A Learning to Rank Approach for Quality-Aware Pseudo-Relevance Feedback

Zheng Ye and Jimmy Xiangji Huang*
Information Retrieval and Knowledge Management Research Lab, School of Information Technology, York University, 4700 Keele Street, Toronto, Ontario M3J 1P3, Canada. E-mail: {yezheng, jhuang}@yorku.ca

Pseudo relevance feedback (PRF) has shown to be effective in ad hoc information retrieval. In traditional PRF methods, top-ranked documents are all assumed to be relevant and therefore treated equally in the feedback process. However, the performance gain brought by each document is different as showed in our preliminary experiments. Thus, it is more reasonable to predict the performance gain brought by each candidate feedback document in the process of PRF. We define the quality level (QL) and then use this information to adjust the weights of feedback terms in these documents. Unlike previous work, we do not make any explicit relevance assumption and we go beyond just selecting "good" documents for PRF. We propose a quality-based PRF framework, in which two quality-based assumptions are introduced. Particularly, two different strategies, relevance-based QL (RelPRF) and improvement-based QL (ImpPRF) are presented to estimate the QL of each feedback document. Based on this, we select a set of heterogeneous document-level features and apply a learning approach to evaluate the QL of each feedback document. Extensive experiments on standard TREC (Text REtrieval Conference) test collections show that our proposed model performs robustly and outperforms strong baselines significantly.

Introduction and Motivation

Relevance feedback (RF) via query expansion (QE) is an effective technique that boosts the retrieval performance of an information retrieval (IR) system by making use of the feedback information. The feedback documents can be obtained by many possible means. In general, there is explicit evidence, such as the labeled relevant documents from real users, or implicit evidence (Sharma & Jansen, 2005), such as the click-through data. Obtaining the feedback information involves extra effort, for example, real-user relevance judgment (Krikon & Kurland, 2011), and is usually expensive. For every given query, the corresponding feedback information is not necessarily available. An alternative solution is pseudo-relevance feedback (PRF), which uses the top-ranked documents in the first-pass retrieval for feedback (Carpineto, de Mori, Romano, & Bigi, 2001; Lavrenko & Croft, 2001; Robertson, Walker, Beaulieu, Gatford, & Payne, 1996; Rocchio, 1971b; Xu & Croft, 2000; Zhai & Lafferty, 2001). Its basic idea is to extract expansion terms from the top-ranked documents to formulate a new query and then process a second-round retrieval. Through query expansion, some relevant documents missed in the first-pass retrieval can then be retrieved so that the overall performance is improved. PRF has been shown to be effective in improving IR performance in a number of IR tasks (Carpineto et al., 2001; Collins-Thompson, 2009; Lavrenko & Croft, 2001; Lin, Lin, Lin, & Zou, 2013; Raman, Udupa, Bhattacharyya, & Bhole, 2010; Robertson et al., 1996; Rocchio, 1971b; Salton & Buckley, 1990; Symonds, Zuccon, Koopman, Bruza, & Sitbon, 2013; White & Marchionini, 2007; Xu & Croft, 2000; Zhai & Lafferty, 2001).

However, PRF can fail when the feedback documents are of low quality, which means that the documents may not help improve the search effectiveness even if they are relevant. A document can be considered relevant when only part of it is relevant. If it is selected as a feedback document, noisy information will be imported and decrease the final performance. Thus, a way to find "good" feedback documents that can improve the effectiveness of PRF is needed. A series of approaches have been proposed for dealing with the problem of low-quality feedback documents (He & Ounis, 2009; Ye, Huang, & Lin, 2011). Most of these studies focused on how to find a set of good feedback documents to improve PRF's effectiveness with a binary mode. Candidate documents will be classified into either a "good" or a "not good" category. For example, classification techniques have been explored to divide the candidate documents for PRF

*Corresponding author.

Received September 20, 2013; revised June 29, 2014; accepted September 12, 2014

 DOI: 10.1002/asi.23430

into bad documents and good documents (He & Ounis, 2009).

In previous work, whether selecting "good" documents or following the traditional PRF methods, the information from feedback documents is not considered thoroughly when selecting feedback terms. Generally, the feedback documents are treated equally after they are chosen. That means that all the feedback terms in these documents are considered reliable equivalently. Actually, the improvements obtained by using different feedback documents can vary extensively, even if they are all of "good" quality. Some documents focusing on the query can provide important feedback terms and help improve the final result significantly, whereas others cannot. Intuitively, terms in these good documents should be considered more relevant and reliable. Thus, it makes sense to investigate the "quality level" (QL) of feedback documents and assign feedback terms in different documents with different weights. However, how to make use of the QLs of feedback documents in PRF has not been well studied in previous work as far as we know.

The main challenges are how to (a) define and measure the QL for each feedback document and (b) incorporate the quality information into the PRF process. In this paper, we define QL as *the contribution of a feedback document to the final performance*. The QL of a document can be determined by many features, so another challenge is how to select features and use them to determine the QL. Traditional heuristic weighting models for the selection of feedback terms can only take advantage of a small number of homogeneous features that may affect information retrieval performance. For example, only the term frequency-inverse document frequency (TFIDF) values of terms are used in the traditional Rocchio model (Rocchio, 1971a). IR is a complex process that may be affected by heterogeneous features. It is therefore necessary to refine this process by taking into account rich heterogeneous features at the document level. Suppose we have these features, we can use them to predict the QL so that we can weight the feedback terms more accurately and flexibly. Because these kinds of features are always heterogeneous, it is difficult to develop a heuristic formula to take into account all the features. Therefore, it makes sense to propose a machine-learning model to estimate the importance of the candidate feedback documents.

In this paper, instead of using the top-k documents obtained in the first-pass retrieval equally, as in traditional PRF work, we propose two QL assumptions and introduce two strategies to estimate the QL of these documents. An intuitive and simple way is the *relevance-based quality level* assumption. In the real world, a document is neither relevant nor irrelevant. Instead, it can operate on different relevance levels. The relevance-based QL assumes the QL of a feedback document is proportional to its relevance level. The other assumption is called the *improvement-based quality level* assumption. A feedback document can improve the final performance with different percentages. We divide the percentages of improvements into several ranges and then assume these ranges as the QLs of documents. The improvement-based QL focuses more on the definition of QL than the relevance-based one, but it is more complex in implementation than the latter. Based on these two assumptions, we exploit machine-learning techniques to further enhance PRF by considering rich features to estimate the quality of feedback documents.

The main contributions of this paper are as follows. First, we formally define a new concept named QL for feedback documents. Second, we introduce two assumptions of QL and propose two strategies to estimate the QL of feedback documents. This can help other researchers go further and make better use of the QL information. Finally, we integrate the QL factor into PRF and propose a learning-to-rank approach for quality-aware PRF. Extensive experiments on five standard TREC (Text REtrieval Conference) data sets show that our proposed framework can outperform strong baselines significantly.

The rest of the paper is organized as follows. In the next section, we review the related work. The Problem Formulation section describes how the QL problem is formulated and the general idea of our proposed framework. Next, A Learning-to-Rank Quality Estmation describes the two assumptions and how we estimate QL. In Experimental Settings we present the results and make a careful analysis of them in Experimental Results. Finally, there are brief conclusions and some thoughts on future directions.

Related Work

In IR, PRF via query expansion is referred to as the techniques, algorithms, or methods that reformulate the original query by adding new terms to achieve better retrieval performance. A classical relevance feedback technique was proposed by Rocchio (1971a) for the SMART retrieval system (Rocchio, 1971b). It takes a set of documents as the feedback information. Unique terms in this set are ranked in descending order of their TFIDF weights. Subsequently, many other relevance feedback techniques and algorithms have been developed, mostly derived under Rocchio's framework. For example, a popular and successful automatic PRF algorithm was proposed (Robertson et al., 1996) in the Okapi system. Amati (2003) proposed a PRF algorithm in the divergence from randomness (DFR) retrieval framework.

In addition, with the development of language model (Ponte & Croft, 1998) in IR, a number of techniques (e.g., Lavrenko & Croft, 2001; Tao & Zhai, 2006; Zhai & Lafferty, 2001) have been developed to fit the language modeling framework. For PRF in the language modeling framework, we always exploit feedback information (e.g., the top-ranked documents set, $F = D_1, D_2, \ldots, D_{|F|}$) to reestimate a more accurate query language model. For example, the model-based feedback approach (Zhai & Lafferty, 2001) is not only theoretically sound but also performs well empirically. Lv and Zhai (2009a) conducted a comparable study of

DOI: 10.1002/asi

five representative state-of-the-art methods for estimating improved query language models in ad hoc information retrieval, including RM3 (a variant of the relevance model [RM]), RM4, DMM (divergence minimization model), SMM (simple mixture model, a variant of model-based feedback approach), and RMM (the regularized mixture model). They found that SMM and RM3 are the most effective in their experiments, and RM3 is more robust in the setting of feedback parameters. The work of Tao and Zhai (2006) is based on the two-component mixture model, to which two modifications have been made so as to eliminate the two parameters that would otherwise need to be set manually. One of the major modifications is to introduce a document-specific mixing coefficient to model potentially different amounts of relevance information in each feedback document, which is similar to the concept of document QL in our work. Unlike from this previous work, however, we not only consider the relevance information but also consider to what degree it is beneficial to the performance of PRF directly. In addition, we use a learning-to-rank approach to address this problem in which a number of salient features are utilized (not only the original query). Our approach can also be easily incorporated into other retrieval frameworks. Last, we focus more on predicting the QL of candidate feedback documents instead of how to set the parameters automatically, as in Tao and Zhai's (2006) work.

However, most of these PRF approaches make a strong assumption that top-ranked documents from the first-pass retrieval are all relevant, and therefore are treated equally in the feedback process. The top-ranked documents are not necessarily good for relevance feedback because they are not evaluated by real users. Following this argument, several studies (He & Ounis, 2009; Lee, Croft, & Allan, 2008) addressed this problem by detecting the right documents for relevance feedback, from which expansion terms are extracted. He and Ounis (2009) proposed detecting good feedback documents by classifying all feedback documents using a variety of features such as the distribution of query terms in the feedback document, the similarity between a single feedback document and all top-ranked documents, or the proximity between the expansion terms and the original query terms in the feedback document. By doing this, PRF is only performed using a selected set of feedback documents, which are predicted to be good among all top-ranked documents. When using machine-learning methods to select good feedback documents, the training data are always scarce. In addition, Lee et al. (2008) proposed a resampling method using clusters to select better documents for PRF. The main idea is to use document clusters to find dominant documents for the initial retrieval set, and to repeatedly feed the documents to emphasize the core topics of a query. Document clusters for the initial retrieval set can represent aspects of a query on especially large-scale web collections because the initial retrieval results may involve diverse subtopics for such collections.

Unlike previous work, we use a finer granularity to demonstrate the "goodness" of a feedback document instead of a binary criterion. Even two "good" documents should be comparable based on their QLs. In this paper, we assume that the contributions/performance gains by different feedback documents are not necessarily the same even if they are evaluated to be relevant by human experts. This is reasonable because different feedback documents can contribute to the overall performance of a retrieval process. Therefore, instead of selecting a refined set of feedback documents, we revisit the traditional PRF framework and propose a new concept, QL. The differences between our work and previous studies are as follows. First, we use QLs to substitute relevance in PRF. The two concepts are related but not the same. Second, we propose several new features to capture the QLs. Finally, we propose two different strategies to evaluate QLs and apply them to PRF.

Problem Formulation

In the first-pass retrieval, given a query Q and a document collection C, a ranked list of document, denoted D, is returned by an information retrieval system. We use d_i to denote the i-th ranked document in D. The top-k documents in D will be used as feedback documents in PRF, which is denoted D_f.

In traditional PRF models, D_f will be all treated as relevant and then used equally to refine the original queries. However, different feedback documents can influence the quality of final queries variously even if they are all relevant. Our goal is to figure out the QLs of feedback documents and use this information to adjust the weights of candidate feedback terms in different documents. Suppose we have training data T, which includes a set of documents labeled with particular QLs. A document d is represented by a vector of n heterogeneous features $<f_1, f_2, f_3 \ldots f_n>$. The selection of features will be introduced in detail below. Based on these features, a learning-to-rank predictor P will be trained and then used to predict the QLs of the top-k documents in D.

In this study, we explore the problem of QLs in the classic Rocchio model. Although Rocchio's model has been used in the information retrieval field for many years, it is still effective in obtaining relevant documents. According to Zhai (2008), BM25 (BM simply stands for "best match") (Robertson, Walker, Jones, Hancock-Beaulieu, & Gatford, 1994) term weighting coupled with Rocchio feedback remains a strong baseline, which is at least as competitive as any language modeling approach for many tasks. This observation is also supported in Miao, Huang, and Ye (2012) and Zhai (2008) as well as in our preliminary experiments of this paper. In the following, we revisit the traditional PRF models and then propose a quality-based PRF framework.

DOI: 10.1002/asi

Our Proposed Framework

Our study is based on a classic framework, Rocchio's model. It models a way of incorporating (pseudo) relevance feedback information into the retrieval process. In the case of PRF, Rocchio's method has the following steps:

1. All documents are ranked for the given query using a particular IR model, for example, the BM25 model (Robertson et al., 1996) in this paper. This step is called *first-pass retrieval*. The $|D_f|$ highest ranked documents are identified as the pseudo relevance set D_f.
2. An expansion weight $w(t, D_f)$ is assigned to each term appearing in the set of the D_f highest-ranked documents. In general, $w(t, D_f)$ is the mean of the weights provided by a weighting model, for example, the TF-IDF weighting model (Salton, Wong, & Yang, 1975) and the KL-divergence weighting model Carpineto et al. (2001). When the KL-divergence weighting model is used with the Rocchio framework, we denote it as $Rocchio_{KL}$ that will be used as a baseline in our experiments.
3. The vector of query terms weight is finally modified by taking a linear combination of the initial query term weights with the expansion weight $w(t, D_f)$ as follows:

$$Q_1 = \alpha * Q_0 + \beta * \sum_{r_i \in D_f} \frac{r_i}{|D_f|} \quad (1)$$

where Q_0 and Q_1 represent the original and first iteration query vectors, r_i is the expansion term weight vector for the i-th feedback document, and α and β are tuning constants controlling how much we rely on the original query and the feedback information.

In our proposed framework, we take the QL factor of feedback documents into account. The original weight of a feedback term is adjusted by the feedback document that it belongs to. Thus, we present our framework as follows:

$$Q_1 = \alpha * Q_0 + \beta * \sum_{r_i \in D_f} \frac{r_i * QLD_i}{|D_f|} \quad (2)$$

where QLD_i is the QL of the i-th feedback document. All the other parameters are the same as in Rocchio's model. In practice, we can always fix α at 1, and only study β to get better performance.

In this paper, we use BM25 as the basic retrieval model in the first-pass retrieval, and the weighting function in our proposed framework is the KL-divergence function. KL-divergence is a popular choice of expansion term weighting, which has been shown to be effective in many state-of-the-art PRF models (Amati, 2003; Carpineto et al., 2001; Ye, Huang, He, & Lin, 2009). KL-divergence measures how a term's distribution in the feedback documents diverges from its distribution in the whole collection. The higher the KL-divergence, the more informative the term. For a unique term in a document d, the KL-divergence weight is given by:

$$KLD(t) = P(t|d) \log_2 \frac{P(t|d)}{P(t|C)} \quad (3)$$

where $P(t|d) = \frac{c(t,d)}{c(d)}$ is the generation probability of term t from D. $c(t, d)$ is the frequency of t in d, and $c(d)$ is the count of words in d. $P(t|C) = \frac{c(t,C)}{c(C)}$ is the collection model. $c(t, C)$ is the frequency of t in collection C, and $c(C)$ is the count of words in the whole collection C.

In this paper, we formulate the problem of QL estimation as a learning-to-rank problem and study how to accurately estimate the quality scores of feedback documents to boost the retrieval performance.

A Learning-to-Rank to Quality Estimation

In the previous section, we formulate the research problem and propose a quality-based feedback framework. The current challenge we face is how to accurately estimate the QL, QLD_i, of each feedback document. Learning-to-rank, also called machine-learned ranking (MLR), is the application of machine learning, typically supervised, semisupervised, or reinforcement learning, which can be used to construct ranking models for information retrieval systems. Training data consist of lists of items with some partial order specified between items in each list. This order is typically induced by giving a numerical or ordinal score or a binary judgment (e.g., "relevant" or "not relevant") for each item. The ranking model's purpose is to rank, that is, produce, a permutation of items in new, unseen lists in a way that is "similar" to rankings in the training data.

The quality of each feedback document correlates with different aspects of the feedback document, which are heterogeneous. It is not easy to establish an empirical formula to integrate all the heterogeneous features. In addition, we also want to go beyond just determining the binary category (good or bad) of a candidate document. Therefore, we use a learning-to-rank model to predict the QL scores QLD_i based on a rich set of salient features detailed in this section. Specifically, to train the learning model we use 2-fold cross-validation in which the data set is partitioned into parts, one for training and the other for testing (see more details below). The learning model is trained on the training set and then applied on the test set to predict the quality level, QLD_i, which will be in our PRF model as described in Equation 2.

In the rest of this section, we first formulate the document quality estimation problem with a learning-to-rank model. Then we describe two QL generation strategies to formally define the QL, which are used for creating the training data set. Finally, we present all the heuristics and the corresponding features to capture the quality of each candidate document for feedback.

DOI: 10.1002/asi

A Learning-to-Rank Model

Here we define our problem as follows: Suppose we are given a set of documents $\mathscr{D}$, queries $\mathscr{Q} = \{Q_i\}_{i=1}^{N}$ and training data $\mathscr{T}$. In addition, we are given a real-valued scoring function (whose output is a real number value) $S_\lambda(D; Q)$, the output of which is QL score parameterized by Λ, a vector of parameters. Given a query Q_i, the quality scoring function $S_\Lambda(D; Q_i)$ is computed for each $D \in \mathscr{D}$ and documents are then ranked in descending order for feedback documents selection according to their quality scores.

The scoring function induces a total ranking $R(\mathscr{D}, Q_i, \Lambda)$ on $\mathscr{D}$ for each query Q_i. For simplicity, we rewrite $R(\mathscr{D}, Q_i, \Lambda)$ as $R_i(\Lambda)$ and let $\mathscr{R}_\Lambda = \{R_i(\Lambda)\}_{i=1}^{N}$ be the set of rankings induced over all of the queries.

Finally, to evaluate a parameter setting, we need an evaluation function $E(\mathscr{R}_\Lambda; \mathscr{T})$ that produces real valued output given a set of ranked lists and the training data. Therefore, our goal is to find the parameter setting Λ that maximizes the evaluation metric E over the parameter space. E(.) could be any performance evaluation metrics (e.g., MAP, NDCG, and P@10). In this paper, we mainly focused on the MAP metric, which is the official metric in the corresponding TREC evaluations. Formally, this can be stated as:

$$\begin{aligned} \hat{\Lambda} &= argmax\Lambda E(\mathscr{R}_\Lambda; \mathscr{T}) \\ s.t.\, &\mathscr{R}_\Lambda \sim S_\Lambda(D; Q) \\ &\Lambda \in M_\Lambda \end{aligned} \tag{4}$$

where $\mathscr{R}_\Lambda \sim S_\Lambda(D; Q)$ denotes that the orderings in $\mathscr{R}_\Lambda$ are induced using quality scoring function S, and M_Λ is the parameter space over Λ. To be more specific, in this work we use a linear feature model as:

$$S(D|Q) = \sum_{i=1}^{k} w_i \cdot f_i(Q, D) \tag{5}$$

where $f_i(Q, D)$ are feature functions and w_i the corresponding weights.

So, given a query and its associated documents, we are trying to find a way to best model the relationship between the documents' QL and query-document features. There are various ways to design and estimate a featured-based model, including statistical classification, logistic regression, and so on. Because this work focuses on ranking the documents with different quality to facilitate better feedback document selection, rather than just classifying them, we adopt the learning-to-rank technique for this purpose. And among various types of learning-to-rank algorithms, we chose one that directly optimizes the parameters in the interest of maximizing the retrieval metric, such as mean average precision or discounted cumulative gain. The main reason for this is our interest in ranking and selecting the really beneficial documents for improving feedback performance. Therefore, it is reasonable to optimize for ranking metrics that reflect the retrieval performance. Some other kinds of learning-to-rank methods designed by minimizing internal defined loss functions might not necessarily obtain the ranking model resulting in the best performance in terms of retrieval metric. A secondary reason is that there is a large number of learning-to-rank algorithms in the literature that are being developed for effectively optimizing ranking functions with respect to retrieval metrics (Liu, 2002).

Among many approaches of this type, we chose to use the coordinate ascent algorithm proposed previously (Metzler & Bruce Croft, 2007), which has been proven to be highly effective for a small number of parameters (Bendersky, Metzler, & Croft, 2010) and is easy to implement. It has good empirical generalization properties.

The coordinate ascent algorithm iteratively optimizes a multivariate objective function by working out a series of one-dimensional line searches. It repeatedly cycles through each parameter of w_i in Equation 5, holding all other parameters fixed while optimizing w_i. This process is carried out iteratively over all parameters until the gain in the target metric is below a certain threshold.

Although we use this algorithm mainly for its simplicity, any other learning-to-rank approaches that estimate the parameters for directly optimizing the retrieval metric will work. Other possible algorithms include SVM_{MAP} (Yue, Finley, Radlinski, & Joachims, 2007), AdaRank (Xu & Li, 2007), LambdaMART (Wu, Burges, Svore, & Gao, 2010), and so on.

QL Strategies

An initial step of our experiments is to create a ground truth, where each candidate feedback document is labeled to a different QL. Our quality model for the feedback documents is then trained based on this ground truth through supervised learning. With respect to this issue, an interesting research question arises: How to define the QL and how to figure it out?

As we discussed previously, documents with different QLs would contribute differently to the PRF process, which will affect the final performance of an information retrieval system. Therefore, it is important to define the document quality to feedback appropriately. In this paper, we investigate two different assumptions: (a) relevance-based QL and (b) improvement-based QL, which are detailed as follows.

Relevance-based QL. In IR, relevance denotes how well a retrieved document or set of documents meets the information needs of the user. Relevance levels can be binary (indicating a result is relevant or that it is not relevant), or graded (indicating results have a varying degree of match between the topic of the result and the information need).

Intuitively, if a document has a higher relevance level to a given query, it is more likely to obtain better performance when this document is used for PRF. Figure 1 plots contribution rates of candidate documents in terms of MAP with different relevance levels on the TREC WT10G and GOV2

DOI: 10.1002/asi

data sets, which solidly support this intuition. Similar trends are also observed on other data sets in our experiments. As we can see, most documents whose relevance level is 2 can improve the performance of PRF, while most of the 0 level documents fail to do so. So it is reasonable to assume that a document with a higher relevance level will also contribute more in the expansion process. Therefore, we can state that a document's quality level to PRF is proportional to its relevance level to the given query.

Improvement-based quality level. In reality, the above assumption may not hold, although it is reasonable. It is possible that two different candidate documents with the same relevance level may contribute differently to PRF, which can be verified to some extent in Figure 1. In addition, even an irrelevant document can sometimes improve the final performance, since it can possibly contain some assistant information for the queries. In Figure 1, about 20% of the 0 level feedback documents can also help improve the performance of PRF, which cannot just be ignored. Meanwhile, some of the top-ranked feedback documents are not really all relevant in experiments. Therefore, we present another assumption which defines document quality to PRF as to how much this document will improve retrieval performance. The borders among different documents are not as clear as in the relevance based strategy. We only focus on the improvements obtained by documents in the training and testing data sets and will not ignore the potential contributions of irrelevant documents. The improvement obtained by a feedback document is calculated as follows:

$$imp(d) = (AP(Q) - AP(Q'))/AP(Q) \quad (6)$$

where $imp(d)$ is the improvement percentage obtained by the feedback document d, $AP(Q)$ is the average performance (AP) of the original query Q, $AP(Q')$ is the AP of the reformulated query Q' by using d.

In particular, given a query, we use each of the candidate documents from the first-pass retrieval to do PRF, and then categorize the documents into different quality levels according to the percentage of improvement by this single document. In this paper, we use a 3-grade strategy: (a) 0 if $imp(d) <= 0$, (b) 1 if $0 < imp(d) <= 0.1$, (c) 2 if $imp(d) > 0.1$.

Heuristics and Features

We propose a set of features guided by heuristics, but our method also allows other features to be explored. In particular, we apply a list of features to assist predicting the document qualities for PRF. The applied features take into account different aspects of feedback documents, which may affect the quality of feedback document, in an attempt to capture the salient characteristics of the good quality documents. The applied features are detailed as follows.

Relevance scores. This intuitive feature uses the relevance score produced by the weighting model for each feedback document. The use of the relevance score feature implies that the higher a document is ranked in the first-pass retrieval, the more chance it can be a high-quality document for relevance feedback. In particular, we use the score obtained by the Okapi BM25 model, which is the basic retrieval mode used in our paper. This feature is denoted `Feature 1:` $f1$.

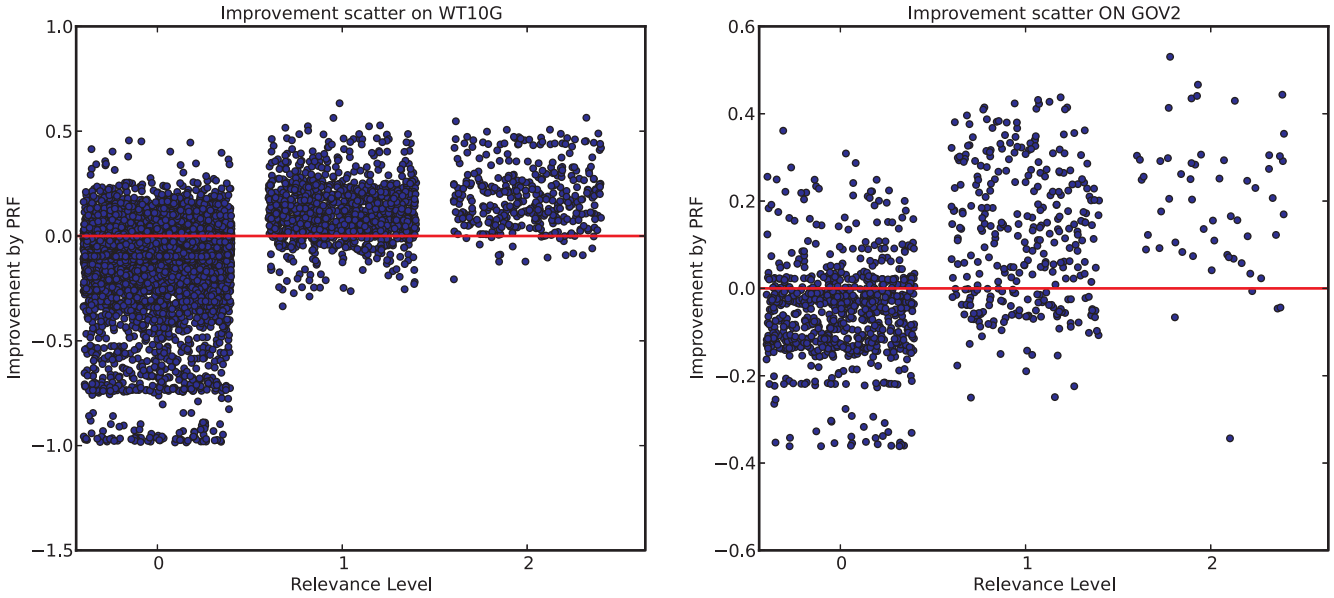

FIG. 1. The scatterplot of improvements on different relevance levels. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

DOI: 10.1002/asi

Proximity scores. Term proximity is an effective measure for modeling term associations, which has been studied extensively in the past few years. Various methods (Büttcher, Clarke, & Lushman, 2006; Clarke, Cormack, & Tudhope, 2000; Daoud & Huang, 2013; Keen, 1991; Lv & Zhai, 2009b; Rasolofo & Savoy, 2003; Scholer, Williams, & Turpin, 2004; Song, Taylor, Wen, Hon, & Yu, 2008; Tao & Zhai, 2007) of integrating proximity information into a retrieval process are introduced in these papers, and it has proven to be useful in discriminating between the relevant and nonrelevant documents. Intuitively, if the proximity among the query terms and informative terms is small, it is more likely that the candidate document is of high quality. In particular, we use two kinds of proximity scores to capture this characteristic.

- Proximity among query terms: We use a recently proposed concept, proximity bigram-term frequency (ptf) (Miao et al., 2012), to measure the proximity of query terms. In particular, a pair of two query terms are viewed as a bigram term, and its frequency is reinterpreted by taking into account the position of the two query terms as follows:

$$K(q_i, q_j) = exp[-(p_i - p_j)^2] \tag{7}$$

  where p_i and p_j are respectively the positions of query term q_i and q_j in a document.
  Besides the average proximity to the query, we also take into account the importance of different query terms. Therefore, the total proximity-based frequency of the query Q is computed as follows:

$$ptf(Q) = \sum_{i,j,i<j} K(q_i, q_j) IDF(q_i, q_j) \tag{8}$$

  where q_i is a query term, $|Q|$ is the number of query terms, and $IDF(q_i, q_j)$ equals $log(N - N_t + 0.5)/(N_t + 0.5)$. N is the number of documents in the collection and N_t is the number of documents that contain both q_i and q_j. This feature is denoted `Feature 2`: $f2$.
- Proximity between the query terms and informative terms: If the most informative terms in a feedback document appear relatively close to the query, it is also more likely the document is of higher quality. Here, we empirically chose the top 35 informative terms. The informative scores are computed by $log(N - N_t + 0.5)/(N_t + 0.5)$. This feature is denoted `Feature 3`: $f3$.

Completeness of query aspects. The completeness of query aspects actually contains three features that we propose to indicate whether a document covers the complete information of a query or not. These features, `Feature 4, 5, 6`: $f4, f5, f6$, are shown as follows:

- Percentage of occurred query terms: the ratio of unique query terms appears in the document: $f4 = \sum_{i=1}^{|Q|} \frac{occur(q_i)}{|Q|}$, where $occur(q_i)$ is a 0–1 function testing whether q_i occurs in this document.
- Weighted percentage of occurred query terms: In this feature, we also take into account the importance of each query terms as follows: $f5 = \sum_{i=1}^{|Q|} \frac{occur(q_i) * IDF(q_i)}{|Q|}$.
- Mitra's Score: In Mitra, Singhal, and Buckley (1998), they proposed different ways to refine the set of documents used in feedback. One of the automatic ways is using term correlation information. The basic idea is that documents that address all the aspects of the query are more likely to be relevant. The equation used to compute a new score for each retrieved document is as follows:

$$f6 = idf(t_1) + \sum_{i=2}^{m} idf(t_i) * \min_{j=1}^{i-1}(1 - P(t_i|t_j)$$

  where $idf(t_j)$ is the inverse document frequency of term t_i if it occurs in document D, and is 0 otherwise. $P(t_i|t_j)$ is estimated based on word occurrences in the top-ranked 100 documents in the first-pass retrieval.

Divergence between query and feedback document. The motivation for using the divergence between query Q and a feedback document d is that we may rely on feedback information more if the query does not represent relevant information well. We estimate the divergence, `Feature 7`: $f7$, as:

$$f7 = P(t|Q) \log_2 \frac{P(t|Q)}{P(t|d)} \tag{9}$$

Discrimination of feedback documents. "Query clarity" is an effective measure to predict query difficulty (Cronen-Townsend, Zhou, & Croft, 2002). Therefore, we expect it to also be useful for estimating the document quality in a similar way. In the definition, the clarity of a query is the Kullback-Leibler divergence of the query model from the collection model. We define the discrimination of feedback documents as the KLdivergence between document model and background models, which can be obtained by Equation 3. To compute the discrimination, we empirically use the top 35 terms and all the terms in the document, which results in `Feature 8`: $f8$ and `Feature 9`: $f9$.

Document length feature. Document length has been recognized as an important factor for adjusting an information retrieval system. For a query, the document length factor can impact relevance based on the scope assumption, that is, some documents may contain more material than others if longer documents are more likely to be retrieved (Huang, Peng, Schuurmans, Cercone, & Robertson, 2003; Huang, Robertson, Cercone, & An, 2000; Zhou, Huang, & He, 2011). Thus, we incorporate it into the regression model as one feature. We do not use the document length directly. We obtain the document length feature *length*, `Feature 10`: $f10$, as follows:

DOI: 10.1002/asi

$$f10 = |dl - avgl| \tag{10}$$

where *dl* is the document length and *avgl* is the average document length of the collection.

Normally, the average document length is used in a weighting function as a pivot to balance its preference of document with different lengths. So, in our case, we use this feature in Formula 10 to capture this possible preference in the procedure of PRF.

Experimental Settings

Test Data Sets

We present five representative test collections used in our experiments, disk4&5, WT2G, WT10G, GOV2, and Robust04, which are different in size and genre. The disk4&5 (no CR) collection contains newswire articles from various sources, such as the Associated Press (AP), Wall Street Journal (WSJ), Financial Times (FT), and so on, which are usually considered high-quality text data with little noise. The WT2G collection is a small web crawl used by the TREC 8 Web track in 1999. The WT10G collection is a medium-size crawl of web documents, which was used in the TREC 9 and 10 Web tracks. It contains 10 gigabytes of uncompressed data. GOV2 is a large crawl of the .gov domains, which has more than 25 million documents with an uncompressed size of 423 gigabytes. This collection was also used in the TREC 2008 Relevance Feedback track that is dedicated to research in relevance feedback algorithms, including pseudo-relevance feedback (Buckley & Robertson, 2008). There are 150 ad hoc query topics, from TREC 2004–2006 Terabyte tracks, associated with GOV2. Robust04 also uses the disk4&5 collection, but includes more topics 301–450 and 601–700. The TREC tasks and topic numbers associated with each collection are presented in Table 1. As we can see from this table, we evaluate the proposed methods with a relatively large number of queries.

In all our experiments, we only use the *title field* of the TREC queries for retrieval. In the process of indexing and querying, each term is stemmed using Porter's English stemmer, and stopwords from InQuery's standard stoplist with 418 stopwords are removed. The MAP (mean average precision) performance measure for the top 1,000 documents is used as an evaluation metric, which is the official measure in the corresponding TREC evaluations. MAP measure reflects the overall accuracy and the detailed descriptions for MAP can be found in Voorhees and Harman (2000).

Baseline Models

To verify the performance of our proposed models, we first compare our model with two representative PRF models, the $Rocchio_{KL}$ model (Rocchio, 1971a) and the RM3 model (Lavrenko & Croft, 2001), in the probabilistic framework and language modeling framework (Lv & Zhai, 2009a), respectively. Both of these PRF models use the top-ranked document for feedback.

Before introducing the two PRF models, we describe the corresponding basic retrieval models for the first-pass retrieval. For the $Rocchio_{KL}$ model introduced in Problem Formulation (above), RelPRF and ImpPRF, BM25 (Robertson et al., 1994) is used as the weighting model in the first-pass retrieval, which makes the comparison fair.

BM25 is a well-known, state-of-the-art model (Huang, Robertson, Cercone, & An, 2000), which achieves good IR performance. In BM25, the weight of a search term is assigned based on its within-document term frequency and query term frequency (Robertson et al., 1996). It was first implemented in the Okapi system. The corresponding weighting function is as follows:

$$w = \frac{(k_1+1)*tf}{K+tf} * \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} * \frac{(k_3+1)*qtf}{k_3+qtf} \oplus k_2 * nq * \frac{(avdl-dl)}{(avdl+dl)} \tag{11}$$

where w is the weight of a query term, N is the number of indexed documents in the data set, n is the number of documents containing a specific term, R is the number of documents known to be relevant to a specific topic, r is the number of relevant documents containing the term, *tf* is within-document term frequency, *qtf* is within-query term frequency, *dl* is the length of the document, *avdl* is the average document length, *nq* is the number of query terms, the k_is are tuning constants, K equals $k_1 * ((1-b) + b * dl/avdl)$, and $\oplus$ indicates that its following component is added only once per document, rather than for each term. More details of $Rocchio_{KL}$ are introduced in Problem Formulation. In addition, we also use BM25 as the basic retrieval model in our proposed PRF models, namely, ImpPRF and RelPRF for fair comparison.

TABLE 1. Overview of the TREC collections used.

Collection	Task	Queries	Docs
disk4&5	TREC6, 7, 8	301–450	528,155
WT2G	TREC8 Web adhoc	401–450	247,491
WT10G	TREC9–10 Web adhoc	451–550	1,692,096
GOV2	TREC04–06 Terabyte Track	701–850	25,178,548
Robust04	Robust04	301–450, 601–700	528,155

DOI: 10.1002/asi

The relevance model (Lavrenko & Croft, 2001) is a representative state-of-the-art method for estimating query language models within language modeling framework (Lv & Zhai, 2009a). We use the language model with Dirichlet smoothing as its first-pass retrieval model. In particular, we use a Dirichlet prior (with a hyperparameter of μ) for smoothing the document language model as shown in Equation 12, which can achieve good performance generally Zhai and Lafferty (2004).

$$p(w|d) = \frac{c(w; d) + \mu p(w|C)}{|d| + \mu} \tag{12}$$

where $c(w; d)$ is the frequency of query term w in document d, $p(w|C)$ is the probability of term w in collection language C, and $|d|$ is the length of document d.

Relevance models do not explicitly model the relevant or pseudo-relevant document. Instead, they model a more generalized notion of relevance R. The formula of RM1 is:

$$p(w|R) \propto \sum_{\theta_D} p(w|\theta_D) p(\theta_D) P(Q|\theta_D) \tag{13}$$

The relevance model $p(w|R)$ is often used to estimate the feedback model θ_F, and then interpolated with the original query model θ_Q to improve its estimation as follows:

$$\theta_{Q'} = (1 - \alpha) * \theta_Q + \alpha * \theta_F \tag{14}$$

The interpolated version of the relevance model is called RM3. For the smoothing of the basic language model, we also use a Dirichlet prior (with a hyperparameter of μ) for smoothing the document language model as shown in Equation 12.

Parameter Training

As we can see, there are several tuning parameters in the basic retrieval models (BM25, LM), baseline PRF models ($Rocchio_{KL}$ and RM3), and our proposed models. It is important to build strong baselines and make fair comparisons. We use the training method in Diaz and Metzler (2006) for both the baselines and our proposed approach. First, for b in BM25 and the smoothing parameter μ in LM, we sweep over in (0.1, 0.2, . . . , 0.9) and (500, 550, . . . , 1,500). Second, for the linear combination parameter β in $Rocchio_{KL}$, and the interpolation parameter α of RM3, we sweep over values in the range of 0.0, 0.1, . . . , 1.0. To evaluate the baseline and our proposed approach, we use 2-fold cross-validation. Two-fold cross-validation partitions the data set into two parts, one (training set) for training the model parameters and the other (test set) for testing the performance of the model learned from the training part. In our experiments, the TREC queries are partitioned by the parity of queries number on each collection. Then the parameters learned on the training set are applied to the test set for evaluation purposes. The quality models introduced in A Learning-to-Rank to Quality Estimation are also trained in this way.

Experimental Results

Comparison of Basic Retrieval Models

As we mentioned in the previous section, the results of both models are obtained by 2-fold cross-validation. Therefore, it is fair to compare them on these five collections. BM25 slightly outperforms LM with the Dirichlet prior on WT2G, and the results of these two models are almost the same on the other three collections. This comparison indicates that the classic BM25 model is generally comparable to LM, and it is reasonable to use them as the basic models of the PRF baselines and our proposed models (Table 2).

Comparison of PRF Models

From Tables 3–7, we present the results of the baseline PRF models and our proposed PRF models with different settings of feedback documents. We denote our PRF model trained by using the relevance-based quality generation strategy as **RelPRF**, the improvement-based quality strategy as **ImpPRF**. The last row in each of these tables is the average performance of each PRF model with different

TABLE 2. Performance of basic retrieval models in terms of MAP.

Basic models	disk4&5	WT2G	WT10G	GOV2	Robust04
BM25	0.2251	0.3132	0.2068	0.2994	0.2492
LM	0.2274	0.3002	0.2056	0.3040	0.2511

TABLE 3. Comparison of the performance of PRF methods in terms of MAP on the disk4&5 collection.

$\|D_f\|$	$Rocchio_{KL}$	RM3	RelPRF	ImpPRF
3	**0.2523**	0.2522	0.2491 (−1.27%, −1.23%)	0.2477 (−1.82%, −1.78%)
5	**0.2576**	0.2538	0.2532 (−1.71%, −0.24%)	0.2523 (−2.06%, −0.59%)
10	**0.2645**	0.2527	0.2628 (−0.64%, 4.00%)	0.2588 (−2.16%, 2.41%)
15	0.2660	0.2518	**0.2683** (0.86%, 6.55%)	0.2619 (−1.54%, 4.01%)
20	0.2677	0.2504	**0.2713** (1.34%, 8.35%)	0.2641 (−1.34%, 5.47%)
30	0.2567	0.2493	**0.2736** (6.58%, 9.75%)	0.2691 (4.83%, 7.94%)
50	0.2521	0.2465	**0.2718** (7.81%, 10.26%)	0.2691 (6.74%, 9.17%)
Ave	0.2596	0.2510	**0.2643**$^{*+}$ (1.81%, 5.30%)	0.2604^{+} (0.3%, 3.75%)

The values in parentheses are the improvements over $Rocchio_{KL}$ and RM3. Ave means the average performance of each PRF model with different $|D_f|$.

DOI: 10.1002/asi

TABLE 4. Comparison of the performance of PRF methods in terms of MAP on the WT2G collection.

$\|D_f\|$	$Rocchio_{KL}$	RM3	RelPRF	ImpPRF
3	**0.3402**	0.3270	0.3323	0.3282
			(–2.32%, 1.62%)	(–3.53%, 0.37%)
5	**0.3389**	0.3243	0.3312	0.3256
			(–2.27%, 2.13%)	(–3.92%, 0.40%)
10	**0.3402**	0.3268	0.3353	0.3335
			(–1.44%, 2.60%)	(–1.97%, 2.05%)
15	0.3393	0.3257	0.3397	**0.3439**
			(0.12%, 4.30%)	(1.36%, 5.59%)
20	0.3390	0.3260	0.3441	**0.3497**
			(1.50%, 5.55%)	(3.16%, 7.27%)
30	0.3293	0.3242	0.3450	**0.3478**
			(4.77%, 6.42%)	(5.62%, 7.28%)
50	0.3058	0.3233	0.3444	**0.3437**
			(12.62%, 6.53%)	(12.39%, 6.31%)
Ave	0.3332	0.3253	**0.3389**$^{*+}$	**0.3389**$^{*+}$
			(1.71%, 4.18%)	(1.71%, 4.18%)

The values in parentheses are the improvements over $Rocchio_{KL}$ and RM3. Ave means the average performance of each PRF model with different $|D_f|$.

TABLE 5. Comparison of the performance of PRF methods in terms of MAP on the WT10G collection.

$\|D_f\|$	$Rocchio_{KL}$	RM3	RelPRF	ImpPRF
3	**0.2335**	0.2311	0.2284	0.2312
			(–2.18%, –1.17%)	(–0.99%, 0.04%)
5	0.2308	0.2271	0.2301	**0.2398**
			(–0.30%, 1.32%)	(3.90%, 5.59%)
10	0.2153	0.2203	0.2318	**0.2396**
			(7.66%, 5.22%)	(11.29%, 8.76%)
15	0.2119	0.2206	0.2338	**0.2368**
			(10.34%, 5.98%)	(11.75%, 7.34%)
20	0.2175	0.2197	**0.2359**	0.2357
			(8.46%, 7.37%)	(8.37%, 7.28%)
30	0.2105	0.2188	**0.2382**	0.2344
			(13.16%, 8.87%)	(11.35%, 7.13%)
50	0.2037	0.2182	**0.2350**	0.2349
			(15.37%, 7.70%)	(15.32%, 7.65%)
Ave	0.2176	0.2223	0.2333^{*+}	**0.2360**$^{*+}$
			(7.21%, 4.95%)	(8.46%, 6.16%)

The values in the parentheses are the improvements over $Rocchio_{KL}$ and RM3. Ave means the average performance of each PRF model with different $|D_f|$.

TABLE 6. Comparison of the performance of PRF methods in terms of MAP on the GOV2 collection.

$\|D_f\|$	$Rocchio_{KL}$	RM3	RelPRF	ImpPRF
3	0.3220	0.3257	**0.3367**	0.3355
			(4.57%, 3.38%)	(4.19%, 3.01%)
5	0.3276	0.3255	0.3419	**0.3428**
			(4.37%, 5.04%)	(4.64%, 5.31%)
10	0.3274	0.3241	**0.3528**	0.3527
			(7.76%, 8.86%)	(7.73%, 8.82%)
15	0.3288	0.3245	0.3544	**0.3545**
			(7.79%, 9.21%)	(7.82%, 9.24%)
20	0.3283	0.3243	0.3554	**0.3554**
			(8.25%, 9.59%)	(8.25%, 9.59%)
30	0.3269	0.3242	0.3552	**0.3561**
			(8.66%, 9.56%)	(8.93%, 9.84%)
50	0.3213	0.3221	0.3549	**0.3561**
			(10.46%, 10.18%)	(10.83%, 10.56%)
Ave	0.3260	0.3243	0.3502^{*+}	**0.3504**$^{*+}$
			(7.42%, 7.99%)	(7.48%, 8.05%)

The values in parentheses are the improvements over $Rocchio_{KL}$ and RM3. Ave means the average performance of each PRF model with different $|D_f|$.

TABLE 7. Comparison of the performance of PRF methods in terms of MAP on the Robust04 collection.

$\|D_f\|$	$Rocchio_{KL}$	RM3	RelPRF	ImpPRF
3	0.2824	**0.2877**	0.2798	0.2766
			(–0.92%, –2.75%)	(–2.05%, –3.86%)
5	**0.2887**	0.2870	0.2880	0.2841
			(–0.24%, 0.35%)	(–1.59%, –1.01%)
10	0.2960	0.2873	**0.2974**	0.2953
			(0.47%, 3.52%)	(–0.24%, 2.78%)
15	0.2990	0.2862	**0.3028**	0.3019
			(1.27%, 5.80%)	(0.97%, 5.49%)
20	0.2980	0.2863	**0.3057**	0.3055
			(2.58%, 6.78%)	(2.52%, 6.71%)
30	0.2986	0.2842	**0.3082**	0.3079
			(3.22%, 8.44%)	(3.11%, 8.34%)
50	0.2953	0.2822	0.3074	**0.3086**
			(4.10%, 8.93%)	(4.50%, 9.36%)
Ave	0.2940	0.2858	**0.2985**$^{+}$	0.2971^{+}
			(1.52%, 4.42%)	(1.06%, 3.95%)

The values in parentheses are the improvements over $Rocchio_{KL}$ and RM3. Ave means the average performance of each PRF model with different $|D_f|$.

settings. We calculate the average MAP scores of each query with a different number of D_f, and then conduct a significance test. In particular, a "*" and a "+" indicate a statistically significant improvement over $Rocchio_{KL}$ and RM3, respectively, according to the Wilcoxon matched-pairs signed-ranks test at the 0.05 level. The bold phase style in a row means that it is the best result. As we mentioned in Baseline Models, the basic retrieval models of ImpPRF, RelPRF, and $Rocchio_{KL}$ are all using BM25 for fair comparison.

First, the $Rocchio_{KL}$ model outperforms the RM3 model on the disk4&5, Robust04, WT2G, and GOV2 collections, but is defeated by the latter on the WT10G collection in terms of average MAP. Both of them have proven effective and have been considered as strong baselines in previous studies. In terms of average MAP performance, our proposed *RelPRF* and *ImpPRF* models are generally better than the two baseline PRF models on all five collections and achieve significantly better results in most cases, especially on larger collections (WT10G, GOV2). The maximum

DOI: 10.1002/asi

improvement is as high as 8.46%. Thus, it is fair to conclude that our proposed models can outperform $Rocchio_{KL}$ and RM3 generally.

Second, the *RelPRF* and *ImpPRF* models perform better than the two basic models in all cases, which confirms their robustness, whereas in some cases $Rocchio_{KL}$ and RM3 decrease the performance when compared to the basic models, for example, $Rocchio_{KL}$ on the WT10G collection with 50 feedback documents. This indicates the robustness of our *RelPRF* and *ImpPRF* models. In addition, with the increase of feedback document size $|D_f|$, the performance of the *RelPRF* and *ImpPRF* models is much stabler than the $Rocchio_{KL}$ model and the RM3 model. When $|D_f|$ is 50, both the baseline models obtain the worst results, whereas the *RelPRF* and *ImpPRF* can still achieve good performance on all five collections. Besides, the optimal $|D_f|$ for the *RelPRF* and *ImpPRF* models is always larger than that for the baseline models. To some extent, this phenomenon proves the effectiveness of the QL factor. Usually, when $|D_f|$ is small the top-k documents are more likely to be of the highest QL, and it is reasonable to treat them equally and obtain good performance. However, when $|D_f|$ increases, the ratio of high QL documents will decrease. Because we have already taken this factor into account, the weights of feedback documents will be adjusted so that we can still obtain satisfying results. The baseline models that still treat the feedback documents equally may fail in this case.

We can also find that the best performance of the *ImpPRF* model is better than the *RelPRF* models on all collections except disk4&5. These experimental results prove that it is more reasonable to consider the improvement gain directly than the relevance level when applying QL, especially on medium or large collections. A possible reason is that the *ImpPRF* model also brings some positive information from the irrelevant documents, as we mentioned in the previous section, and most of the feedback documents are actually irrelevant according to previous studies.

To summarize, the PRF models all generally outperform the basic models (BM25 and LM). Meanwhile, the performance of the baseline PRF models, namely, the $Rocchio_{KL}$ model and the RM3 model, is generally comparable on all five collections. Moreover, our proposed models, *RelPRF* and *ImpPRF*, make significant improvements over the baseline models, and extensive experiments have shown the robustness of them especially when a relatively larger value of $|D_f|$ is used. Finally, the *ImpPRF* model can obtain better results than the *RelPRF* model in most cases, which indicates our improvement-based QL assumption is more reasonable than the relevance-based QL assumption. This can be a heuristic reference for further research.

Robustness Analysis

As we can see from the previous experiments, the number of feedback documents $|D_f|$ can greatly impact the performance of the PRF models, and the choice of $|D_f|$ turns out to be a challenge problem because it is difficult to determine the optimal number of feedback documents. In this section, we further analyze the robustness of our proposed PRF models with respect to D_f. In the experiments, other parameters are optimized by using 2-fold cross validation explained in Parameter Training. In particular, 2-fold cross-validation partitions the data set into two parts, one (training set) for training the model parameters and the other (test set) for testing the performance of the model learned from the training part.

From Figure 2, one can see the robustness of each method. Generally, the performance of all methods increases at the beginning when the number of feedback documents $|D_f|$ grows up. However, there is no unique optimal value of $|D_f|$ for all of them. The performance of each method starts to continuously drop after a peak. For example, RelPRF obtains the best value when $|D_f|$ is 30, whereas RM3 performs the best when $|D_f|$ is 5 on the disk4&5 collection. Meanwhile, the best performance of RelPRF is much better than that of RM3 on this collection. This indicates that the feedback documents with lower ranks can also help.

In Figure 2, the best values of $|D_f|$ of RelPRF and ImpPRF are larger than those for $Rocchio_{KL}$ and RM3 in most cases. This indicates that our proposed methods can make better use of feedback documents with lower ranks. Furthermore, after the peak point, the curves of RelPRF and ImpPRF fall down much smoother than those of the baselines. When $|D_f|$ is 50, which means 50 feedback documents are selected, the performance of our proposed methods are much better than $Rocchio_{KL}$ and RM3 on all five collections. Thus, it is clear that our RelPRF and ImpPRF methods perform more robustly on different $|D_f|$ values. We also observe that our proposed methods can obtain the best results on all five collections that are of different sizes and quality. This is solid evidence that the RelPRF and ImpPRF methods can make better use of the feedback documents to improve the overall performance and consistently get good results. The QL helps make PRF perform better.

In addition, it is interesting to note that the performance of both ImpPRF and RelPRF first increases with the increase of D_f (this is especially obvious on the disk4&5, WT2G, and GOV2 collections), and then after the peak point it stays relatively stable on all collections. The reasons are two-fold. First, the increase in the first phase is due to the utilization of more useful feedback documents, although the optimal values of D_f are different on different collections. In other words, the performance of PRF models can be increased by using more useful feedback documents. Second, because ImpPRF and RelPRF have explicitly estimated the qualities of candidate feedback documents, it is possible to reduce the negative impact of feedback documents with lower qualities. This leads to the stability of our proposed models after the peak points, so it is always safe to choose a relatively larger value of D_f. This feature of our proposed models makes it

DOI: 10.1002/asi

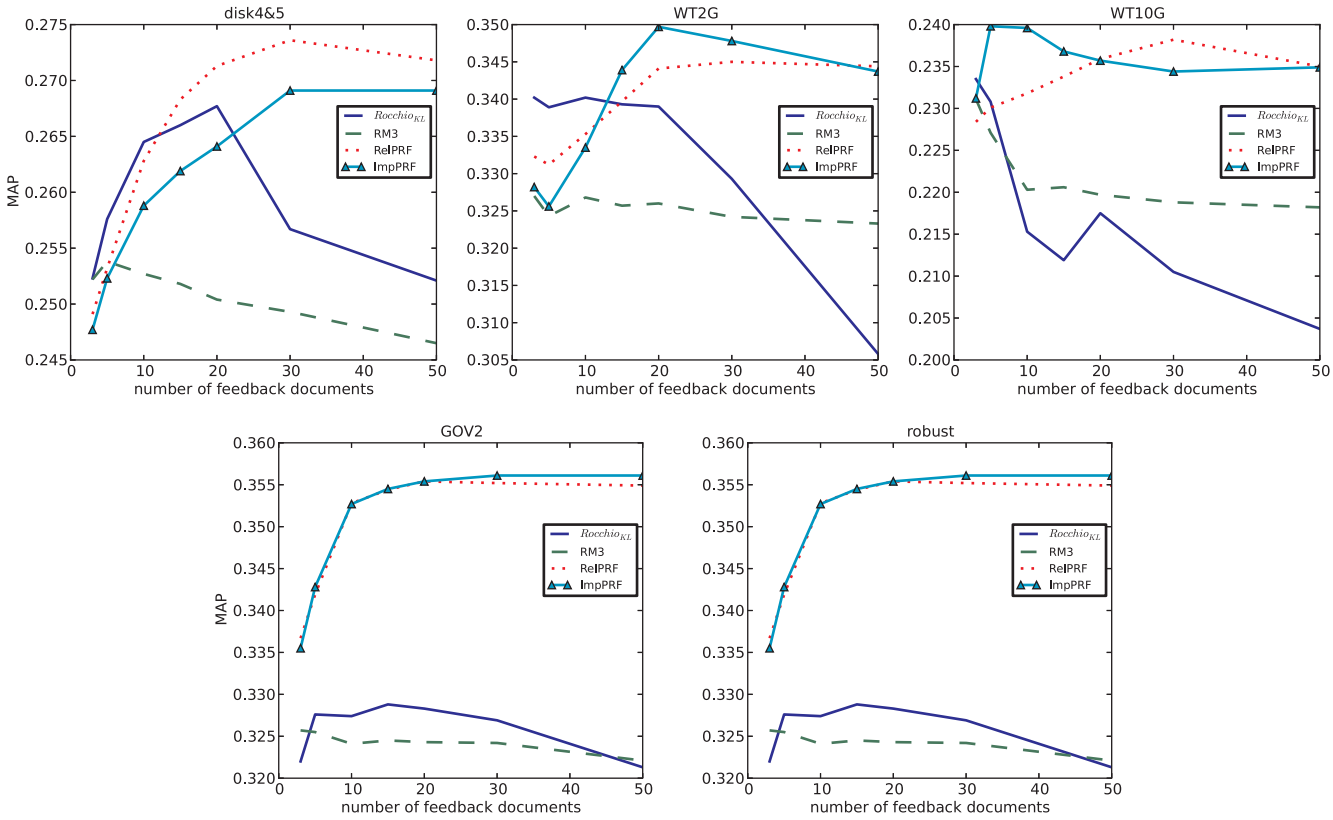

FIG. 2. Robustness comparison. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

viable to address the problem of the selection of the number of feedback documents.

In summary, the proposed models, ImpPRF and RelPRF, can utilize a sufficient number of useful feedback documents, and can also reduce the negative impact of feedback documents with lower qualities. Generally, ImpPRF and RelPRF reach their best performance when D_f is in the range of [20, 30]. These values can be used as the empirical optimal when no training data are available, although larger values of $|D_f|$ do not necessarily harm the retrieval performance in terms of MAP.

Influence of Control Parameter β

Recall that we incorporate the feedback information to derive a better representation of the query, as shown in Equation 2. How much we rely on the feedback information is controlled by the parameter β. In our preliminary experiments, we found that the control parameter β in Equation 2 plays an important role in obtaining good performance. In this section, we empirically study the influence of this parameter on all the test collections, and suggest some empirical settings when training data are not available. In particular, Figure 3 depicts its influence over different numbers of feedback documents. In this set of experiments, we use 35 terms to expand the original query since our model achieves very good performance under this setting generally. When $\beta = 0$, it means we do not use any feedback information. In other words, the MAP score at $\beta = 0$ is actually the baseline of BM25 without query expansion. It is of note that when β takes a large value, it approaches the performance resulting from using only the feedback information.

As we can see from Figure 3, in general the IR performance can always be boosted when the feedback documents are used to expand the original query. Although the setting of β can affect the retrieval performance significantly, it is always safe to set β to a value around 0.8 on all our test collections. Similar results can be observed over other numbers of feedback documents, which are not presented in this paper.

Discussion of Feature Importance

We use a popular tree-based ensemble model (Geurts, Ernst, & Wehenkel, 2006) to evaluate the importance of the features used, which combines the predictions of many tree models (200 in our paper) to improve robustness over a single model. According to Deng and Runger (2012), tree ensembles, consisting of multiple trees, are believed to be significantly more accurate than a single tree in feature selection. The quality of the selected features may be limited because the accuracy of a single tree model may be limited.

DOI: 10.1002/asi

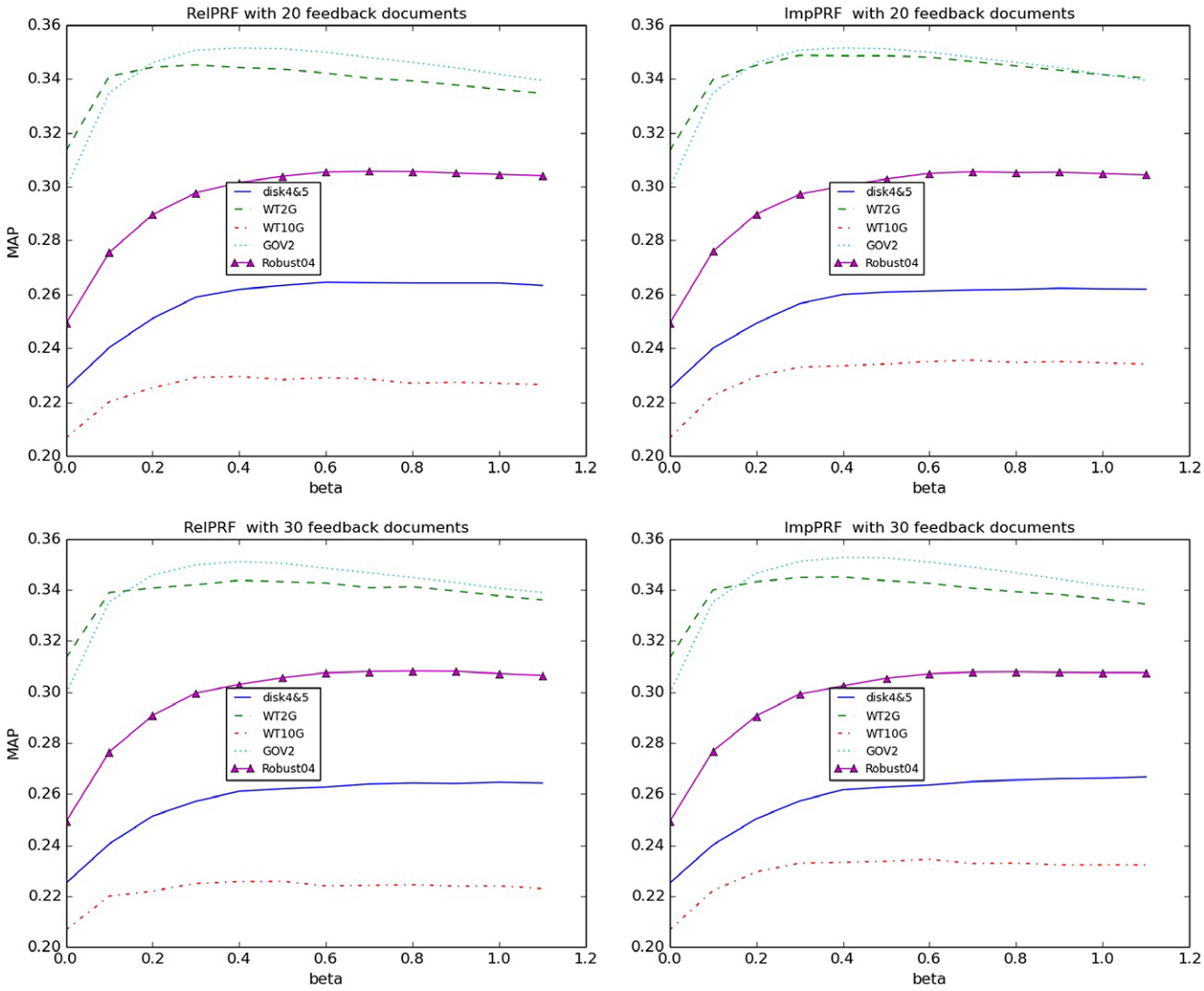

FIG. 3. Influence of the control parameter β for the ImpPRF model. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

In Figure 4, we can see clearly that seven of the ten features have more weights than the other features. The three features that are not so important as the others are percentage of occurred query terms, weighted percentage of occurred query terms, and Mitra's score. These three features belong to the completeness of query aspects. This is reasonable, since usually the top-k feedback documents contain all the query terms, and the weights of these terms and Mitra's scores of these documents are relatively close. Thus, these features will not make significant differences in the feedback documents.

The weights of the other seven features are generally close on the five collections, and they do not vary much in all cases. On one hand, this indicates that all these features can influence the QLs of documents, so it is necessary to consider all of them when evaluating feedback documents. On the other hand, it is difficult to find a solid rule to explain and compare the impacts of these features according to our experimental results. As we mentioned previously, heuristic methods cannot handle these heterogeneous features well. So it is better to use machine-learning methods, for example, learning-to-rank in this paper.

Features 2–6, which are proximity of query terms, proximity of query terms, and informative terms, percentage of occurred query terms, weighted percentage of occurred query terms, and Mitra' score, respectively, are first explored in this paper. Features 2 and 3 are proved to be effective when compared to other features. Thus, proximity information is also an important factor to estimate the QLs of feedback documents.

In general, most features applied are related to the QLs of documents. Meanwhile, it is better to consider machine-learning ways to utilize these heterogeneous features since it is hard to find explicit rules for their usage. The

DOI: 10.1002/asi

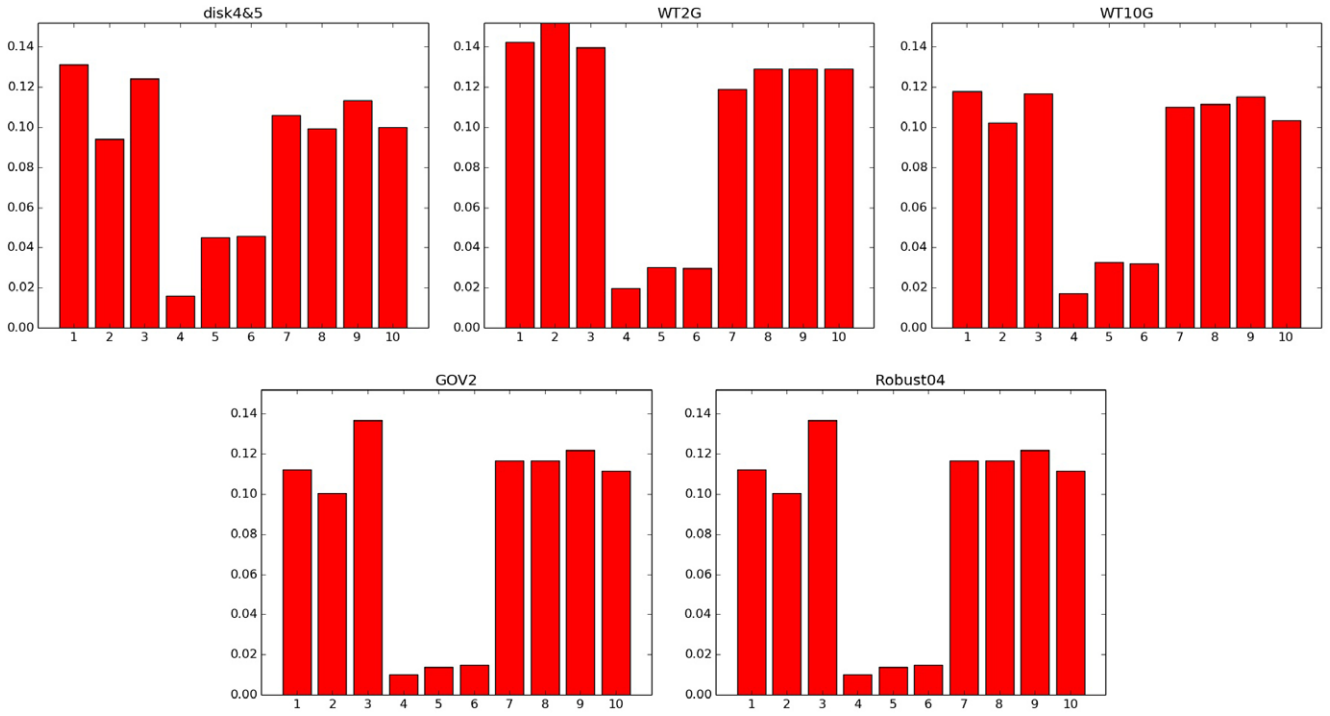

FIG. 4. Feature importance on five test collections. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

estimation of QLs can be complex, so we still need more investigations for further research.

Comparison With Filtering-Based Method

We also compared our proposed model with a filter-based PRF method, in which classification methods are used to refine the PRF at the document level. To the best of our knowledge, this is the closest research to our work. In particular, naive Bayes and logistic regression are used to filter out bad feedback documents. The corresponding models are respectively denoted NaiveFilter and LogisticFilter in this paper. As we mentioned previously, our method does not make any relevance assumption, and estimates a quality score for each candidate document rather than filtering out any of them. To make the comparison fair, we train our model on the 75 odd-numbered topics on the GOV2 collection, and use 50 of the 75 even-numbered topics for testing, which is the same setting as in He and Ounis (2009).

According to the results in Table 8, the NaiveFilter and LogisticFilter PRF approaches show a high sensitivity to $|D_f|$. On one hand, the retrieval performance of our RelPRF and ImpPRF approaches remain stable on GOV2. This indicates that our proposed model is indeed able to accurately estimate the document quality. In addition, on average ImpPRF outperforms the filtering-based approach by as much as 6.14%, although the best performance is achieved by LogisticFilter when $|D_f|$ equals 50. Overall, our feedback approach has been shown to be robust and effective with a varying size of the pseudo-feedback set.

TABLE 8. Comparison with NaiveFilter and LogisticFilter on GOV2.

$\|D_f\|$	NaiveFilter	LogisticFilter	RelPRF	ImpPRF
10	0.3403	0.3429	0.3546	**0.3560**
20	0.3152	0.3213	0.3541	**0.3561**
30	0.3124	0.3021	0.3550	**0.3569**
50	0.3636	**0.3698**	0.3547	0.3563
80	0.3462	0.3485	0.3535	**0.3555**
Ave	0.3355	0.3369	0.3543	**0.3561**
			(5.60%, 5.16%)	(6.14%, 5.70%)

Bold font indicates best result.

This is an encouraging finding, in that the size of the pseudo-relevant set is an important parameter of PRF, which has a direct impact on PRF's retrieval performance (Carpineto, Romano, & Giannini, 2002). On the other hand, our proposed framework is able to achieve effective and robust retrieval performance regardless of the optimal pseudo-relevant set size.

Comparison With the Best TREC Results

The Text REtrieval Conference (TREC)[1] holds a series of workshops focusing on different IR research tasks, or tracks. The TREC Conference series is cosponsored by the National Institute of Standards and Technology (NIST) Information Technology Laboratory's (ITL) Retrieval Group of the

[1]http://trec.nist.gov/

DOI: 10.1002/asi

TABLE 9. Performance of ImpPRF, RelPRF, and the best TREC results in terms of MAP.

Methods	disk4&5	WT2G	WT10G	GOV2	Robust04
BestTREC	0.2851	0.3329	0.2155	0.3614	0.3331
RelPRF	0.2736	0.3444	0.2382	0.3554	0.3082
ImpPRF	0.2691	0.3497	0.2396	0.3561	0.3086

Information Access Division (IAD). It also provides corresponding test collections for the evaluation of different IR tasks, and participants will develop different methods and submit their results to the TREC organizers. In this subsection, we present the performance of our proposed models and the best results for each collection used in our experiments in Table 9, which can be found at http://www.evaluatIR.org/ introduced in Armstrong, Moffat, Webber, and Zobel (2009). The results of RelPRF and ImpPRF are obtained with the settings $|D_f| = 30$ as suggested in Robustness Analysis and other parameters are set as described in Baseline Models.

In the first row of Table 9, we provide the best results from the corresponding TREC conferences for reference. RelPRF and ImpPRF outperform the best TREC systems on WT2G and WT10G, whereas they are not as good as the best TREC results on disk4&5 and GOV2. However, our results are close to the best ones. However, on the Robust04 collection, the best result is obviously better than ours. It is of note that it is difficult to make solid cross-comparisons with the results reported in the corresponding TREC conferences, because the results were generated by different teams, methods, heuristics, and settings. Some of the heuristics are collection-dependent. For example, they may use different stemmers, stoplists. Some of the participants may also use multiple IR techniques. In addition, sometimes it is also difficult to duplicate their experiments. Taking the best result in Robust04 from Queens College (Voorhees, 2005) as a running example, it combined four components to produce the result, including the basic retrieval model, 2-word phrase, PRF, and web-assistant technique (Kwok, Grunfeld, Sun, Deng, & Dinstl, 2004).

Our results are generated in a uniform way without collection-based parameter settings, and we understand that combining such techniques could further improve the results (Ye, Huang, & Miao, 2012). The main reason why we do not employ all of these is that we want to focus on evaluating the proposed method by comparison with similar approaches rather than developing the best-performing system.

We believe it is more reasonable to compare with results that are generated on the same basis and easy to duplicate. Therefore, we will not provide further analysis of the cross-comparison with the best TREC results, just this table to better understand our models.

In addition, in this revision we add a table of results with different evaluation metrics, as follows in Table 10 for reference to readers. We also provide the experiments for how many queries were hurt versus helped by each technique with improvement percentages in Figure 5, as that in Lang, Metzler, Wang, and Li (2010) and Terra and Warren (2005).

TABLE 10. Comparison of different PRF methods in terms of MAP, P@5, and P@20 when the number of feedback documents is 30.

$\|D_f\|$	$Rocchio_{KL}$	RM3	RelPRF	ImpPRF
DISK4&5				
MAP	0.2567	0.2493	**0.2736**	0.2691
P@5	0.4813	0.4747	0.4853	**0.4893**
P@20	0.3960	0.3820	0.4027	**0.4040**
WT2G				
MAP	0.3293	0.3242	0.3450	**0.3478**
P@5	0.4840	0.5080	**0.5520**	0.5320
P@20	0.4030	0.3900	0.4060	**0.4080**
WT10G				
MAP	0.2105	0.2188	**0.2382**	0.2344
P@5	0.3700	0.3940	**0.4080**	0.3980
P@20	0.2760	0.2985	**0.2940**	0.2930
GOV2				
MAP	0.3269	0.3242	0.3552	**0.3561**
P@5	0.6282	0.6416	**0.6644**	0.6631
P@20	0.5648	0.5735	**0.6215**	0.6211
Robust04				
MAP	0.2986	0.2842	**0.3082**	0.3079
P@5	0.5012	0.4803	0.5108	**0.5124**
P@20	0.3867	0.3701	0.3990	**0.4016**

Note. Bold fond indicates the best performance with different settings.

Conclusions and Future Work

In this paper we proposed a learning-to-rank approach for quality-aware pseudo-relevance feedback. Unlike previous work, we do not make explicit relevance assumptions and we go beyond selecting "good" documents for PRF. In addition, we introduce two quality-based assumptions and model the QL of feedback documents to obtain better retrieval performance. Specifically, two different strategies, relevance-based QL and improvement-based QL, are presented to estimate the QL for each feedback document. Based on this, we select several heterogeneous document features and apply a learning approach to estimate the QL of each feedback document, which results in two quality-based PRF models: ImpPRF and RelPRF.

A variety of document features, including the distribution of query terms in the feedback document, the similarity between a single feedback document and all top-ranked documents, the proximity between the expansion terms and the original query terms in the feedback document, are applied to facilitate the quality estimation of the feedback documents. Supported by extensive experimental results, our feedback framework provides impressive retrieval performance, compared to several strong PRF baselines that use all top-ranked documents for relevance feedback. We also study the robustness of the proposed models with respect to the number of feedback documents D_f. The experimental results indicate that ImpPRF and RelPRF can utilize a sufficient number of useful feedback documents, and can also reduce the negative impact of feedback documents with

DOI: 10.1002/asi

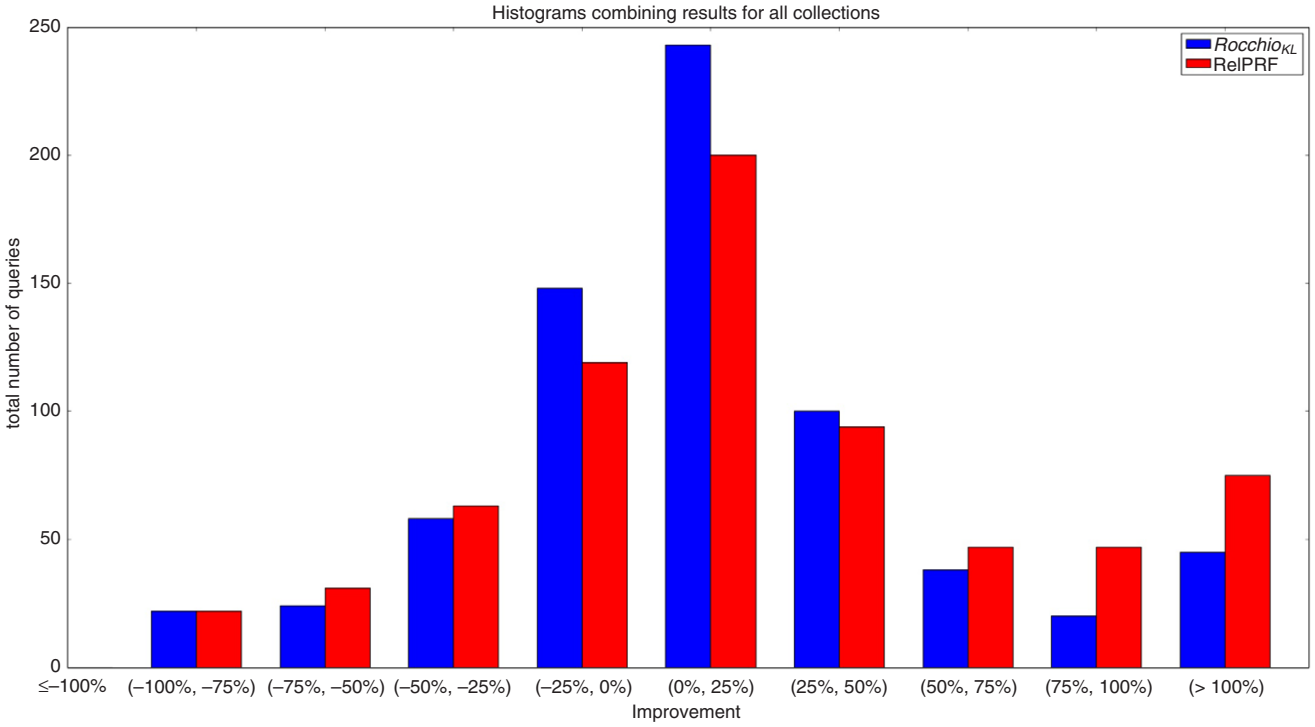

FIG. 5. Histograms combining the results for all collections. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

lower qualities. In addition, we investigate the influence of important parameters in our models, and suggest some empirical settings of these parameters when no training data are available.

In future work, there are several interesting further research directions. First, we plan to enhance the quality estimation by utilizing more training data from other collections. Second, we will study the correlation of different features for the quality estimation process. Third, we also plan to investigate the influence of different grades of QL. Finally, our model can be viewed as a way to properly weight the candidate feedback documents, so another possible investigation is to integrate our models with models dedicated to select good feedback terms. We believe the overall performance can be further improved in that way.

Acknowledgments

This research is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Early Researcher Award/Premiers Research Excellence Award, and the IBM Shared University Research (SUR) Award. We also would like to thank IBM Canada for providing IBM BladeCenter blade servers to conduct experiments reported in the paper. We thank the anonymous reviewers for their thorough comments on this paper.

References

Amati, G. (2003). Probabilistic models for information retrieval based on divergence from randomness. Ph.D. thesis, Department of Computing Science, University of Glasgow, Glasgow, Scotland.

Armstrong, T.G., Moffat, A., Webber, W., & Zobel, J. (2009). Improvements that don't add up: Ad-hoc retrieval results since 1998. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09 (pp. 601–610). New York: ACM.

Bendersky, M., Metzler, D., & Croft, W.B. (2010). Learning concept importance using a weighted dependence model. In Proceedings of WSDM '10 (pp. 31–40). New York: ACM.

Buckley, C., & Robertson, S.E. (2008). Relevance feedback track overview: TREC 2008. In Proceedings of TREC 2008 (pp. 1–3). Gaithersburg, MD: National Institute of Standards and Technology.

Büttcher, S., Clarke, C.L.A., & Lushman, B. (2006). Term proximity scoring for ad-hoc retrieval on very large text collections. In Proceedings of SIGIR '06 (pp. 621–622). New York: ACM.

Carpineto, C., de Mori, R., Romano, G., & Bigi, B. (2001). An information-theoretic approach to automatic query expansion. ACM Transactions on Information Systems, 19(1), 1–27.

Carpineto, C., Romano, G., & Giannini, V. (2002). Improving retrieval feedback with multiple term-ranking function combination. ACM Transactions on Information Systems, 20(3), 259–290.

Clarke, C.L., Cormack, G.V., & Tudhope, E.A. (2000). Relevance ranking for one to three term queries. Information Processing & Management, 36(2), 291–311.

Collins-Thompson, K. (2009). Reducing the risk of query expansion via robust constrained optimization. In Proceedings of CIKM '09 (pp. 837–846). New York: ACM.

Cronen-Townsend, S., Zhou, Y., & Croft, W.B. (2002). Predicting query performance. In Proceedings of SIGIR '02 (pp. 299–306). New York: ACM.

Daoud, M., & Huang, J.X. (2013). Modeling geographic, temporal, and proximity contexts for improving geotemporal search. Journal of the American Society for Information Science, 64(1), 190–212.

Deng, H., & Runger, G.C. (2012). Feature selection via regularized trees. In Proceedings of IJCNN 12 (pp. 1–8). Beijing, China: IEEE.

Diaz, F., & Metzler, D. (2006). Improving the estimation of relevance models using large external corpora. In Proceedings of SIGIR '06 (pp. 154–161). Beijing, China: IEEE.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3–42.

He, B., & Ounis, I. (2009). Finding good feedback documents. In Proceedings of CIKM '09 (pp. 2011–2014). Beijing, China: IEEE.

Huang, X., Robertson, S.E., Cercone, N., & An, A. (2000). Probability-based Chinese text processing and retrieval. Computational Intelligence, 16(4), 552–569.

Huang, X., Peng, F., Schuurmans, D., Cercone, N., & Robertson, S.E. (2003). Applying machine learning to text segmentation for information retrieval. Information Retrieval 6(3–4), 333–362.

Keen, E.M. (1991). The use of term position devices in ranked output experiments. The Journal of Documentation; Devoted to the Recording, Organization and Dissemination of Specialized Knowledge, 47(1), 1–22.

Krikon, E., & Kurland, O. (2011). Utilizing minimal relevance feedback for ad hoc retrieval. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11 (pp. 1099–1100). New York: ACM.

Kwok, K.-L., Grunfeld, L., Sun, H., Deng, P., & Dinstl, N. (2004). Trec 2004 robust track experiments using pircs. In TREC (pp. 1–7). Gaithersburg, MD: National Institute of Standards and Technology.

Lang, H., Metzler, D., Wang, B., & Li, J.-T. (2010). Improved latent concept expansion using hierarchical Markov random fields. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10 (pp. 249–258). New York: ACM.

Lavrenko, V., & Croft, W.B. (2001). Relevance based language models. In Proceedings of SIGIR '01 (pp. 120–127). New York: ACM.

Lee, K.S., Croft, W.B., & Allan, J. (2008). A cluster-based resampling method for pseudo-relevance feedback. In Proceedings of SIGIR '08 (pp. 235–242). New York: ACM.

Lin, C., Lin, C., Lin, Z., & Zou, Q. (2013). Hybrid pseudo-relevance feedback for microblog retrieval. Journal of Information Science, 39(6), 773–788.

Liu, T.-Y. (2002). Learning to rank for information retrieval. Foundations and Trends in Information Retrieval, 3(3), 225–331.

Lv, Y., & Zhai, C. (2009a). A comparative study of methods for estimating query language models with pseudo feedback. In Proceedings of CIKM '09 (pp. 1895–1898). New York: ACM.

Lv, Y., & Zhai, C. (2009b). Positional language models for information retrieval. In Proceedings of SIGIR '09 (pp. 299–306). New York: ACM.

Metzler, D., & Bruce Croft, W. (2007). Linear feature-based models for information retrieval. Information Retrieval, 10(3), 257–274.

Miao, J., Huang, J.X., & Ye, Z. (2012). Proximity-based rocchio's model for pseudo relevance. In Proceedings of SIGIR '12 (pp. 535–544). Beijing, China: IEEE.

Mitra, M., Singhal, A., & Buckley, C. (1998). Improving automatic query expansion. In Proceedings of SIGIR '98 (pp. 206–214). New York: ACM.

Ponte, J.M., & Croft, W.B. (1998). A language modeling approach to information retrieval. In Proceedings of SIGIR '98 (pp. 275–281). New York: ACM.

Raman, K., Udupa, R., Bhattacharyya, P., & Bhole, A. (2010). On improving pseudo-relevance feedback using pseudo-irrelevant documents. In Proceedings of ECIR '10 (pp. 573–576). Beijing, China: IEEE.

Rasolofo, Y., & Savoy, J. (2003). Term proximity scoring for keyword-based retrieval systems. In F. Sebastiani (Ed.), Advances in information retrieval of lecture notes in computer science (Vol. 2633, pp. 207–218). Berlin/Heidelberg, Germany: Springer.

Robertson, S., Walker, S., Beaulieu, M., Gatford, M., & Payne, A. (1996). Okapi at TREC-4. In Proceedings of TREC-4 (pp. 73–96). Gaithersburg, MD: National Institute of Standards and Technology.

Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., & Gatford, M. (1994). Okapi at TREC-3. In Proceedings of TREC-3 (pp. 109–126). Beijing, China: IEEE.

Rocchio, J. (1971a). Relevance feedback in information retrieval (pp. 313–323). Englewood Cliffs, NJ: Prentice-Hall.

Rocchio, J.J. (1971b). Relevance feedback in information retrieval. In G. Salton (Ed.), The SMART retrieval system: Experiments in automatic document (pp. 313–323). Englewood Cliffs, NJ: Prentice-Hall.

Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, 41(5), 288–297.

Salton, G., Wong, A., & Yang, C. (1975). A vector space model for information retrieval. Journal of American Society for Information Retrieval, 18(11), 613–620.

Scholer, F., Williams, H., & Turpin, A. (2004). Query association surrogates for web search. Journal of the American Society for Information Science and Technology, 55(7), 637–650.

Sharma, H., & Jansen, B.J. (2005). Automated evaluation of search engine performance via implicit user feedback. In R.A. Baeza-Yates, N. Ziviani, G. Marchionini, A. Moffat, & J. Tait (Eds.), SIGIR (pp. 649–650). Salvador, Brazil: ACM.

Song, R., Taylor, M., Wen, J.-R., Hon, H.-W., & Yu, Y. (2008). Viewing term proximity from a different perspective. In C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, & R.W. White (Eds.), Advances in Information retrieval of lecture notes in computer science (Vol. 4956, pp. 346–357). Berlin/Heidelberg, Germany: Springer.

Symonds, M., Zuccon, G., Koopman, B., Bruza, P.D., & Sitbon, L. (2013). Term associations in query expansion: A structural linguistic perspective. In ACM International Conference on Information and Knowledge Management (CIKM 2013) (pp. 1189–1192). New York: ACM.

Tao, T., & Zhai, C. (2006). Regularized estimation of mixture models for robust pseudo-relevance feedback. In Proceedings of SIGIR '06 (pp. 162–169). New York: ACM.

Tao, T., & Zhai, C. (2007). An exploration of proximity measures in information retrieval. In Proceedings of SIGIR '07 (pp. 295–302). New York: ACM.

Terra, E., & Warren, R. (2005). Poison pills: Harmful relevant documents in feedback. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM '05 (pp. 319–320). New York: ACM.

Voorhees, E.M. (2005). The trec robust retrieval track. SIGIR Forum, 39(1), 11–20.

Voorhees, E.M., & Harman, D. (2000). Overview of the sixth text retrieval conference. Information Processing & Management, 36(5), 3–35.

White, R.W., & Marchionini, G. (2007). Examining the effectiveness of real-time query expansion. Information Processing & Management, 43(3), 685–704.

Wu, Q., Burges, C.J., Svore, K.M., & Gao, J. (2010). Adapting boosting for information retrieval measures. Information Retrieval, 13(3), 254–270.

Xu, J., & Croft, W.B. (2000). Improving the effectiveness of information retrieval with local context analysis. ACM Transactions on Information Systems, 18(1), 79–112.

Xu, J., & Li, H. (2007). Adarank: A boosting algorithm for information retrieval. In Proceedings of SIGIR '07 (pp. 391–398). New York: ACM.

Ye, Z., Huang, X., He, B., & Lin, H. (2009). York University at TREC 2009: Relevance feedback track. In Proceedings of TREC 2009 (pp. 1–6). Gaithersburg, MD: National Institute of Standards and Technology.

Ye, Z., Huang, J.X., & Lin, H. (2011). Finding a good query-related topic for boosting pseudo-relevance feedback. Journal of the Association for Information Science and Technology, 62(4), 748–760.

Ye, Z., Huang, J.X., & Miao, J. (2012). A hybrid model for ad-hoc information retrieval. In Proceedings of SIGIR'12 (pp. 1025–1026). Beijing, China: IEEE.

Yue, Y., Finley, T., Radlinski, F., & Joachims, T. (2007). A support vector method for optimizing average precision. In Proceedings of SIGIR '07 (pp. 271–278). New York: ACM.

Zhai, C. (2008). Statistical language models for information retrieval a critical review. Foundations and Trends in Information Retrieval, 2(3), 137–213.

Zhai, C., & Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. In Proceedings of CIKM '01 (pp. 403–410). New York: ACM.

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems, 22(2), 179–214.

Zhou, X., Huang, J.X., & He, B. (2011). Enhancing ad-hoc relevance weighting using probability density estimation. In Proceedings of SIGIR '11 (pp. 175–184). Beijing, China: IEEE.

DOI: 10.1002/asi