A Simple Term Frequency Transformation Model for Effective Pseudo Relevance Feedback

Zheng Ye, Jimmy Xiangji Huang
Information Retrieval and Knowledge Management Research Lab
School of Information Technology
York University, Toronto, Canada
{yezheng, jhuang}@yorku.ca

ABSTRACT

Pseudo Relevance Feedback is an effective technique to improve the performance of ad-hoc information retrieval. Traditionally, the expansion terms are extracted either according to the term distributions in the feedback documents; or according to both the term distributions in the feedback documents and in the whole document collection. However, most of the existing models employ a single term frequency normalization mechanism or criteria that cannot take into account various aspects of a term's saliency in the feedback documents. In this paper, we propose a simple and heuristic, but effective model, in which three term frequency transformation techniques are integrated to capture the saliency of a candidate term associated with the original query terms in the feedback documents. Through evaluations and comparisons on six TREC collections, we show that our proposed model is effective and generally superior to the recent progress of relevance feedback models.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models, Relevance feedback

General Terms

Algorithms, Performance, Experimentation

Keywords

Term Frequency Transformation; Pseudo Relevance Feedback

1. INTRODUCTION AND MOTIVATION

Users often issue very short queries to describe their information need, which leads to the absence of some important terms from the queries. Thus, users could get a poor coverage of relevant documents. To solve this problem, pseudo relevance feedback (PRF) via query expansion (QE) is an effective technique for boosting the overall performance in Information Retrieval (IR). It assumes that top-ranked documents in the first-pass retrieval are relevant, and then used as feedback documents in order to refine the representation of original queries by adding potentially related terms or adjusting the weights of query terms. PRF has been shown to be effective in improving IR performance [6, 10, 14, 17, 29, 32, 34, 35, 42, 44, 46] in a number of IR tasks.

In general, the expansion terms are weighted and extracted either according to the term distributions in the feedback documents (i.e. one tries to extract the most frequent terms); or according to both the term distributions in the feedback documents and in the whole document collection (i.e. to extract the most specific terms in the feedback documents). Normally, the term frequency in a document determines its importance in that document, while inverse document frequency in the whole collections is used to estimate its importance globally. The term frequency is always normalized according to the length of the document that contains it. However, most of the existing models employ a single term frequency normalization mechanism or criteria that cannot take into account various aspects of a term's saliency in the feedback documents. When estimating the weight of a candidate expansion term, how the other terms are distributed are largely unexplored. First, for example, the original query itself is usually ignored in the process of expansion term selection. In other words, the term associations between candidate terms and the query terms have been ignored in most of traditional PRF models. Term proximity is an effective measure for term associations, which has been studied extensively in the past few years. Most of these studies focus on the term proximity within the original query and adapt this in ranking documents [5, 9, 15, 20, 30, 38, 40], which has proven to be useful in discriminating between the relevant and non-relevant documents. So it is promising to take into account the distribution of candidate expansion terms in combination with that of the original query terms. Second, although document length-based normalization can balance the weight of a term in different feedback documents well, the importance of the documents are not well utilized. Third, when one estimates the importance of a term in different documents, normal term frequency normalization methods only consider the frequency of itself and the document length. The distributions of other terms are always ignored, while we believe that it will affect the importance of the current term in a given document.

In this paper, we propose a uniform and heuristic model, in which three term frequency transformation techniques are used to capture the local saliency of a candidate term asso-

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

http://dx.doi.org/10.1145/2600428.2609636 .

ciated in the feedback documents. In particular, three kinds of term frequency transformation techniques are then integrated to capture the overall saliency. First, besides the traditional term frequency normalization to overcome the length difference of documents, we take into account the weights of feedback documents to get a weighted and normalized term frequency. Second, we propose to use a relative term frequency transformation to capture the relative importance of a term in a given document. Third, we use a kernel-base term frequency transformation to capture the closeness to the original query.

The main contributions can be summarized as follows: 1) expansion terms are no longer selected merely based on term distributions regardless of original query and other terms in the same feedback documents. We can expect the selected terms to be more closely related to the original query, and thus have a higher impact on the effectiveness; 2) three different term frequency transformation techniques are combined in a heuristic way. 3) our model is simple, yet effective since we only have to transform the term frequency based on information in the feedback documents.

We evaluate our model on six TREC collections and compare it to the traditional PRF models. The experimental results show that the retrieval effectiveness can be improved significantly and empirical parameter settings are suggested while no training data is available.

The remainder of this paper is organized as follows: in Section 2 we review the related work. In Section 3, three transformation methods for capturing different aspects of term frequency (TF) and our proposed model are presented in details. In Section 4, we introduce the settings of the experiments. In Section 5, the experimental results are presented and discussed. Finally, we conclude our work briefly and present future research directions in Section 6.

2. RELATED WORK

PRF via query expansion is referred to as the techniques or algorithms that reformulate the original query by adding new terms and adjust their weights, in order to obtain a better query. With the refined query, usually better retrieval performance can be expected. PRF has been shown to be effective with various retrieval models [6, 10, 14, 17, 29, 32, 34, 35, 42, 44, 46, 43, 25]. There are a large number of studies on the topic of PRF. Here we mainly review the work about PRF which is the most related to our research.

The Rocchio's model [34] is one of the earliest work of PRF models, which was developed in 1971 for the Smart retrieval system. It provides a framework for implementing (pseudo) relevance feedback via improving the query representation in vector space retrieval model. In the following decades, a number of PRF models were developed, mostly derived from Rocchio's framework. For example, Carpineto *et al.* proposed an information-theoretic approach to automatic query expansion evaluated under the vector space model. Another popular and successful automatic PRF model was proposed by Robertson *et al.* [32, 31]. Amati *et al.* [3] proposed a query expansion algorithm in the divergence from randomness (DFR) retrieval framework.

In addition, with the development of language model [27] in IR, a number of PRF (e.g. [17, 39, 46]) have been developed to fit in the language modeling framework. For example, the model based feedback approach [46] is not only theoretically sound, but also performs well empirically. The essence of model based feedback is to update the probability of a term in the query language model by making use of the feedback information. Much like model-based feedback, relevance models [17] also estimate an improved query model. The difference between the two approaches is that relevance models do not explicitly model the relevant or pseudo-relevant document. Instead, they model a more generalized notion of relevance [22]. Lv *et al.* [19] have conducted a comparable study of 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 language model), RM4, DMM, SMM (a variant of model-based feedback approach), and RMM [17, 39, 46]. They found that SMM and RM3 are the most effective in their experiments, and RM3 is more robust to the setting of feedback parameters.

Most of these PRF approaches estimate a value of the importance (or probability) of a candidate expansion term based on its own distribution or statistics (e.g. term frequency, collection term frequency and document frequency). However, when estimating the value of a term, how the original query terms and other terms in the same document are distributed was not considered together. Unlike previous work, we not only use the raw term frequency to capture the saliency of a term, but also take advantage of the distribution information of other terms. One of the information to utilize is the distributions of original query terms in combination with that of expansion terms. In particular, we model the closeness in terms of term proximity. In the following, we review related work of term proximity in IR.

Term proximity is the co-occurrences of terms within a specified distance, which could measure the closeness of terms. A large amount of work has been done to integrate term proximity into both probabilistic and language models, which are characterized by the distance of the original query terms in documents. For example, Allan and Ballesteros [1] proposed phrases indexing instead of terms, and obtained some improvements on TREC datasets. However, this approach cannot handle the scenario in which the query terms are not adjacent to each other. A more relaxed approach [15, 16, 8] attempted to introduce "NEAR" operator to quantify the proximity of query terms. Hawking and Thistlewaite [12] proposed a similar one, which evaluated text segments containing all query terms. In addition, Song *et al.* [38] grouped query terms into phrases and the contribution of a term is determined by how many query terms appear in the context phrases. Under the language modeling framework, Zhao *et al.* [50] used a query term's proximate centrality as a hyper parameter in the Dirichlet language model. Lv *et al.* [20] integrated the positional and proximity information into the language model by a different way. They defined a positional language model at each position in documents by creating virtual documents based on term propagation. With probabilistic models, Zhao *et al.*[48, 49] introduced a pseudo term, called cross term, to measure the association of multiple query terms.

Although there have been plenty of efforts in integrating proximity heuristic into existing retrieval models, work on how to utilize this information for PRF is still limited. Lv *et al.* [21] presented two methods to estimate the joint probability of a term w with the query Q at every position in each feedback document, which extended the relevance model [17], and significant improvements were obtained on

two large collections. Miao *et al.* [24] employed proximity heuristic in a formalistic framework which extensively differs from the language modeling framework. Unlike previous work, we propose a TF transformation method to capture this feature, which is then integrated into the PRF procedure.

Another aspect that could affect the performance of PRF is the quality of feedback documents. In most of traditional PRF approaches, there is a very strong assumption that top ranked documents from the first-pass retrieval are all relevant. In fact, the top ranked documents are not necessarily to be good for PRF since they are not evaluated by real users. Therefore, the candidate expansion terms should be assigned different weights. Several studies ([13, 18]) have investigated this problem by detecting the right documents for PRF, from which expansion terms are extracted. In ([13]), He et al. proposed to detect 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. In addition, Lee et al. ([18]) proposed a re-sampling 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. In this study, we model this problem by transforming the TF of the candidate terms to consider the importance of different feedback documents.

3. A TERM FREQUENCY TRANSFORMATION MODEL

We first give the notations and conventions used in this paper. Then, a term frequency (TF) transformation model for PRF is proposed.

3.1 Notation and Conventions

Given a query Q and a document collection C, a list of ranked documents in descending order, denoted as D, is returned by an information retrieval system. This step is always call first-pass retrieval in the process of PRF. We use d_i to denote the i-th ranked document in D. After the first-pass retrieval, the top-k documents in D will be used as feedback documents in PRF, which is denoted as D_f.

In traditional PRF models, each d_i in D_f will all be treated as relevant. The goal is to utilize these feedback documents to expand the original queries and adjust their weights in order to derive a refined query $Q1$. With $Q1$, we could expect better retrieval performance.

3.2 Enhancement of Rocchio's Model

In this study, we explore the techniques of term frequency transformation in the classic Rocchio's model. Although the Rocchio's model has been introduced in the information retrieval field for many years, it is still very effective in obtaining relevant documents and most of the state-of-the-art PRF approaches are derived from Rocchio's model. According to ([45]), "BM25 ([33]) 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 ([45, 24]) as well as in our preliminary experiments of this paper. In the following, we revisit the traditional Rocchio's models and enhance it with three TF transformation techniques.

The Rocchio's model provides a way of incorporating (pseudo) relevance feedback information into the retrieval process. In case of pseudo relevance feedback, Rocchio's method has the following steps:

1. All documents are ranked for the given query using a particular Information Retrieval model (for example the BM25 model [33]). 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 [36] and the KL-Divergence weighting model [7]).

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. We enhance the Rocchio's model by refining the estimation of r_i described in the following section. In practice, we can always fix α to 1, and only study β in order to get better performance.

3.3 Our Proposed Model

As we can see from the previous section, the most important part within this framework is to calculate the vector r_i for d_i. Namely, how to weight the candidate terms in a feedback document. The traditional approach uses the so-called TF-IDF weighting function to address this problem. However, most of these approaches only normalize the term frequency according to the length of feedback documents. Other aspects failed to be captured in this simple, yet effective framework as discussed in the introduction. So in this paper, we propose different transformation techniques and investigate how to integrate them to obtain a still simple and efficient, but more performing PRF approach. The resulting weighting framework is as follows:

$$w(t, d_i) = \sum_{j=0}^{n} (\lambda_j * tf_j(t)) * IDF(t) \quad (2)$$

where $tf_j(t)$ is the j-th transformation technique and λ_j is the importance of j-th transformation technique. $IDF(t)$ is the inverse document frequency of term t in the collection. In this study, we mainly focus on term frequency transformation techniques. So we simple use the IDF formula from BM25 as follows:

$$IDF(t) = \log \frac{N - n(t) + 0.5}{n(t) + 0.5} \quad (3)$$

where N is the total number of documents in the collection, and $n(t)$ is the number of documents containing t. It is of note that other variants of IDF can also be here, and performance improvement may be expected.

In the following, we present three methods for term frequency transformation, which are used for pseudo relevance feedback.

3.3.1 Weighted TF Transformation

In traditional TF-IDF model and its variants, the term frequency is always normalized according to the length of the document [37, 33, 4]. All these methods have shown to be simple, yet effective to make term frequency comparable in different documents for the first-pass retrieval.

However, feedback documents play different roles in the process of PRF. More specifically, some feedback document may be much more important that other ones. So it is necessary to take into account the quality or importance of a candidate feedback to PRF. In order to address this problem, we define a weighted term frequency by integrating the importance of a candidate document and the normalized term frequency according to the length of this document. The resulting formula, denoted as $TF1$, is as follows:

$$TF1(t,d) = lntf(t,d) * imp(d) \tag{4}$$

where $lntf(t)$ denotes the traditional length-based normalized term frequency, and $imp(d)$ is the importance of document d. For $lntf(t)$, we use the following formula proposed in [4]:

$$lntf(t,d) = tf(t,D) * log_2(1 + \frac{avdl}{len(d)}) \tag{5}$$

where $len(d)$ is the length of document d and $avdl$ is the average document length in the collection. Other TF normalization functions, such as the Robertson TF [33], are also viable. Here, the reason we choose formula 5 is simply because it is not only effective but also parameter-free such that we can focus on evaluating main framework of our proposed PRF model.

For $imp(d)$, without extra knowledge about the candidate feedback document, the best bet is to believe scores of the documents returned in the first-pass retrieval. In particular, we use the normalized scores returned the BM25 model as follows:

$$imp(d) = \sum_{t \in Q} \frac{(k_1+1) * tf(t,d)}{K + tf(t,d)} * \frac{(k_3+1) * qtf(t)}{k_3 + qtf(t)} * IDF(t) \tag{6}$$

where k_1 and k_3 are tuning constants which depend on the dataset used and possibly on the nature of the queries. K equals $k_1 * ((1-b) + b * dl/avdl)$, and dl is the length of the document. In our experiments, the values of k_1, k_3 are default to 1.2 and 8, respectively, which is the recommended setting in [32].

3.3.2 Kernel-based TF Transformation

In the process of PRF, the raw frequency of a term or its length-based normalization variants could be used to estimate its importance in a feedback document. However, it cannot capture the characteristic that whether a candidate term occurs near or far away from the query, which may cause the selected expansion terms not relevant to the query topic. Thus, we propose a Kernel-based term frequency (ktf) transformation method, which models the frequency of a term as well as the closeness to the query in terms of proximity.

In [21, 48], a pseudo term, namely Cross Term, is introduced to model term proximity within original query for boosting retrieval performance. kernel-based method to count the term frequency in a document. There are a number of kernel functions (e.g. Gaussian, Triangle, Cosine, and Circle [48]) which were used for measuring the proximity. The Gaussian kernel has been shown to be effective in most cases. So, in this paper, we adapt the concept of Cross Term with the Gaussian kernel by proposing a kernel-based TF transformation method, which captures the saliency of a candidate term brought not only by its occurrences but also the closeness to the original query. The resulting kernel-based TF between a candidate expansion term t and a query term q is as follows:

$$K(t,q) = exp[\frac{-(p_t - p_q)^2}{2\sigma^2}] \tag{7}$$

where p_t and p_q are respectively the positions of candidate term t and query term q in a document, σ is a tuning parameter which controls the scale of Gaussian distribution.

In this method, beside the average proximity to the query, we also take into account the importance of different query terms. Therefore, we build a representational vector for the query, in which each dimension is the weight of a query term by the inverse document frequency formula below, and then the kernel-based term frequency, denoted as $TF2$, in the Kernel-based method is computed as follows:

$$TF2(t) = \sum_{i=1}^{|Q|} K(t,q_i) IDF(q_i) \tag{8}$$

where q_i is a query term, $|Q|$ is the number of query terms, and $IDF(q_i)$ is the same as in Equation 6. N is the number of documents in the collection, and N_t is the number of documents that contain q_i.

3.3.3 Relative TF Transformation

When comparing the importance of a term in difference documents, normal term frequency methods only consider the frequency of a term itself and the document length. The distributions of other terms are always ignored, while we believe that it will affect the importance of the current term. Similar to [26], let d_1 and d_2 be two documents of equal lengths, the frequency values of t in d_1 and d_2 are the same; but d_2 has more distinct terms and even some other terms have higher frequency that t. One could imagine an extreme case that all other terms in d_i occur only once. So should the weight values of t in these two documents be the same?

With traditional TF normalization techniques, these two documents will be assigned with the same weights, which makes the ranking infeasible in this case. Intuitively, however, d_2 mentions the query term t more frequently than d_1, so t in d_2 is more likely to be most important term than in d_1. In other words, for d_1 it is possible that it talks about a topic not related to the query term t since other terms occur more frequently, while d_2 has a higher chance to talk about the query term t.

In order to capture the saliency of a term in this aspect, we use a relative TF transformation method, denoted as $TF3$

as follows:

$$TF3(t) = \frac{log_2(1 + tf(t,d))}{log_2(1 + atf(d))} \quad (9)$$

where $tf(t,d)$ is the raw term frequency of term t in document d, and $atf(d)$ is the average term frequency of document d. This formula was also used in [37, 26] to normalize the tf values. Defferent from previous work, here we use it to transform the raw TF in the scenario of PRF.

3.3.4 Normalization and Combining

As we can see from Equation 2, the three different TF transformation methods are linearly combined in our proposed model. In order to make the tuning of parameter simple, we need to normalize these three aspects, and the normalization method $f(x)$ is suggested to meet the following property: 1) when the $TF = 0$, $f(TF) = 0$; 2) $f(TF+1) > f(TF)$, but $f(TF+2) - f(TF+1) > f(TF+1) - f(TF)$; it means the weight of a term increases as the increase of TF, but the improvement has a diminishing effect; 3) $f(x)$ maps TF into a specified scale.

One of the possible functions that satisfy the above requirements is as follows:

$$TF^{'}(t) = \frac{TF(t)}{1 + TF(t)} \quad (10)$$

This popular sub-linear TF normalization method has an upper-bound of 1, and puts TF into a range of 0 to 1. It also has the effect of reducing the influence of extreme values or outliers in the data without removing them from the data set. Other normalization methods may also be viable, and we leave this issue for further study in future work.

4. EXPERIMENTAL SETTINGS

4.1 Test Collections and Evaluation Metrics

In this section, we describe six representative test collections used in our experiments: Disk1&2, Disk4&5, WT2G, WT10G, GOV2 and Robust04, which are different in size and genre. The Disk1&2, Disk4&5 collection contains newswire articles from various sources, such as Association Press (AP), Wall Street Journal (WSJ), Financial Times (FT), etc., which are usually considered as high-quality text data with little noise. The WT2G collection is a general Web crawl of Web documents, which has 2 Gigabytes of uncompressed data. This collection was used in the TREC 8 Web track. 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 very large crawl of the *.gov* domain, which has more than 25 million documents with an uncompressed size of 423 Gigabytes. This collection has been employed in the TREC 2004, 2005 and 2006 Terabyte tracks. There are 150 ad-hoc query topics, from TREC 2004 - 2006 Terabyte tracks, associated to GOV2. In our experiments, we use 100 topics in TREC 2005 - 2006. The TREC tasks and topic numbers associated with each collection are presented in Table 1.

In all the experiments, we only use the *title field* of the TREC queries for retrieval. It is closer to the actual queries used in the real application and feedback is expected to be the most useful for short queries [46].

In the process of indexing and querying, each term is stemmed using Porter's English stemmer [28], and stopwords from InQuery's standard stoplist [2] with 418 stopwords are removed. The MAP (Mean Average Precision) performance measure for the top 1000 documents is used as evaluation metric, as is commonly done in TREC evaluations. The MAP metric reflects the overall accuracy and the detailed descriptions for MAP can be found in [41]. We take this metric as the primary single summary performance for the experiments, which is also the main official metric in the corresponding TREC evaluations. To emphasize on the top retrieved documents, we also include P@k in the evaluation measures, which measures precision at fixed low levels of retrieved results, such as 10 or 20 documents. This is referred to as "Precision at k", for example "Precision at 20". It has the advantage of not requiring any estimate of the size of the set of relevant documents but the disadvantages that it is the least stable of the commonly used evaluation measures and that it does not average well, since the total number of relevant documents for a query has a strong influence on precision at k.

Table 1: the TREC tasks and topic numbers associated with each collection.

Collection	Task	Queries	Docs
disk1&2	TREC1,2,3	51-200	741,856
Disk4&5	TREC 2004	301-450	528,155
WT2G	TREC8	401-450	247,491
WT10G	TREC9,10	451-550	1,692,096
GOV2	TREC04-06	701-850	25,178,548
Robust04	Robust04	301-450,601-700	528,155

4.2 Baseline Models

In our experiments, we compare our model with a information-theoretic approach [7] (denoted as $Rocchio_{KL}$) developed under the Rocchio's framework in combination with the basic model BM25 as shown in Equation 6 and Rocchio's feedback model. According to ([45]), "BM25 term weighting coupled with Rocchio feedback remains a strong baseline which is at least as competitive as any language modeling approach for many tasks". In addition, we also compare the proposed models with the state-of-the-art feedback models in language modeling (LM) retrieval framework. In particular, for the basic language model, we use a Dirichlet prior (with a hyperparameter of μ) for smoothing the document language model, which can achieve good performance generally [47].

For PRF in language modeling framework, we first compare our proposed model with the relevance language model [17, 19], which is a representative and state-of-the-art approach for re-estimating query language models for PRF [19]. Relevance language 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:

$$p(w|R) \propto \sum_{\theta_D} p(w|\theta_D)p(\theta_D)P(Q|\theta_D) \quad (11)$$

The relevance model $p(w|R)$ is often used to estimate the feedback language model θ_F, and then interpolated with the original query model θ_Q in order to improve its estimation as follows: $\theta_{Q'} = (1-\alpha) * \theta_Q + \alpha * \theta_F$. This interpolated version of relevance model is called `RM3`. Lv *et al.* [19] systematically compared five state-of-the-art approaches for estimating query language models in ad-hoc retrieval, in which RM3

not only yields impressive retrieval performance in both precision and recall metric, but also performs steadily. In particular, we apply Dirichlet prior for smoothing document language models [46].

4.3 Parameter Settings

As we can see from all the PRF retrieval models in our experiments, there are several controlling parameters to tune. In order to build strong baseline, the parameter b in BM25 and μ in LM are optimized as follows. For the smoothing parameter μ in LM with Dirichlet prior, we sweep over values from 500 to 2000 with an interval of 50. Meanwhile, we sweep the values of b for BM25 from 0 to 1.0 with an interval of 0.05. In order to find the optimal parameter setting for fair comparisons, we use the training method presented in [11] for all the PRF baselines and our models, which is popular in the IR domain for building strong baselines. To evaluate the baselines and our proposed approach, we use 2-fold cross-validation, in which the TREC queries are partitioned into two sets by the parity of their numbers on each collection. Then, the parameters learned on the training set are applied to the test set for evaluation purpose as in [23].

Specifically, for parameters in PRF models, we evaluate all PRF models with different settings of feedback document size ($|D_f| \in \{5, 10, 15, 20, 30, 50\}$). We sweep the number of expansion terms over ($k \in \{10, 15, 20, 25, 30, 35, 50\}$), and the interpolation parameter ($\beta, \lambda_1, \lambda_2, \lambda_3, \in \{0.0, 0.1, \ldots, 1.0\}$).

5. EXPERIMENTS AND ANALYSIS

5.1 Comparison of Basic Retrieval Models

As we mentioned in the previous section, the results of both basic models are optimized using the same method. Therefore, it is fair to compare them on these six collections.

As we can see from Table 2, BM25 slightly outperforms LM with Dirichlet prior on the Disk1&2, WT10G and WT2G collections, while LM is superior on the other three collections. The performance of these two basic models are generally comparative, and no significant difference is observed. So it is reasonable to use them as the basic models of the PRF baselines and our proposed model.

5.2 Comparison with PRF Baseline Models

In Table 3, we present the results of the baseline PRF models and our proposed PRF model with different settings of feedback documents. We denote our PRF model as `TF-PRF`. The last row in each of these tables is the average performance of each PRF model with different settings. We calculate the average MAP scores of each query with different number of D_f, and then conduct significant 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 Section 4.2, the basic retrieval models of $Rocchio_{KL}$ is using BM25 for fair comparison.

First, both of $Rocchio_{KL}$ and RM3 have proven effective and been considered as strong baselines in previous studies. The $Rocchio_{KL}$ model outperforms the RM3 model on the disk4&5, Robust04, WT2G, GOV2 and Robust04 collections, but defeated by the latter on the WT10G collection in terms of average MAP. In terms of average MAP and P@20, our proposed `TF-PRF` model is generally better than the two baseline PRF models on all collections, and achieve significant better results in most cases. The maximum average improvement is as high as 8.00% and 8.25 in terms of MAP and P@20, respectively. Thus it is fair to conclude that our proposed models can outperform $Rocchio_{KL}$ and RM3 generally.

Second, with the increase of feedback document size $|D_f|$, the performance of our proposed `TF-PRF` model 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 while our proposed `TF-PRF` model can still get very good performance on all the six collections. Besides, the optimal $|D_f|$ for our proposed `TF-PRF` model is always larger than that for the baseline models. To some extent, this phenomenon proves the effectiveness of the the weighted term frequency transformation method. Usually, when $|D_f|$ is very small, the top-k documents are more likely to be relevant, and it is reasonable to treat them equally and obtain good performance. However, when $|D_f|$ increases, the ratio of relevant documents will decrease. Since we have already taken this factor into account, the weights of feedback documents will be adjusted so that we can still obtain satisfying results. On the contrary, the baseline models which still treat the feedback documents equally may fail in this case.

Third, it is also interesting to note on Disk1&2 that as the increase of $|D_f|$, the performance of all the PRF models increases. When $|D_f|$ increases to 50, there is only a slight decrease. It means most of the feedback documents are at least not harmful to PRF, which differs from all other collections. This is the only case that our `TF-PRF` model is slightly inferior to $Rocchio_{KL}$, though it still performs significantly better that RM3. The main reason is that the $TF1$ transformation method down-weight the term frequency of terms in lower-ranked documents.

To summarize, all the PRF models 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 the five collections, except Disk1&2. Moreover, our proposed model, `TF-PRF`, makes significant improvements over the baseline models and extensive experiments have shown the effectiveness of `TF-PRF` especially when a relatively larger value of $|D_f|$ is used.

5.3 Robustness Analysis

As we can see from the above experiments, the number of feedback documents $|D_f|$ can greatly impact the performance of PRF models, and the choice of $|D_f|$ turns out to be a challenge problem since it is hard to determine the optimal number of feedback documents. In this section, we further analyze their robustness of our proposed PRF models with respect to $|D_f|$.

From Figure 5.3, it is clear to have a picture about 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, TF-PRF obtains the best value when $|D_f|$ is 30 while RM3 performs the best when $|D_f|$ is 5 on the disk4&5 collection. Meanwhile, the best performance of TF-PRF is much better than that of RM3 on this collection.

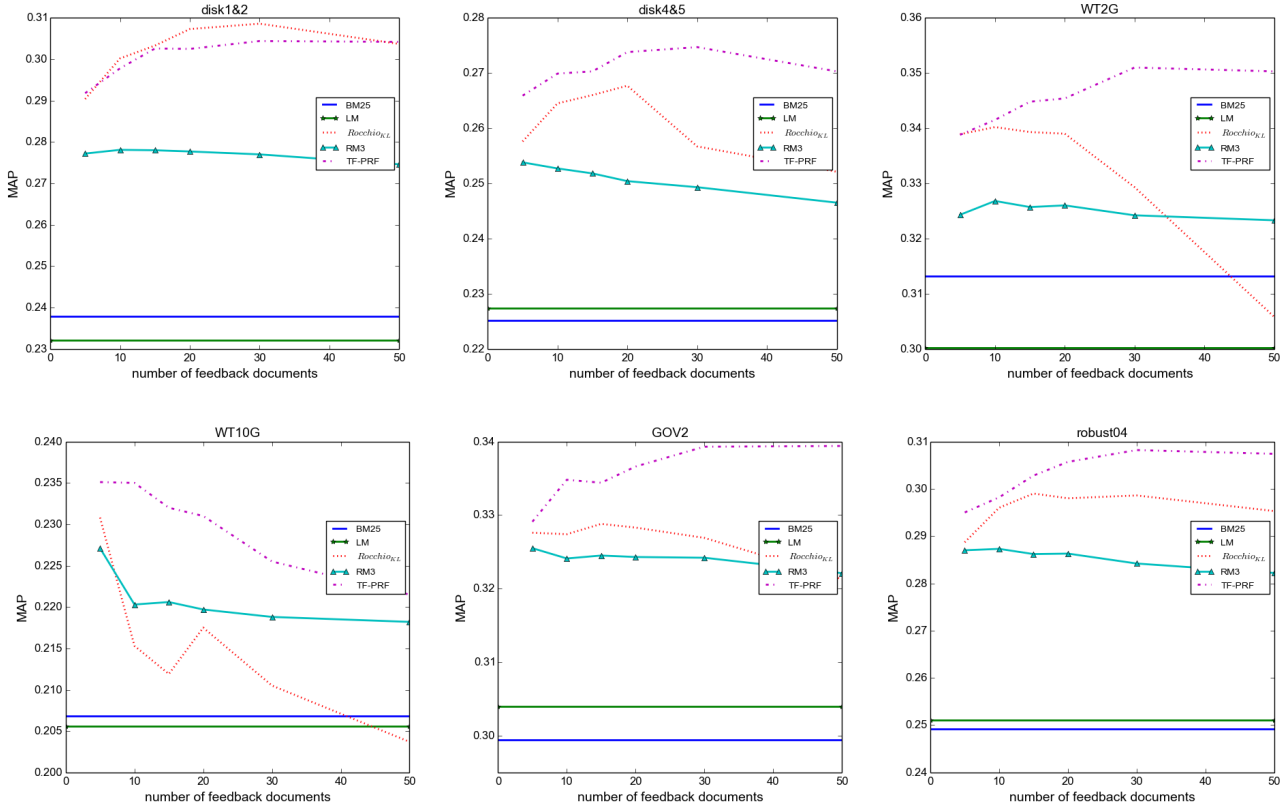

Figure 1: Robustness Comparison in terms of MAP

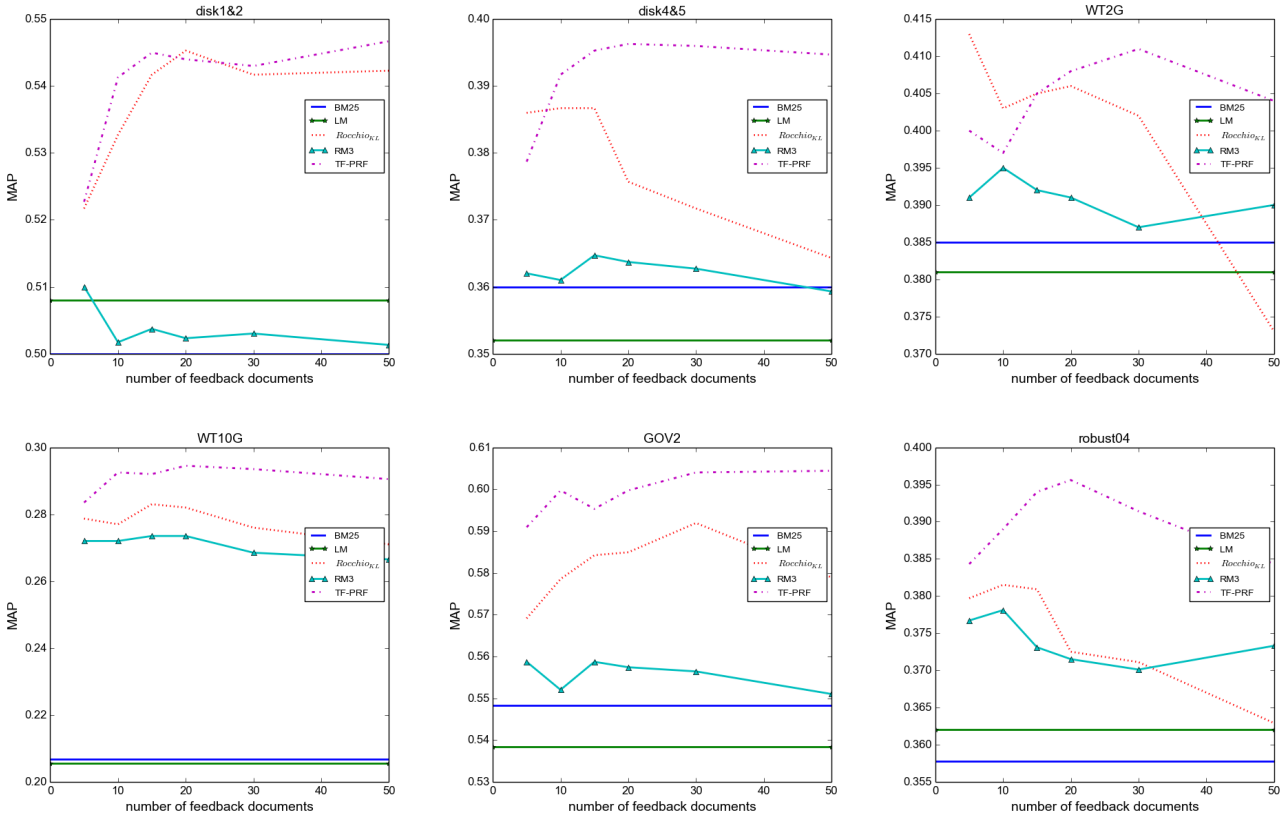

Figure 2: Robustness Comparison in terms of P@20

Table 2: Performance of basic retrieval models in terms of MAP.

Basic Models	disk1&2	disk4&5	WT2G	WT10G	GOV2	Robust04
BM25	0.2378	0.2251	0.3132	0.2068	0.2994	0.2492
LM	0.2320	0.2274	0.3002	0.2056	0.3040	0.2511

Table 3: Comparison of the performance of PRF methods in terms of MAP and P@20. The values in the parentheses are the improvements over $Rocchio_{KL}$ and RM3 respectively. "Ave" in the last column means the average performance of each PRF model with different $|D_f|$. A "*" indicates a statistically significant improvement over the classic Rocchio's model, and a "+" indicates a statistically significant improvement over the RM3 model according to the Wilcoxon matched-pairs signed-ranks test at the 0.05 level. The bold phase style means that it is the best result.

	Models/docs	5	10	15	20	30	50	Average
MAP								
Disk1&2	$Rocchio_{KL}$	0.2904	**0.3002**	**0.3033**	**0.3073**	**0.3086**	0.3037	**0.3023**
	RM3	0.2772	0.2781	0.2780	0.2777	0.2770	0.2746	0.2771
	TF-PRF	**0.2918**$^{+}$ (0.48%, 5.27%)	0.2978^{+} (-0.80%, 7.08%)	0.3026^{+} (-0.23%, 8.85%)	0.3025^{+} (-1.56%, 8.93%)	0.3044^{+} (-1.36%, 9.89%)	**0.3042**$^{+}$ (0.16%, 10.78%)	0.3006^{+} (-0.56%, 8.46%)
Disk4&5	$Rocchio_{KL}$	0.2576	0.2645	0.2660	0.2677	0.2567	0.2521	0.2608
	RM3	0.2538	0.2527	0.2518	0.2504	0.2493	0.2465	0.2508
	TF-PRF	**0.2659**$^{*+}$ (3.22%, 4.77%)	**0.2699**$^{*+}$ (2.04%, 6.81%)	**0.2703**$^{+}$ (1.62%, 7.35%)	**0.2738**$^{*+}$ (2.28%, 9.35%)	**0.2747**$^{*+}$ (7.01%, 10.19%)	**0.2703**$^{*+}$ (7.22%, 9.66%)	**0.2708**$^{*+}$ (3.85%, 8.00%)
WT2G	$Rocchio_{KL}$	**0.3389**	0.3402	0.3393	0.3390	0.3293	0.3058	0.3321
	RM3	0.3243	0.3268	0.3257	0.3260	0.3242	0.3233	0.3251
	TF-PRF	0.3388 (-0.03%, 4.47%)	**0.3415**$^{+}$ (0.38%, 4.50%)	**0.3448**$^{+}$ (1.62%, 5.86%)	**0.3454**$^{+}$ (1.89%, 5.95%)	**0.3510**$^{*+}$ (6.59%, 8.27%)	**0.3503**$^{*+}$ (14.55%, 8.35%)	**0.3453**$^{*+}$ (3.98%, 6.23%)
WT10G	$Rocchio_{KL}$	0.2308	0.2153	0.2119	0.2175	0.2105	0.2037	0.2150
	RM3	0.2271	0.2203	0.2206	0.2197	0.2188	0.2182	0.2208
	TF-PRF	**0.2351**$^{+}$ (1.86%, 3.52%)	**0.2350**$^{*+}$ (9.15%, 6.67%)	**0.2320**$^{*+}$ (9.49%, 5.17%)	**0.2310**$^{*+}$ (6.21%, 5.14%)	**0.2255**$^{*+}$ (7.13%, 3.06%)	**0.2215**$^{*+}$ (8.74%, 1.51%)	**0.2300**$^{*+}$ (7.01%, 4.18%)
GOV2	$Rocchio_{KL}$	0.3276	0.3274	0.3288	0.3283	0.3269	0.3213	0.3267
	RM3	0.3255	0.3241	0.3245	0.3243	0.3242	0.3221	0.3241
	TF-PRF	**0.3291** (0.46%, 1.11%)	**0.3348**$^{*+}$ (2.26%, 3.30%)	**0.3344**$^{+}$ (1.70%,3.05%)	**0.3366**$^{*+}$ (2.53%, 3.79%)	**0.3393**$^{*+}$ (3.79%, 4.66%)	**0.3394**$^{*+}$ (5.63%, 5.37%)	**0.3356**$^{*+}$ (2.72%, 3.54%)
Robust04	$Rocchio_{KL}$	0.2887	0.2960	0.2990	0.2980	0.2986	0.2953	0.2959
	RM3	0.2870	0.2873	0.2862	0.2863	0.2842	0.2822	0.2855
	TF-PRF	**0.2950**$^{*+}$ (2.18%, 2.79%)	**0.2982**$^{+}$ (0.74%, 3.79%)	**0.3028**$^{+}$ (1.27%, 5.80%)	**0.3057**$^{*+}$ (2.58%, 6.78%)	**0.3082**$^{*+}$ (3.22%, 8.44%)	**0.3074**$^{*+}$ (4.10%, 8.93%)	**0.3029**$^{+}$ (2.35%, 6.08%)
P@20								
Disk1&2	$Rocchio_{KL}$	0.5217	0.5327	0.5417	0.5453	0.5417	0.5423	0.5376
	RM3	0.5100	0.5017	0.5037	0.5023	0.5030	0.5013	0.5037
	TF-PRF	**0.5227**$^{+}$ (0.19%, 2.49%)	**0.5413**$^{+}$ (1.61%, 7.89%)	**0.5450**$^{+}$ (0.61%, 8.20%)	**0.5440**$^{+}$ (-0.24%, 8.30%)	**0.5430**$^{+}$ (0.24%, 7.95%)	**0.5467**$^{+}$ (0.81%, 9.06%)	**0.5405**$^{+}$ (0.54%, 7.30%)
Disk4&5	$Rocchio_{KL}$	0.3860	0.3867	0.3867	0.3757	0.3717	0.3643	0.3785
	RM3	0.3620	0.3610	0.3647	0.3637	0.3627	0.3593	0.3622
	TF-PRF	**0.3787**$^{+}$ (-1.89%, 4.61%)	**0.3917**$^{+}$ (1.29%, 8.50%)	**0.3953**$^{*+}$ (2.22%, 8.39%)	**0.3963**$^{*+}$ (5.48%, 8.96%)	**0.3960**$^{*+}$ (6.54%, 9.18%)	**0.3947**$^{*+}$ (8.34%, 9.85%)	**0.3921**$^{*+}$ (3.59%, 8.25%)
WT2G	$Rocchio_{KL}$	**0.4130**	**0.4030**	0.4050	0.4060	0.4020	0.3730	0.4008
	RM3	0.3910	0.3950	0.3920	0.3910	0.3870	0.3900	0.3910
	TF-PRF	0.4000^{+} (-3.14%, 2.30%)	0.3970^{+} (-1.49%, 0.51%)	**0.4050**$^{+}$ (0.00%, 3.32%)	**0.4080**$^{+}$ (0.49%, 5.35%)	**0.4110**$^{*+}$ (2.24%, 6.20%)	**0.4040**$^{*+}$ (8.31%, 3.59%)	**0.4042**$^{+}$ (0.85%, 3.38%)
WT10G	$Rocchio_{KL}$	0.2787	0.2770	0.2830	0.2820	0.2760	0.2710	0.2780
	RM3	0.2720	0.2720	0.2735	0.2735	0.2685	0.2665	0.2710
	TF-PRF	**0.2835**$^{+}$ (1.72%, 4.23%)	**0.2925**$^{*+}$ (5.60%, 7.54%)	**0.2920**$^{*+}$ (3.18%, 6.76%)	**0.2945**$^{*+}$ (4.43%, 7.68%)	**0.2935**$^{*+}$ (6.34%, 9.31%)	**0.2905**$^{*+}$ (7.20%, 9.01%)	**0.2911**$^{*+}$ (4.73%, 7.41%)
GOV2	$Rocchio_{KL}$	0.5691	0.5785	0.5842	0.5849	0.5919	0.5789	0.5813
	RM3	0.5587	0.5520	0.5587	0.5574	0.5564	0.5510	0.5557
	TF-PRF	**0.5909**$^{*+}$ (3.83%, 5.76%)	**0.5997**$^{*+}$ (3.66%, 8.64%)	**0.5953**$^{*+}$ (1.90%, 6.55%)	**0.5997**$^{*+}$ (2.53%, 7.59%)	**0.6040**$^{*+}$ (2.04%, 8.55%)	**0.6044**$^{*+}$ (4.40%, 9.69%)	**0.5990**$^{*+}$ (3.05%, 7.79%)
Robust04	$Rocchio_{KL}$	0.3797	0.3815	0.3809	0.3725	0.3711	0.3629	0.3748
	RM3	0.3767	0.3781	0.3731	0.3715	0.3701	0.3733	0.3738
	TF-PRF	**0.3843**$^{+}$ (1.21%, 2.10%)	**0.3890**$^{*+}$ (1.97%, 2.88%)	**0.3940**$^{*+}$ (3.44%, 5.60%)	**0.3956**$^{*+}$ (6.20%, 6.48%)	**0.3914**$^{*+}$ (5.47%, 5.76%)	**0.3843**$^{+}$ (5.90%, 2.95%)	**0.3898**$^{*+}$ (4.00%, 4.28%)

In addition, the best values of $|D_f|$ of `TF-PRF` 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. Furthermore, after the peak point, the curve of `TF-PRF` falls 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 most collections. Thus, it is clear that our proposed method performs more robustly with respect to $|D_f|$. We also observe that our proposed model can obtain the best performance on all the six collections which are of different sizes and quality. This is a solid evidence that the `TF-PRF` model can make better use of the feedback documents to improve the overall performance and constantly get good results.

In addition, it is also interesting to note that the performance of `TF-PRF` 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, though 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, since `TF-PRF` uses a weighted term frequency 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 viable to address the problem of the selection of the number of feedback documents, while still keep good performance.

Table 4: Comparison with PRM1, PRM2 and PRoc on Tera06 dataset. The bold phase style means that it is the best result.

	PRoc3	PRM1	PRM2	TF-PRF
MAP	0.3283	0.3322	0.3319	**0.3371** (2.68%, 1.48%, 1.57%)
P@10	0.5800	0.5306	0.5490	**0.6248** (7.72%, 17.8%, 13.8%)
P@30	0.5260	0.4884	0.4871	**0.5678** (7.95%, 16.3%, 16.5%)
P@100	0.3756	0.3671	0.3741	**0.4326** (15.2%, 17.8%, 15.6%)

In summary, the proposed model can utilize sufficient number of useful feedback documents, and can also reduce the negative impact of feedback documents with lower qualities. Generally, TF-PRF can reach their best performance when $|D_f|$ is in the range of $[20, 30]$. These values can be used as empirical optimal when no training data is available, although even larger values of $|D_f|$ do not necessarily harm the retrieval performance in terms of MAP and P@20. We also evaluate our proposed model on other data sets and similar results are observed. Due to the limit of space, we did not include all the results here.

5.4 Comparison with the Recent Progress

In this section, we compare our model with the recent progress related to this paper, namely the proximity-based Rocchio's model (PRoc) [24] and the position relevance model (PRM) [21]. PRoc extends the classic Rocchio's model by integrating proximity information between expansion term and original query such that terms that occur closer to the original query will be given more weight in the process of relevance feedback. Unlike PRoc, PRM is developed under the language modeling framework. It extends the relevance model, which takes into account term positions and proximity with the similar intuition that words closer to query words are more likely to be related to the query topic, and assigns more weights to candidate expansion terms closer to the query.

To make the comparison fair, we train our parameters on the Terabyte05 topics and use Terabyte06 [1] topics on the GOV2 collection for testing as Lv. *et al.* did in [21]. Since we do not give results for the Million Query Track so far, we do not compare our method with PRM on the ClueWeb collection with the topics of this track. In [21], parameter μ in the Dirichlet smoothing is set to an optimal value of 1500, and b in our basic model, BM25, is set to 0.3 as PRoc in [24]. As we mentioned previously, the performance of BM25 and LM with Dirichlet smoothing does not differ significantly on the GOV2 collection. Therefore, this setting will not affect the comparison. Since PRoc3 is the most robust and performs the best generally among PRoc's three variants, it is selected to make this comparison. There are two versions of PRM, PRM1 and PRM2, which behave differently with different evaluation measures. So the results of both PRM1 and PRM2 are obtained directly from [21] for fair comparison.

[1]http://trec.nist.gov/data/terabyte.html

First, as we can see from Table 4, the TF-PRF model outperforms PRoc3 and PRM1 in terms of all metrics, which indicates the general effectiveness of our model. Second, it is also interesting to notice that TF-PRF is markedly superior to PRoc3, PRM2 and PRM1 by up to 17.8% improvement in terms of P@10, P@30 and P@100 which is more significant than on MAP. It shows that our model has more advantages in applications that emphasize the top results. In summary, our model is at least comparable to the recent progress in both probabilistic model and language model framework in MAP, and significantly better in P@10, P@30 and P@100.

6. CONCLUSIONS AND FUTURE WORK

In this paper, a new feedback model, TF-PRF, is proposed by incorporating three different term frequency transformation methods into the classic Rocchio's model. Specifically, we present three term frequency transformation methods to capture the saliency of an expansion terms from different local aspects. Then, three frequency measures, namely weighted term frequency, relative term frequency and kernel-based term frequency, are integrated for capturing the overall saliency of expansion terms.

Experiment results on six standard TREC data sets show that the proposed TF-PRF model is very effective and robust, and significantly outperforms strong baseline PRF models in different retrieval frameworks. Meanwhile, our proposed TF-PRF is at least competitive to the most recent work, the PRoc model and the PRM model. Compared to the $Rocchio_{KL}$ model, the proposed model is also less sensitive to the setting of parameter $|D_f|$, the number of feedback documents. Additionally, we carefully analyze the robustness of our model with respect to the number of feedback documents, and an empirical rule to set this parameter is suggested.

There are several interesting future research directions to further explore. We would like to study more term frequency transformation techniques and try to use machine learning models to further optimize the PRF procedure. Another possible research direction is to study how the parameters in our model can be set automatically. It is also interesting to study different normalization and combination methods for integrating the three TF transformation techniques proposed in this paper, and to evaluate our models on more collections (e.g. ClueWeb).

7. ACKNOWLEDGMENTS

This research is supported by the research grant from the Natural Sciences & Engineering Research Council (NSERC) of Canada and the Early Researcher Award/ Premier's Research Excellence Award. We thank five anonymous reviewers for their thorough review comments on this paper.

8. REFERENCES

[1] J. Allan, L. Ballesteros, J. Callan, W. Croft, and Z. Lu. Recent experiments with inquery. In *Proceedings of the 4th Text Retrieval Conference*, pages 49–64, 1995.

[2] J. Allan, M. E. Connell, W. B. Croft, F. Feng, D. Fisher, and X. Li. INQUERY and TREC-9. In *Proceedings of TREC*, 2000.

[3] G. Amati. Probabilistic models for information retrieval based on divergence from randomness. *Ph.D. thesis, Department of Computing Science, University of Glasgow*, 2003.

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

[5] S. Büttcher, C. L. A. Clarke, and B. Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In

Proceedings of SIGIR '06, pages 621–622, New York, NY, USA, 2006. ACM.

[6] G. R. C. Carpineto, R. de Mori and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems (TOIS)*, 19(1):1–27, 2001.

[7] C. Carpineto, R. de Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Trans. Inf. Syst.*, 19(1):1–27, 2001.

[8] C. Clarke, G. Cormack, and F. Burkowski. Shortest substring ranking (MultiText experiments for TREC-4). In *Proceedings of the 4th Text REtrieval Conference*, pages 295–304, 1996.

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

[10] K. Collins-Thompson. Reducing the risk of query expansion via robust constrained optimization. In *Proceedings of CIKM '09*, pages 837–846, New York, NY, USA, 2009. ACM.

[11] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 154–161, New York, NY, USA, 2006. ACM.

[12] D. Hawking and P. Thistlewaite. Proximity operators - So near and yet so far. In *Proceedings of the 4th Text Retrieval Conference*, pages 131–143, 1995.

[13] B. He and I. Ounis. Finding good feedback documents. In *Proceedings of CIKM '09*, pages 2011–2014, 2009.

[14] X. Huang, Y. R. Huang, M. Wen, A. An, Y. Liu, and J. Poon. Applying data mining to pseudo-relevance feedback for high performance text retrieval. In *Proceedings of ICDM '06*, pages 295–306. IEEE Computer Society, 2006.

[15] E. M. Keen. The use of term position devices in ranked output experiments. *J. Doc.*, 47:1–22, 1991.

[16] E. M. Keen. Some aspects of proximity searching in text retrieval systems. *J. Inf. Sci.*, 18:89–98, 1992.

[17] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of SIGIR '01*, pages 120–127, New York, USA, 2001. ACM.

[18] K. S. Lee, W. B. Croft, and J. Allan. A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of SIGIR '08*, pages 235–242, New York, NY, USA, 2008. ACM.

[19] Y. Lv and C. Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of CIKM '09*, pages 1895–1898, New York, NY, USA, 2009. ACM.

[20] Y. Lv and C. Zhai. Positional language models for information retrieval. In *Proceedings of SIGIR '09*, pages 299–306, New York, NY, USA, 2009. ACM.

[21] Y. Lv and C. Zhai. Positional relevance model for pseudo-relevance feedback. In *Proceeding of SIGIR '10*, pages 579–586. ACM, 2010.

[22] D. Metzler and W. B. Croft. Latent concept expansion using markov random fields. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–318, New York, NY, USA, 2007. ACM.

[23] D. Metzler, J. Novak, H. Cui, and S. Reddy. Building enriched document representations using aggregated anchor text. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 219–226, New York, NY, USA, 2009. ACM.

[24] J. Miao, J. X. Huang, and Z. Ye. Proximity-based rocchio's model for pseudo relevance. In *Proceedings of SIGIR '12*, pages 535–544, 2012.

[25] T. Miyanishi, K. Seki, and K. Uehara. Improving pseudo-relevance feedback via tweet selection. In *Proceedings of the 22Nd ACM International Conference on Conference on Information Knowledge Management*, CIKM '13, pages 439–448, New York, NY, USA, 2013. ACM.

[26] J. H. Paik. A novel tf-idf weighting scheme for effective ranking. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 343–352, New York, NY, USA, 2013. ACM.

[27] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR '98*, pages 275–281, New York, NY, USA, 1998. ACM.

[28] M. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14:130–137, 1980.

[29] K. Raman, R. Udupa, P. Bhattacharyya, and A. Bhole. On improving pseudo-relevance feedback using pseudo-irrelevant documents. In *Proceedings of ECIR '10*, pages 573–576, 2010.

[30] Y. Rasolofo and J. Savoy. Term proximity scoring for keyword-based retrieval systems. In F. Sebastiani, editor, *Advances in Information Retrieval*, volume 2633 of *Lecture Notes in Computer Science*, pages 79–79. Springer Berlin / Heidelberg, 2003.

[31] S. E. Robertson and K. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1977.

[32] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne. Okapi at TREC-4. In *The Forth Text REtrieval Conference (TREC-4)*, 1995.

[33] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of TREC-3*, pages 109–126, 1994.

[34] J. J. Rocchio. Relevance feedback in information retrieval. In *G. Salton, The SMART retrieval system: Experiments in automatic document*, pages 313–323, 1971.

[35] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.

[36] G. Salton, A. Wong, and C. Yang. A vector space model for information retrieval. *Journal of American Society for Information Retrieval*, 18(11):613–620, November 1975.

[37] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, Zurich, Switzerland, 1996.

[38] R. Song, M. Taylor, J.-R. Wen, H.-W. Hon, and Y. Yu. Viewing term proximity from a different perspective. In *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 346–357. Springer Berlin / Heidelberg, 2008.

[39] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proceedings of SIGIR '06*, pages 162–169, New York, NY, USA, 2006. ACM.

[40] T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *Proceedings of SIGIR '07*, pages 295–302, New York, USA, 2007. ACM.

[41] E. M. Voorhees and D. Harman. Overview of the sixth text retrieval conference. *Information Processing and Management: an International Journal*, 36:3–35, July 2000.

[42] R. W. White and G. Marchionini. Examining the effectiveness of real-time query expansion. *Inf. Process. Manage.*, 43(3):685–704, 2007.

[43] H. Wu and H. Fang. An incremental approach to efficient pseudo-relevance feedback. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 553–562, New York, NY, USA, 2013. ACM.

[44] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.*, 18(1):79–112, 2000.

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

[46] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of CIKM '01*, pages 403–410. ACM, 2001.

[47] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

[48] J. Zhao, J. X. Huang, and B. He. CRTER: using cross terms to enhance probabilistic information retrieval. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 155–164, New York, USA, 2011. ACM.

[49] J. Zhao, J. X. Huang, and Z. Ye. Modeling term associations for probabilistic information retrieval. *ACM Trans. Inf. Syst.*, 32(2):7, 2014.

[50] J. Zhao and Y. Yun. A proximity language model for information retrieval. In *Proceedings of the 32th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 291–298, New York, USA, 2009. ACM.