Learning Concept Importance Using a Weighted Dependence Model

Michael Bendersky
Dept. of Computer Science
University of Massachusetts
Amherst, MA
bemike@cs.umass.edu

Donald Metzler
Yahoo! Labs
4401 Great America Pkwy.
Santa Clara, CA
metzler@yahoo-inc.com

W. Bruce Croft
Dept. of Computer Science
University of Massachusetts
Amherst, MA
croft@cs.umass.edu

ABSTRACT

Modeling query concepts through term dependencies has been shown to have a significant positive effect on retrieval performance, especially for tasks such as web search, where relevance at high ranks is particularly critical. Most previous work, however, treats all concepts as equally important, an assumption that often does not hold, especially for longer, more complex queries. In this paper, we show that one of the most effective existing term dependence models can be naturally extended by assigning weights to concepts. We demonstrate that the weighted dependence model can be trained using existing learning-to-rank techniques, even with a relatively small number of training queries. Our study compares the effectiveness of both endogenous (collection-based) and exogenous (based on external sources) features for determining concept importance. To test the weighted dependence model, we perform experiments on both publicly available TREC corpora and a proprietary web corpus. Our experimental results indicate that our model consistently and significantly outperforms both the standard bag-of-words model and the unweighted term dependence model, and that combining endogenous and exogenous features generally results in the best retrieval effectiveness.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

Weighted dependence model, query concept weighting

1. INTRODUCTION

WSDM'10, February 4–6, 2010, New York City, New York, USA.

Search queries come in many different flavors, depending on the user's information need and the particular search task. Most modern search engines allow users to express their information needs as free text queries. Although this is a convenient interface for users, it places a heavy burden on the search engine to properly interpret the user's intent. For example, for the query "american airlines reservations", the search engine should identify that the query is made up of the concepts "american airlines" and "reservations" and that the key focus of the query is "reservations". This information, if properly identified and incorporated into the underlying retrieval model, can be used to retrieve highly relevant documents.

However, most traditional information retrieval models, such as language modeling and BM25, utilize very simple user query models. These models tend to treat query terms as independent and of uniform importance. Simple heuristics, such as inverse document frequency (*idf*), are integral parts of these models and can be thought of as a simple query term weighting model, but they are very rigid and are based on a single data source. Furthermore, it is not clear if *idf* is an appropriate measure of importance for phrases and other generic concepts [26]. Recent research has shown that modeling query term dependencies and using non-uniform query term weighting (beyond *idf*) can significantly improve retrieval effectiveness, especially on very large collections and for long, complex queries [3, 17, 20]. To our knowledge, no work exists on simultaneously modeling query term dependencies and weighting generic query term concepts (e.g., unigrams, bigrams, etc.) in a unified, trainable framework. It is precisely this problem that we tackle in this paper.

Our proposed model extends the Markov Random Field model (MRF) for information retrieval [20] by automatically learning query concept weights. By making use of the MRF model, we go beyond the query term independence assumptions made by traditional retrieval models. Furthermore, our proposed extension is a generic framework for learning the importance of query term concepts in a way that directly optimizes an underlying retrieval metric. It is important to note that this is quite different from query segmentation approaches [5, 11, 30]. Optimizing segmentation accuracy is not guaranteed to optimize retrieval effectiveness. By implementing concept weighting directly into the underlying retrieval model we avoid the issue of metric divergence [24]. As we will show, this strategy yields strong retrieval effectiveness gains.

As an illustration of such metric divergence, Table 1 shows an actual example of unigram and bigram concept impor-

Concept	Weight
civil	0.0619
war	0.1947
battle	0.0913
reenactments	0.3487
civil war	0.1959
war battle	0.2458
battle reenactments	0.0540

Table 1: Concept weights generated for query "civil war battle reenactments".

tances learned within our proposed model for the query "civil war battle reenactments". If, instead, the weighting was done based on the output of a query segmenter, then it is likely that the phrase "civil war" and perhaps "battle reenactments" would be given large weights. However, our proposed model assigns high weights to the unigram "reenacments" and the bigram "war battle", which happen to be the most *discriminative* (between relevant and non-relevant documents) concepts, not the most *likely* concepts in terms of query segmentation.

There are three primary contributions of our work. First, we propose a straightforward, effective extension of the MRF model that dynamically weights query concepts. We will show that the model can be automatically trained using standard learning to rank approaches. Second, we show that effective weighting of query concepts can be derived using a combination of endogenous (i.e., internal to the collection) and exogenous (i.e., external to the collection) query concept features. Finally, we conduct an extensive evaluation on several publically available TREC test collections and a real-world web search test collection from a commercial search engine. Our experiments show that our proposed approach is consistently and signficantly better than the current publicly disclosed state-of-the-art text matching model for web search across all test collections.

The rest of this paper is laid out as follows. First, in Section 2 we discuss previous work on term dependencies, concept weighting and learning to rank techniques. Next, Section 3 reviews the MRF model and describes our proposed extension. Then, in Section 4, we present our experimental results. Finally, Section 5 concludes the paper and describes possible directions of future work.

2. RELATED WORK

Modeling atomic query concepts through term dependencies, or proximities, proved to have a significant positive impact on retrieval effectiveness on both TREC and web corpora [2, 4, 9, 20, 23, 25, 31]. Most of this work, however, was restricted to modeling term dependencies, rather than weighting them. In other words, all concept matches in the query had the same impact on the document score. While this assumption is reasonable for short keyword queries, it is much less reasonable for longer, more complex queries.

Recent work, focused on verbose queries, started to explore the direction of assigning varying *document independent* weights to query concepts. Kumaran and Carvalho [14] address this by automatically removing extraneous terms that may have a negative effect on the overall retrieval performance of a query. Bendersky and Croft [3] use a supervised discovery method for "key concepts" in verbose queries, and use a ranking approach that integrates the weighted key concepts with the original query. They find that weighted key concepts approach outperforms the standard bag-of-words model, however its performance is on par with the sequential dependence model (described in Section 3.1) that does not use any concept weighting [20]. Most recently, Lease [17] extended his previous work on term weighting [18] to show that incorporating learned term weights in a sequential dependence model improves the retrieval performance over the unweighted variant for verbose description queries on a number of TREC collections.

An additional information retrieval method that is related to our work is pseudo relevance feedback (PRF), which can be viewed as both query expansion and query term weighting technique [16]. Recently, researchers separately focused on both modeling term dependencies [21] and term weighting [7] within the PRF framework. While we do not focus on PRF-based concept weights in this paper, our concept importance framework is general enough to readily incorporate such weights as additional features.

There are two major differences of our approach with the afforementioned previous work. First, our proposed extension to the MRF model provides a principled retrieval framework that, unlike previously proposed methods, naturally combines both term and phrase weights. Second, the model parameters are estimated by directly maximizing the underlying retrieval metric such as MAP or DCG. This differentiates our work from previous methods for concept weighting that employed indirect parameter estimation, maximizing metrics not directly related to retrieval performance such as classification accuracy [3, 7] or expected query model performance [18, 17]. We show that the direct optimization approach allows us to achieve consistent performance gains over a range of query types, while previous work on concept weighting was mainly concentrated on verbose [3, 17] or expanded [7] queries.

Direct optimization of an underlying retrieval metric ties our work to learning-to-rank approaches for information retrieval (LR4IR) (see Liu [19] for a recent survey). Our formulation of metric optimization is similar to some previous work, and thus allows us to build upon the existing direct optimization methods [22]. The novelty of our method lies in the fact that we are not limited to a linear combination of pairwise query-document features, as is usually the case in LR4IR [19]. Instead, we can also use *individual concept features* to effectively learn a concept weighting model in a similar, yet much more flexible, way than that proposed by Gey [10]. As we will show, this approach allows us to improve upon retrieval models that use only query-document dependent features.

3. WEIGHTED DEPENDENCE MODEL

The Markov Random Field model for information retrieval forms the basis of our concept importance weighting [20]. Our use of the MRF model is motivated by the fact that it consistently demonstrates state-of-the-art retrieval effectiveness for a wide variety of search tasks, especially web search. This is illustrated by the fact that the top performing submissions at the Text REtrieval Conference (TREC) web search evaluations for the past five years (Terabyte Track 2004-2006 [8], Million Query Track 2007-2008 [1]) have made use of the model. Thus, the model is one of the most effective publicly disclosed text matching models for web search.

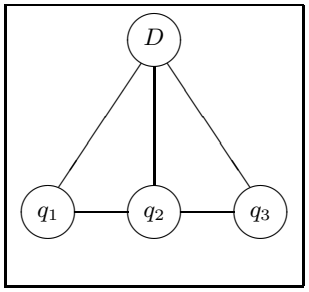

Figure 1: Example Markov random field model for three query terms under the sequential dependence assumption.

In this section we first provide a brief overview of the MRF model and the current best practice for using it. Then, we discuss our proposed extension of the model and how it can be used to learn highly effective weights for various types of query concepts.

3.1 Markov Random Field Model

Markov Random Fields are undirected graphical models that define a joint probability distribution over a set of random variables. A MRF is defined by a graph G, where the nodes in the graph represent random variables and the edges define the dependence semantics between the random variables. In the context of information retrieval, we are interested in modeling the joint distribution over a document random variable D and query term random variables $q_1, \ldots, q_N$ (denoted as Q). An example MRF is shown in Figure 1. In this model, adjacent query terms are dependent on each other since they share an edge, but non-adjacent query terms (e.g., q_1 and q_3) are independent given D.

The joint distribution over the document and query terms is generally defined as:

$$P_{G,\Lambda}(Q, D) = \frac{1}{Z_\Lambda} \prod_{c \in Cliques(G)} \psi(c; \lambda_c) \qquad (1)$$

where $Cliques(G)$ is the set of cliques in G, each $\psi(c; \lambda_c)$ is a non-negative potential function defined over clique configuration c that measures the 'compatibility' of the configuration, Λ is a set of parameters that are used within the potential functions, and Z_Λ normalizes the distribution.

Therefore, to instantiate the MRF model, one must define a graph structure and a set of potential functions. Metzler and Croft propose three different graph structures that make different dependence assumptions about the query terms [20]. The *full independence* variant places no edges between query terms, the *sequential dependence* variant places edges between adjacent query terms (see Figure 1), and the *full dependence* variant places edges between all pairs of query terms. We employ the sequential dependence variant in this work, as it has been shown to provide a good balance between effectiveness and efficiency [20].

Under the sequential dependence assumption, there are two types of cliques that we are interested in defining potential functions over. First, there are cliques involving a single term node and the document node. The potentials for these cliques are defined as follows:

$$\psi(q_i, D; \Lambda) = \exp[\lambda_T f_T(q_i, D)]$$

It is common practice for MRF potential functions to have this type of exponential form, since potentials, by definition, must be non-negative. Here, $f_T(q_i, D)$ is a feature function defined over the query term q_i and the document D, and λ_T is a free parameter. The subscript T denotes that these potentials are defined over *terms*.

The other cliques that we are interested in are those that contain two (adjacent) query term nodes and the document node. The potentials over these cliques are defined as:

$$\psi(q_i, q_{i+1}, D; \Lambda) = \exp[\lambda_O f_O(q_i, q_{i+1}, D) + \lambda_U f_U(q_i, q_{i+1}, D)]$$

where $f_O(q_i, q_{i+1}, D)$ and $f_U(q_i, q_{i+1}, D)$ are feature functions and λ_O and λ_U are free parameters. These potentials are made up of two distinct components. The first considers ordered (i.e., exact phrase) matches and is denoted by the O subscript. The second, denoted by the U subscript, considers unordered matches.

We use the feature functions defined in Table 2, which have been successfully used by researchers in the past [3, 17, 20]. In the table, $tf_{\#1(q_i, q_{i+1}, D)}$ is the number of times that the exact phrase $q_i\ q_{i+1}$ matches in the document and $tf_{\#uw8(q_i, q_{i+1}, D)}$ is the number of times that both terms q_i and q_{i+1} occur (ordered or unordered) within a window of 8 positions in the document. The collection frequencies (cf) are defined analogously. Of course, different potential functions are possible, however an exhaustive exploration of these functions is beyond the scope of this paper.

After making the sequential dependence assumption and substituting the potentials $\psi(q_i, D; \Lambda)$, $\psi(q_i, q_{i+1}, D; \Lambda)$ into Equation 1, documents can be ranked according to:

$$P(D|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} f_T(q, D) + \lambda_O \sum_{q_i, q_{i+1} \in Q} f_O(q_i, q_{i+1}, D) + \lambda_U \sum_{q_i, q_{i+1} \in Q} f_U(q_i, q_{i+1}, D) \qquad (2)$$

Conceptually, this is a weighted combination of a unigram score, an exact bigram match score, and an unordered window bigram match score. Throughout the remainder of this paper we will refer to the ranking method in Equation 2 as the *sequential dependence model*. It has been shown that the parameters $\lambda_T = 0.8$, $\lambda_O = 0.1$, $\lambda_U = 0.1$ are very robust and are optimal or near-optimal across a wide range of retrieval tasks [20, 22].

3.2 Weighted Sequential Dependence Model

One of the primary limitations of the sequential dependence model, as just defined, is the fact that all matches of the same type (e.g., term, ordered window, or unordered window) are treated as being equally important. This is the result of the massive parameter tying that is done in Equation 2. Instead, it would be desirable to weight, *a priori*, different terms (or bigrams) within the query differently based on *query-level evidence*. For example, in a verbose query, there will likely be a few concepts (terms or phrases) within the query that will carry the most weight. While the sequential dependence model would treat all of the concepts as equally important, we would like to be able to weight the concepts appropriately, with regard to each other.

Weighting	Description
$f_T(q_i, D) = \log\left[\frac{tf_{q_i,D}+\mu\frac{cf_{q_i}}{\|C\|}}{\|D\|+\mu}\right]$	Weight of unigram q_i in document D.
$f_O(q_i, q_{i+1}, D) = \log\left[\frac{tf_{\#1(q_i,q_{i+1}),D}+\mu\frac{cf_{\#1(q_i,q_{i+1})}}{\|C\|}}{\|D\|+\mu}\right]$	Weight of exact phrase "q_i q_{i+1}" in document D.
$f_U(q_i, q_{i+1}, D) = \log\left[\frac{tf_{\#uw8(q_i,q_{i+1}),D}+\mu\frac{cf_{\#uw8(q_i,q_{i+1})}}{\|C\|}}{\|D\|+\mu}\right]$	Weight of unordered window q_i q_{i+1} (span = 8) in document D.

Table 2: Summary of language modeling-based unigram and concept weighting functions. Here, $tf_{e,D}$ is the number of times concept e matches in document D, $cf_{e,D}$ is the number of times concept e matches in the entire collection, $|D|$ is the length of document D, and $|C|$ is the total length of the collection. Finally, μ is a weighting function hyperparameter that is set to 2500.

There are several ways to model this in the MRF model, but perhaps the most straightforward is to allow the parameters λ to depend on the concept that they are being applied to, rather than some global weight. This can be achieved by defining the potentials within the model as follows:

$$\begin{aligned}\psi(q_i, D; \Lambda) &= \exp[\lambda(q_i) f_T(q_i, D)] \\ \psi(q_i, q_{i+1}, D; \Lambda) &= \exp[\lambda(q_i, q_{i+1}) f_O(q_i, q_{i+1}, D) + \\ &\quad \lambda(q_i, q_{i+1}) f_U(q_i, q_{i+1}, D)]\end{aligned}$$

where $\lambda(q_i)$ is a parameter that depends on term q_i and $\lambda(q_i, q_{i+1})$ is a parameter that depends on the bigram q_i, q_{i+1}. In this setting, each term and bigram has a separate weight associated with it that is *independent of the document*. This parameter should be some measure of the general importance of the concept with respect to the rest of the query.

Although this formulation of the model is more general, it results in an infeasibly large number of parameters, since each λ now depends on the identity of one (or two) query terms. This was not a problem in the original formulation of the sequential dependence model, because it was assumed that all of the λ parameters, for a given match type, were tied to the same value, resulting in just three parameters. Our proposed solution is in the middle ground between these two extremes. We assume that the parameters λ take on a parameterized form. For simplicity, we assume the following weighted linear form:

$$\lambda(q_i) = \sum_{j=1}^{k_u} w_j^u g_j^u(q_i)$$

$$\lambda(q_i, q_{i+1}) = \sum_{j=1}^{k_b} w_j^b g_j^b(q_i, q_{i+1})$$

where $g^u(q_i)$ and $g^b(q_i, q_{i+1})$ are features defined over unigrams and bigrams, respectively. Similarly, w^u and w^b are free parameters that must be estimated. If there are k_u unigram features and k_b bigram features, then we have a total of $k_u + k_b$ total parameters to estimate, compared to three in the sequential dependence model. The features $g^u(q_i)$ and $g^b(q_i, q_{i+1})$ are document independent and should be useful for determining the relative importance of the concept within the context of the query.

When the parameters λ have this parametric form, the final MRF ranking function can be shown to have the following form:

$$\begin{aligned}P(D|Q) \quad \stackrel{rank}{=} \quad & \sum_{i=1}^{k_u} w_i^t \sum_{q \in Q} g_i^u(q) f_T(q, D) + \\ & \sum_{i=1}^{k_b} w_i^b \sum_{q_j, q_{j+1} \in Q} g_i^b(q_j, q_{j+1}) f_O(q_j, q_{j+1}, D) + \\ & \sum_{i=1}^{k_b} w_i^b \sum_{q_j, q_{j+1} \in Q} g_i^b(q_j, q_{j+1}) f_U(q_j, q_{j+1}, D) \quad (3)\end{aligned}$$

which we call the *weighted sequential dependence model*. It is important to note that this model can easily be extended to handle other dependence assumptions, including the so-called full dependence assumption [20] and other models that focus on key dependencies in the queries [4].

3.3 Concept Importance Features

In this section, we describe the features used for determining the importance of a term or a bigram in a weighted sequential dependence model. Recall that parameters $\lambda(q_i)$ and $\lambda(q_i, q_{i+1})$, which determine the concept weights, are represented as a weighted linear combination of features $g^u(q_i)$ and $g^b(q_i, q_{i+1})$. These features are defined over concepts (either terms or bigrams) and are independent of a specific document. This fact allows us to combine the statistics of the underlying document corpus with the statistics of various external data sources to achieve a potentially more accurate weighting. Accordingly, we divide the features used for concept importance weighting into two main types, based on the type of information they are using.

The first type, the *endogenous*, or collection-dependent, features are akin to standard weights used in information retrieval. They are based on collection frequency counts and document frequency counts calculated over a particular document corpus on which the retrieval is performed.

The second type, the *exogenous*, or collection-independent, features are calculated over an array of external data sources. The use of such sources was found to be beneficial for information retrieval models in previous work [2, 3, 15, 18]. Some of these data sources provide better coverage of terms, and can be used for smoothing sparse concept frequencies calculated over smaller document collections. Others provide more focused sources of information for determining concept importance. In this paper, we use three external data sources: (i) a large collection of web n-grams, (ii) a sample of a query log, and (iii) Wikipedia. Although there are numer-

Data Source	Feature	Description
`Collection`	cf_e	Collection frequency for concept e
	df_e	Document frequency for concept e
`G-Grams`	$gf(e)$	n-gram count of concept e
`MSN Query Log`	$qe\_cnt(e)$	Count of exact matches of a concept e and a query in the log
	$qp\_cnt(e)$	Count of times concept e occurs within a query in the log
`Wikipedia Titles`	$we\_cnt(e)$	Does a concept e appear as a Wikipedia title?
	$wp\_cnt(e)$	Count of times concept e occurs within a Wikipedia title.

Table 3: Statistics used to estimate term importance for a concept e. Concept e is either a query term q_i or a sequential query term pair q_iq_{i+1}.

ous additional data sources that could be potentially used, we intentionally limit our attention to these three sources as they are available for research purposes, and can be easily used to reproduce the reported results.

The first source, *Google n-grams* corpus[1], contains the frequency counts of English n-grams generated from approximately 1 trillion word tokens of text from publicly accessible Web pages. We expect these counts to provide a more accurate frequency estimator, especially for smaller corpora, where some concept frequencies may be underestimated due to the collection size.

In addition, we use a large sample of a query log consisting of approximately 15 million queries[2]. We use this data source to estimate how often a concept occurs in user queries. Intuitively, we assume a positive correlation between an importance of a concept for retrieval and the frequency with which it occurs in queries formulated by search engine users.

Finally, our third external data source is a snapshot of Wikipedia article titles[3]. Due to the large volume and the high diversity of topics covered by Wikipedia (~ 3 million articles in English alone), we assume that important concepts will often appear as (a part of) article titles in Wikipedia.

Table 3 details the statistics used for determining concept weights. As described above, these statistics are based either on the collection or on one of the external data sources. These statistics are used to compute term and bigram features ($g^u(q_i)$ and $g^b(q_i, q_{i+1})$, respectively) in the weighted sequential dependence model (see Equation 3).

For computing the term features, we calculate the statistics presented in the Table 3 for all query terms q_i. This provides us with 7 features $g^u(q_i)$ for determining term importance weights.

To compute the bigram features, we calculate the statistics presented in the Table 3 for all sequential query term pairs q_i, q_{i+1}. For computing collection statistics, we use both the "exact phrase" matches and "unordered window" matches, as described in Table 2. In addition, as bigram features, we compute a ratio $\frac{s(q_iq_{i+1})}{s(q_i)s(q_i+1)}$ for every statistic s in the Table 3. Overall, the combination of the above statistics, provides us with 18 features $g^b(q_i, q_{i+1})$ for determining bigram importance weights.

3.4 Parameter Estimation

In this section, we detail our method for estimating the parameters in the weighted dependence model in Equation 3. There are various ways to estimate the parameters w in the weighted dependence model, including maximum likelihood estimation, maximum *a posteriori* estimation, and discriminative estimation approaches. In this work, we choose to directly optimize the parameters for the retrieval metric of interest, such as mean average precision or discounted cumulative gain.

The primary reason for performing a direct optimization, is our interest in using the weighted dependence model for ranking. Therefore, ranking metrics are the appropriate metrics to optimize for. Another reason is that there is a large and growing body of literature on the learning to rank methods for information retrieval that are being developed for effectively optimizing ranking functions with respect to retrieval metrics [19].

In this work, we employ a very simple, yet highly effective, learning to rank approach that directly optimizes the underlying retrieval metric. It is easy to see that our ranking function is *linear* with respect to w's. Therefore, we make use of the coordinate-level ascent algorithm that was originally proposed in [22], which is easy to implement, efficient for a small number of parameters (as is the case here), and has good empirically verified generalization properties.

The coordinate-level ascent algorithm iteratively optimizes a multivariate objective function by performing a series of one-dimensional line searches. It repeatedly cycles through each parameter w_i in Equation 3, holding all other parameters fixed while optimizing w_i. This process is performed iteratively over all parameters until the gain in the target metric is below a certain threshold.

Although we use this algorithm primarily for its simplicity, any number of other learning to rank approaches that estimate the parameters for linear models can be used. Other possible algorithms include ranking SVMs [13] and SVM^{MAP} [33].

Framing the parameter estimation problem in this manner has several benefits. First, it allows us to make use of any training data that is available, in the form of editorial judgments or click logs, to learn a highly effecive model. Second, the model can be seamlessly integrated into any existing linear learning to rank model. Most other textual features, such as BM25, are highly non-linear with respect to their parameters (i.e., b and k_1). When integrating such a feature into a linear learning to rank model, the b and k_1 parameters would likely have to be estimated in a special manner, because of the non-linearity. However, if the features have a linear form, as is the case in Equation 3, it is straightforward to estimate all the parameters using the existing optimization framework.

[1]Available from Linguistic Data Consortium catalog.
[2]Available as a part of Microsoft 2006 RFP dataset.
[3]Available at: http://download.wikimedia.org/enwiki/

4. EVALUATION

In this section we present the experimental results of our work. We start by detailing the experimental setup in Section 4.1. Next, in Section 4.2, we perform a comprehensive evaluation of our method using several publicly available corpora used at the Text REtrieval Conference (TREC), including newswire and web collections. Finally, to illustrate the benefits of the proposed techniques for web search, in Section 4.3 we test the performance of our method using a proprietary web corpus and a large sample of user queries.

4.1 Experimental setup

All initial indexing and retrieval are implemented using an open-source search engine Indri[4] [29]. The structured query language implemented in Indri natively supports term proximities and custom term weighting schemes, which provides a flexible and convenient platform for evaluating the performance of our method.

During indexing, the documents are stemmed using Porter stemmer. Queries are stopped using a short list of 35 stopwords. For TREC *description* queries (see Section 4.2) an additional set of 17 stopwords is used, designed to remove some of the frequent stop patterns (e.g., "find information") and to improve the performance of the initial retrieval step.

The retrieval experiments are set-up as follows. For all TREC collections, we obtain an initial list of top-1000 results retrieved by an unweighted sequential dependence model. This initial ranking provides a very competitive baseline, as the sequential dependence model was consistently shown to outperform the standard bag-of-word models [17, 20]. We append all the non-retrieved relevant documents to the top-1000 list, and use the resulting document set for training and evaluating all the compared retrieval models, which is a standard practice in evaluating learning to rank methods.

For the proprietary web corpus, we only index web pages that have relevance judgments for our query samples. Training and evaluation of the retrieval models is done using this set of judged web pages, which is a common evaluation practice for this type of test collection. There are, on average, 27 judgments per query.

We compare the performance of our weighted sequential dependence model (`WSD`) to two baseline retrieval models. The first is the query-likelihood model [27] (`QL`), a standard bag-of-words retrieval model implemented by the Indri search engine. The second is the unweighted sequential dependence model (`SD`) as described in Section 3.1. All the initial retrieval parameters are set to default Indri values, which reflect the best-practice settings. All the training/evaluation on both TREC an web corpora is done using five fold cross-validation. The statistical significance of the differences in the performance is determined using a two-sided Wilcoxon sign test, with $\alpha < 0.05$.

We measure the performance using standard retrieval metrics for TREC and web corpora. For TREC, which uses binary relevance judgments, we use precision at top 10 documents retrieved (prec@10), a preference-based measure (b-pref) and a mean average precision at top 1000 documents retrieved (MAP). See, for instance, Buckley and Voorhees [6], for a detailed description of these measures. When estimating the parameters for the `WSD` model, we directly optimize MAP, which is known to be a stable measure [6].

[4]http://www.lemurproject.org/indri/

Name	# Docs	TREC Topic Numbers
ROBUST04	528,155	301-450, 601-700
W10g	1,692,096	451-550
GOV2	25,205,179	701-850

Table 4: Summary of TREC collections and topics used in Section 4.2.

For the web corpus, which uses graded relevance judgments, we use the discounted cumulative gain measure (DCG) [12] at ranks 1, 5, and at the total depth of the ranked list. Relevance is judged as either Perfect, Excellent, Good, Fair, or Bad. The corresponding DCG gains for these grades are 10, 7, 3, 0.5, and 0, respectively.

In the direct optimization of the weighted dependence model, we use the normalized DCG (nDCG) at total depth of the ranked list as the target metric. During the development phase, we found that the results attained by optimizing this metric were more stable over all ranks than the results attained by optimizing for DCG at a particular rank. This can be attributed to the fact that nDCG incorporates information about the entire ranked list, whereas DCG@1 and DCG@5 only consider the top ranked documents. A similar finding was also reported previously by Yilmaz and Robertson [32].

4.2 Evaluation on TREC corpora

In this section, we describe the retrieval results obtained by our model on three standard TREC collections. A summary of the corpora used for these experiments is shown in Table 4. We note that collections vary both by type (ROBUST04 is a newswire collection, while W10g and GOV2 are web collections), number of documents and number of available topics, thus providing a diverse experimental setup for assessing the robustness of the proposed weighted dependence model.

In our evaluation we use both the *title* and the *description* portions of TREC topics as queries. *Title* queries are generally short, and can be viewed as a keyword queries on the topic. *Description* queries are generally more verbose and syntactically richer natural language expressions of the topic. For instance queries *pet therapy* and *How are pets or animals used in therapy for humans and what are the benefits?* are examples of title and description queries on the same topic, respectively.

4.2.1 Retrieval results

Table 5 shows the summary of the retrieval results for the three TREC collections on both *title* and *description* queries. It is evident that both sequential dependence models (`SD` and `WSD`) significantly outperform the query likelihood model `QL` in almost all the cases on all the metrics. This verifies the positive impact of term dependencies on the retrieval performance.

From the two sequential dependence models, weighted sequential dependence model (`WSD`) significantly outperforms the unweighted one (`SD`) on all collections in terms of MAP (which is used as our metric for direct optimization). The gains in MAP range between 1.6% and 24.1%, and are statistically significant for all collections and both query types.

It is interesting to note that even on prec@10 and b-pref metrics, which are not directly optimized, `WSD` is more effective than `SD` in all but two comparisons (prec@10 for GOV2).

title	ROBUST04			W10g			GOV2		
	prec@10	b-pref	MAP	prec@10	b-pref	MAP	prec@10	b-pref	MAP
QL	42.25	25.45	24.93	25.60	18.62	19.04	53.42	35.30	30.19
SD	44.10*	27.10*	26.61* *(+6.7/—)*	**28.90***	19.76*	20.63* *(+8.3/—)*	**57.85***	37.71*	32.47* *(+7.5/—)*
WSD	**44.62***	**27.49**$^*_\dagger$	**27.21**$^*_\dagger$ *(+9.2/+2.3)*	**28.90***	**21.32**$^*_\dagger$	**22.20**$^*_\dagger$ *(+16.6/+7.6)*	57.79*	**38.10***	**33.38**$^*_\dagger$ *(+10.6/+2.8)*

desc	ROBUST04			W10g			GOV2		
	prec@10	b-pref	MAP	prec@10	b-pref	MAP	prec@10	b-pref	MAP
QL	**42.69**	25.37	25.07	32.70	20.47	19.71	51.68	35.70	26.06
SD	41.77	25.90	25.58 *(+2.0/—)*	36.10*	20.73	20.32 *(+3.1/—)*	**53.56**	36.22	26.94* *(+3.4/—)*
WSD	**42.69**	**27.02**$^*_\dagger$	**27.18**$^*_\dagger$ *(+8.4/+6.3)*	**37.10***	**24.33**$^*_\dagger$	**25.23**$^*_\dagger$ *(+28.0/+24.1)*	51.81	**36.64**$^*_\dagger$	**27.38**$^*_\dagger$ *(+5.1/+1.6)*

$*$ Statistically significant difference with QL
† Statistically significance difference with SD

Table 5: Comparison of retrieval results for *title* (top table) and *description* (bottom table) TREC queries with query likelihood (QL), sequential dependence model (SD) and weighted sequential dependence model (WSD). Numbers in parentheses indicate % improvement in MAP over QL/SD (if available).

Gains are as high as 2.8% for prec@10 and 17.4% for bpref. We expect that even higher gains for both prec@10 and b-pref can be attained by WSD by directly training the model for these measures of interest rather than MAP.

4.2.2 Feature analysis

In this section we perform a detailed feature analysis, in order to identify the key elements in the success of the weighted sequential model, as compared to its unweighted counterpart.

Unigrams and Bigrams.

Table 6 compares the impact on the retrieval effectiveness of the importance weights assigned by WSD to either unigrams or bigrams in the sequential dependence model. Recall that the weighted sequential dependence model WSD is derived from its unweighted counterpart by replacing the constants λ_T, λ_O,, and λ_U in Equation 2 with concept dependent parameters $\lambda(q_i)$ and $\lambda(q_i, q_{i+1})$, as shown in Equation 3.

The WSD-UNI model, shown in Table 6, is obtained by replacing λ_T with the term dependent $\lambda(q_i)$, while fixing the values of λ_O and λ_U to those of the unweighted sequential dependence model. Alternatively, WSD-BI model is obtained by replacing λ_O and λ_U with the term dependent $\lambda(q_i, q_{i+1})$, while fixing the value of λ_T.

Table 6 compares the performance of both WSD-UNI and WSD-BI models to the performance of the fully weighted sequential dependence model (WSD). We note that while, in general, both WSD-UNI and WSD-BI outperform SD, in most cases WSD-UNI outperforms WSD-BI as well. This indicates that a unigram weighting has more impact on the retrieval performance than the bigram weighting. This result is in line with previous results reported by Lease for TREC collections [17], which showed that by solely weighting unigrams, one can significantly outperform the unweighted sequential model baseline.

Another important finding shown in Table 6, is that WSD, which combines both unigram and bigram weights, outperforms WSD-UNI in 5 out of 6 comparisons, and always outperforms WSD-BI. In addition, WSD attains statistically significant differences in comparison with WSD-UNI for *description* queries on a large web collection GOV2. This fact underscores the importance of weighting for all the concepts in the sequential dependence model.

Endogeneous and Exogenous Features.

Recall from Section 3.3 that WSD uses two types of features for estimating concept importance: endogeneous (collection-dependent) and exogeneous (collection-independent). While applying collection-dependent features for term weighting has been extensively studied in traditional information retrieval [28], the research on combining them with external sources of information is more recent [3, 18]. Therefore, it is interesting to examine the contribution of each of these feature types to the overall model performance.

Table 7 compares the performance of the weighted sequential dependence model when either only endogeneous (WSD-ENDO) or only exogeneous (WSD-EXO) features are used to the performance of the fully weighted sequential dependence model (WSD). It is evident from Table 7 that using either the endogeneous or the exogeneous features results in comparable performance, and both of them outperform the unweighted dependence model. This indicates that both of these features are useful for learning the optimal weights for WSD. In both cases, however, their combination results in gains in MAP in 4 out of 6 comparisons. In addition, we found that both WSD-ENDO and WSD-EXO display statistically significant differences with WSD on a large web collection GOV2 for *description* queries.

4.3 Evaluation on a commercial web corpus

Previous research has shown that modeling sequential term dependencies has a significant positive impact on retrieval performance in the web search setting [2, 20, 23]. Given the retrieval performance gains obtained from using the weighted variant of the sequential dependence model demonstrated on TREC collections in the previous section, the following set of experiments explores whether these gains can be directly transferred into a web search setting. To this end, in this section we test the ranking with a weighted sequential de-

title	ROBUST04	W10g	GOV2
	MAP	MAP	MAP
WSD	**27.21**	**22.20**	33.38
WSD-UNI	26.85†	21.88	**33.43**
WSD-BI	26.75†	20.65†	32.58

desc	ROBUST04	W10g	GOV2
	MAP	MAP	MAP
WSD	**27.18**	**25.23**	**27.38**
WSD-UNI	27.17	24.86	26.77†
WSD-BI	26.02†	20.43†	27.00

† Statistically significant difference with WSD

Table 6: Comparison of retrieval results for *title* (left) and *description* (right) TREC queries with either only unigram features (WSD-UNI), only bigram features (WSD-BI) or both.

title	ROBUST04	W10g	GOV2
	MAP	MAP	MAP
WSD	**27.21**	**22.20**	33.38
WSD-ENDO	26.85†	21.76	32.81
WSD-EXO	27.01	21.19†	**33.54**

desc	ROBUST04	W10g	GOV2
	MAP	MAP	MAP
WSD	27.18	**25.23**	**27.38**
WSD-ENDO	27.07	23.28	26.95†
WSD-EXO	**27.33**	24.68	27.33†

† Statistically significant difference with WSD

Table 7: Comparison of retrieval results for *title* (left) and *description* (right) TREC queries with either only endogenous features (WSD-ENDO), only exogenous features (WSD-EXO) or both.

pendence model on a proprietary web corpus provided by a large commercial search engine.

To differentiate between the effect of concept weighting on queries of varying length, as was done in the case of TREC corpora, we divide the queries into three groups based on their length. Length is defined as a number of word tokens separated by space in the query.

The first group of queries (*Len-2*) includes very short queries of length two. The second group (*Len-3*) includes queries of length three. The third group (*Len-4+*) consists of more verbose queries of length varying between four and twelve.

While the queries in the first two groups mostly have a navigational intent, the queries in the third group tend to be more complex informational queries. For each group, we randomly sample 1,000 web search queries for which relevance judgments are available. We then train and evaluate (using five fold cross-validation) a separate sequential dependence model and weighted sequential dependence model for each group.

4.3.1 Retrieval results

Table 8 shows the summary of the retrieval results on the three query groups. To demonstrate the impact on the relevance at the top ranks of the retrieved list we report the DCG@1 and DCG@5 measures. To demonstrate the overall ranking quality, we report the results for DCG at unlimited depth (denoted DCG).

Table 8 demonstrates two important findings. First, including term dependence information is highly beneficial for queries of all lengths. SD attains up to 15.4% improvement over QL, which is a bag-of-words model. This result is highly significant, given the large size of our query set.

Second, concept weighting results in significant improvements for longer (*Len-4+*) queries, and its performance is comparable for shorter queries to the performance of the unweighted dependence model (slight improvement on *Len-2* and slight decrease in performance on *Len-3*). For group *Len-4+*, WSD attains improvement of close to 2.5% for DCG@1 and DCG@5. This is a highly significant improvement, especially when taking into account the importance of relevance at top ranks for the web search task.

4.3.2 Feature analysis

Similarly to the feature analysis performed in Section 4.2.2 for TREC corpora, in this section we analyze the importance of different weights and features in the weighted sequential model for the web corpus.

Unigrams and bigrams.

Table 9 compares the impact on the retrieval effectiveness of the importance weights assigned by WSD to either unigrams or bigrams in the sequential dependence model. Notice that, contradictory to what was observed in Table 6 for the TREC data, the bigram weights have more impact on the retrieval effectiveness than the unigram weights.

For short queries in groups *Len-2* and *Len-3*, using bigram weights alone and omitting the unigram weights results in a slightly higher DCG at all measured ranks than using the fully weighted dependence model.

A likely explanation for this effect is the dominance of navigational intent for short queries in web search. TREC topics, including the short *title* queries, mostly have an informational intent and often consist of several separate concepts of unequal importance (e.g., "abandoned mine reclamation"). Short two-three word web queries, on the other hand, often consist of a single navigational bigram ("yahoo mail"), or a bigram followed by an auxiliary term ("yahoo mail login").

Compared to the first two groups, using both unigram and bigram weights in queries in group *Len-4+* results in a better performance than using either of them alone, which is in line with the results for the TREC collections.

We hypothesize that this stems from the fact that a higher percentage of these queries have an informational intent, and they contain both unigram and bigram concepts of varying importance ("best metal songs of the 80s").

Overall, as evident from Table 9, the impact of concept weights is influenced both by the query type and by the collection. While the weighted sequential model can naturally incorporate weighted and unweighted concepts, the optimal weighting policy has to be determined using training on the available data.

	Len-2			Len-3			Len-4+		
	DCG@1	DCG@5	DCG	DCG@1	DCG@5	DCG	DCG@1	DCG@5	DCG
QL	0.803	2.231	10.750	0.784	2.290	8.204	0.629	1.691	5.844
SD	0.926	2.733	11.539 *(+7.3/—)*	**1.008**	**2.971**	**9.139** *(+11.4/—)*	0.864	2.383	6.681 *(+14.3/—)*
WSD	**0.929**	**2.754**	**11.585** *(+7.8/+0.4)*	0.995	2.929	9.087 *(+10.8/-0.6)*	**0.884**	**2.443**	**6.741** *(+15.4/+0.9)*

- All the differences are statistically significant

Table 8: Comparison of retrieval results over a sample of web queries with query likelihood (QL), sequential dependence model (SD) and weighted sequential dependence model (WSD). Numbers in parentheses indicate % improvement in DCG over QL/SD (if available).

Endogenous and exogenous features.

Table 10 compares the performance of the weighted sequential dependence model when either only endogeneous (WSD-ENDO) or only exogeneous (WSD-EXO) features are used to the performance of the fully weighted sequential dependence model (WSD). It is evident from Table 10 that using either the endogeneous or the exogeneous features results in most cases in comparable performance. Similarly to WSD, both of them outperform the unweighted dependence model on queries in group *Len-4+*.

For shorter queries in the first two groups combining the two types of features results in a better performance than using either one in isolation. For queries in a group *Len-4+* using endogenous features alone results in a slightly better performance than the WSD.

We note that the impact of exogeneous features on the overall retrieval performance of the web queries might be potentially boosted by including additional external sources, instead of just three, as is done in our work. For instance, a larger and a more recent sample of user queries than the one used in this study could be employed. However, in the current work we intentionally adhere to using a query log available to other researchers, in order to promote the reproducibility of our results on public data.

As a general "rule of thumb" strategy, a combination of both endogenous and exogenous features appears to be the preferred option both for the TREC and for the web corpora.

5. CONCLUSIONS AND FUTURE WORK

This paper presented a novel extension of the Markov Random Field model for information retrieval. The proposed model provides a robust, effective mechanism for learning query concept importance weights. We showed that parameter estimation in our model can be framed as a learning to rank problem, allowing us to learn concept weights that ultimately directly optimize an underlying information retrieval metric of interest. We also showed that endogenous and exogenous features, such as web-based n-grams and query log information, can be useful for learning document-independent concept importance weights.

In our experiments, we used the proposed framework to learn unigram and bigram importance weights. Our experimental results showed that our proposed model consistently and significantly outperforms the state-of-the-art sequential dependence model across several TREC test collections as well as a web collection from a commercial search engine.

There are several possible directions of future work. First, it would be interesting to incorporate novel sources of concept importance features. For instance, it may be possible to exploit information from click logs to derive better importance estimates.

Additionally, it may be useful to use learning to rank framework similar to the one proposed in this paper to automatically learn parameterized versions of f_T, f_O, and f_U feature functions (see Table 2). The resultant model would make use of a completely learned concept weighting function, which would include both document-dependent (concept score) and document-independent (concept importance) components. Such a function may produce even better retrieval effectiveness than the weighted sequential dependence model proposed in our work.

6. ACKNOWLEDGMENT

This work was supported in part by the Center for Intelligent Information Retrieval, in part by ARRA NSF IIS-9014442, and in part by Yahoo!. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

7. REFERENCES

[1] J. Allan, J. Aslam, B. Carterette, V. Pavlu, and E. Kanoulas. Million query track 2008 overview. In *Proc. 16th Text REtrieval Conference*, 2008.

[2] J. Bai, Y. Chang, H. Cui, Z. Zheng, G. Sun, and X. Li. Investigation of partial query proximity in web search. In *Proc. 17th International Conference on World Wide Web*, pages 1183–1184, 2008.

[3] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *Proc. 31st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2008.

[4] M. Bendersky, W. B. Croft, and D. A. Smith. Two-stage query segmentation for information retrieval. In *Proc. 32nd Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2009.

[5] S. Bergsma and Q. Wang. Learning noun phrase query segmentation. In *Proc. of EMNLP-CoNLL*.

[6] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2004.

[7] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proc. 31st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2008.

[8] C. Clarke, F. Scholar, and I. Soboroff. Overview of the TREC 2005 terabyte track. In *Proc. 14th Text REtrieval Conference*, 2006.

[9] R. Cummins and C. O'Riordan. Learning in a pairwise term-term proximity framework for information retrieval.

	Len-2			Len-3			Len-4+		
	DCG@1	DCG@5	DCG	DCG@1	DCG@5	DCG	DCG@1	DCG@5	DCG
WSD	0.929	2.754	11.585	0.995	2.929	9.087	0.884	**2.443**	**6.741**
WSD-UNI	0.926	2.743	11.556	1.005	2.963	9.132	0.864	2.379	6.677
WSD-BI	**0.930**	**2.758**	**11.602**	**1.012**	**2.967**	**9.132**	**0.888**	2.409	6.711

- All the differences are statistically significant

Table 9: Comparison of retrieval results for a sample of web queries with either only unigram features (WSD-UNI), only bigram features (WSD-BI) or both.

	Len-2			Len-3			Len-4+		
	DCG@1	DCG@5	DCG	DCG@1	DCG@5	DCG	DCG@1	DCG@5	DCG
WSD	**0.929**	**2.754**	**11.585**	**0.995**	**2.929**	**9.087**	0.884	2.443	6.741
WSD-ENDO	0.911	2.687	11.487	0.989	2.924	9.085	**0.885**	**2.455**	**6.760**
WSD-EXO	0.929	2.749	11.575	0.992	2.919	9.079	0.884	2.439	6.732

- All the differences are statistically significant

Table 10: Comparison of retrieval results for a sample of web queries with either only endogenous features (WSD-ENDO), only exogenous features (WSD-EXO) or both.

In *Proc. 32nd Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2009.

[10] F. Gey. Inferring probability of relevance using the method of logistic regression. In *Proc. 17th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1994.

[11] J. Guo, G. Xu, H. Li, and X. Cheng. A unified and discriminative model for query refinement. In *Proc. of 31st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*. ACM, 2008.

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

[13] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. 8th Ann. Intl. ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

[14] G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *Proc. 32nd Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2009.

[15] K. L. Kwok and M. Chan. Improving two-stage ad-hoc retrieval for short queries. In *Proc. 21st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 250–256, 1998.

[16] V. Lavrenko and W. B. Croft. Relevance models in information retrieval. Number 13 in Information Retrieval Book Series, pages 11–56. Kluwer, 2003.

[17] M. Lease. An improved Markov Random Field model for supporting verbose queries. In *Proc. 32nd Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2009.

[18] M. Lease, W. B. Croft, and J. Allan. Regression rank: Learning to meet the opportunity of descriptive queries. In *Proc. 31st European Conf. on Information Retrieval*, 2009.

[19] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3), 2009.

[20] D. Metzler and W. B. Croft. A Markov Random Field model for term dependencies. In *Proc. 28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 472–479, 2005.

[21] D. Metzler and W. B. Croft. Latent concept expansion using Markov Random Fields. In *Proc. 30th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2007.

[22] D. Metzler and W. B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[23] G. Mishne and M. de Rijke. Boosting web retrieval through query operations. In *Proc. 27th European Conf. on Information Retrieval*, pages 502–516, 2005.

[24] W. Morgan, W. Greiff, and J. Henderson. Direct maximization of average precision by hill-climbing with a comparison to a maximum entropy approach. Technical report, MITRE, 2004.

[25] J. Peng, C. Macdonald, B. He, V. Plachouras, and I. Ounis. Incorporating term dependency in the dfr framework. In *Proc. 30th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2007.

[26] J. Pickens and W. B. Croft. An exploratory analysis of phrases in text retrieval. In *Proc. of RIAO 2000*, 1999.

[27] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. 21st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 275–281, 1998.

[28] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[29] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*, 2004.

[30] B. Tan and F. Peng. Unsupervised query segmentation using generative language models and wikipedia. In *Proc. 17th International Conference on World Wide Web*. ACM, 2008.

[31] T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *Proc. 30th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2007.

[32] E. Yilmaz and S. Robertson. On the choice of effectiveness measures for learning to rank. In *SIGIR 2009 Workshop on Learning to Rank for Information Retrieval*, 2009.

[33] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proc. 30th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2007.