d) \geq 0.3$.

Figure 5 plots the average subtopic recall for our evaluation set. As expected, *IA-Select* outperforms our algorithm on S-recall for the highest ranked documents. Nonetheless, our algorithm outperforms the original search engine ranking, and on average covers over one-half of the subtopics within the top 10 results.

7.4.2 *MRR-IA*

S-recall assumes all subtopics are equally important. In reality we know that certain subtopics are often considerably more likely than others. To evaluate the effectiveness of our algorithm identifying such subtopics and presenting them early, we consider the "intent aware" Mean Reciprocal Rank (MRR-IA) metric defined in [1]. MRR-IA measures the traditional mean reciprocal rank over each subtopic, weighted by their probability of user intent. Again, we use the threshold 0.3 to determine when a document satisfies a subtopic.

Figure 6 shows our algorithm outperforms the original search engine ranking and a small decrease in performance (approximately -6%) with respect to *IA-Select* at $n = 10$. This indicates that our algorithm is still able to identify the most probable subtopics and present at least one document from each early in the ranking, thus performing well for a majority of users even when we assume one relevant document is sufficient.

7.5 Smoothing IA-Select

As noted earlier, we can partially address the weakness in *IA-Select* by imposing limits on the maximum score assigned to any particular subtopic. We now evaluate the effects of varying that limit on expected hits, MRR-IA, and S-recall. For these experiments, we modified the Bayesian update step of *IA-Select* (shown in Equation 4) to multiply the conditional utility U of each subtopic by $(1 - \min(\Pr(T_i|d), L))$, where L is the maximum allowable score. Figure 7 shows the effects of smoothing on IA-Select for various limits on the maximum subtopic scores. The general trend shows that, as we decrease the maximum allowable subtopic score, the expected hits increase as the other metrics decrease. It is unclear how to intelligently select a proper "smoothing" value for any particular number of relevant documents required.

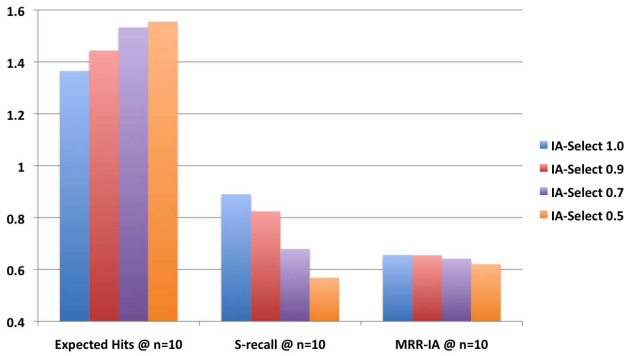


Figure 7: Effects of Smoothing on IA-Select

8. CONCLUSION AND FUTURE WORK

In this paper we focused on diversifying search results for ambiguous informational queries, where users often require multiple relevant documents. We presented a model for user satisfaction with a set of search results, represented by the expected number of hits, or user clicks on relevant documents, in the top n. We studied the problem of how, when faced with an ambiguous query, a search engine can use probabilistic knowledge of user intent, document classification, and how many relevant documents a user will require to return a document set which maximizes the probability of satisfaction for an average user. Experiments show our *Diversity-IQ* algorithm outperforms a commercial search engine and the state-of-the-art *IA-Select* diversification algorithm on the expected hits metric, while performing comparably on metrics designed under single relevant document assumptions. Our algorithm also helps overcome the limitations of *IA-Select*, performing well regardless of the subtopic scores assigned by the document classification function.

For future work, we would like to investigate a more detailed model for a user's page requirements by considering query-dependent or query-class dependent distributions. That is, using a single distribution for page requirements is a simplification, and query-dependent $\Pr(J = j|U, q)$ or query-class dependent $\Pr(J = j|U, C(q))$ distributions may help improve the model by allowing for different estimates of user need based on perceived goals (e.g. product purchase vs. product research). As an extreme example, for navigational queries we may want to assign this distribution as $\Pr(J = 1|U, C(q) = nav) \approx 1.0$.

9. ACKNOWLEDGEMENTS

This work is partially supported by NSF grants, IIS-0534784 and IIS-0347993. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding institutions.

10. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, pages 5–14, 2009.

[2] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

[3] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[4] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.

[5] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *SIGIR*, pages 429–436, 2006.

[6] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *SIGIR*, pages 659–666, 2008.

[7] W. S. Cooper. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems, 1968.

[8] C. Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA, 1998.

[9] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101:5228–5235, April 2004.

[10] B. J. Jansen and A. Spink. How are we searching the world wide web?: a comparison of nine search engine transaction logs. *Inf. Process. Manage.*, 42(1):248–263, 2006.

[11] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *WWW*, pages 391–400, 2005.

[12] F. Liu, C. Yu, and W. Meng. Personalized web search by mapping user queries to categories. In *CIKM*, pages 558–565, 2002.

[13] B. U. Oztekin, G. Karypis, and V. Kumar. Expert agreement and content based reranking in a meta search environment using mearf. In *WWW*, pages 333–344, 2002.

[14] X.-H. Phan and C.-T. Nguyen. http://gibbslda.sourceforge.net/.

[15] A. Pretschner and S. Gauch. Ontology based personalized search. In *ICTAI*, pages 391–398, 1999.

[16] F. Qiu, Z. Liu, and J. Cho. Analysis of user web traffic with a focus on search activities. In *WebDB*, 2005.

[17] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *SIGIR*, pages 691–692, 2006.

[18] D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW*, pages 13–19, 2004.

[19] M. Sanderson. Ambiguous queries: test collections need more sense. In *SIGIR*, pages 499–506, 2008.

[20] R. Song, Z. Luo, J.-Y. Nie, Y. Yu, and H.-W. Hon. Identification of ambiguous queries in web search. *Inf. Process. Manage.*, 45(2):216–229, 2009.

[21] M. Steyvers and T. Griffiths. Probabilistic topic models. In *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.

[22] E. Voorhees. Overview of the trec 2004 robust retrieval track. In *TREC*, 2004.

[23] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *SIGIR*, pages 115–122, 2009.

[24] C. X. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR*, pages 10–17, 2003.