Proximity-based Rocchio's Model for Pseudo Relevance Feedback

Jun Miao[1], Jimmy Xiangji Huang[2], Zheng Ye[2]
Information Retrieval and Knowledge Management Research Lab
[1]Department of Computer Science & Engineering, [2]School of Information Technology
York University, Toronto, Canada
jun@cse.yorku.ca, {jhuang, yezheng}@yorku.ca

ABSTRACT

Rocchio's relevance feedback model is a classic query expansion method and it has been shown to be effective in boosting information retrieval performance. The selection of expansion terms in this method, however, does not take into account the relationship between the candidate terms and the query terms (e.g., term proximity). Intuitively, the proximity between candidate expansion terms and query terms can be exploited in the process of query expansion, since terms closer to query terms are more likely to be related to the query topic.

In this paper, we study how to incorporate proximity information into the Rocchio's model, and propose a proximity-based Rocchio's model, called `PRoc`, with three variants. In our PRoc models, a new concept (proximity-based term frequency, ptf) is introduced to model the proximity information in the pseudo relevant documents, which is then used in three kinds of proximity measures. Experimental results on TREC collections show that our proposed PRoc models are effective and generally superior to the state-of-the-art relevance feedback models with optimal parameters. A direct comparison with positional relevance model (PRM) on the GOV2 collection also indicates our proposed model is at least competitive to the most recent progress.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance, Experimentation

Keywords

Pseudo Relevance Feedback, Rocchio's Model, Proximity-based Term Frequency, Query Expansion

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

1. INTRODUCTION

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. Although PRF has been shown to be effective in improving IR performance [4, 6, 9, 13, 23, 26, 28, 30, 36, 37, 40, 42] in a number of IR tasks, traditional PRF can also fail in some cases. For example, when some of the feedback documents have several incoherent topics, terms in the irrelevant contents are likely to misguide the feedback models by importing noisy terms into the queries. This could influence the retrieval performance in a negative way. It is partially because the query itself was ignored in the process of expansion term selection. To be more specific, the term associations between candidate terms and the query terms have been ignored in 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 [3, 5, 8, 10, 15, 24, 31, 33]. Various methods 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 non-relevant documents.

In the field of PRF, based on the assumption that terms closer to the query terms are more likely to be relevant to the query topic, there are several studies which investigated how to give more weight to these terms in the process of pseudo relevance feedback. For example, to this end, Vechtomova *et al.* [34] imported a distance factor which combined with Mutual Information (MI) for selecting query expansion terms. Lv *et al.* [16] proposed a positional relevance model (PRM) by using position and proximity information to solve this problem and obtain significant performance.

However, as far as we are aware, there is little work done on incorporating proximity information into the traditional Rocchio's feedback model. Although the Rocchio's model has been introduced in the information retrieval field for many years, it is still effective in obtaining relevant documents. According to [41], "BM25 [27] 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 our preliminary experiments of this paper. Therefore, it

is promising to make an extension of the Rocchio's model to take into account the proximity information.

In addition, it is unknown how to tackle the challenge of modeling the traditional statistics of expansion terms and the relationship between expansion terms and the query terms in a unified framework. In this paper, we propose a proximity-based feedback model based on the traditional Rocchio's model, called `PRoc`. Unlike the PRM model, we focus on the proximity of terms rather than the positional information. In our method, we estimate the weights of candidate expansion terms by taking their distance from query terms into account. Specifically, if a term is far away from the query terms in the feedback documents, it is proposed to be punished by discounting its weight because it is likely to be irrelevant to the query topic.

The main contribution of this paper is as follows. First, we study how to adapt the traditional Rocchio's model [28] for proximity information, and propose a proximity-based feedback model, called `PRoc`, in which the traditional statistics of expansion terms and the proximity relationship between expansion terms and the query terms are taken into account. Second, we propose to use three proximity measures. Unlike previous methods, the importance of query terms has been taken into account to measure the proximity information by proposing a new concept, namely proximity-based term frequency. Finally, extensive experiments on standard TREC collections have been conducted to evaluate our proposed feedback model from different aspects. We compare our proposed feedback model with strong feedback baselines. Our model can achieve significantly better performance over RM3 and the classic Rocchios' model.

The remainder of this paper is organized as follows: in Section 2 we review the related work. In Section 3, three methods for measuring the proximity and our proposed model, PRoc, are presented in details. In Section 4, we introduce the settings of the experiments. In Section 5, the experimental results are presented and discussed. A direct comparison is made to compare PRoc with the most recent work PRM. Finally, we conclude our work with a brief conclusion and future research directions in Section 6.

2. RELATED WORK

2.1 Pseudo Relevance Feedback

In information retrieval, PRF via query expansion is referred to as the techniques, algorithms or methodologies that reformulate the original query by adding new terms into the query, in order to achieve a better retrieval performance. 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.

A classical relevance feedback technique was proposed by Rocchio in 1971 for the Smart retrieval system [28]. It is a framework for implementing (pseudo) relevance feedback via improving the query representation, in which a set of documents are utilized as the feedback information. Unique terms in this set are ranked in descending order of their TFIDF weights. In the following decades, many other relevance feedback techniques and algorithms were developed, mostly derived under Rocchio's framework. For example, a popular and successful automatic PRF algorithm was proposed by [26] in the Okapi system; Amati *et al.* [2] proposed a query expansion algorithm in his divergence from randomness (DFR) retrieval framework. In addition, with the development of language model [21] in IR, a number of techniques (e.g. [13, 32, 42]) have been developed to fit in the language modeling framework. In addition, Robertson [25] proposed a theoretical feedback model that supports the assumption by using of the difference of term distributions to select and re-weight the expansion terms.

For PRF in the language modeling framework, we always exploit feedback information (e.g., the top-ranked documents set, $F = D_1, D_2, \ldots, D_{|F|}$), in order to re-estimate a more accurate query language model. For example, the model based feedback approach [42] 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 [13] 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 [18]. Lv *et al.* [14] 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 [13, 32, 42]. They found that SMM and RM3 are the most effective in their experiments, and RM3 is more robust to the setting of feedback parameters.

However, most of these PRF approaches estimate the importance (or probability) of the candidate expansion terms based on their statistics, while the proximity information is always ignored.

2.2 Term Proximity in Previous Work

Term proximity is the co-occurrences of terms within a specified distance. Particularly, the distance is the number of intermediate terms in a document. Plenty of work has been done to integrate term proximity into both probabilistic and language models. Keen [10, 11] firstly attempted to import term proximity in the Boolean retrieval model by introducing a "NEAR" operator. Buttcher *et al.* [3] proposed an integration of term proximity scoring into Okapi BM25 and obtain improvements on several collections. Rasolofo *et al.* [24] added additional weight to the top documents which contain query term pairs appearing in a window through a two-phase process, but the improvement is somewhat marginal. Song *et al.* [31] presented a new perspective on term proximity. Query terms are grouped into phrases, and the contribution of a term is determined by how many query terms appear in the context phrases. In order to make it clear that how we could model proximity and incorporate a proximity measure, Tao *et al.* [33] systemically studied five proximity measures and investigated how they perform in the KL-divergence retrieval model and the Okapi BM25 retrieval model. Under the language modeling framework, Zhao *et al.* [45] used a query term's proximate centrality as a hyper parameter in the Dirichlet language model. Lv *et al.* [15] 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 create virtual documents based on term probation.

All the above work focuses on how to utilize the proxim-

ity information of query terms to avoid documents which contain scattered query terms. This kind of documents should be punished because they are very likely to be irrelevant. For example, a document contains both "Japan" and "Earthquake" is possible to be irrelevant to the topic "Earthquake in Japan" if these two terms are not close in the context. It could be biased to only "Earthquake" and mention some technologies in "Japan" about "Earthquake". Term proximity is effective to discriminate against these types of documents. Although there have been plenty of efforts in integrating proximity heuristic into existing retrieval models, research work about how to utilize this information for pseudo relevance feedback is still limited. Vechtomova *et al.* [34] combined several distance factors with Mutual Information for selecting query expansion terms from windows or passages surrounding query term occurrences. However, marginal improvements were observed in the experiments. Lv *et al.* [16] presented two methods to estimate the joint probability of a term w with the query Q at every position in each feedback document. This is an extension of the state-of-the-art relevance model [13], and significant improvements were obtained on two collections. Besides the work presented [16, 34], it is difficult to find other systematical work about formally modeling term proximity heuristic in the context of pseudo feedback, especially in the classic Rocchio's model.

In this paper, we propose PRoc models which integrates the proximity information of terms into the traditional Rocchio's framework. Three kinds of proximity measures are introduced to estimate relevance and importance of the candidate terms. In order to integrate term proximity into the Rocchio's model, we re-interpret the definition of term frequency and introduce a new concept, proximity term frequency. Unlike in [34], we conduct our study on a mature feedback model which has proven to be effective. Instead, the work of Vechtomova *et al.* was based on the Mutual Information which failed to perform as well as the traditional feedback model. In contrast to the work in [16], we try to employ proximity heuristic in a formalistic framework which extensively differs from the language modeling framework. Indeed, we do not concern ourselves with the position of each candidate term as in [16]. Meanwhile, to confirm the effectiveness of our model, we compare the performance of PRoc with that of PRM in Section 5. The experimental results show that our proposed PRoc is at least competitive to the most recent work, PRM.

3. PROXIMITY-BASED ROCCHIO'S MODEL

In this section, we present the proposed proximity-based Rocchio's model, called PRoc. Specifically, we first briefly introduce the traditional Rocchio's model, and present the adaption of Rocchio's model for proximity information by proposing a new concept, namely proximity-based term frequency (ptf). Then we describe in details about how to adopt ptf in three investigated proximity measures.

3.1 Adaption of Rocchio's Model

Rocchio's model [28] is a classic framework for implementing (pseudo) relevance feedback via improving the query representation. It models a way of incorporating (pseudo) relevance feedback information into the vector space model (VSM) in IR. In case of pseudo relevance feedback, the Rocchio's model without considering negative feedback documents has the following steps:

1. All documents are ranked for the given query using a particular retrieval model. This step is called *first-pass retrieval*, from which the $|R|$ highest ranked documents are used as the feedback set.
2. Each document in the feedback set R is represented as a weighted term vector, annotated by r, by a certain weighting function, for example originally by the TFIDF weights [29].
3. The representation of the query is finally refined by taking a linear combination of the initial query term vector with the feedback document vector:
$$Q_1 = \alpha * Q_0 + \beta * \sum_{r \in R} \frac{r}{|R|} \quad (1)$$
where Q_0 and Q_1 represent the original and first iteration query vectors, r is the expansion term weight vector, and α and β are tuning constants controlling how much we rely on the original query and the feedback information. In practice, we can always fix α at 1, and only study β in order to get better performance.

Many other relevance feedback techniques and algorithms [2, 4, 26] are also derived under the Rocchio's framework. For example, Carpineto *et al.* proposed to compute the weight of candidate expansion terms based on the divergence between the probability distributions of terms in the top ranked documents and the whole collection. In this paper, we also take advantage of this distributional view. But we re-interpret the definition of *term frequency* in the KLD formula 2 instead of the distribution estimated from a set of top documents. We use the following function to rank the candidate terms:
$$score(w) = P(w|d) * log(\frac{P(w|d)}{P(w|C)}) \quad (2)$$
where $P(w|d)$ is the probability of candidate expansion term w in feedback document d, $P(w|C)$ is the probability in the retrieval collection C.

Traditionally, candidate terms are ranked by their weights in the feedback documents, and the weights are affected by term frequencies in these documents extensively. However, the normal term frequency cannot capture the characteristic that whether a candidate term occurs near or far away from the query, such that the candidate term may not be relevant to the query topic. In other words, if the occurrence of a term is far away from the query terms, it should not be counted in the effective term frequency because this term is very likely to be irrelevant to the query topic. Thus, we propose a new concept, proximity term frequency (ptf), which models the frequency of a term as well as the semantics to the query in terms of proximity. In order to adapt the proximity information, we re-interpret the definition of *term frequency* by proposing three kinds of proximity measures: window-based method, kernel-based method and the hyperspace analogue to language method. The main research challenge now we are facing is how to evaluate *ptf*. In the following subsections, we introduce three measures to compute the ptf. Meanwhile, the importance of query terms is also taken into account. A very frequent query term is likely to be close to many candidate terms, which makes it difficult to distinguish the related feedback terms. Inverse document frequency (idf) of query terms is integrated to calculate ptf.

3.2 Window-based Method

The first method adopts a simple window-based n-gram frequency counting method, which has been popular in previous studies on using term proximity for IR (e.g. [17, 20]).

The basic idea of the window-based n-gram counting method is to segment the document into a list of sliding windows, with each window having a fixed window size *wSize*. If a document has a length of *l*, and the window size is set to *wSize*, the document is then segmented into *l-wSize* sliding windows, where each window contains *wSize* consecutive tokens. For example, if a document has four tokens A, B, C, and D, and the window size is 3, there are two windows in this document, namely A, B, C and B, C, D. The n-gram frequency is then defined as the number of windows in which all n-gram terms co-occur. There could be two variants of the n-gram models, namely the *ordered* and *unordered* n-gram models. The ordered n-gram model takes the order of occurrences of the n-gram terms into account. For the same n-gram terms, the n-grams in which the composing n-gram terms appear in different orders are considered as different n-grams. In contrast, the unordered model ignores the order of occurrences of the n-gram terms. Actually, only a rough distance between terms is considered in this measure. If two terms are in the window, they are strongly related and the co-occurrence is counted in ptf.

$$ptf(t) = \sum_{i=1}^{|Q|} C(t, q_i) IDF(q_i) \tag{3}$$

where q_i is a query term, C(t, q_i) is the number of windows in which the candidate term and the query terms co-occur, $|Q|$ is the number of query terms, and $IDF(q_i)$ equals to $log(N - N_t + 0.5)/(N_t + 0.5)$. N is the number of documents in the collection, and N_t is the number of documents that contain q_i.

The n-gram counting method has the advantage of being straight-forward, and can be easily deployed in practice. It does not take the actual distance between query terms into account directly, and any n-gram terms appear together within a window is counted as one occurrence of the n-gram. If a term is very close to a query term, its co-occurrence count with the query term will be more than that of a term far away from this query term in the sliding windows. This variant of PRoc is denoted by `PRoc1` in the rest of this paper.

3.3 Kernel-based Method

Following to previous studies [16, 44], an alternative method we use is a 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 [44]) which were used for measuring the proximity. Gaussian kernel has been shown to be effective in most cases. In this paper, we also use the Gaussian kernel to measure the proximity between a candidate expansion term t and a query term q.

$$K(t, q) = exp[\frac{-(p_t - p_q)^2}{2\sigma^2}] \tag{4}$$

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 other words, σ has a similar effect as the parameter $wSize$ in window-based method. In order to keep the consistency with other proximity measures, we also use $wSize$ to denote σ. Different from the window-based method, the kernel-based method is a soft proximity measure. In particular, even if the appearance of a candidate term and a query term is not in a window of $wSize$, its weight can still be slightly boosted.

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 proximity-based term frequency ptf in the Kernel-based method is computed as follows:

$$ptf(t) = \sum_{i=1}^{|Q|} K(t, q_i) IDF(q_i) \tag{5}$$

where q_i is a query term, $|Q|$ is the number of query terms, and $IDF(q_i)$ is the same as in PRoc1. N is the number of documents in the collection, and N_t is the number of documents that contain q_i. The second variant of PRoc is denoted by `PRoc2` in the rest of this paper.

3.4 HAL Method

The Hyperspace Analogue to Language (HAL) [12] is a computational modeling of psychological theory of word meaning by considering context only as the words that immediately surround the given word. The basic motivation is that when a human encounters a new concept, its meaning is derived via other concepts occurred within the same context.

An shown in [39], the HAL Space is automatically built from a corpus of text, defined as follows: for each term in a specified vocabulary V, a $|V| \times |V|$ matrix is built by moving a sliding windows of length $wSize$ across the corpus, where $|V|$ is the number of terms in vocabulary V. All words within the window are considered as co-occurring with each other with strengths inversely proportional to the distance between them. The weightings of each co-occurred terms are accumulated over the corpus. Then, a term can be represented by a semantic vector, in which each dimension is the weight for this term and other terms as follows:

$$HAL(t'||t) = \sum_{k=1}^{wSize} w(k) n(t, k, t') \tag{6}$$

where k is the distance from term t' to t, n(t, k, t') is the co-occurrence frequency within the sliding windows when the distance equals k, and $w(k) = wSize - k + 1$ denotes the strength.

In this paper, we adapt the the original HAL model similarly as in [12]. In particular, in order to measure the proximity between a candidate expansion term and the original query, we restrict the context to the query terms, not all the co-occurred terms in the feedback documents. With this adaption, the resulting vector for each candidate term denotes a proximity relationship with the entire query. Like the Kernel-based method, we also take into account the importance factor of query terms in the same way. Then, the HAL based ptf is as follows:

$$ptf(t) = vec(t) \cdot vec(Q) = \sum_{i=1}^{|Q|} HAL(t||q_i) IDF(q_i) \tag{7}$$

$IDF(q_i)$ is the as in PRoc1

In the adaption of proximity information in PRF, ptf replaces the traditional term frequency in our approach.

The weighted HAL model includes the information of term distances and co-occurrence frequencies completely. It is the first time that this linguistic model is adopted to measure the proximity. The third variant of PRoc is denoted by `PRoc3` in the rest of this paper.

4. EXPERIMENTAL SETTINGS

4.1 Test Collections

In this section, we describe four representative test collections used in our experiments: Disk4&5, WT2G, WT10G, and GOV2, which are different in size and genre. The 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.

The GOV2 collection, which has 426 Gigabytes of uncompressed data, is crawled from the .gov domain. This collection has been employed in the TREC 2004, 2005 and 2006 Terabyte tracks. GOV2 is a very large crawl of the *.gov* domain, which has more than 25 million documents with an uncompressed size of 423 Gigabytes. 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. As we can see from this table, we evaluate the proposed approach with a relative large number of queries. 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 [42].

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

Collection	Task	Queries	Docs
Disk4&5	TREC 2004, Robust	301-450	528,155
WT2G	TREC8, Web ad-hoc	401-450	247,491
WT10G	TREC9, 10, Web ad-hoc	451-550	1,692,096
GOV2	TREC04-06, Web ad-hoc	701-850	25,178,548

In the process of indexing and querying, each term is stemmed using Porter's English stemmer [22], and stopwords from InQuery's standard stoplist [1] 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 [35]. 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.

4.2 Baseline Models

In our experiments, we compare our PRoc models with the traditional combination of BM25 and Rocchio's feedback model. In addition, we also compare the proposed models with the state-of-the-art feedback models in the KL-divergence 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 as shown in Equation 8, which can achieve good performance generally [43]. Besides, this is also utilized as the basic model in [16].

$$p(w|d) = \frac{c(w_d) + \mu p(w|C)}{|d| + \mu} \tag{8}$$

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 C and $|d|$ is the length of document d. We train the parameter in the document language model in all the experiments in order to make fair comparisons, and focus on evaluating different ways of approaching the query-related topic for PRF.

For PRF in language modeling framework, we first compare our proposed model with the relevance language model [13, 14], which is a representative and state-of-the-art approach for re-estimating query language models for PRF [14]. 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:

$$p(w|R) \propto \sum_{\theta_D} p(w|\theta_D)p(\theta_D)P(Q|\theta_D) \tag{9}$$

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:

$$\theta_{Q'} = (1-\alpha) * \theta_Q + \alpha * \theta_F \tag{10}$$

This interpolated version of relevance model is called `RM3`. Lv *et al.* [14] 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 [42].

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 find the optimal parameter setting for fair comparisons, we use the training method presented in [7] for both the baselines and our proposed models, which is popular in the IR domain for building strong baselines. In particular, first, for the smoothing parameter μ in LM with Dirichlet prior, we sweep over values from 500 to 2000 with an interval of 100. Meanwhile, we sweep the values of b for BM25 from 0 to 1.0 with an interval of 0.1. Second, for parameters in PRF models, we empirically set the number of top documents to 20 in baseline PRF approaches and our PRoc models, the number of expansion terms ($k \in 10, 20, 30, 50$), and the interpolation parameter ($\beta \in 0.0, 0.1, \ldots, 1.0$). The window size for PRoc models are from 10 to 1500 with an interval 10. 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 [19].

Table 2: BM25 vs LM on the four TREC collections

	disk4&5	WT2G	WT10G	GOV2
BM25	0.2216	0.3124	0.2055	0.3034
LM	0.2247	0.2995	0.2063	0.3040

5. EXPERIMENTS AND ANALYSIS

5.1 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 four collections. BM25 slightly outperforms LM with Dirichlet prior on the WT2G collection. The results of these two models are almost the same over the Disk4&5, WT10G and GOV2 collections. This comparison indicates that the classic BM25 model is generally comparative to LM, and it is reasonable to use them as the basic models of the PRF baselines and our proposed model.

5.2 Comparison with PRF Models

From Table 2 to Table 6, we can clearly see that the average performance of PRF models is superior to the basic models in most cases. The classic Rocchio's model achieves improvements of 13.31%, -0.41%, 2.34% and 4.22% over BM25 on the Disk4&5, WT2G, WT10G and GOV2 collections, while RM3 obtains significant improvements over LM (4.05%, 7.98%, 3.64% and 3.32%) on all the four collections[1]. The effectiveness of pseudo relevance feedback is reconfirmed in this set of experiments. The classic Rocchio's model, fails to obtain improvement on the WT2G collection. This indicates that the Rocchio's model is not so robust as RM3 in this case. However, the Rocchio's model outperforms RM3 on the Disk4&5 collection significantly while RM3 performs better than the classic Rocchio's model on the WT2G collection. On the WT10G and GOV2 collections, their results are very close. Therefore, the Rocchio's model is generally comparable to RM3 so that it is still competitive to be a strong baseline.

In general, the performance of our proposed PRoc models is close on all the four collections, and all of them obtain more improvements over the basic models than the Rocchio's model and RM3. Specifically, from Table 3 to Table 6, we observe that all the three proximity-based Rocchio's models outperform the classic Rocchio's model (2.00% - 11.54%) and state-of-the-art RM3 (4.78% - 12.75%) significantly on all the four collections, which demonstrates the effectiveness of the three PRoc models. Although the measures in our PRoc models are different, all of them can successfully model the proximity information to some extent. Furthermore, the PRoc models perform more robustly than the classic Rocchio's model. It indicates that proximity plays an important role in discriminating relevant expansion terms from irrelevant ones.

In addition, from Table 4 we observe that PRoc3 outperforms the other two on the WT2G collection. On the other three collections, the performance of all the three PRoc models is very close. Generally, PRoc3 is slightly more effective than the other two PRoc models.

[1]The computation of these percentages is based on the average performance of the Rocchio's model and RM3 in Table 3∼6 and MAPs in Table 2.

5.3 Effectiveness of Window Size

In our proposed PRoc models, there are two important parameters: (1) β in the feedback models controlling how much we rely on the original query and the feedback information and (2) window size parameter $wSize$ in the calculation of the proximity-based frequency. In our preliminary experiments, we observed that the influence of β is similar to that in [38], which investigated this parameter thoroughly. Since we mainly focus on the study of proximity evidence, detailed discussion about β will not be made in this paper.

$wSize$ is a key parameter for most proximity measures because it determines the distance in which terms are considered to be related. Thus, how to find an appropriate window size is very important for adapting the proximity measures. In this section, we attempt to discover some useful evidence for obtaining optimal $wSize$ values. Particularly, σ in PRoc2 is also interpreted as the window size.

From Figure 1 to 4, we show how the performance of our PRoc models changes with $wSize$ on different collections. We investigate a large range of $wSize$ from 10 to 1500, and the numbers of expansion terms are 10, 20, 30 and 50. Generally, the values of $wSize$ affect the performance of all the PRoc models extensively. In the second subfigure of Figure 4, the best MAP is 0.3205 when number of expansion terms is 50, and it falls to 0.2286 when $wSize$ is 1500. Almost 30% of performance is lost in this case.

For PRoc1, PRoc2 and PRoc3, although they are based on different measures, their curves fluctuate similarly on the same collection with different numbers of expansion terms. In contrast, the curves of each PRoc model are various extensively on different collections. This demonstrates that the influence of $wSize$ is collection-based. However, the best $wSize$ values for PRoc models are not the same, not even close to each other. For example, on the disk4&5 collection, optimal $wSize$ values for PRoc1 are 80, 50, 100 and 80 over 10, 20, 30 and 50 expansion terms, and the corresponding values for PRoc1 on WT10G is 100, 30, 10 and 10. Thus, the optimal values of $wSize$ depend on the proximity measures and the collections.

Another phenomenon is that the more the expansion terms are selected, the more the performance is affected by $wSize$. Normally, the performance of PRoc models drops when $wSize$ takes a relatively large value. However, to what extent the performance is affected is determined by the number of expansion terms. Specifically, in Figure 2, while $wSize$ is relatively small, the performance of PRoc models with 50 expansion terms is the best. However, when $wSize$ is larger than 200, the curves for 50 expansion terms are constantly below all the others. As an additional example, when the number of expansion terms is 30, the performance of PRoc models is the second worst in most cases when $wSize$ is 1500.

This is reasonable because the accumulated influence of proximity information for 50 expansion terms is larger than that of the small numbers. When $wSize$ increases, it is very likely that more noise is adopted in the expansion term selection. The more expansion term are there, the more noisy information is involved. Thus, the $wSize$ must be set very carefully when the number of expansion is larger than 30 in our case.

Additionally, the influence of $wSize$ on PRoc1 is more significant than on the other two over WT10G. Meanwhile, $wSize$ affects PRoc2 more significantly over GOV2 than PRoc1 and PRoc3. Overall, the PRoc3 model is less sensi-

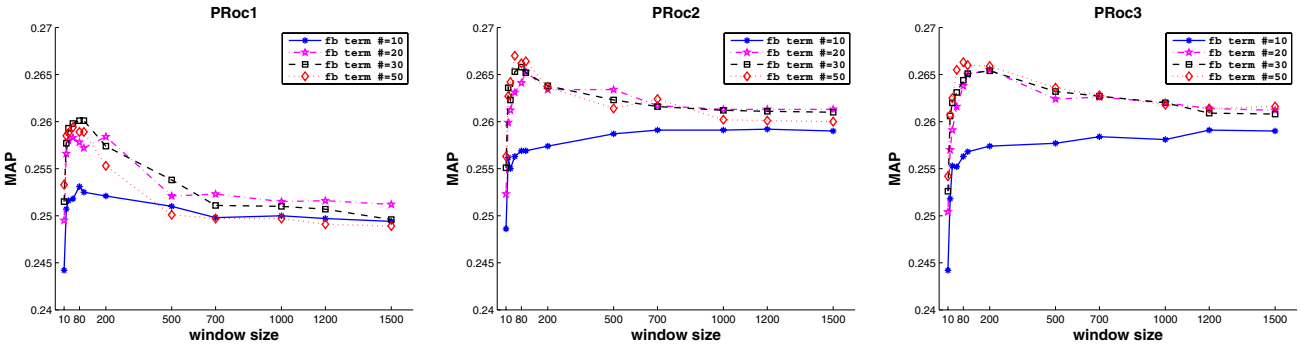

Figure 1: PRoc1, PRoc2 and PRoc3 over disk4&5 with 10, 20, 30 and 50 expansion terms

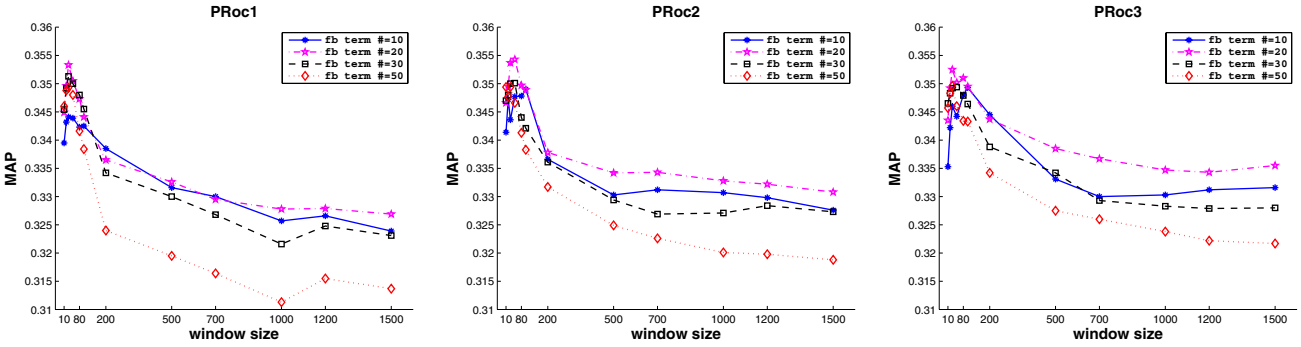

Figure 2: PRoc1, PRoc2 and PRoc3 over WT2G with 10, 20, 30 and 50 expansion terms

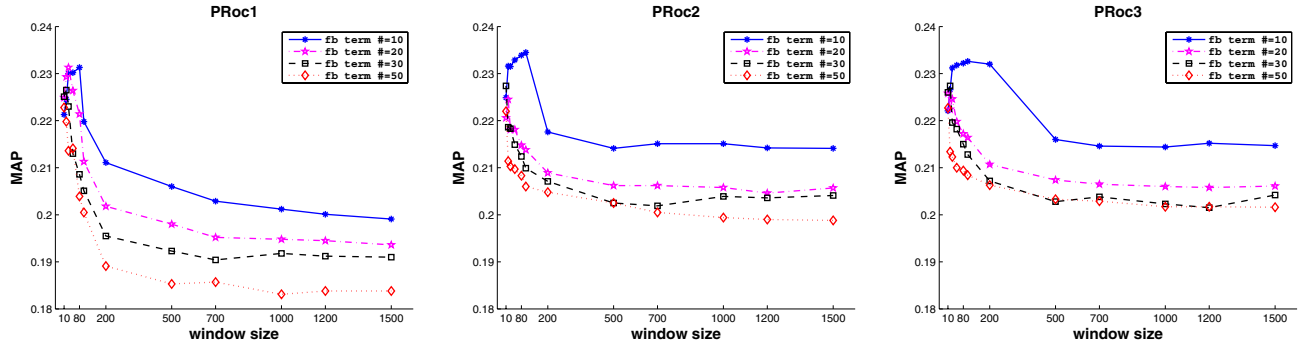

Figure 3: PRoc1, PRoc2 and PRoc3 over WT10G with 10, 20, 30 and 50 expansion terms

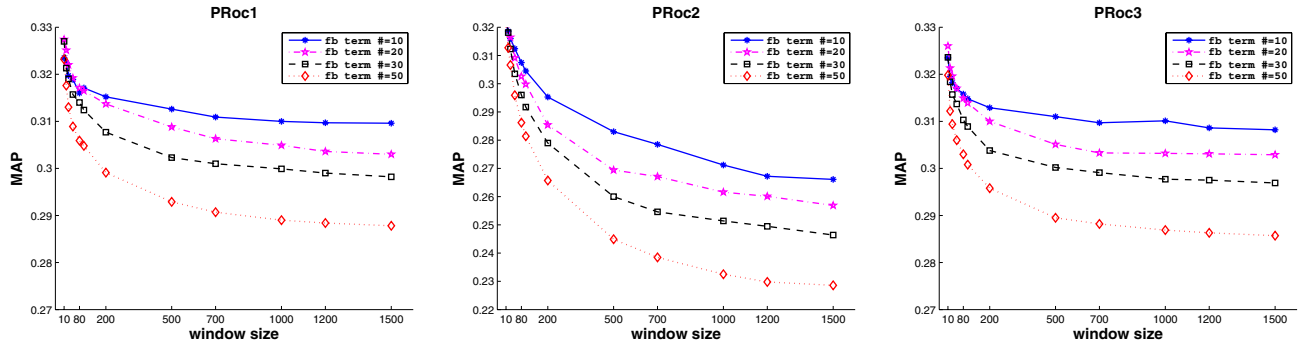

Figure 4: PRoc1, PRoc2 and PRoc3 over GOV2 with 10, 20, 30 and 50 expansion terms

Table 3: PRoc compares with BM25+Rocchio and LM+RM3 on Disk4&5. The percentages in the parenthesis are the improvement gains over the classic Rocchio's model and RM3. A "*" indicates a statistically significant improvement over the classic Rocchio's model baseline, and a "+" indicates a statistically significant improvement over the RM3 model baseline 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.

# of feeback terms	PRoc1	PRoc2	PRoc3	BM25 + Rocchio	LM + RM3
10	0.2509^{+}	0.2523^{+}	$\mathbf{0.2571}^{*+}$	0.2463	0.2289
20	0.2567^{+}	0.2596^{+}	$\mathbf{0.2647}^{*+}$	0.2504	0.2326
30	0.2589^{+}	0.2595^{+}	$\mathbf{0.2662}^{*+}$	0.2545	0.2356
50	0.2578^{+}	0.2602^{+}	$\mathbf{0.2662}^{*+}$	0.2533	0.2382
Average	0.2561 (2.00%, 9.54%,)	0.2579 (2.71%, 10.31%,)	0.2636 (4.98%, 12.75%,)	0.2511	0.2338

Table 4: PRoc compares with BM25+Rocchio and LM+RM3 on WT2G

# of feeback terms	PRoc1	PRoc2	PRoc3	BM25 + Rocchio	LM + RM3
10	$\mathbf{0.3415}^{*+}$	$\mathbf{0.3415}^{*+}$	0.3385^{*+}	0.3091	0.3212
20	$\mathbf{0.3501}^{*+}$	0.3403^{*+}	0.3424^{*+}	0.311	0.3225
30	0.3452^{*+}	0.3456^{*+}	$\mathbf{0.3461}^{*+}$	0.3082	0.3255
50	0.3513^{*+}	0.3406^{*+}	$\mathbf{0.3525}^{*+}$	0.3162	0.3242
Average	0.3470 (11.54%, 7.30%,)	0.3420 (9.93%, 5.75%,)	0.3449 (10.86%, 6.65%,)	0.3111	0.3234

Table 5: PRoc compares with BM25+Rocchio and LM+RM3 on WT10G

# of feeback terms	PRoc1	PRoc2	PRoc3	BM25 + Rocchio	LM + RM3
10	0.2290^{*+}	0.2267^{*+}	$\mathbf{0.2308}^{*+}$	0.2143	0.2098
20	0.2272^{*+}	0.2219^{*+}	$\mathbf{0.2287}^{*+}$	0.2147	0.2168
30	0.2260^{*+}	$\mathbf{0.2264}^{*+}$	0.2261^{*+}	0.2084	0.2151
50	0.2202^{*+}	0.2206^{*+}	$\mathbf{0.2245}^{*+}$	0.2039	0.2136
Average	0.2256 (7.28%, 5.52%,)	0.2239 (6.47%, 4.72%,)	0.2275 (8.18%, 6.41%,)	0.2103	0.2138

Table 6: PRoc compares with BM25+Rocchio and LM+RM3 on GOV2

# of feeback terms	PRoc1	PRoc2	PRoc3	BM25 + Rocchio	LM + RM3
10	0.3271^{*+}	$\mathbf{0.3294}^{*+}$	0.3288^{*+}	0.3126	0.3071
20	0.3303^{*+}	$\mathbf{0.3325}^{*+}$	0.3315^{*+}	0.3164	0.3141
30	$\mathbf{0.3323}^{*+}$	0.3316^{*+}	0.3320^{*+}	0.3175	0.3167
50	0.3242^{*+}	$\mathbf{0.3313}^{*+}$	$\mathbf{0.3313}^{*+}$	0.3181	0.3183
Average	0.3285 (3.89%, 4.58%,)	0.3312 (4.74%, 5.44%,)	0.3309 (4.65%, 5.35%,)	0.3162	0.3141

tive than the other two PRoc models according to our experiments.

In summary, a big challenge is to find an optimal value since the value space is very large without any constraints. It is very time-consuming to try every possible values in the relevance feedback process. In order to narrow the value space of $wSize$, we attempt to find a rule to direct the searching of optimal values. Based on our experimental results, we conjecture that there are two factors affecting the choice of $wSize$: the average document length (ADL) and the size of a collection. Intuitively, if the average document length is large, it is more likely to have more than one topic in a document which leads to involve more irrelevant terms. Besides, as the increase of the size of collection, it is more likely to bring irrelevant documents into the feedback process.

In order to minimize the negative influence of noise, the values of $wSize$ should be relatively small especially when the ADL or collection size is large. Only the closest terms will be considered to avoid the selection of irrelevant terms. In our experiments, this rule is supported by some evidence. The ADL of Disk4&5 is 334 and there are only 528,155 documents in this collection. The performance of all the PRoc models is not affected by $wSize$ so extensively as that on the other collections. When there are only 10 expansion terms, the optimal value of $wSize$ can be as large as 1500. On WT10G, which has 1,692,096 documents and an average document length of 426, the optimal $wSize$ values are larger than 50 but smaller than 200. For WT2G, even it has less documents (247,491) than other collections, the optimal $wSize$ values are in a range of (30, 50) because of its long ADL (722). GOV2 is the largest collection with 25,178,548 documents in our experiments, and its ADL(679) is only smaller than that of WT2G. As a result, the optimal values of $wSize$ for GOV2 is the smallest one. It is always 10 in our case. In summary, we can use this rule to narrow the search space of optimal $wSize$ values. If a collection has plenty of documents or its ADL is large (e.g., more than 700), it is always good for us to start from 20 or smaller. Otherwise, we can try a larger starting value like 50 or more.

5.4 Comparison with PRM

We also compare our proposed model with the recently developed position relevance model (PRM) [16], which is an extension of the relevance model. In particular, PRM takes into account term positions and proximity with the intuition that words closer to query words are more likely to

Table 7: PRoc compares with the classic Rocchio's model, RM3, PRM1 and PRM2 on Tera06 dataset. The bold phase style means that it is the best result.

	PRoc3	Rocchio	RM3	PRM1	PRM2
MAP	0.3283	0.3156	0.3131	**0.3322**	0.3319
P@10	**0.5800**	**0.5800**	0.5043	0.5306	0.5490
P@30	**0.5260**	0.5167	0.4660	0.4884	0.4871
P@100	**0.3756**	0.3664	0.3576	0.3671	0.3741

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 [2] topics on the GOV2 collection for testing as Lv. *et al.* did in [16]. 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 [16], parameter μ in the Dirichlet smoothing is set to an optimal value of 1500, and we set b in our basic model, BM25, empirically to 0.3 [44]. 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, it is selected to make this comparison. There are two versions of PRM, PRM1 and PRM2. The results of RM3, PRM1 and PRM2 are directly from [16].

In Table 7, PRoc3 outperforms the classic Rocchio's model and RM3 significantly in terms of the MAP metric, and it is only slightly inferior to PRM1 and PRM2 by 1.19% and 1.1% respectively. On the P@10, P@30 and P@100 metrics, PRoc3 obtains the best results over all the other four models and outperforms RM3, PRM1 and PRM2 significantly. All these significant tests are based on the Wilcoxon matched-pairs signed-ranks test at the 0.05 level. This shows that the retrieval accuracy of our proposed model is better than that of the PRF models in the language modeling framework in this case. Since the results of PRM1 and PRM2 are optimized, it is reasonable to state that our propose model is at least comparable to the most recent progress.

6. CONCLUSIONS AND FUTURE WORK

In this paper, a novel feedback model `PRoc` is proposed by incorporating proximity information into the classic Rocchio's model. Specifically, we model the statistics of expansion terms and their proximity relationship with query terms by introducing a new concept ptf. Three proximity measures, namely window-based method, kernel-based method and the HAL method, are then proposed for evaluating the relationship between expansion terms and query terms. The corresponding PRoc models based on these measures, `PRoc1`, `PRoc2` and `PRoc3`, are evaluated extensively on four standard TREC collections. In general, PRoc is very effective and outperforms the state-of-the-art feedback models in different frameworks. Comparing the three variants of PRoc, PRoc3 is more effective and robust than PRoc1 and PRoc2. Meanwhile, our proposed PRoc is at least competitive to the most recent work, PRM. Additionally, we carefully analyze the influence of the parameter of $wSize$, and an empirical rule to narrow the value space of the window size is suggested.

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

In the future, we will try to discover more about how to effectively adapt proximity into the probabilistic retrieval models. Another possible research direction is to find the exact relationship between the window size factor and the information of collections, e.g., the length distribution of documents. It is also interesting to apply our work to other retrieval frameworks, like DFR or the language modeling framework.

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 four anonymous reviewers for their thorough review comments on this paper.

8. REFERENCES

[1] James Allan, Margaret E. Connell, W. Bruce Croft, Fangfang Feng, David Fisher, and Xiaoyan Li. INQUERY and TREC-9. In *TREC*, 2000.

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

[3] Stefan Büttcher, Charles L. A. Clarke, and Brad Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *Proceedings of the 29th annual international ACM SIGIR conference*, SIGIR '06, pages 621–622, New York, NY, USA, 2006. ACM.

[4] G. Romano 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.

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

[6] Kevyn Collins-Thompson. Reducing the risk of query expansion via robust constrained optimization. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 837–846, New York, NY, USA, 2009. ACM.

[7] Fernando Diaz and Donald Metzler. Improving the estimation of relevance models using large external corpora. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference*, pages 154–161, New York, NY, USA, 2006. ACM.

[8] Ben He, Jimmy Xiangji Huang, and Xiaofeng Zhou. Modeling term proximity for probabilistic information retrieval models. *Inf. Sci.*, 181(14):3017–3031, 2011.

[9] Xiangji Huang, Yan Rui Huang, Miao Wen, Aijun An, Yang Liu, Josiah Poon. applying data mining to pseudo-relevance feedback for high performance text retrieval. In *IEEE ICDM 2006*, page 295–306, 2006.

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

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

[12] Ruth Ann Atchley Kevin Lund, Curt Burgess. Semantic and associative priming in high-dimensional semantic space. In *Proceedings of the 17th CogSci*, pages 660–665, 1995.

[13] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference*, SIGIR '01, pages 120–127, New York, USA, 2001. ACM.

[14] Yuanhua Lv and ChengXiang Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1895–1898, New York, NY, USA, 2009. ACM.

[15] Yuanhua Lv and ChengXiang Zhai. Positional language models for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference*, SIGIR '09, pages 299–306, New York, NY, USA, 2009. ACM.

[16] Yuanhua Lv and ChengXiang Zhai. Positional relevance model for pseudo-relevance feedback. In *Proceeding of the 33rd SIGIR conference*, SIGIR '10, pages 579–586. ACM, 2010.

[17] Donald Metzler and W. Bruce Croft. A markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual International ACM SIGIR Conference*, pages 472–479, New York, NY, USA, 2005. ACM.

[18] Donald Metzler and W. Bruce Croft. Latent concept expansion using markov random fields. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference*, pages 311–318, New York, NY, USA, 2007. ACM.

[19] Donald Metzler, Jasmine Novak, Hang Cui, and Srihari 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.

[20] Vassilis Plachouras and Iadh Ounis. Multinomial randomness models for retrieval with document fields. In *Proceedings of ECIR*, pages 28–39, 2007.

[21] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference*, pages 275–281, New York, NY, USA, 1998. ACM.

[22] M. Porter. An algorithm for suffix stripping. *Program*, 14:130–137, 1980.

[23] Karthik Raman, Raghavendra Udupa, Pushpak Bhattacharyya, and Abhijit Bhole. On improving pseudo-relevance feedback using pseudo-irrelevant documents. In *ECIR*, pages 573–576, 2010.

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

[25] Stephen E. Robertson. On term selection for query expansion. *Journal of Documentation*, 46:359–364, January 1991.

[26] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Mike Gatford, and A. Payne. Okapi at TREC-4. In *TREC*, 1995.

[27] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *TREC*, pages 109–126, 1994.

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

[29] Gerald Salton. *The SMART Retrieval System.* Prentice Hall, New Jersey, 1971.

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

[31] Ruihua Song, Michael Taylor, Ji-Rong Wen, Hsiao-Wuen Hon, and Yong Yu. Viewing term proximity from a different perspective. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen White, editors, *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 346–357. Springer Berlin / Heidelberg, 2008.

[32] Tao Tao and ChengXiang Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169, New York, NY, USA, 2006. ACM.

[33] Tao Tao and ChengXiang Zhai. An exploration of proximity measures in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference*, SIGIR '07, pages 295–302, New York, USA, 2007. ACM.

[34] Olga Vechtomova and Ying Wang. A study of the effect of term proximity on query expansion. *Journal of Information Science*, 32(4):324–333, August 2006.

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

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

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

[38] Zheng Ye, Ben He, Xiangji Huang, and Hongfei Lin. Revisiting rocchio's relevance feedback algorithm for probabilistic models. pages 151–161. AIRS, 2010.

[39] Zheng Ye, Xiangji Huang, and Hongfei Lin. A bayesian network approach to context sensitive query expansion. In *SAC*, pages 1138–1142, 2011.

[40] Zheng Ye, Jimmy Xiangji Huang, and Hongfei Lin. Finding a good query-related topic for boosting pseudo-relevance feedback. *Journal of the American Society for Information Science and Technology (JASIST)*, 62(4):748-760, 2011.

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

[42] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410. ACM, 2001.

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

[44] Jiashu Zhao, Jimmy Xiangji Huang, and Ben He. CRTER: using cross terms to enhance probabilistic information retrieval. In *Proceedings of the 34th international ACM SIGIR conference*, SIGIR '11, pages 155–164, New York, USA, 2011. ACM.

[45] Jinglei Zhao and Yeogirl Yun. A proximity language model for information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference*, SIGIR '09, pages 291–298, New York, USA, 2009. ACM.