On Statistical Analysis and Optimization of Information Retrieval Effectiveness Metrics

Jun Wang and Jianhan Zhu
Department of Computer Science, University College London, UK
wang.jun@acm.org, j.zhu@cs.ucl.ac.uk

ABSTRACT

This paper presents a new way of thinking for IR metric optimization. It is argued that the optimal ranking problem should be factorized into two distinct yet interrelated stages: the relevance prediction stage and ranking decision stage. During retrieval the relevance of documents is not known a priori, and the *joint* probability of relevance is used to measure the uncertainty of documents' relevance in the collection as a whole. The resulting optimization objective function in the latter stage is, thus, the expected value of the IR metric with respect to this probability measure of relevance. Through statistically analyzing the expected values of IR metrics under such uncertainty, we discover and explain some interesting properties of IR metrics that have not been known before. Our analysis and optimization framework do not assume a particular (relevance) retrieval model and metric, making it applicable to many existing IR models and metrics. The experiments on one of resulting applications have demonstrated its significance in adapting to various IR metrics.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H3.1Content analysis and Indexing; H.3.3 Information Search and Retrieval

General Terms

Algorithms, Experimentation, Measurement, Performance

1. INTRODUCTION

In Information Retrieval Modelling, the main efforts have been devoted to, for a specific information need (query), automatically scoring individual documents with respect to their relevance states. Representative examples include the Probabilistic Indexing model that studies how likely a query term is assigned to a relevant document [17], the RSJ model that derives a scoring function on the basis of the log-ratio of probability of relevance [20], to name just a few. And yet, given the fact that in many practical situations relevance information is not steadily available, major developments have shifted their focus to estimating text statistics in the documents and queries and then building up the link through these statistics[12, 21, 34]. For example, scoring functions such as TF·IDF, Vector Space Model, and the Divergence from Randomness (DFR) model [1] have been developed [16]. A practical approximation of the RSJ model led to the popular BM25 scoring function [21]. Another direction in probabilistic modelling was to build a "language model" of a document and assess its likelihood of generating a given query [34]; a query language model is also covered under the Kullback-Leibler divergence based loss function [15].

Despite the efforts for retrieval, when in the evaluation phase, many IR tasks have evaluation criteria that go beyond simply counting the number of relevant documents in a ranked list. Measuring IR effectiveness by different metrics is critical because, for different retrieval goals, we need to capture different aspects of retrieval performance. In the case where the preference goes strongly towards early-retrieved documents, MRR (Mean Reciprocal Rank) is a good measure [28], whereas if we try to capture a broader summary of retrieval performance, MAP (Mean Average Precision) becomes suitable [13]. Thus, there is a gap between the underlying (ranking) decision process of retrieval models and the final evaluation criterion used to measure success in a task. Ideally, it is desirable to have retrieval systems adapted to the specific IR effectiveness metrics.

In fact, IR researchers have already started to explore the opportunity. One extreme case is *learning to rank*; it directly constructs a document ranking model from training data, bypassing the step of estimating the relevance states of individual documents [8]. Under this paradigm, some attempts have been made to directly optimizing IR metrics such as NDCG (Normalized Discounted Cumulated Gain) and MAP [23, 33]. However, it is known that some evaluation metrics are less informative than others [4]. As argued in [32], some IR metrics thus do not necessarily summarize the (training) data well; if we begin optimizing IR metrics right from the data, the statistics of the data may not be fully explored and utilized.

A somewhat opposite direction is to focus still on designing a scoring function of a document, but with the acknowledgement of various retrieval goals and the final rank context. The "less is more" model proposed in [10] is one of the examples. By treating the previously retrieved documents as non-relevant when calculating the relevance of documents for the current rank position, the algorithm is shown to be equivalent to maximizing the Reciprocal Rank measure. In [35], a more general and flexible treatment in this direction is proposed. In the framework, Bayesian decision theory is applied to incorporate various ranking strategies through predefined loss functions. Despite its generality, the resulting IR models, however, lack the ability of directly incorporating IR metrics into the rank decision.

In this paper, we argue that regarding the retrieval task solely as either optimizing IR metrics or deriving a (rele-

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.

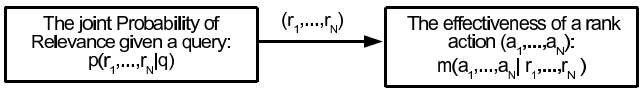


Figure 1: The two distinct stages in the statistical document ranking process.

vance) scoring function presents a partial view of the underlying problem; a more unified view is to divide the retrieval process into two distinct stages, namely relevance prediction and ranking decision optimization stages, and solve them sequentially. In the first stage, the aim is to estimate the relevance of documents as accurate as possible, and summarize it by the joint probability of documents' relevance. Only in the second stage is the rank preference specified, possibly by an IR metric. The rank decision making is a stochastic one due to the uncertainty about the relevance. As a result, the optimal ranking action is the one that maximizes the expected value of the IR metric. We shall show that statistical analysis of the expected value of IR metrics gives insight into the properties of the metrics. One of the findings is that AP (Average Precision) encourages documents whose relevance is positively correlated with previous retrieved documents, while RR (Reciprocal Rank) does otherwise. It follows that if a rank achieves superior results on AP, it must pay with inferiority on RR. Apart from a theoretical contribution, our experiments on TREC data sets demonstrate the significance of our probabilistic framework.

The remainder of the paper is organized as follows. We first establish our optimization scheme, and study major expected IR metrics and practical issues. We then provide an empirical evaluation, and finally conclude our work.

2. STATISTICAL RANKING MECHANICS

In this section, we present the framework of optimizing IR metrics in the situation where the relevance of documents is unknown. To keep our discussion simple, we consider binary relevance, while graded relevance can be extended similarly. Given an information need, let us assume each document in the corpus is either relevant or non-relevant. We denote them jointly as a vector $\mathbf{r} \equiv (r_1, ..., r_k, ..., r_N) \in \{0,1\}^N$, where $k = \{1, ..., N\}$, N denotes the number of documents. $r_k = 1$ if document k is relevant; otherwise r_k=0.

Our view is the following: firstly the IR model should focus on estimating the relevance of documents. The relevance in this stage is the "true" topical relevance [18], different from the user "perceived" relevance that will be qualified in the next stage. In statistical modelling, we assign to every possible relevance state $\mathbf{r}$ a number $p(\mathbf{r}|q)$, which we interpret as the probability that a user, who issues query q, will find the documents' relevance states as $\mathbf{r}$. Given the observation so far (the query, the user's interaction etc), the posterior probability $p(\mathbf{r}|q)$ presents our (or the IR model's) belief about the relevance states of the documents in the collection as a whole. Note that we use the *joint distribution of relevance* instead of the marginal distribution $p(r_k|q)$ to cover the dependency of relevance among documents.

It is argued that only in the second stage does the retrieval model make a ranking decision under the uncertainty specified by the joint probability of relevance. To formulate this, we follow the terminology in natural language processing [6]; a ranking order is represented by a vector $\mathbf{a} \equiv (a_1, ..., a_i, ..., a_N)$, where $a_i \in \{1, ..., N\}$. If a document k is in rank position i, then $a_i = k$. The retrieval task is, thus, to find an optimal rank order $\mathbf{a}$ to maximize a certain retrieval objective. Formally, an IR metric (measure) $m(\mathbf{a}|\mathbf{r})$ is defined as a score function of $\mathbf{a}$ given $\mathbf{r}$. A good metric should be able to measure the user's gain or utility of a rank order $\mathbf{a}$ when the true relevance states of all the documents, $\mathbf{r}$, are known. $m(\mathbf{a}|\mathbf{r})$ can be also seen as a measure of the user's perceived relevance in the context of a ranked list. For example, Precision concerns a solution that finds relevant documents as many as possible in the list regardless of their order, while Reciprocal Rank (inverse of the rank of the first relevant document retrieved) makes sure to retrieve the first relevant document as early as possible regardless of the rank positions of remaining relevant documents.

Given the fact that different IR effectiveness metrics are useful for capturing different aspects of retrieval quality, it is desirable to optimize $\mathbf{a}$ with respect to the specific metric m. Bayesian decision theory suggests that the optimal rank order $\hat{\mathbf{a}}$ is obtained by maximizing the expected IR metric:

$$\hat{\mathbf{a}} = \operatorname*{argmax}_{\mathbf{a}} E_{\mathbf{r}}[m|q] = \operatorname*{argmax}_{\mathbf{a}} \sum_{\mathbf{r}\in\{0,1\}^N} m(\mathbf{a}|\mathbf{r}) p(\mathbf{r}|q), \quad (1)$$

where $E[\cdot|q]$ denotes an expectation with respect to a conditional distribution $p(\cdot|q)$. The subscript $\mathbf{r}$ indicates it is averaged over all possible $\mathbf{r}$. Eq. (1) shows that: firstly the true relevance state of documents, $\mathbf{r}$, is generated from probability $p(\mathbf{r}|q)$ estimated by an IR model. Under the relevance state $\mathbf{r}$, the score of a given rank order $\mathbf{a}$ is calculated. $E_{\mathbf{r}}[m|q]$, the expected score of the rank order, is the one averaging over all possible relevance states of $\mathbf{r}$. Finally, the optimal rank order is chosen by maximizing $E_{\mathbf{r}}[m|q]$.

Although the formulation can be thought of as a special instantiation of the general retrieval decision framework in [15, 35], our underlying idea and development are quite different from their instantiated models. The advantage is that, as illustrated in Figure 1, in our framework, the IR metric (utility) relies only on the true relevance and ranking order, while (relevance) IR models are for estimating the relevance. Decoupling them is essential to directly use any retrieval metric and plug it into the optimization procedure. More discussion can be found in Section 4.

To obtain Eq. (1), we analyze the expected IR metrics $E_{\mathbf{r}}[m|q]$ in Section 2.1 and present a practical implementation and maximization (search) method in Section 2.2.

2.1 Analysis of Expected IR metrics

2.1.1 Expected Average Precision

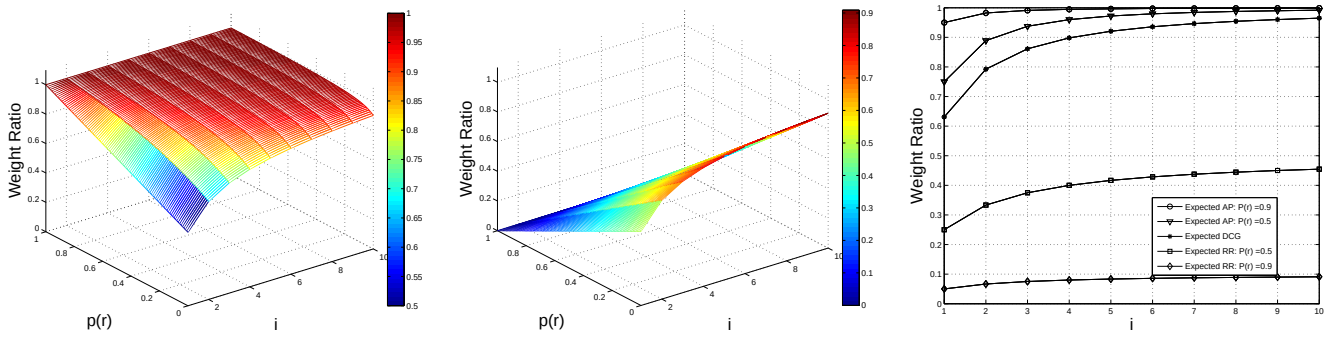
Average Precision (AP) is a widely-adopted metric. For each query, it is the average of the precision scores obtained across rank positions where each relevant document is retrieved; relevant documents that are not retrieved receive a precision score of zero [7]. The metric, in fact, is the area under the Precision-Recall curve, capturing a broad summary of retrieval performance with a single value [4].

By definition, the Average Precision measure is as follows:

$$m_A(\mathbf{a}|\mathbf{r}) \equiv \frac{1}{N_R} \sum_{i=1}^{M} r_{a_i} \frac{(1 + \sum_{j=1}^{i-1} r_{a_j})}{i}, \quad (2)$$

where $M \leq N$ ($\sum_{j=1}^{i-1} r_{a_j} \equiv 0$ when i=1). N_R is the number of relevant documents, and its expected value equals $\sum_{i=1}^{N} p(r_{a_i} = 1)$, the summation of the marginal probability of relevance. For simplicity, we define $p(r_{a_i} = 1) \equiv p(R_{a_i})$ in the remainder of the paper. Because during retrieval $\mathbf{r}$ is hidden, $m_A(\mathbf{a}|\mathbf{r})$ cannot be calculated exactly. Instead, its expected value under the joint probability of relevance is derived by making use of the properties of expectation (Throughout this paper the expectation is all conditioned on a given query q and with respect to $\mathbf{r}$. For simplicity, we drop the subscript $\mathbf{r}$ and notation q in $E[\cdot]$ from now on):

$$E[m_A] = \sum_{N_R} p(N_R|q) E[m_A|N_R] \quad (3)$$



(a) (b) (c)

Figure 2: (a) The adaptive weight w_i^A of the expected Average Precision, (b) The adaptive weight w_i^R of the expected Reciprocal Rank, and (c) Comparison of the weights in different expected IR metrics.

$$= \sum_{N_R} p(N_R|q)\Big(\frac{1}{N_R}\sum_{i=1}^{M}(\frac{E[r_{a_i}|N_R]}{i} + \sum_{j=1}^{i-1}\frac{E[r_{a_i}r_{a_j}|N_R]}{i})\Big)$$

$$= \sum_{N_R} p(N_R|q)\Big(\frac{1}{N_R}\sum_{i=1}^{M}(\frac{E[r_{a_i}|N_R]}{i} + \sum_{j=1}^{i-1}\frac{Cov(r_{a_i}, r_{a_j}|N_R) + E[r_{a_i}|N_R]E[r_{a_j}|N_R]}{i})\Big),$$

where $Cov(r_{a_i}, r_{a_j}|N_R)$ denotes the correlation between the relevance values of documents at rank i and j given the number of relevant documents is N_R. Eq. (3) shows that the expected AP can be interpreted as: for the given query, an IR model first estimates the number of relevant documents in the collection, and then estimates the expected AP for that number of relevant documents. The final expected measure is the average, weighted by $p(N_R|q)$, across all the possible numbers of relevant documents.

We can obtain more insight into the expected AP by making a simple approximation to the average over N_R. By assuming that the posterior distribution of N_R is sharply peaked around the most probable value (the mode) $\hat{N}_R$, we can use the mode to approximate the average [5]. This gives:

$$E_{\mathbf{R}}[m_A] \approx \frac{1}{\hat{N}_R}\sum_{i=1}^{M}\Big(w_i^A p(R_{a_i}) + \sum_{j=1}^{i-1}\frac{Cov(r_{a_i}, r_{a_j})}{i}\Big), \quad (4)$$

where $E[r_{a_i}] = \sum_{r_{a_i}} r_{a_i} p(r_{a_i}) = p(R_{a_i})$, the marginal probability of a document's relevance at rank i. Note that the equation removes the dependency of $\hat{N}_R$ because the conditional expectation and variance are well approximated by the non-conditional ones when $P(\hat{N}_R|q) \approx 1$. To simplify the equation, we also define $w_i^A \equiv \frac{1+\sum_{j=1}^{i-1} p(R_{a_j})}{i}$, which is regarded as an adaptive weight of rank i.

The first term in this simple approximation indicates that the expected AP is a weighted average of the scores across all rank positions, and as we increase the marginal probability of relevance $p(R_{a_i})$ in the ranked list, the expected AP increases. Furthermore, because the weight ratio:

$$\frac{w_{i+1}^a}{w_i^a} = \frac{i}{i+1}(1 + \frac{p(R_{a_i})}{1+\sum_{j=1}^{i-1} p(R_{a_j})}) \quad (5)$$

is in the range between $\frac{i}{1+i}$ and $\frac{2i}{1+i}$.The ratio is adaptive to the expected relevance (defined as $\sum_{j=1}^{i-1} p(R_{a_j})$) received so far. To get the insight into it, we approximate the weight by setting $p(R_{a_i})$ all equal to $p(r)$. We plot the weight ratio against the marginal distribution $p(r)$ and rank position i in Figure 2 (a). It illustrates that when we have more confidence about the relevance of the early retrieved documents ($p(r)$ approaches one), the weight ratio becomes near one. As a result, the metric is less worried about the early retrieved documents, thus putting equal weights to the later-retrieved documents. This is similar to the Precision metric. But once less confident documents ($p(r)$ approaches zero) are retrieved, particularly in the top ranked positions, the weight ratio approaches its lower bound $\frac{i}{i+1}$. As a consequence, the weight penalizes more the later-retrieved relevant documents, and the ratio of the expected AP behaves more like that of the expected DCG, which will be discussed later.

The second term in Eq. (4) indicates that a document will contribute more to the expected AP if its relevance is more positively correlated with those of previous retrieved documents. The consequence is that it will push positively correlated documents up in the ranked list. This is an interesting finding because it shows that the expected AP is in fact *nonlinear* – it models well the dependencies between documents' relevance and incorporates them in deciding the preferred rank order. The rational of encouraging positively correlated relevant documents is that if a document is relevant, it is likely that its positively correlated documents are also relevant. It theoretically explains why pseudo relevance feedback, i.e., the top ranked documents are generally likely to be relevant, and finding other documents similar to these top ranked ones helps improve MAP [24].

2.1.2 Expected DCG and Precision

Discounted Cumulative Gain (DCG) is another popular measure for ranking effectiveness, especially in web search. DCG measures the usefulness, or gain, of a document based on its (graded) relevance[14] (for the moment, let us consider r_{a_i} to cover the graded relevance too); the gain is accumulated from the top of the result list to the bottom. To penalize late-retrieved relevant documents, the gain of each result is discounted by a function of its rank position. By definition, we have the DCG measure as:

$$m_D(\mathbf{a}|\mathbf{r}) = \sum_{i=1}^{M} w_i^D g(r_{a_i}), \quad (6)$$

where w_i^D is the discount weight for rank position i, and $g(r_{a_i})$ is a gain function mapping the relevance value to the retrieval gain. Unlike the expected AP, the expect DCG is linear with respect to rank positions. We thus have:

$$E_{\mathbf{r}}[m_D] = \sum_{i=1}^{M} w_i^D E[g(r_{a_i})] \quad (7)$$

Since $g(r_{a_i})$ is infinitely differentiable in the neighborhood

of the mean of r_{a_i}, i.e., $\hat{r}_{a_i} \equiv E[r_{a_i}]$, the mean of $g(r_{a_i})$ can be represented by a Taylor power series as:

$$\begin{aligned} E[g(r_{a_i})] =& E[g(\hat{r}_{a_i})] + E[(r_{a_i} - \hat{r}_{a_i})g'(\hat{r}_{a_i})] + \\ & E[\frac{1}{2}(r_{a_i} - \hat{r}_{a_i})^2 g''(\hat{r}_{a_i})] + ... \\ =& g(\hat{r}_{a_i}) + 0 + \frac{1}{2}Var(r_{a_i})g''(\hat{r}_{a_i}) + ... \\ \approx& g(\hat{r}_{a_i}) + Var(r_{a_i})\frac{g''(\hat{r}_{a_i})}{2}, \end{aligned} \tag{8}$$

The expected DCG is thus approximated by:

$$E_{\mathbf{r}}[m_D] \approx \sum_{i=1}^{M} w_i^D \Big(g(\hat{r}_{a_i}) + \frac{1}{2}g''(\hat{r}_{a_i})VAR(r_{a_i}) \Big), \tag{9}$$

where $VAR(r_{a_i})$ denotes the variance of r_{a_i}. Eq. (9) shows that the expected value of DCG is determined by both the mean and variance of the relevance of documents at rank positions from 1 to M. Whether it should add variance or minus variance depends on the sign of the second derivative of the gain function. In the case of graded relevance, if consider highly relevant documents more valuable than marginally relevant documents and give them more gain, we can then use a gain function like $g(r_{a_i}) = 2^{r_{a_i}} - 1$. In this case, we need to add variance.

It is shown that when $w_1^D > w_2^D ... > w_M^D$, the document with the highest score of $g(\hat{r}_{a_i}) + \frac{1}{2}g''(\hat{r}_{a_i})VAR(r_{a_i})$ is retrieved first, the document with the next highest score is retrieved second, and so on. It is common to define $w_i^D \equiv \frac{1}{\log_2(i+1)}$. Compared to the adaptive weight in the expected AP, it penalizes more the late-retrieved relevant documents. Figure 2 (c) compares their weight ratios.

Precision at M is a special case of DCG, where the discount is a constant and the gain function is linear. Thus, the expected Precision measure is

$$E[m_P] = \frac{1}{M}\sum_{i=1}^{M} E(r_{a_i}) \equiv \frac{1}{M}\sum_{i=1}^{M} p(R_{a_i}) \tag{10}$$

2.1.3 Expected Reciprocal Rank

In the cases like web search and question answering tasks, we quite often expect a relevant document to be retrieved as early as possible [10, 28]. Expected Search Length and Reciprocal Rank (RR) are strongly biased towards early-retrieved documents. This section analyzes RR, while Expected Search Length can be derived similarly. RR is the inverse of the rank of the first relevant document and bounded between 0 and 1. It is formally defined as:

$$\begin{aligned} m_R(\mathbf{a}|\mathbf{r}) =& r_{a_1}\frac{1}{1} + r_{a_2}(1 - r_{a_1})\frac{1}{2} \\ & + r_{a_3}(1 - r_{a_1})(1 - r_{a_2})\frac{1}{3} + ... \\ =& \sum_{i=1}^{N} \frac{r_{a_i}}{i}\prod_{j=1}^{i-1}(1 - r_{a_j}) = \sum_{i=1}^{N}\frac{1}{i}v_i r_{a_i}, \end{aligned} \tag{11}$$

where we define $v_i \equiv \prod_{j=1}^{i-1}(1 - r_{a_j})$, a function of the relevance values of documents ranked above i; ($v_i \equiv 1$ when $i = 1$). Conceptually, RR measure can be thought of as a weighted average of relevance values at different rank positions, where the weights are adaptive to earlier retrieved documents.

The expected value of the RR measure is the following:

$$\begin{aligned} E[m_R] =& E[\sum_{i=1}^{M}\frac{1}{i}v_i r_{a_i}] = \sum_{i=1}^{M}\frac{E[v_i r_{a_i}]}{i} \\ =& \sum_{i=1}^{M}\frac{E[v_i]E[r_{a_i}] + Cov(r_{a_i}, v_i)}{i} \\ =& \sum_{i=1}^{M}\big(w_i^R p(R_{a_i}) + \frac{1}{i}Cov(r_{a_i}, v_i)\big), \end{aligned} \tag{12}$$

where, similarly, we consider $\frac{E[v_i]}{i}$ as an adaptive weight and denote it as w_i^R. It can be approximated by assuming that the irrelevance of documents above rank i is independent when calculating w_i^R, i.e., $w_i^R \equiv \frac{E[v_i]}{i} \approx \frac{1}{i}\prod_{j=1}^{i-1}(1 - p(R_{a_j}))$. Thus $w_i^R > w_{i+1}^R$. On the one hand, similar to the expected DCG, the weight w_i^R is a discount factor penalizing late retrieved relevant documents. As a result, maximizing the measure intends to push documents that have high marginal distribution of relevance $p(r_j)$ to the top. However, the penalty is much larger than the ones in expected DCG and expected AP. To see this, let us again approximate the weight by setting $p(R_{a_i}) \equiv p(r)$. The weight ratio is compared with those of the expected AP and expected DCG in Figure 2 (c). It shows that expected RR has the smallest weight ratio, while expected AP has the largest. Expected DCG is the one in the middle.

One the other hand, the weight is updated in a completely different way compared to expected AP. Figure 2 (b) plots the weight ratio against the marginal distribution $p(r)$ and rank position i. Different from expected AP, the weight ratio of expected RR becomes larger when $p(r)$ is larger, reinforcing the discount further. As a consequence, it entirely focuses on the quality of a few early retrieval documents. For example, the upper bound for w_3^R is $\frac{1}{12}$. If we consider $p(R_{a_i}) > 0.5$ for $i = \{1, 2, 3\}$, while for DCG it usually equals $\frac{1}{\log_2 4} = \frac{1}{2}$ and for expected AP even larger.

The covariance bit in Eq. (12) shows that overall the expected value of RR increases when relevance of a document is more positively correlated with v_i, the product of non-relevancies $(1 - r_{a_j})$ of the documents above. The effect is that negatively correlated documents will have higher expected RR than positively correlated documents. Such effect will be discounted by a factor $1/i$ at rank i. This is an entirely opposite preference compared to the expected AP. To see this, suppose we have two documents to rank:

$$\begin{aligned} & E[m_{RR}] \\ =& E[R_{a_1}] + \frac{E[R_{a_2}]}{2} - \frac{E[r_{a_1}r_{a_2}]}{2} \\ =& p(R_{a_1}) + \frac{p(R_{a_2})}{2} - \frac{Cov[r_{a_1}, r_{a_2}] + p(R_{a_1})p(R_{a_2})}{2} \\ =& p(R_{a_1}) + w_2^R p(R_{a_2}) - \frac{Cov[r_{a_1}, r_{a_2}]}{2}, \end{aligned} \tag{13}$$

where $w_2^R = \frac{(1 - p(R_{a_1}))}{2}$. It shows that negatively correlated document has a higher value of the expected RR, confirming the findings in [10, 29] that the RR metric is optimized by diversifying the ranked list of documents.

2.1.4 A General View

Through our analysis, it can be seen that the expected IR metrics roughly have two components. A unified definition is given as follows:

$$E[m(\mathbf{a}|\mathbf{r})] \propto \sum_{i=1}^{M}\Big(W_i p(R_{a_i})\Big) + \sum_{i=1}^{M}\frac{V(r_{a_i}, ..., r_{a_1})}{i}, \tag{14}$$

where W_i is the discount weight in position i, and V is a

Table 1: A unified view of expected IR metrics.

	Precision	DCG	AP	RR
Definition:	$\sum_{i=1}^{M} r_{a_i}$	$\sum_{i=1}^{M} \frac{2^{r_{a_i}}-1}{\log_2(i+1)}$	$\frac{1}{N_R}\sum_{i=1}^{M} r_{a_i}\frac{(1+\sum_{j=1}^{i-1} r_{a_j})}{i}$	$\sum_{i=1}^{M} \frac{r_{a_i}}{i}\prod_{j=1}^{i-1}(1-r_{a_j})$
	Expected Precision	Expected DCG	Expected AP	Expected RR
W_i	1	$\frac{1}{\log_2(i+1)}$	$\frac{1+\sum_{j=1}^{i-1} p(R_{a_j})}{i}$	$\frac{\prod_{j=1}^{i-1}(1-p(R_{a_j}))}{i}$
$V(r_{a_i},...,r_{a_1})$	0	0	$\sum_{j=1}^{i-1} Cov(r_{a_i}, r_{a_j})$	$Cov(r_{a_i}, \prod_{j=1}^{i-1}(1-r_{a_j}))$

function defining the correlation between documents. The specific definitions with respect to different metrics are summarized in Table 1. Notice that for DCG, in the case of binary relevance, $g(r_{a_i}) = 2^{r_{a_i}} - 1$ can be approximated as a linear function, and the variance bit vanishes in Eq. (9).

The first bit is a *linear* one with respect to the marginal probability $p(R_{a_i})$. Strictly speaking, this is untrue as W is adaptive to previously retrieved documents. But since the weight ratio W_{i+1}/W_i is usually smaller than one, the maximum value of the first bit is still achieved by ranking in the decreasing order of the marginal probability of relevance. This is identical to what the Probability Ranking Principle has suggested [19]. We call it the *general ranking preference*. The second bit makes the IR metrics different from each other. It is called the *specific ranking preference*. A more detailed discussion and comparison about it is presented in Section 3.1 through a simulation.

2.2 Practical Considerations

Stack Search Maximizing Eq. (14) is a non-trivial task because it needs to search over all possible ranking combinations. We use stack search similar to [30], which keeps a list of the best n ranking combinations as candidates seen so far. These candidates are incomplete solutions till rank i. It then iteratively expands each of the best partial solutions by adding a document at rank $i+1$. For each candidate, we select top-n documents that have the maximum increases of the expected IR metric in Eq. (14). We then put all resulting partial solutions (in this case, $n \times n$) onto the stack and then trim the resulting list of partial solutions to the top n candidates again. We repeat the loop until the end of the rank list is reached. The solution is the one having the maximum value among the candidate solutions. Such a sequential update may not necessarily provide a global optimization solution, but it provides an excellent trade off between accuracy and efficiency by adjusting n. When n is 1, it goes back to the greedy approach. When we increase n, better solutions may be found at the expense of more computational cost. For details refer to [30].

IR Model Calibration To calculate the expected IR metrics during retrieval, we need to estimate the joint probability of relevance. An obvious solution is to directly estimate it from the (training) data [20]. Relevance information is, however, not steadily available in many practical situations to build a robust relevance model. In this paper, we intend to conduct an indirect estimation using existing IR models. It is observed that in many text retrieval experiments that the calculated ranking scores can serve as robust indicators of documents' relevance with respect to queries. Thus, a mapping function can be developed to map from the ranking scores to the probability of relevance. Similar to [29], the joint probability of relevance $p(\mathbf{r}|q)$ is summarized by the marginal probability $p(r_{a_i}|q)$ and covariance $Cov[r_{a_i}, r_{a_j}]$.

Let us first look at $p(r_{a_i}|q)$, and treat it as the utility of ranking scores. We expect the utility, defined as u, to be a non-decreasing function of the ranking score. Thus the first derivative $u' > 0$. It is also expected that u has a maximum value as the ranking score increases. Thus the second derivative $u'' < 0$. Our experiment (Section 3.2) on TREC data has confirmed our intuition. Applying an exponential utility function ($u' > 0$ and $u'' < 0$) [2] gives the mapping function as:

$$p(R_{a_i}|q) \equiv u(s) = 1 - e^{-bs}, \tag{15}$$

where $u(s)$, in the range $[0, 1)$, is the utility of the ranking score s, where $s \geq 0$. b denotes a constant. For the empirical study of the mapping, we refer to Section 3.2.

The next question is how to estimate the covariance

$$Cov[r_{a_i}, r_{a_j}] = \rho(r_{a_i}, r_{a_j})\sqrt{Var[r_{a_i}]Var[r_{a_j}]}, \tag{16}$$

where $Var[r_{a_i}] = (1-p(R_{a_i}))p(R_{a_i})$ if r_{a_i} follows a Bernoulli distribution. The correlation coefficient $\rho(r_{a_i}, r_{a_j})$ models the dependency of relevance between documents at rank i and j. During retrieval, it is reasonable to use the documents' score correlation to estimate the relevance correlation, i.e., $\rho(r_{a_i}, r_{a_j}) \approx \rho(s_{a_i}, s_{a_j})$. Strictly speaking, the score correlation is query-dependent. A practical solution is, however, to approximate it by sampling queries and calculating the correlation between documents' ranking scores from an IR model. In our implementation, we construct each of these queries by randomly sampling query terms from the vocabulary of a data set.

For the expected RR, we need to compute the covariance between document a_i and variable v_i, where v_i is the "meta-relevance" of previously retrieved $i-1$ documents, i.e., $v_i \equiv \prod_{j=1}^{i-1}(1 - r_{a_j})$ as defined in Section 2.1.3. In our implementation, we aggregate the content of the top i-1 documents as a meta document, and estimate the correlation between r_{a_i} and v_i as 1 minus the correlation between the meta document's ranking score and document a_i's ranking score.

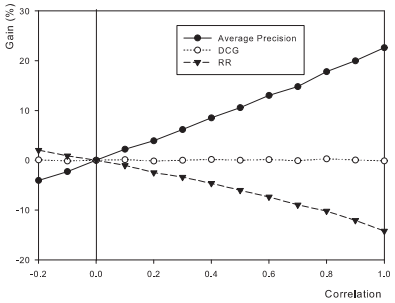


Figure 3: By adjusting the correlation between documents from -0.2 to 1.0, the gain on performance for average precision, DCG, and RR, respectively.

3. EXPERIMENTS

3.1 Simulation

In this section, we carried out a simulation as a confirmation of our analysis about the effect of correlation between different documents' relevance on a range of IR metrics. The relevance states of documents were generated for 10,000 trials. At each trial, for each rank position i, we kept

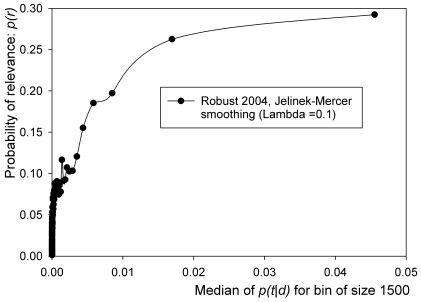


Figure 4: Probability that a result from each bin is relevant against the median of each bin.

the marginal probability of relevance $p(R_{a_i}|q)$ unchanged and generated the relevance/nonrelevance states of the document. The samples were then randomly perturbed so that the correlation between each pair of variables increases from negative to positive (x axis in Figure (3)). For each sample in each trial we calculated the value of an IR metric. We then averaged the metric values across all the trials to obtain the average value. We used the value of the IR metric when the correlation is set as zero as the basis for calculating the gain on the metric when the correlation changes. The results for AP, DCG, and RR are shown in Figure (3). It confirms our derivation of the expected DCG that it is insensitive to correlation. AP value increases when correlation increases, whereas RR does otherwise.

We tried with different settings such as the number of documents, and marginals etc, and got similar findings to the reported above. Previous empirical studies on TREC data have found out that one cannot optimize both the RR and AP metrics at the same time [24, 29]. The analytical forms and the simulation provide direct evidence that the AP metric encourage positively correlated documents whereas the RR metric encourages the opposite.

3.2 IR Model Calibration

In this section, TREC data is used to get an insight into how the mapping function u looks like. Similar to the experimental setup in [22], we measured the utility of ranking scores by the probability that documents given the ranking scores are judged relevant. Documents were binned based on their ranking scores for analysis; we judged the probability that a randomly picked document from each bin is judged as relevant. More specifically, we ran the Jelinek-Mercer smoothing language model on the TREC2004 Robust Track 249 topics with the parameter λ set as its typical value 0.1 [34]. The top 1000 documents were returned for each topic, and there were in total 241,606 results returned for these 249 queries, among which there are 7,029 relevant documents out of a total number of 17,412 relevant documents in the track. The queries contain different numbers of terms. To making the ranking scores comparable across queries, we normalized the ranking scores for all results of each query by dividing these ranking scores by the number of terms in the query.

We sorted the 241,606 results in the descending order in terms of their scores, and divided this ranked list into bins of 1,500 results each, yielding 161 bins: the first 160 bins containing 1,500 results each, and the last bin containing the 1606 documents with the lowest scores. We selected the median score in each bin to represent the bin. In Figure 4, the utility of each bin, i.e., the probability that a randomly chosen result from the bin is relevant, is estimated as the number of relevant documents in each bin divided by the bin size. The data points are based on the pairs of the median of each bin and probability of relevance, and the data points are connected by smoothed curves.

Table 2: Overview of six TREC collections.

Name	Description	Size	# Docs	Topics
TREC8	TREC disks 4&5 minus CR	1.86 GB	528,155	401-450
Robust 2004	TREC disks 4&5 minus CR	1.86 GB	528,155	301-450 and 601-700 minus 672
Robust Hard	TREC disks 4&5 minus CR	1.86 GB	528,155	50 difficult Robust2004 topics
WT10g	TREC Web collection	11 GB	1,692,096	501-550
CSIRO	CSIRO crawl	4.2 GB	370,715	1-50 minus 8 unjudged topics
.Gov	2002 crawl of .gov domain	18 GB	1,247,753	551-600

Figure 4 confirms our intuition that the mapping function is approximately a concave curve ($u' > 0$ and $u'' < 0$) and fitting Eq. (15) to the data in Figure 4 gives b= 9.133. Our experiments showed that the performance of our approach is robust with respect to the choice of b, and a value of b anywhere between 7.0 and 12.0 results in negligible changes of the performance on all the test collections. For the remaining experiments, we fix the parameter b as 9, while bearing in mind that tuning it from training data might have potentials for further performance improvement.

3.3 Performance

We continued our empirical study of the proposed probabilistic retrieval framework, focusing on understanding its ability of optimizing IR metrics. Dirichlet and Jelinek-Mercer smoothing language models were chosen as the two baseline IR models since they are frequently reported for good performance on TREC test collections [34]. For each query, the ranking score of each document, calculated by either of the two IR models, is normalized by dividing them over the number of terms in the query. It is used as the input to estimate the marginal probabilities and covariance on the basis of the discussion in Section 2.2. The stack search is then applied to find an optimal ranking list that maximizes a given IR metric in Eq. (14). For the stack search, we simply set n=1, i.e., equivalent to a greedy approach, while leaving this line of research to future work.

Standard stemming and stopword removing were carried out for both queries and documents. The smoothing parameters of the language models were tuned for the optimal performance for a metric on each data set. The results are reported on six TREC test collections, described in Table 2. TREC8, Robust 2004, and Robust 2004 Hard topics are three plain text collections, and TREC 2001 ad hoc task on WT10g data, TREC 2007 enterprise track document search task on CSIRO data, and TREC 2002 topic distillation task on .Gov data are on three Web collections.

The results in Table 3 indicate that if we choose a certain IR metric to maximize, we obtained in most cases the best performance on this metric than optimizing other metrics and the baselines. More specifically, our approach always had the best performance with respect to MAP and MRR when the objective was to maximize the expected AP and RR, respectively. When we aimed to optimize the expected DCG, our approach improved the baseline on 8 out of 12 occasions in terms of NDCG. It is worth mentioning that no parameter was needed when optimizing the metrics. Without any parameter tuning, our approach consistently outperformed the two baseline models, and eight improvements are statistically significant.

Recall the analysis in Section 2 that the expected AP and RR have a rather "opposite" rank preference (utility) – the expected AP favors a document whose relevance is positively correlated with those of the documents ranked above, whereas the expected RR suggests otherwise. Table 3 demonstrates that the optimization of the expected RR always leads to better performance on MRR than optimization

Table 3: Performance on MAP, NDCG and MRR when the objective is to optimize AP, DCG, and RR, respectively. We used the Dirichlet and Jelinek-Mercer smoothing language models, whose smoothing parameters were tuned for the optimal performance of a metric on each data set, as the baselines in optimization. We highlight the highest performance in bold. A Wilcoxon signed-rank test (p <0.05) is conducted and statistically significant improvements over the baselines are marked with $*$.

TREC8	MAP	NDCG	MRR
Dirichlet (Baseline)	0.224	0.428	0.606
Maximize AP	**0.236***	0.428	0.602
Maximize DCG	0.224	**0.44**	0.615
Maximize RR	0.189	0.436	**0.628**
Jelinek-Mercer (Baseline)	0.228	0.404	0.458
Maximize AP	**0.239***	**0.44***	0.469
Maximize DCG	0.227	0.416	0.476
Maximize RR	0.196	0.404	**0.477**

Robust2004	MAP	NDCG	MRR
Dirichlet (Baseline)	0.221	0.410	0.596
Maximize AP	**0.227**	**0.412**	0.593
Maximize DCG	0.219	0.411	0.593
Maximize RR	0.208	0.391	**0.597**
Jelinek-Mercer (Baseline)	0.221	0.401	0.542
Maximize AP	**0.228**	**0.412**	0.593
Maximize DCG	0.22	0.406	0.543
Maximize RR	0.18	0.364	**0.546**

Robust hard	MAP	NDCG	MRR
Dirichlet (Baseline)	0.088	0.21	0.393
Maximize AP	**0.089**	0.21	0.387
Maximize DCG	**0.089**	**0.235***	0.399
Maximize RR	0.076	0.23	**0.410**
Jelinek-Mercer (Baseline)	0.09	0.225	0.36
Maximize AP	**0.092**	0.23	0.358
Maximize DCG	0.09	**0.245***	0.37
Maximize RR	0.087	0.24	**0.374**

WT10g	MAP	NDCG	MRR
Dirichlet (Baseline)	0.202	0.4	0.550
Maximize AP	**0.204**	0.392	0.546
Maximize DCG	0.199	**0.405**	0.551
Maximize RR	0.181	0.316	**0.552**
Jelinek-Mercer (Baseline)	0.168	0.360	0.472
Maximize AP	**0.176**	**0.376**	0.48
Maximize DCG	0.168	0.360	0.472
Maximize RR	0.153	0.36	**0.481**

CSIRO	MAP	NDCG	MRR
Dirichlet (Baseline)	0.398	0.692	0.782
Maximize AP	**0.408**	0.692	0.785
Maximize DCG	0.395	0.692	0.779
Maximize RR	0.367	0.636	**0.789**
Jelinek-Mercer (Baseline)	0.374	0.684	0.849
Maximize AP	**0.384**	**0.704**	0.85
Maximize DCG	0.371	0.676	0.850
Maximize RR	0.349	0.644	**0.870**

.Gov	MAP	NDCG	MRR
Dirichlet (Baseline)	0.147	0.272	0.419
Maximize AP	**0.151**	0.272	0.417
Maximize DCG	0.148	**0.293***	**0.428**
Maximize RR	0.132	0.238	0.427
Jelinek-Mercer (Baseline)	0.167	0.286	0.45
Maximize AP	**0.187***	**0.306***	0.449
Maximize DCG	0.169	0.286	0.444
Maximize RR	0.147	0.245	**0.454**

of the expected AP, and vice versa. The result supports our theoretical finding that RR and AP are two different types of metrics, and optimizing either of them cannot lead to the optimal performance of the other.

Table 3 also shows that optimization of AP can sometimes lead to better performance on NDCG than direct optimization of DCG. Similar finding appeared in the learning to rank paradigm, and it was argued that the reason is due to the fact that MAP is more informative than DCG [32]. Yet, we think that the informative explanation, although true in learning to rank, does not necessarily hold in our probabilistic framework since we do not use IR metrics to summarize the training data. Our belief is supported by the results from the simulation in Section 3.1 that the expected DCG is invariant to the changes of relevance correlation between documents; and as a result, optimzing AP (prompting documents whose relevance is positively correlated with previous documents) shouldn't do any better than directly optimizing DCG for the NDCG metric. We thus believe the somewhat contradicted finding in the real data set may be attributed to the estimation of the joint probability of relevance, more specifically the relevance correlation, given the fact we used textual content to infer relevancy. As the *cluster hypothesis* suggests that relevant documents tend to be similar to each other to form clusters [25], a document is likely to be relevant if it is similar to relevant documents. As a result, the expected AP biases towards putting documents *similar* with each other in the top rank positions. When top ranked documents are relevant, these other documents are also likely to be relevant - their marginal probabilities of relevance might be higher than the estimated. As a result, metrics such as NDCG and Precision are improved.

Finally, we provide a further account of RR and AP, the two differently behaving metrics. Recall that in Figure 2 the properties of the expected RR and AP were depicted by adjusting the weight functions w_i^A and w_i^R using a single parameter $p(r)$. Figure (5) used TREC8 test collection to further show the effect of $p(r)$ on the resulting MRR and MAP performance. For comparison, the performance of the baseline Dirichlet smoothing language model, and the exact optimization of RR, MAP and DCG was also plotted.

It shows that adjusting $p(r)$ to approximate AP is very stable since the solution keeps roughly the same for all eight values of $p(r)$. This could be explained by the fact that the weight ratio between w_{i+1}^A and w_i^A saturates at 1 for

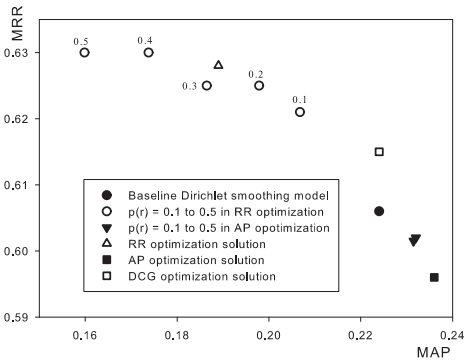


Figure 5: MRR v.s. MAP

all values of $p(r)$ when i increases above 4. By contrast, the RR approximation is more volatile with respect to $p(r)$. As $p(r)$ increases from 0.1 to 0.5, the MRR performance increases whereas the MAP performance decreases. This is due to the fact that as $p(r)$ decreases, the weight ratio of RR becomes similar to that of DCG and AP. $p(r)$ can be used to trade off between the performance of MAP and MRR. When $p(r) = 0.3$ and 0.4, the performance on MRR even slightly exceeds that on the exact optimization of RR. This suggests that there might be still scope to improve our stack search algorithm by setting n higher than 1.

4. LINKS TO OTHER WORK

To complement Section 1, we continue the discussion of related work. In the learning to rank paradigm, optimizing IR metrics is conducted in a discriminative manner where Support Vector Machines or Neural Networks were commonly used [23, 33]. By contrast, we study the problem in a probabilistic framework where the intention is to combine both the generative and discriminative processes. Our formulation of optimal ranking also fundamentally departs from the idea in [26], where a probability distribution over document permutations (rank) is defined, and the expectation of IR metrics is considered under this distribution. In this paper, we, however, believe that the expectation of IR metrics should be with respect to a distribution of relevance, because the uncertainty comes only from the fact that we cannot know the relevance of documents with absolute certainty.

For the purpose of evaluation, the estimation of IR metrics, particularly MAP, has been investigated in the past.

For example, to reduce the variability of test collection, a normalization technique was introduced [11]; to deal with incomplete judgements, sampling approaches were proposed [3, 31]. Empirically, their error rates were measured [7]; and the uncertainty from the variability of relevance judgments in TREC were also examined [27]. By contrast, our study is for the purpose of retrieval, and thus the IR metric estimation and optimization were explored in a complete different situation where the relevance is not known a priori.

The most relevant work can be found in [10, 15, 35]. The study in [10] argued that in some tasks users would be satisfied with a limited number of relevant documents, rather than requiring all relevant documents. The authors therefore proposed to maximize the probability of finding *a* relevant document among the top n. By treating the previously retrieved documents as non-relevant ones, their algorithm is equivalent to optimizing Reciprocal Rank. A more general solution is proposed in [35] on the basis of the Bayesian rank decision framework in [15]. In their solutions, different rank preferences are expressed by different utility functions and can be incorporated when calculating the score for each of the documents. The two ideas are close in spirit to the Maximal Marginal Relevance (MMR) criterion in [9], and can be called "marginal relevance" IR models because they are designed to calculate the additional information a document contributes in a result list. But unfortunately this framework does not allow the capacity to model and optimize different IR metrics.

This paper takes a rather different view, although similar to [15, 35] we also follow the Bayesian decision theory. We argue that the rank utility is nothing to do with the (relevance) model parameters but only with the hidden true topical relevance; and the relevance states of documents need to be estimated before knowing any user (rank) utility. A good IR metric could be able to specify one type of rank utilities. Once we summarize our belief about the true relevance by the joint probability of relevance, the utility, expressed by an evaluation metric, can be estimated under such uncertainty, and the optimal decision is the one that optimizes that expected value. The two distinct retrieval steps do not assume a particular (relevance) retrieval model, making it applicable to many existing IR models and IR metrics.

Our work is also related to the portfolio theory of document ranking [29]. By an analogy with the financial problems, they argued that an optimal rank order is the one that balances the overall relevance (mean) of the ranked list against its risk level (variance). This paper follows the idea of using mean and variance to summarize a distribution and to analyze the expected IR metrics. Our analytical forms of expected IR metrics on the basis of the mean and variance reveal some interesting properties that have not been shown in the past.

5. CONCLUSIONS

In this paper, we have studied the statistical properties of expected IR metrics when the relevance of documents is unknown. An implementation based on our analysis and the two-stage framework has already shown its ability of optimizing major IR metrics in a probabilistic framework. In the future, it is of great interest to seek its usage in web search where click-through data can be viewed as indirect evidence of documents' relevance. Also, during evaluation, the "Cranfield paradigm" considers relevance as deterministic values, either binary or graded ones. It is, however, more general to consider IR evaluation as a stochastic process too. Thus, despite the fact that our study of the expected IR metrics is for retrieval, the analysis and development are also relevant to evaluation if the disagreement between relevance assessors needs to be modelled.

6. REFERENCES

[1] G. Amati and C. J. V. Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, 2002.

[2] K. Arrow. *Aspects of the Theory of Risk-Bearing*. Helsinki: Yrjö Hahnsson Foundation, 1965.

[3] J. A. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In *SIGIR*, 2006.

[4] J. A. Aslam, E. Yilmaz, and V. Pavlu. The maximum entropy method for analyzing retrieval measures. In *SIGIR*, 2005.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[6] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 1993.

[7] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *SIGIR*, 2000.

[8] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML '05*, 2005.

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

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

[11] G. V. Cormack and T. R. Lynam. Statistical precision of information retrieval evaluation. In *SIGIR*, 2006.

[12] W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Document Retrieval Systems*, 1988.

[13] D. Harman. Overview of the second text retrieval conference (trec-2). In *HLT '94*, 1994.

[14] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 2002.

[15] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR*, 2001.

[16] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[17] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *J. ACM*, 1960.

[18] S. Mizzaro. Relevance: The whole history. *Journal of the American Society of Information Science*, 1997.

[19] S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, pages 294–304, 1977.

[20] S. E. Robertson and K. Spärck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–46, 1976.

[21] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*, 1994.

[22] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *SIGIR*, pages 21–29, 1996.

[23] M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: optimizing non-smooth rank metrics. In *WSDM*, 2008.

[24] S. Tomlinson. Early precision measures: implications from the downside of blind feedback. In *SIGIR*, 2006.

[25] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, London, UK, 1979.

[26] M. N. Volkovs and R. S. Zemel. Boltzrank: learning to maximize expected ranking gain. In *ICML '09*, 2009.

[27] E. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Information Processing and Management*, pages 315–323. ACM Press, 1998.

[28] E. M. Voorhees. The TREC-8 question answering track report. In *TREC-8*, pages 77–82, 1999.

[29] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *SIGIR*, 2009.

[30] Y. Wang and A. Waibel. Decoding algorithm in statistical machine translation. In *EACL*, 1997.

[31] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating ap and ndcg. In *SIGIR*, 2008.

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

[33] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR*, 2007.

[34] C. Zhai. Statistical language models for information retrieval a critical review. *Found. Trends Inf. Retr.*, 2(3):137–213, 2008.

[35] C. Zhai and J. D. Lafferty. A risk minimization framework for information retrieval. *Inf. Process. Manage.*, 42(1):31–55, 2006.