Representative & Informative Query Selection for Learning to Rank using Submodular Functions

Rishabh Mehrotra
Dept of Computer Science
University College London, UK
r.mehrotra@cs.ucl.ac.uk

Emine Yilmaz
Dept of Computer Science
University College London, UK
emine.yilmaz@ucl.ac.uk

ABSTRACT

The performance of Learning to Rank algorithms strongly depend on the number of labelled queries in the training set, while the cost incurred in annotating a large number of queries with relevance judgements is prohibitively high. As a result, constructing such a training dataset involves selecting a set of candidate queries for labelling. In this work, we investigate query selection strategies for learning to rank aimed at actively selecting unlabelled queries to be labelled so as to minimize the data annotation cost. In particular, we characterize query selection based on two aspects of *informativeness* and *representativeness* and propose two novel query selection strategies (i) Permutation Probability based query selection and (ii) Topic Model based query selection which capture the two aspects, respectively. We further argue that an ideal query selection strategy should take into account both these aspects and as our final contribution, we present a submodular objective that couples both these aspects while selecting query subsets. We evaluate the quality of the proposed strategies on three real world learning to rank datasets and show that the proposed query selection methods results in significant performance gains compared to the existing state-of-the-art approaches.

Categories and Subject Descriptors

H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval—*Learning to Rank*

Keywords

Learning to Rank, Query Selection, Active Learning, Submodularity

1. INTRODUCTION

Most modern search technologies are based on machine learning algorithms that learn to rank documents given a query, an approach that is commonly referred to as "learning to rank". Learning to Rank algorithms aim to learn ranking functions that achieve good ranking objectives on test data. Such learning methods require labelled data for training. As is the case with many supervised learning algorithms, the performance of Learning to Rank algorithms are often highly correlated with the amount of labelled training data available[1][17][7].

Constructing such labelled training data for learning-to-rank tasks incurs prohibitive costs since it requires selecting candidate queries, extracting features from query-document pairs and annotating documents in terms of their relevance to these queries (annotations are used as labels for training). The major bottleneck in constructing learning-to-rank collections is annotating documents with query specific relevance grades. It is essential therefore, both for the efficiency of the construction methodology and for the efficiency of the training algorithm, that only a small subset of queries be selected. The query selection, though, should be done in a way that does not harm the effectiveness of learning.

Active Learning algorithms help reduce the annotation costs by selecting a subset of informative instances to be labelled. Unlike traditional algorithms, active learning strategies for ranking algorithms are more complex because of the inherent query-document pair structure embodied in ranking datasets, non-smooth cost functions, etc., hence these cannot be applied directly in ranking setting.

Existing approaches for active learning for ranking have focused on selecting documents [1], selecting queries [17] or balancing number of queries with depth of documents judged using random query selection [27].

In this work, we focus on selecting subset of queries to be labelled so as to minimize the data annotation cost. Prior work on selecting queries made use of expected loss optimization [17] to estimate which queries should be selected but their approach is limited to rankers that predict absolute graded relevance which is not the case with modern Learning to Rank algorithms since many of them induce a ranking and not absolute labels [4]. Apart from the learning to rank setting, query selection has also received significant attention for evaluation setting [13] wherein the goal was to find a subset of queries that most closely approximates the system evaluation results that would be obtained if instead documents for the full set of queries was judged instead. However, it was shown by Aslam *et a.l*[1] that learning to rank and evaluation of retrieval systems are quite different from each other and that datasets constructed for evaluating quality of retrieval systems are not necessarily good for training and vice versa. Therefore, query selection strategies that are directly devised for learning to rank purposes are needed.

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.
© 2015 ACM. ISBN 978-1-4503-3621-5/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766462.2767753.

Intuitively, an optimal subset of queries constructed for learning to rank should have two characteristics: (i) *informativeness*, which measures the ability of an instance (query) in reducing the uncertainty of a statistical model (ranking model) and (ii) *representativeness*, which measures if an instance (query) well represents the possible input patterns of unlabelled data (unlabelled queries) [22]. Most existing active learning for ranking algorithms solely focus on the informativeness aspect of queries without considering the representativeness aspect which can lead to possible selection of *noisy* queries, not quite representative of the whole population of queries; thus, significantly limiting the performance of query selection.

In this work, we focus on query selection strategies for learning to rank and propose novel query selection algorithms aimed at finding an optimal subset of queries to be labelled. Since problems associated with subset selection are generally NP-Hard or NP-Complete[12], we approximate the solution by an iterative query selection process so as to minimize the data annotation cost without severely degrading the performance of the ranking model.

We describe two paradigms of query selection strategies based on the aspects of *informativeness* and *representativeness* described above and propose novel query selection techniques: Permutation Probability based query selection and query selection based on topic models which capture these two aspects, respectively. We further present a new algorithm based on defining a submodular objective that combines the powers of the two paradigms. Submodular functions have the characteristic of diminishing returns [19], which is an important attribute of any query-subset selection technique since the value-addition from individual queries should ideally decrease as more and more queries are selected. Thus, not only are submodular functions natural for query subset selection, they can also be optimized efficiently and scalably such that the result has mathematical performance guarantees.

We show that our proposed algorithms result in significant improvements compared to state-of-the-art query selection algorithms thereby helping in reducing data annotation costs.

2. RELATED WORK

Active Learning for Labelling Cost Reduction:
A number of active learning strategies have been proposed for the traditional supervised learning setting, a common one being *uncertainty sampling* which selects the unlabelled example about which the model is most uncertain how to label. Some of the others adopt the idea of reducing the generalization error and select the unlabelled example that has the highest effect on the test error, i.e. points in the maximally uncertain and highly dense regions of the underlying data distribution[9]. A comprehensive active learning survey can be found in [22].

Reducing judgment effort for learning to rank has received significant amount of attention from the research community. Learning to rank methods are quite different than approaches used for classification as they require optimizing nonsmooth cost functions such as NDCG and AP [24]. Moreover, owing to the unique *query-document* structure which inherent to the learning to rank setting, it is not straightforward to extend the models devised for traditional supervised learning settings to ranking problems. In recent years, active learning has been actively extended to rank learning and can be classified into two classes of approaches: document level and query level active learning.

Document Selection for Learning to Rank:
Based on uncertainty sampling, Yu *et al*[28] selected the most ambiguous document pairs, in which two documents received close scores predicted by the current model, as informative examples. Donmez *et al.*[8] chose those document pairs, which if labelled could change the current model parameters significantly. Silva *et al* [23] proposed a novel document level active sampling algorithm based on association rules, which does not rely on any initial training seed.

Query Selection for Learning to Rank:
For query level active learning, Yilmaz *et al.* [27] empirically showed that having more queries but shallow documents performed better than having less queries but deep documents. They balance number of queries with depth of documents judged using random query selection. Cai *et al.* [5] propose the use of Query-By-Committee (QBC) based method to select queries for ranking adaptation but omit the evaluation of the query selection part and focussed on the ranking adaptation results instead. Long *et al.* [17] introduced an expected loss optimization (ELO) framework for ranking, where the selection of query and documents were integrated in a principled 2 staged active learning framework and most informative queries selected by optimizing the expected DCG loss but the proposed approach is limited to rankers that predict absolute graded relevance and hence not generalizable to all rankers. Authors in [2] adapt ELO to work with any ranker by introducing a calibration phase where a classification model is trained over in the validation data. Moreover, they show that estimating expected loss in DCG is more robust than NDCG even when the final performance measure is NDCG.

Thus, QBC attempts to capture the informativeness aspect of queries by selecting queries which minimize the disagreement among a committee of rankers while the Expected loss optimization based approach formulates informativeness in terms of expected DCG loss; both these approaches fail to capture the representativeness aspect of queries which we show outperforms both these approaches.

Submodular Maximization:
Submodularity is a property of set functions with deep theoretical and practical consequences. Submodular maximization generalizes to many well-known problems, e.g., maximum weighted matching, max coverage, and finds numerous applications in machine learning and social networks. In Information Retrieval, submodular objectives have been majorly employed for diversified retrieval[29] & learning from implicit feedback[21]. A seminal result of Nemhauser *et al.* [19] states that a simple greedy algorithm, based on a submodular objective, produces solutions competitive with the optimal (intractable) solution. In fact, if assuming nothing but submodularity, no efficient algorithm produces better solutions in general [10].

3. QUERY SELECTION STRATEGIES

Our aim is to *actively* select the optimal subset of unlabelled queries for obtaining relevance judgements so as to reduce data annotation costs. Intuitively, the selected queries should have two major properties: informativeness & representativeness. We describe both these properties below and provide intuitions motivating each.

3.1 Informativeness

Informativeness measures the ability of an instance in reducing the uncertainty of a statistical model[22]. Ideally, the selected queries should be maximally informative to the ranking model. In learning to rank setting, Informativeness based query selection focusses on greedily selecting queries which are most informative to the current version of the ranking model.

Different notions of informativeness can be encapsulated by different techniques depending on how query-level informativeness is quantified. Two possible measures of capturing a query's informativeness include: (i) *Uncertainty* based informativeness & (ii) *Disagreement* based informativeness.

Uncertainty based informativeness quantifies the query-level information as the uncertainty associated with the optimal document ranking order for that query. Query selection strategies focusing on uncertainty reduction would greedily select the query instance about which the current ranking model is most uncertain about, thereby trying to reduce the overall uncertainty associated with the ranking model.

Disagreement based informativeness, on the other hand, quantifies the query-level informativeness as the disagreement in this query's document rankings among a committee of ranking models. The key idea here is that the maximally informative query is one about whose document rankings, the committee of ranking models maximally disagree; hence obtaining relevance labels for such a query would provide the maximum information. Among the existing approaches for query selection for ranking models, the Query-by-Committee [5] attempts to capture the Informativeness aspect of queries based on a disagreement measure.

3.2 Representativeness

Representativeness measures if an instance well represents the overall input patterns of unlabelled data [22]. Web search queries can span a multitude of topics and information needs, with even a small dataset containing a broad set of queries ranging from simple navigational queries to very specific domain-dependent queries. In learning to rank settings, this implies that selected queries should have strong correlation with the remaining queries, as without this correlation there is no generalizability and predictive capability. Different notions of representativeness can be defined covering different characteristics of individual queries. Improving the representativeness of the selected query subset improves the coverage aspect of the query collection - the more representative selected queries are, the more they cover the entire query collection.

3.3 Informativeness vs Representativeness

Selecting queries solely based on their informativeness aspects could possibly lead to selection of *noisy* queries. In line with the Meta-Search Hypothesis [14][15], rankers tend to agree on relevant documents and disagree about nonrelevant docs. Hence, the queries that a ranker is unsure about or there is big disagreements across rankers are likely to be the ones that contain a lot of nonrelevant documents. Such noisy, *outlier* queries which majorly have non-relevant documents would lead to maximal disagreement and uncertainty among ranking models, and thus would be wrongly labelled maximally informative. Also, the set of informative queries might not necessarily represent the set of all possible queries, which lead to less coverage of the unlabelled query set.

On the other hand, selecting queries based on representativeness aspects could lead to the selection of a query that is very similar to the a query already in the labelled set and hence, does not provide much information to the ranking model. Despite being representative, such queries possibly offer redundant information to the ranking models. Ideally, a query selection algorithm should take into account both these aspects while selecting queries. Existing work has majorly looked into selecting queries by considering informativeness based on disagreement among rankers (Query-by-Committee) or informativeness in terms of expected DCG loss (Expected loss optimization). Both these approaches fail to capture the representativeness aspect of queries. In addition to a novel informativeness approach based on uncertainty reduction, we present a representativeness based approach and finally couple both these aspects for query selection via a joint submodular objective which jointly incorporates informativeness & representativeness.

As our first contribution, we present a novel informativeness based query selection scheme (§ 4) based on permutation probabilities of document rankings which tries to reduce uncertainty among rankers. While no existing query selection scheme for learning to rank incorporates the representativeness aspect of queries, we propose a LDA topic model based query selection scheme (§ 5) which captures the representative aspect of queries while constructing the query subset. An ideal query subset would have both informative & representative queries. As our third contribution, we combine the two paradigms of representativeness & informativeness by proposing a coupled model based on submodular functions(§ 6).

4. CAPTURING INFORMATIVENESS VIA PERMUTATION PROBABILITIES

Our first novel query selection scheme is aimed at capturing the informative-aspect of queries. We maintain a committee of ranking models $C = \{\theta^1, \theta^2, ..., \theta^C\}$ which are trained on a randomly selected subset from the current labelled set, and thus contain different aspects of the training data depending on the queries in their subset. It is to be noted that these ranking models could be generated using any learning to rank algorithm. Given the set of currently labelled query instances, our goal is to pick the next query (q^*) from the set of unlabelled queries by selecting the maximally informative query instance. The query-level informativeness is defined in terms of the uncertainty associated with the optimal document ranking orders among the $|C|$ ranking models. We follow a similar approach as outlined by Cai *et al.*[5] to maintain a committee of rankers. However, unlike Query-By-Committee [5] which encapsulates informativeness via ranker disagreements, our approach presents an alternate view of informativeness based on uncertainty reduction wherein a ranking model's uncertainty for the query's document ranking order is defined based on the concept of permutation probabilities.

More specifically, each committee ranking model is allowed to score the documents associated with each query following which a permutation probability is calculated on the ranking obtained on sorting these document scores. Thus, each query gets a permutation probability score by each committee member. The most informative query is considered to be the query instance which minimizes the maximum permutation probability of document scores given by each ranking model committee member.

We postulate that a query which has the minimum permutation probability score from among the maximum scores assigned between the different ranking models is maximally informative in the sense that even the best ranker among the committee is highly uncertain about its document rankings and hence this query obtained the least permutation probability score among the set of unlabelled candidate queries. We select a query for which the probability with respect to the most certain (maximum permutation probability) model is minimal, i.e., a query for which even the most certain committee member has minimum confidence.

To define permutation probabilities, we make use of the Plackett-Luce model [20]. The Plackett-Luce (P-L) model is a distribution over rankings of items (documents) which is described in terms of the associated ordering of these items (documents). We define $P(\pi|\upsilon)$ as the probability of obtaining the ranking order (π) based on the score (υ_k) assigned to each document (k) by the ranking model learnt thus far. For each query, we rank the documents based on the scores assigned by model learnt so far and calculate the probability of the ranking order obtained (π) using the permutation probability defined as follows:

$$P(\pi|\upsilon) = \prod_{i=1,\ldots,K} \frac{\upsilon_{\omega_i}}{\upsilon_{\omega_i} + \cdots + \upsilon_{\omega_K}} \tag{1}$$

where each ranking π has an associated ordering of document scores $\omega = (\omega_1, \cdots, \omega_K)$ and an ordering is defined as a permutation the K document indices with υ_{ω_i} representing the score assigned to document i (at rank ω_i) by the ranking model. We make use of a committee of ranking models and select the maximally informative query based on a greedy min-max algorithm described next.

4.1 Min-Max PL Probability Algorithm

Building a Min-Max PL Probability based selection system involves two components: (i) building a committee of ranking models that are well diversified and compatible with the currently labelled data and (ii) computing permutation probabilities by each committee member for each query in the unlabelled set of queries & selecting maximally informative query as per the min-max score.

Committee Construction:
Following the work of [5], we use query-by-bagging approach to construct the members. Given the set of currently labelled instances, bagging generates C partitions of sub samples by sampling uniformly with replacement, and then the committee can be constructed by training each of its members on one portion of the sub-sample partitions. We randomly initialize the initial set of labelled queries with a small base set of queries and their labelled documents. We sample with replacement for C times in the set of labelled queries and train a ranking model on each subset of queries. Such a sampling procedure allows us to create various different training datasets that each represent a subset of the data possibly having very different characteristics than each other. These C models represents our C committee members. We set the size of each subset to be 50 % of the current labelled subset size at each step. The maximally informative query q^* is selected for annotation which obtains the lowest min-max score, the calculation of which is described below.

Calculating min-max score:
For each query q in the candidate set of unlabelled queries, the C committee members return C ranked lists. Following the construction of $|C|$ ranking models, for each ranking model per query, we sort the documents based on the scores given by the ranking model and compute the permutation probability of obtaining this ranking order.

Thus, each query has $|C|$ permutation probability scores. In order to minimize the overall uncertainty associated with the ranking models, we select the maximally informative query q^*, i.e., the query that has the minimum value of the permutation probability assigned by its *most certain* committee member, i.e., the committee member that has the highest permutation probability score associated with the query's document ranking order. Thus,

$$q^* = argmin_{q \in D_u} \Bigg[max_{c \in C} \Bigg\{ P(\pi_q^c | \upsilon_q^c) \triangleq \prod_{k=1,\ldots,K} \frac{\upsilon^c_{\omega_k}}{\upsilon^c_{\omega_k} + \cdots + \upsilon^c_{\omega_K}} \Bigg\} \Bigg] \tag{2}$$

where each ranking π_q^c has an associated ordering $\omega = (\omega_1^c, \cdots, \omega_K^c)$ and an ordering is defined as a permutation the K document indices with $\upsilon^c_{\omega_k}$ representing the score assigned to document k by the ranking model c.

5. CAPTURING REPRESENTATIVENESS VIA LDA TOPICS

A major drawback associated with pure-Informativeness based models is that often they tend to select outlier queries. As is confirmed by the Meta-Search Hypothesis [14][15], rankers tend to agree on relevant documents but disagree on non-relevant documents. In such a scenario, an outlier query which majorly has non-relevant documents would lead to maximal disagreement and uncertainty in the ranking model, and thus will be wrongly labelled maximally informative. This motivates the need for considering the representativeness aspect of queries.

The information-seeking behaviour of users tend to vary based on the search task at hand [25] which suggests that the importance of feature weights for queries belonging to different tasks or topics are likely to be very different. The relative importance of different features are likely to be very different for different tasks. For example, queries belonging to a topic such as news would warrant high authority websites to be ranked higher (i.e., larger weight on the pagerank score) while queries belonging to (say) educational informational content would prefer the documents better matched with their query terms be ranked higher (i.e., larger weight on the relevance features such as BM25). To capture these diverse variations in the feature weights, the training set should ideally be composed of representative queries from different tasks. This makes it necessary that the labelled set of queries have representative queries spanning the entire array of different topics. We propose a Latent Dirichlet Allocation (LDA) [3] topic model based query selection scheme which tries to capture this insight by selecting representative queries which are most *topically* similar to the set of unlabelled queries.

Based on this intuition, we conjecture that representative queries would be those that are most similar to the set of unlabelled queries in terms of their topical distribution. To capture the heterogeneity among all queries in the search logs, we make use of the concept of latent topics. We learn these latent topics from the collection of queries and represent each query as a probability distribution over these latent topics. We train an LDA model, a generative model which posits that each document (query in our case) is a mixture of a small number of topics and that each word's (query term's) creation is attributable to one of the document's (query's) topics. Each query is represented as a feature vector corresponding to its distribution over the LDA topics. To find representative queries, we select the query with the maximum average similarity from among the unlabelled set of queries, i.e.,

$$q^* = argmax_q \frac{1}{|D_u|} \sum_{q_i \in D_u} sim(T_q, T_{q_i}) \tag{3}$$

where $|D_u|$ represents the number of queries in the unlabelled set D_u; T_q represents the query q's feature vector in the LDA topic space and $sim(T_q, T_{q_i})$ can be any similarity score between queries; we use the cosine similarity between the topic-space representations of queries q and q_i.

6. COMBINING REPRESENTATIVENESS & INFORMATIVENESS

The approaches discussed so far have looked at either the *informativeness* of queries and selected queries which are most informative in terms of their ability reduce the uncertainty of the ranking model or they have focussed on *representativeness* of queries and selected representative queries spanning the entire array of different topics. As we discussed earlier in subsection 3.3, optimizing for only one of the two criteria for query selection could significantly limit the performance of query selection by selecting suboptimal query subsets. In this section we present a way of combining the two objectives by means of submodular functions and propose a submodular objective which jointly captures the notions of representativeness and informativeness.

6.1 Submodular Functions

Submodular functions are discrete functions that model laws of diminishing returns and can be defined as follows:[19]: Given a finite set of objects (samples) $Q = \{q_1, ..., q_n\}$ and a function $f : 2^S \rightarrow \Re^+$ that returns a real value for any subset $S \subseteq Q$, f is submodular if given $S \subseteq S'$, and $q \notin S'$

$$f(S+q) - f(S) \geq f(S'+q) - f(S') \tag{4}$$

That is, the incremental "value" of q decreases when the set in which q is considered grows from S to S'. A function is *monotone submodular* if $\forall S \subseteq S'$, $f(S) \leq f(S')$. Powerful guarantees exist for such subtypes of monotone submodular function maximization. Though NP-hard, the problem of maximizing a monotone submodular function subject to a cardinality constraint can be approximately solved by a simple greedy algorithm [19] with a worst-case approximation factor $(1 - e^{-1})$. This is also the best solution obtainable in polynomial time unless P=NP [10].

6.2 Problem Formulation

Submodularity is a natural model for query subset selection in Learning to Rank setting. Indeed, an important characteristic of any query-subset selection technique would be to decrease the value-addition of a query $q \in Q$ based on how much of that query has in common with the subset of queries already selected (S). The value $f(q|S)$ of a query in the context of previously selected subset of queries S further diminishes as the subset grows $S' \supseteq S$. In our setting, each $q \in Q$ is a distinct query, Q corresponds to the entire collection of queries and S corresponds to the subset of queries already selected from Q.

Mathematically, the query subset selection problem can be formulated as selecting the subset of queries S which maximizes the value of $f(S)$ where $f(S)$ captures both the representativeness aspect as well as the informativeness aspects of queries. We next describe in detail the construction of such a monotone submodular function and later present a greedy algorithm to approximately solve the problem of query subset selection.

6.3 Submodular Query Selection

We model the quality of the query subset in terms of both the representativeness & informativeness. To capture both these traits, we model the quality of the query subset as:

$$F(S) = \beta\Phi(S) + (1-\beta)\Psi(S) \tag{5}$$

where $\Phi(S)$ captures the representativeness aspect of the query subset (S) with respect to the entire query set Q while $\Psi(S)$ rewards selecting informative queries. The parameter β controls the trade-off between the importance of representativeness & informativeness while selecting queries. A single weighting scheme would not be suitable for all problems since depending on the constituent queries, size of the overall dataset and the size of the subset that needs to be selected, different weighting schemes would produce different results. The function $F(S)$ will be monotone submodular if each of $\Phi(S)$ and $\Psi(S)$ are individually monotone submodular. We defer an in-depth analysis of the trade-off between representativeness & informativeness aspects to subsection 8.1 and next describe the details of both these functions.

6.3.1 Representativeness: $\Phi(S)$

$\Phi(S)$ can be interpreted either as a set function that measures the similarity of query subset S to the overall query set Q, or as a function representing some form of "representation" of Q by S. Most naturally, $\Phi(S)$ should be monotone, as representativeness improves with a larger subset. $\Phi(S)$ should also be submodular: consider adding a new query to two query subsets, one a subset of the other. Intuitively, the increment when adding a new query to the small subset should be larger than the increment when adding it to the larger subset, as the information carried by the new query might have already been covered by those queries that are in the larger subset but not in the smaller subset. Indeed, this is the property of diminishing returns.

We employ the same functional form of $\Phi(S)$ as was adopted by Lin *et al.*[16]. Specifically, a saturated coverage function is defined as follows:

$$\Phi(S) = \sum_{q \in Q} min \{C_q(S), \alpha C_q(Q)\} \tag{6}$$

where $C_q(S)$ is a set based function defined as $C_q(S) : 2^S \rightarrow \Re$ and $0 \leq \alpha \leq 1$ is a threshold co-efficient. Intuitively, $C_q(S)$ measures how *topically* similar S is to query q or how much of the query q is covered by the subset S. Building on top of the earlier proposed LDA topic model based query selection, we define the coverage function $C_q(S)$ in terms of the *topical coverage* of queries. More specifically,

$$C_q(S) = \sum_{q' \in S} w_{q,q'} \tag{7}$$

where $w_{q,q'} \geq 0$ measures the topical similarity between queries q and q'. Since $C_q(S)$ measures how *topically* similar S is to query q, summing $C_q(S)$ $\forall q \in Q$ would measure how similar the current subset S is to the overall set of queries Q. It is important to note that $C_q(Q)$ is just the largest value $C_q(S)$ can ever obtain because Q is the set of all the queries we have and it maximally represents all the information we have. We call a query q *saturated* by the subset of queries S when $min \{C_q(S), \alpha C_q(Q)\} = \alpha C_q(Q)$. When q is saturated in this way, any new query cannot further improve the coverage even if it is very similar to the query q. Thus, this gives other queries which are not yet saturated a higher chance of being better covered and hence the resulting subset tends to better cover the entire set of queries Q.

6.3.2 *Informativeness:* $\Psi(S)$

The $\Phi(S)$ function described above intuitively captures the notion of coverage or representativeness by selecting subset of queries S which are topically most representative of the entire set of queries Q. While representativeness is an important trait, we also wish to capture the informativeness aspect of queries and select queries which are most informative to the current version of the ranking model. We formulate the functional form of $\Psi(S)$ based on top of the earlier proposed ways of encapsulating query-level informativeness in terms of either ranker disagreements or model uncertainity, or both. As a precursor, it is worth mentioning that to define the function $\Phi(S)$ we make use of LDA topic model which gives us k-topics and we associate each query to one of these k-topics. Formally, we define the $\Psi(S)$ function as follows:

$$\Psi(S) = \sum_{i=1}^{K} \sqrt{\sum_{q \in P_i \cap S} \Upsilon_q} \tag{8}$$

where $P_i, i = 1, ..., K$ is the topical-partition of the set of queries Q into K-topics and Υ_q captures the informativeness carried by the query q based on the current ranking model. The function $\Psi(S)$ rewards topical-diversity along with valuing informativeness since there is usually more benefit to selecting a query from a topic not yet having one of its query already chosen. As soon as a query is selected from a topic, other queries from the same topic start having diminishing gain owing to the square root function ($\sqrt{2}+\sqrt{1} > \sqrt{3}+\sqrt{0}$). It is easy to show that $\Psi(S)$ is submodular by the composition rule. The square root is non-decreasing concave function. Inside each square root lies a modular function with non-negative weights (and thus is monotone). Applying the square root to such a monotone submodular function yields a submodular function, and summing them all together retains submodularity.

The informativeness of a query Υ_q can be defined based on the metrics proposed earlier. To incorporate the informativeness aspects of queries, we experiment with various different formulations of the singleton-query rewards (Υ_q) include the following::

- Disagreement Score for a query - this allows us to capture information about the disagreement about the document rankings for a query among a committee of ranking models [5]
- Uncertainty associated with the query - this allows us to capture the ranking model's uncertainty about the query's document rankings 4
- Combination of uncertainty & disagreement.

Based on empirical analysis, we find that the disagreement based reward functions perform better than the rest of the formulations across all datasets, so we skip the performance comparisons among these.

6.4 Greedy Optimization

Having defined the individual functions based on the different paradigms, we formulate the overall query subset selection problem as the selection of the subset S of queries which maximizes the following function:

$$\begin{aligned} F(S) \quad &= \quad \beta \sum_{q \in Q} min \left\{ \sum_{q' \in S} w_{q,q'}, \alpha \sum_{q' \in Q} w_{q,q'} \right\} \\ &+ \quad (1-\beta) \sum_{i=1}^{K} \sqrt{\sum_{q \in P_i \cap S} \Upsilon_q} \end{aligned} \tag{9}$$

Modelling the query selection problem in such an objective provides many advantages. Firstly, the submodular formulation provides a natural way of coupling the different aspects of query selection. Secondly, the above formulation can be optimized efficiently and scalably given the monotone submodular form of the function $F(S)$. Assuming we wish to select a subset of N queries from the total unlabelled set of Q queries, the problem reduces to solving the following optimization problem:

$$S^* = \underset{S \subseteq Q, |S| \leq N}{\text{argmax}} \; F(S) \tag{10}$$

While solving this problem exactly is NP-complete [10], techniques like ILP [18] can be used but scaling it to bigger datasets becomes prohibitive. Since the function $F(S)$ is submodular, it can be shown that a simple greedy algorithm will have a worst-case guarantee of $f(S^*) \geq (1-\frac{1}{e})F(S_{opt}) \approx 0.63F(S_{opt})$ where S_{opt} is the optimal and S^* is the greedy solution [10]. This constant factor guarantee has practical importance. First, a constant factor guarantee stays the same as N grows, so the relative worst-case quality of the solution is the same for small and for big problem instances. Second, the worst-case result is achieved only by very contrived and unrealistic function instances - the typical case is almost always much better. The greedy solution works by starting with an empty set and repeatedly augmenting the set as

$$S \leftarrow S \cup \underset{q \in Q \setminus S}{\text{argmax}} \, F(q|S) \tag{11}$$

	MQ2007 Dataset					
nQueries	**SF**	**LDA**	**PL**	**ELO**	**QBC**	**RDM**
30	**0.496**$^{*\&}$	0.495	0.493	0.493	0.493	0.482
50	**0.502**$^{*\&}$	0.501	0.496	0.494	0.490	0.485
100	0.509	0.504	**0.510**$^{*\&}$	0.506	0.500	0.501
150	**0.518**$^{*\&}$	0.517	0.510	0.511	0.506	0.507
250	**0.528**$^{*\&}$	0.527	0.519	0.517	0.513	0.516
350	0.527	**0.528***	0.523	0.520	0.525	0.523
400	**0.531**$^{*\&}$	**0.531**$^{*\&}$	0.526	0.523	0.526	0.524
500	**0.535**$^{*\&}$	0.531	0.530	0.528	0.527	0.526

	MQ2008 Dataset					
nQueries	**SF**	**LDA**	**PL**	**ELO**	**QBC**	**RDM**
30	**0.730***	0.728	0.722	0.716	0.728	0.714
50	**0.735***	0.731	0.731	0.720	0.734	0.721
100	**0.741**$^{*\&}$	0.740	0.739	0.724	0.735	0.733
150	**0.743***	0.742	0.740	0.729	0.742	0.734
250	0.745	0.745	**0.748**$^{*\&}$	0.735	0.746	0.740
350	**0.751**$^{\&}$	0.749	0.745	0.749	0.747	0.744
400	**0.753**$^{*\&}$	0.750	0.748	0.746	0.747	0.745

	OHSUMED Dataset					
nQueries	**SF**	**LDA**	**PL**	**ELO**	**QBC**	**RDM**
25	**0.466***	0.459	0.463	0.463	0.465	0.432
30	0.464	0.466	**0.473**$^{*\&}$	0.454	0.455	0.462
35	0.463	**0.478**$^{*\&}$	0.476	0.467	0.458	0.463
40	**0.478**$^{*\&}$	0.460	0.468	0.455	0.469	0.460
45	**0.481**$^{*\&}$	0.473	0.456	0.455	0.464	0.473
50	**0.484**$^{*\&}$	0.466	0.467	0.467	0.472	0.464

Figure 1: Performance evaluation based on NDCG@10 scores for the different algorithms; SF: Submodular function based query selection, LDA: LDA Topic Model based query selection, PL: Permutation Probability Based Query Selection, ELO: expected loss minimization baseline, QBC: Query-By-Committee baseline, RDM: Random query selection baseline. nQueries is the number of queries in the labelled set = base set + actively labelled queries. * and & indicates a statistically significant result (t-test, p≤0.05) when compared to ELO & QBC respectively.

until we select the N number of queries in the subset we intended.

Overall, we select query subsets based on the aforementioned formulations; we next describe in detail the experimental evaluation performed to compare the performances of the three proposed approaches against state-of-the-art baselines.

7. EXPERIMENTAL EVALUATION

We evaluate the proposed query selection strategies on web search ranking and show that the proposed techniques can result in good performance with much fewer labelled queries. We next describe our experimental settings along with the baselines, dataset and evaluation metrics used.

7.1 Compared Approaches

We compare the performance of the proposed query selection strategies against existing state-of-the-art approaches. The compared approaches include:

- **Query-By-Committee (QBC)**: The Query-By-Committee (QBC) approach involves maintaining a committee of models wherein each member is then allowed to vote on the labellings of query candidates. The most informative query is considered to be the instance about which the committee members most disagree. QBC based query selection strategy was used in [5] for ranking adaptation.
- **Expected Loss Optimization (ELO)**: Based on the ELO framework described by Long et al [17], we implemented the query-selection phase of the originally proposed 2-phase active learning framework to select queries wherein the most informative queries are selected by optimizing the expected DCG loss. As is mentioned in the original paper, we use score-range normalization to calculate the gain function. For details, please refer to [17].
- **Random Query Selection (RDM)**: Queries are selected randomly for labelling from among the set of unlabelled queries. It is to be noted that random query selection is the primary method used in most settings [6].
- **Permutation Probability Model (PL)**: Our first proposed approach (§ 4) based on capturing informativeness of queries via the uncertainty reduction principle.
- **Topic Model (LDA)**: Our second proposed approach (§ 5) based on selecting *representative* queries which are most *topically* similar to the set of unlabelled queries.
- **Submodular Model (SF)**: Our final proposed approach (§ 6) based on the coupled submodular objective which incorporates both the aspects of query informativeness & representativeness.

7.2 Dataset

We use three commonlyused real-world learning to rank datasets: (i) MQ2007; (ii) MQ2008 from LETOR 4.0 which uses query sets from Million Query track of TREC 2007, TREC 2008 and (iii) the OHSUMED test collection, a subset of the MEDLINE database, which is a bibliographic database of important, peer-reviewed medical literature maintained by the National Library of Medicine. It is worth mentioning that the proposed approaches make use of query term information which is not available in many other ranking datasets, hence we restrict our evaluation to these three datasets having query term information. There are ∼1700 queries in MQ2007, ∼800 queries in MQ2008 and ∼100 queries in the OHSUMED dataset. The MQ2007 & MQ2008 datasets are of notable size and query selection indeed makes sense in the such datasets; the OHSUMED dataset, on the other hand, has too few queries to select from which isn't ideal for a query selection scenario. Nevertheless, we compare performances across all datasets.

We adopt a 5-fold cross validation scheme with each fold divided into three parts, one each for training, validation and testing in the ratio 3:1:1. Each query-document pair is represented using 46 features [45 in case of the OHSUMED dataset) along with the relevance score from among {0,1,2}. The test set is used to evaluate the different query selection strategies while active learning is performed on queries from the training set.

7.3 Experimental Setting

We start with a base set of 40 labelled queries randomly sampled from the entire query set; the rest of the queries form the candidate set. We make use of $C = 4$ committee members (where applicable) each of which is constructed based on the procedure described earlier (subsection 4.1). To learn the initial ranking models for each of the committee members, we randomly select a sample of 20 queries from the base set of 40 queries and build a ranking model based

	MQ2008 Dataset						MQ2007 Dataset					
% Queries	SF	LDA	PL	ELO	QBC	RDM	SF	LDA	PL	ELO	QBC	RDM
∼5%	0.726	**0.731***	0.721	0.729	0.730	0.728	**0.514**$^{*\&}$	0.505	0.486	0.501	0.498	0.498
∼10%	0.735	**0.737**$^{*\&}$	0.733	0.734	0.734	0.726	**0.508***	0.498	0.501	0.503	0.507	0.496
∼25%	**0.738**$^{*\&}$	0.732	0.733	0.731	0.730	0.727	**0.513**$^{*\&}$	0.509	0.511	0.507	0.511	0.504
∼50%	**0.745**$^{*\&}$	0.731	0.734	0.735	0.734	0.728	**0.516**$^{*\&}$	0.510	0.505	0.509	0.514	0.505

Table 1: Generalizability across different Learning to Rank algorithm: NDCG performance based on ADARANK algorithm. Performance evaluation based on NDCG@10 scores for the different algorithms; SF: Submodular function based query selection, LDA: LDA Topic Model based query selection, PL: min-max Plackett-Luce Based Query Selection, ELO: expected loss minimization baseline, QBC: Query-By-Committee baseline, RDM: Random query selection baseline. % Queries is the % of queries in the labelled set = base set + actively labelled queries. * and & indicates a statistically significant result (t-test, p≤0.05) when compared to ELO & QBC respectively.

on these queries as training data. We first focus on LambdaMART [11], (a state-of-the-art learning to rank algorithm that was the winner of the Yahoo! Learning to Rank challenge [4]) to build ranking models used in the initial part of our experiments. We later show (subsection 8.2) that the queries selected by this method could also be used by other Learning to Rank algorithms.

The entire experiment is repeated multiple times over the 5 folds on each dataset. We perform batch mode Active Learning for queries by selecting a batch of top 10 queries from the candidate set of queries based on the query selection criterion at each round and iteratively add them to our base set. Queries having no relevant documents were ignored while calculating the different metrics. Based on empirical estimation, the threshold parameter in equation 6 was initialized as $\alpha = 0.8$. For our initial results, we evaluate the performance of the proposed query selection strategies based on their NDCG@10 values in the test set. We later analyse the generalizability of our approach on a different metric (MAP).

8. RESULTS

We compare the NDCG@10 performance of the test set against the number of queries in training set (base queries plus actively selected) in Figure 1 for the different datasets and compare the performance of the proposed query selection schemes against the QBC, ELO and Random baselines (statistically significant results are highlighted in the respective tables). For all the methods, the NDCG@10 values tends to increase with the number of iterations which is in line with the intuition that the quality of the ranking model is positively correlated with the number of examples in the training set.

While min-max PL based query selection stems from the same class of approaches (informativeness based) like the two baselines ELO & QBC, it performs better than both these baselines in most cases; this is in line with our initial claim of capturing informative queries from an alternate view of informativeness based on uncertainty reduction. We observe that LDA Topic Model based query selection performs better than existing baselines as well as the PL model which suggests that the quality of the queries selected by this scheme is better than those selected by other strategies which are mostly based on the informativeness aspect. Perhaps selecting queries based on the informativeness results in some noisy *outlier* queries getting selected, a case which LDA topic model based query selection avoids by selecting representative queries. The minor fluctuations and occasional dip in the NDCG values on adding more queries to the labelled set could be explained by the fact that some queries are indeed noisy and selecting such queries induces noise in the ranking models, which results in a slightly worse ranker performance.

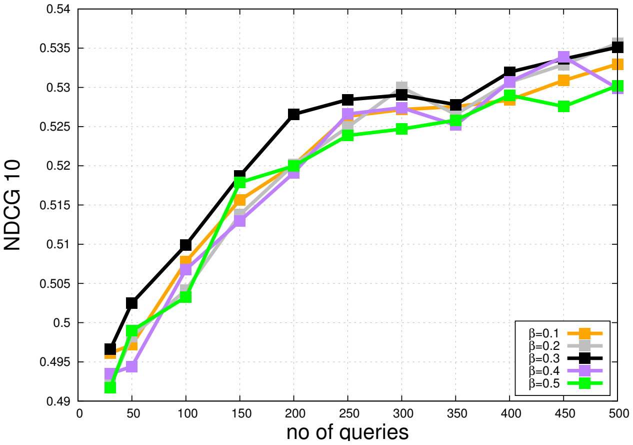

Figure 2: Tradeoff analysis between Informativeness & Representativeness for the MQ2007 datasets. The β coefficient in equation 5 controls the relative importance of the two aspects.

Finally, we observe from the results that the submodular objective (SF) outperforms the baselines as well as (in most cases) our own proposed purely informativeness & purely representativeness based query selection schemes across the different datasets. While purely informativeness based methods tend to select noisy queries, purely representativeness based methods might possibly select queries which are representative but add redundant information. Hence, selecting queries based on the coupled aspects selects queries which are not only representative of other unselected queries, but are also informative to the ranking model.

8.1 Trade-Off between Informativeness & Representativeness

Our main motivation behind introducing the submodular objective was to couple the notions of informativeness and representativeness in a joint coherent manner. Indeed, an ideal subset of queries would be a fine blend of queries which convey the maximal amount of *information* to the ranking model while at the same time, be characteristic of the unselected set of queries. In Figure 2, we present a example analysis on one of the datasets of the relative importance of the two aspects and how they contribute to the overall ranking performance. As can be seen in the figure, a relative weight-

	MQ2008 Dataset						MQ2007 Dataset					
% Queries	**SF**	**LDA**	**PL**	**ELO**	**QBC**	**RDM**	**SF**	**LDA**	**PL**	**ELO**	**QBC**	**RDM**
∼5%	**0.354**$^{*\&}$	**0.354**$^{*\&}$	0.344	0.341	0.340	0.342	**0.164***	0.147	0.147	0.154	0.163	0.149
∼10%	**0.371**$^{*\&}$	0.328	0.354	0.362	0.361	0.352	**0.164**$^{*\&}$	0.146	0.155	0.148	0.160	0.159
∼25%	0.362	**0.369**$^{*\&}$	0.357	0.357	0.362	0.361	**0.165**$^{*\&}$	0.158	0.161	0.157	0.161	0.160
∼50%	0.360	**0.371**$^{*\&}$	0.367	0.369	0.346	0.365	**0.166**$^{*\&}$	**0.166**$^{*\&}$	0.160	0.153	0.135	0.159

Table 2: Generalizability across different Learning to Rank algorithm: AP performance based on Adarank algorithm. Performance evaluation based on NDCG@10 scores for the different algorithms; SF: Submodular function based query selection, LDA: LDA Topic Model based query selection, PL: min-max Plackett-Luce Based Query Selection, ELO: expected loss minimization baseline, QBC: Query-By-Committee baseline, RDM: Random query selection baseline. % Queries is the % of queries in the labelled set = base set + actively labelled queries. * and & indicates a statistically significant result (t-test, p≤0.05) when compared to ELO & QBC respectively.

	MQ2007		MQ2008		OHSUMED	
Algorithm	**SS**	**LCR**	**SS**	**LCR**	**SS**	**LCR**
SF	∼ 370	63%	∼ 330	57%	∼ 45	29%
LDA	∼ 390	61%	∼ 400	48%	∼ 55	14%
PL	∼ 490	51%	∼ 510	35%	∼ 55	14%
ELO	∼ 560	44%	∼ 520	34%	∼ 60	6%
QBC	∼ 620	39%	∼ 540	31%	∼ 60	6%
RDM	∼ 720	29%	∼ 570	27%	∼ 60	6%

Table 3: The performance in terms of the Labelling Cost Reduction (LCR) and the Saturated Size (SS) for the various compared approaches.

	OHSUMED Dataset					
nQueries	**SF**	**LDA**	**PL**	**ELO**	**QBC**	**RDM**
30	0.473	0.469	**0.478**	0.477	**0.478**	0.466
40	**0.478**	**0.478**	0.475	0.472	0.477	0.467
50	**0.478**	0.466	0.469	0.477	0.477	0.473

Figure 3: Results on the OHSUMED dataset with LamdaMART Learning to Rank algorithm. * and & indicates a statistically significant result (t-test, p≤0.05) when compared to ELO & QBC respectively.

ing scheme of $\beta = 0.3$ (which weighs representativeness-vs-informativeness in 3:7 proportions) works best for query selection. This highlights that while representativeness is important, selecting informative queries from the different topics indeed helps. Also, it must be noted that the informativeness term in Equation 9 not only contains contributions from query's singleton informativeness reward, but also has contributions from the topical segregation of queries into partitions. Overall, we chose the coefficient $\beta = 0.3$ to weigh the contributions from the two aspects while reporting results. It is to be noted that domain knowledge about the dataset in consideration can be used to vary β accordingly, depending on the desired proportion of representativeness & informativeness.

For a milder sized dataset (MQ2008), putting more weight on informativeness helps initially while the relative contributions tend toequal out once a certain threshold of queries have been selected. Overall, the general weighting factor or $\beta = 0.3$ works well consistently across different datasets.

8.2 Generalizability Across Learning Algorithms & Metrics

For initial results shown before, the query selection method uses LambdaMART as the learning to rank algorithms optimized for the NDCG metric. Since the labelled learning to rank dataset generated as a result of the query selection process could potentially be used in any future ranking systems, the selected queries should ideally be usable by any learning to rank algorithm, optimized for any metric. We analyze such generalization performance in these sets of experiments. While the initial set of results presented above were NDCG values based on LambdaMART ranking algorithm optimizing for NDCG metric, we divert from our original setting and present results on a different ranker: AdaRank [26] in table 1. Similar results for the OHSUMED dataset can be seen in Fig 3. Additionally, we demonstrate the performance of the proposed query selection strategies on a different metric (MAP) and report results in Table 2. Overall, we can see that the proposed query selection methodologies consistently perform better than the baselines across different ranking algorithms and metrics.

8.3 Labelling Cost Reduction

We next analyse the reduction in labelling cost achieved as compared to the case where the entire set of unlabelled queries were labelled. The performance of the ranking function trained with the whole labelled data set is referred to as the optimal performance. When the performance of the active learning model obtained with the proposed algorithms is comparable to the optimal performance, we call the size of training data as the saturated size (SS). Table 3 highlights the *approximate* labelling cost reduction (LCR) results obtained via the proposed query selection techniques. The %-ages were calculated based on the average number of queries in the training set. The corresponding values were calculated using the LambdaMART implementation with NDCG metric. Experimental evaluation shows the proposed query selection algorithms indeed require less number of queries to be labelled than baseline methods to achieve comparable ranking performance. It is worth mentioning that at some point, adding more queries to the labelled training set doesn't help improve ranking performance, as can be seen by the results of the RDM algorithm in the table: with about 720 labelled queries out of 1015 queries, the algorithm is able to demonstrate comparable ranking performance.

9. CONCLUSION & FUTURE WORK

We formulated approaches to the query selection problem into two classes: *informativeness* based and *representativeness* based strategies and proposed two novel query selection strategies, one from each class respectively: permutation probability based and LDA Topic Model based query selection. Additionally, we argued that an ideal query selection scheme should incorporate insights from both the aspects and presented a principled way of coupling information from the two aspects. Based on rigorous experiments we demonstrated the efficacy of the proposed query selection schemes. A possible line of future work could look at enriching the representativeness aspect by adding document level information to the topic model.

10. REFERENCES

[1] J. A. Aslam, E. Kanoulas, V. Pavlu, S. Savev, and E. Yilmaz. Document selection methodologies for efficient and effective learning-to-rank. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 468–475. ACM, 2009.

[2] M. Bilgic and P. N. Bennett. Active query selection for learning rankers. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1033–1034. ACM, 2012.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[4] C. J. Burges, K. M. Svore, P. N. Bennett, A. Pastusiak, and Q. Wu. Learning to rank using an ensemble of lambda-gradient models. In *Yahoo! Learning to Rank Challenge*, 2011.

[5] P. Cai, W. Gao, A. Zhou, and K.-F. Wong. Relevant knowledge helps in choosing right teacher: active query selection for ranking adaptation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 115–124. ACM, 2011.

[6] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. In *Yahoo! Learning to Rank Challenge*, 2011.

[7] O. Chapelle, Y. Chang, and T.-Y. Liu. Future directions in learning to rank. In *Yahoo! Learning to Rank Challenge*, 2011.

[8] P. Donmez and J. G. Carbonell. Optimizing estimated loss reduction for active sampling in rank learning. In *Proceedings of the 25th international conference on Machine learning*, pages 248–255. ACM, 2008.

[9] P. Donmez, J. G. Carbonell, and P. N. Bennett. Dual strategy active learning. In *Machine Learning: ECML 2007*, pages 116–127. Springer, 2007.

[10] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 1998.

[11] Y. Ganjisaffar, R. Caruana, and C. V. Lopes. Bagging gradient-boosted trees for high precision, low variance ranking models. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 85–94. ACM, 2011.

[12] Y. Hamo and S. Markovitch. The compset algorithm for subset selection. In *IJCAI*, pages 728–733, 2005.

[13] M. Hosseini, I. J. Cox, N. Milic-Frayling, M. Shokouhi, and E. Yilmaz. An uncertainty-aware query selection model for evaluation of ir systems. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 901–910. ACM, 2012.

[14] J. H. Lee. Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 180–188. ACM, 1995.

[15] J. H. Lee. Analyses of multiple evidence combination. In *ACM SIGIR Forum*. ACM, 1997.

[16] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010.

[17] B. Long, O. Chapelle, Y. Zhang, Y. Chang, Z. Zheng, and B. Tseng. Active learning for ranking through expected loss optimization. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM, 2010.

[18] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

[19] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-i. *Mathematical Programming*, 1978.

[20] R. L. Plackett. The analysis of permutations. *Applied Statistics*, pages 193–202, 1975.

[21] K. Raman, P. Shivaswamy, and T. Joachims. Online learning to diversify from implicit feedback. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

[22] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52:55–66, 2010.

[23] R. Silva, M. A. Gonçalves, and A. Veloso. Rule-based active sampling for learning to rank. In *Machine Learning and Knowledge Discovery in Databases*, pages 240–255. Springer, 2011.

[24] M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 2008.

[25] R. W. White and D. Kelly. A study on the effects of personalization and task information on implicit feedback performance. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 297–306. ACM, 2006.

[26] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.

[27] E. Yilmaz and S. Robertson. Deep versus shallow judgments in learning to rank. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 662–663. ACM, 2009.

[28] H. Yu. Svm selective sampling for ranking with application to data retrieval. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 354–363. ACM, 2005.

[29] Y. Yue and C. Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems*, 2011.