A Study of Document Weight Smoothness in Pseudo Relevance Feedback

Peng Zhang[1], Dawei Song[1], Xiaochao Zhao[2], and Yuexian Hou[2]

[1] School of Computing, The Robert Gordon University, United Kingdom
[2] School of Computer Sci & Tec, Tianjin University, China
{p.zhang1,d.song}@rgu.ac.uk;{0.25eye,krete1941}@gmail.com

Abstract. In pseudo relevance feedback (PRF), the document weight which indicates how important a document is for the PRF model, plays a key role. In this paper, we investigate the smoothness issue of the document weights in PRF. The term *smoothness* means that the document weights decrease smoothly (i.e. gradually) along the document ranking list, and the weights are smooth (i.e. similar) within topically similar documents. We postulate that a reasonably smooth document-weighting function can benefit the PRF performance. This hypothesis is tested under a typical PRF model, namely the Relevance Model (RM). We propose a two-step document weight smoothing method, the different instantiations of which have different effects on weight smoothing. Experiments on three TREC collections show that the instantiated methods with better smoothing effects generally lead to better PRF performance. In addition, the proposed method can significantly improve the RM's performance and outperform various alternative methods which can also be used to smooth the document weights.

Key words: Pseudo relevance feedback, Document weight smoothness, Query language model, Relevance Model

1 Introduction

Pseudo relevance feedback (PRF) assumes that the top n (e.g., 30) documents in the first-round retrieval are all relevant to the query. Due to its automatic manner and effective performance, PRF has been widely applied in information retrieval (IR), where the PRF (i.e., top n) documents are often used to derive a new query model that expands the original query [5, 8]. The document weight, which represents the weight of a PRF document in the query expansion (QE) model, is a key factor for the QE performance [8].

In this paper, we investigate the smoothness issue of document weights in QE. The term *smoothness* in this paper does not refer to the smoothness of document language models [11, 14, 9]. Instead, it is with respect to the PRF document weights used in QE and means that, firstly, the document weights decrease smoothly (i.e. gradually or slowly) from the top-ranked document to the subsequent ones along the rank list, and secondly, the document weights are smooth within the topically similar documents, i.e., the topically similar

documents should have similar weights. We postulate that a reasonably smooth document weighting function (weighting function for short) can benefit the QE performance. First, it can reduce the risk when some topmost-ranked documents with very high weights are not truly relevant. Second, considering the inter-document similarity can make smoother the conventional weighting functions which often take into account the query-document similarity only.

We test the above hypothesis under the Relevance Model (RM) [5], which is a typical language model based QE approach [8]. In RM, effectively the document weight consists of two components: the document relevance score and the document prior. The former represents the initial document relevance probability, while the latter is the prior probability of selecting the corresponding document to estimate the RM. In implementation (e.g. in the RM1 [5]), the document prior is set to be uniform, and the document relevance score is the query-likelihood (QL) [11, 14, 8]. Empirical evidence (see Section 2) shows that the QL scores decrease rapidly along the topmost-ranked $k(k < n)$ documents, where the truly relevant ones, however, are often quite randomly distributed. Moreover, the QL scoring function only considers the query-document similarity but ignores the inter-document similarity.

In the literature, various methods have been proposed to smooth the document relevance score or revise the document prior, thereby adjusting the document weights. Based on the clustering hypothesis [12], the score regulation method [2–4] forces the topically related documents to have similar relevance scores. In a similar manner, the graph-based smoothing framework proposed in [9] can also smooth the document relevance scores. To the best of our knowledge, neither of the above methods has been used to smooth the relevance scores for query expansion. Moreover, they do not explicitly consider the score smoothness along the document rank list. As for revising the document prior, the rank-related prior was proposed in [6] by utilizing the document rank and document length. This method, however, does not consider the inter-document similarity.

In this paper, we propose a two-step document weight smoothing method to obtain smoother weighting functions. The first step is to smooth the weights of the topmost documents in order to prevent the document weights from dropping sharply along the rank list. The second step aims to further smooth the document weights of all the PRF documents, by considering the inter-document similarity. Specifically, we allocate the weights of topmost-ranked documents to the lower-ranked documents which are statistically similar to the topmost ones. In this step, different weight allocation strategies as well as different similarity measures are considered and analyzed in terms of their effects on document weight smoothness, thus instantiating several smoothing methods each with different smoothing effect. Experiments on three TREC collections show that the methods with better smoothing effects generally give better QE performance. In addition, the proposed smoothing method can significantly improve the performance of the RM, and outperform three comparative methods, i.e., the score regulation approach [2, 3], the graph-based smoothing approach [9] and the rank-related prior approach [6], for revising the document weights in the RM.

2 Document Weight and Its Smoothness

In this section, we start with a description of the Relevance Model (RM) and a re-formulation of the model for explicitly applying revised (e.g. smoothed) weighting functions in it. We then provide empirical evidence, showing that the query-likehood scores (as RM's document weights) are not reasonably smooth along the rank list.

2.1 The Relevance Model (RM)

For each given query $q = (q_1, q_2, \cdots, q_m)$, based on the corresponding PRF document set M ($|M| = n$), the Relevance Model (RM) [5] estimates an expanded query model:

$$p(w|\theta_R) = \sum_{d \in M} p(w|\theta_d) \frac{p(q|\theta_d)p(\theta_d)}{\sum_{d' \in M} p(q|\theta_{d'})p(\theta_{d'})} \tag{1}$$

where $p(w|\theta_R)$ is the estimated relevance model[1]. A number of terms with top probabilities in $p(w|\theta_R)$ will be used to estimate the query expansion (QE) model (i.e. the expanded query model). In Equation 1, $p(w|\theta_d)$ is the probability of a term w in the language model θ_d for a document d, $p(\theta_d)$ is d's prior probability, and $p(q|\theta_d)$ is the query likelihood (QL) [11, 14]:

$$p(q|\theta_d) = \prod_{i=1}^{m} p(q_i|\theta_d) \tag{2}$$

In RM, the weighting function is:

$$f(d, q) = \frac{p(q|\theta_d)p(\theta_d)}{\sum_{d' \in M} p(q|\theta_{d'})p(\theta_{d'})} \tag{3}$$

where the QL relevance score $p(q|\theta_d)$ and document prior $p(\theta_d)$ are integrated to form the document weight. The $f(d, q)$ plays a key role in RM since it distinguishes the RM from a mixture of document language model (say $\sum_{d \in M} p(w|\theta_d)$).

To apply revised weighting functions under the RM framework, we re-formulate the RM as:

$$p(w|\tilde{\theta}_R) = \sum_{d \in M} p(w|\theta_d)\tilde{f}(d, q) \tag{4}$$

where $\tilde{f}(d, q)$ denotes any revised document-weighting function that satisfies $\sum_{d \in M} \tilde{f}(d, q) = 1$, and different $\tilde{f}(d, q)$ will derive different QE models.

2.2 Smoothness of QL as the Document Weight

In the RM [5, 8], since the document prior $p(\theta_d)$ is assumed to be uniform, it turns out that the weighting function is the normalized query likelihood (QL):

$$f(d, q) = f_{QL}(d, q) = \frac{p(q|\theta_d)}{\sum_{d' \in M} p(q|\theta_{d'})} \tag{5}$$

[1] This formulation is equivalent to RM1 in [5], but some notations are slightly different. We adopt the similar notations used in the recent work [8, 2] related to RM.

query id	$f_{QL}(d_1)/r$	$f_{QL}(d_2)/r$	$f_{QL}(d_3)/r$	$f_{QL}(d_4)/r$
#151	0.206/0	0.167/1	0.106/1	0.064/0
#152	0.153/0	0.097/1	0.085/0	0.075/1
#153	0.232/0	0.185/1	0.103/1	0.090/1

Table 1. Topmost $k(k = 4)$ documents' QL weights $f_{QL}(d_i)$ and the relevance judgements ($r = 1$ means truly relevant, while $r = 0$ means non-relevant.)

Data	Queries	$Mf_{QL}(d_1)/Mr$	$Mf_{QL}(d_2)/Mr$	$Mf_{QL}(d_3)/Mr$	$Mf_{QL}(d_4)/Mr$
WSJ8792	151-200	0.256/0.600	0.123/0.580	0.084/0.560	0.063/0.440
AP8889	151-200	0.235/0.580	0.104/0.560	0.075/0.560	0.057/0.480
ROBUST2004	601-700	0.252/0.650	0.125/0.490	0.085/0.500	0.063/0.480

Table 2. Topmost $k(k = 4)$ documents' mean QL weights $Mf_{QL}(d_i)$ and the mean relevance judgements Mr, with respect to all the queries.

The normalized QL scores are called as QL weights in this paper. From Equation 5, it turns out that the QL weights do not take into account the inter-document similarity. Therefore, we just present the empirical evidence showing that the QL weights are not reasonably smooth along the PRF rank list. We start with a small example on one TREC collection, and then present more statistical data on three TREC collections. In both cases, for each query, the top $n = 30$ documents are selected for the PRF documents. Note that the following data correspond to the topmost $k(k < n)$ documents. Therefore, the sum of QL weights of these topmost documents for each query will not necessary be 1.

The data in Table 1 are from the WSJ8792 collection and three queries. For each query, the QL weights of the topmost $k = 4$ documents are listed. As shown in Table 1, the QL weights decrease rapidly along the rank list, e.g., for all three queries, the weights of d_1 are about twice the d_3's weights and three times the d_4's weights. All the d_1s, however, are non-relevant.

Next, we provide more statistical data about the topmost 4 documents, denoted as d_i $(1 \leq i \leq 4)$ for different sets of queries from three TREC collections. First, we define two statistics:

$$Mf_{QL}(d_i) = \frac{1}{|Q|} \sum_{q \in Q} f_{QL}(d_i, q) \quad and \quad Mr(d_i) = \frac{1}{|Q|} \sum_{q \in Q} r(d_i, q) \tag{6}$$

where Q denotes the set of all involved queries, $|Q|$ is the number of queries, and $r(d_i, q) = 1$ if d_i is truly relevant to q, and 0 otherwise. The $Mf_{QL}(d_i)$ computes the d_i's mean QL weight, and the $Mr(d_i)$ denotes d_i's mean relevance judgement. The values of these two statistics are summarized in Table 2, which shows that the mean QL weights drop rapidly along the topmost 4 documents. The truly relevant documents, however, are rather randomly distributed, since the mean relevance judgements decrease quite slowly. This indicates that the QL weights are not reasonably smooth on these collections and queries.

3 Two-step Weight Smoothing

Now, we propose a two-step weight smoothing method, in which the first step is to smooth the sharply dropping weights within the topmost-ranked documents,

and the second step is to allocate the weights in the topmost-ranked documents to the lower-ranked documents, based on the similarity between these two parts.

3.1 Smoothing Topmost Weights

Recall that in Section 2.2, we presented the initial evidence that document weights drop rapidly along the topmost-ranked $k(k < n)$ documents. To solve this problem, our basic idea is to smooth every adjacently-ranked document pair subsequently, along the rank list. Specifically, given the topmost document list $d_1 d_2 \cdots d_k$, set $\tilde{f}(d_1, q) = f(d_1, q)$, and then from the document index $i = 1$ to $k-1$, smooth $\tilde{f}(d_i, q)$ and $f(d_{i+1}, q)$ as follows:

$$\tilde{f}(d_i, q) \leftarrow \tilde{f}(d_{i+1}, q) \leftarrow avg(\tilde{f}(d_i, q), f(d_{i+1}, q)) \tag{7}$$

The average operation (i.e. avg) can reduce the difference between d_i's weight and d_{i+1}'s weight subsequently. For example, consider the document weights of the query 151 in Table 1. The strategy is to first change $\tilde{f}(d_1, q)$ (0.2060) and $f(d_2, q)$ (0.1670) to their average weight 0.1865, then set the $\tilde{f}(d_2, q)$ (0.1865) and $f(d_3, q)$ (0.1060) to the average weight 0.1462, finally revise the $\tilde{f}(d_3, q)$ (0.1462) and $f(d_3, q)$ (0.0640) as the average weight 0.1051. The revised weights for these four documents are 0.1865, 0.1462, 0.1051 and 0.1051, which are smoother than the original weights, in the sense that the d_1's weight has been reduced, while the weights of d_2 and d_3 have been relatively improved.

The above strategy can retain the weight sum of topmost k documents, making the weight sum of all PRF documents to be always 1. Actually, we also considered smoothing the original weighting function by interpolating it with a uniform weighting function which assigns the same weight for every document. However, one more parameter (i.e., the interpolation coefficient) was required to control the smoothing, and according to our prior study the experimental results were not so good.

3.2 Improving Lower Weights

In this step, we aim to allocate the weights of topmost-ranked $k(k<n)$ documents to the lower-ranked documents, according to the similarity between these two parts of documents. This is not only to further smooth the document weights, but also to improve the ranks [2] of those documents which are truly relevant but have lower weights. Recall that usually the topmost-ranked documents (e.g, the first 5 documents) are more likely to be truly relevant, since the corresponding retrieval precision (e.g., P@5) is often relatively higher compared with the average precision of all the PRF documents. According to the clustering hypothesis [12], the weight allocation methods, in which the allocation is actually based on the similarity value with respect to the topmost-ranked documents, could boost the weights of the truly relevant documents which may have lower initial weights. In the following, we present two weight allocation methods (WAs) with

[2] Here, we assume that the higher rank corresponds to the higher weight.

different smoothing effects. Note that in the formulation of WAs, $\tilde{f}(d,q)$ is the weighing function obtained from the previous step (see Section 3.1).

Linear Weight Allocation (LWA) To see the basic idea, let us consider one topmost-ranked document d_t, and a lower-ranked document d_l. Our basic idea is to keep d_t's weight unchanged, and meanwhile improve d_l's weight based on the similarity between d_t and d_l, which is measured by $sim(d_l, d_t)$ [3]. Specifically, LWA lets d_l have $(1 - sim(d_l, d_t))$ proportion of its own weight and $sim(d_l, d_t)$ proportion of d_t's weight, and the allocation can be formulated as:

$$\tilde{f}_{LWA}(d_l, q) = (1 - sim(d_l, d_t))\tilde{f}(d_l, q) + sim(d_l, d_t)\tilde{f}(d_t, q) \tag{8}$$

where $\tilde{f}_{LWA}(d_l, q)$ is the LWA weight for the d_l. For the d_t, LWA retains its own weight, meaning that $\tilde{f}_{LWA}(d_t, q) = \tilde{f}(d_t, q)$. Therefore, the Equation 8 can also represent the LWA weight of d_t due to the fact that $sim(d_t, d_t) = 1$.

Next, if considering all the k topmost documents, for any PRF document d, we have

$$\tilde{f}_{LWA}(d, q) = \frac{1}{Z} \times \sum_{d_t \in M_t} (1 - sim(d, d_t))\tilde{f}(d, q) + sim(d, d_t)\tilde{f}(d_t, q) \tag{9}$$

where $\tilde{f}_{LWA}(d, q)$ denotes the LWA weighting function, Z is the normalization factor, and the M_t is the set of the topmost k documents.

Nonlinear Weight Allocation (NLWA) In addition to LWA, we propose a nonlinear version of weight allocation, called NLWA, which has the same basic idea as LWA. The difference between NLWA and LWA is the specific allocation strategy. For a topmost document d_t and a lower one d_l, the NLWA weights are formulated as:

$$\tilde{f}_{NLWA}(d, q) = \sqrt{\tilde{f}(d, q)}\sqrt{\tilde{f}(d_t, q)}sim(d, d_t) \tag{10}$$

where d can be d_t or d_l. In a similar manner as for the LWA, if considering all the topmost documents, for any PRF document d, the NLWA weighting function is:

$$\tilde{f}_{NLWA}(d, q) = \frac{1}{Z} \times \sum_{d_t \in M_t} \sqrt{\tilde{f}(d, q)}\sqrt{\tilde{f}(d_t, q)}sim(d, d_t) \tag{11}$$

Analyzing WAs' Effects on Smoothing Generally, the LWA weights are smoother than the NLWA weights. For simplicity, our analysis on WAs' smoothing effects is only based on any two documents d_t and d_l, where d_t is ranked higher than d_l. Let $s = sim(d_t, d_l)$, $\tilde{f}(l) = \tilde{f}(d_l, q)$ and $\tilde{f}(t) = \tilde{f}(d_t, q)$, where $0 < \tilde{f}(l) < \tilde{f}(t)$. According to Equation 8, we have $\tilde{f}_{LWA}(l) = (1-s)\tilde{f}(l) + s\tilde{f}(t)$, and from the Equation 10, we can obtain $\tilde{f}_{NLWA}(l) = s\sqrt{\tilde{f}(l)\tilde{f}(t)}$. Then, the quotient of d_l's LWA weight and d_l's NLWA weight is:

$$\frac{\tilde{f}_{LWA}(l)}{\tilde{f}_{NLWA}(l)} = \frac{1-s}{s}\sqrt{\frac{\tilde{f}(l)}{\tilde{f}(t)}} + \sqrt{\frac{\tilde{f}(t)}{\tilde{f}(l)}} \tag{12}$$

[3] Generally, *sim* can be any similarity metric with values on [0, 1].

Since $\frac{1-s}{s}\sqrt{\frac{\tilde{f}(l)}{\tilde{f}(t)}} > 0$ and $\sqrt{\frac{\tilde{f}(t)}{\tilde{f}(l)}} > 1$, we can get:

$$\frac{\tilde{f}_{LWA}(l)}{\tilde{f}_{NLWA}(l)} > 1 \tag{13}$$

It turns out that d_l's LWA weight is larger than its NLWA weight. Since d_t's weight is unchanged in both LWA and NLWA, we can conclude that LWA makes the weight difference between d_t and d_l smaller than NLWA.

Similarity Measurements with different Smoothing Effects The similarity metric that we adopt is the Cosine similarity between the $tf \times idf$ vectors of two documents. Here, we set two specific options for the Cosine similarity: the first option (S1) is the similarity based on the document vectors with all the terms, while the second option (S2) is the similarity based on the document vectors with query terms removed. Since query terms often have high term frequency in the PRF documents, the similarity values in S1 are generally larger than those in S2. If similarity values are larger, the lower-ranked documents can have more weights allocated by the WAs. Therefore, the S1 can lead to smoother document weights than the S2.

Overall Smoothing Effect Analysis Since different similarity measurement options have different smoothing effects, it is necessary to investigate different combinations of the weight allocation method (LWA or NLWA) and the similarity measurement option (S1 or S2) for the PRF. Accordingly, we denote the four resulting methods as LWA_S1, LWA_S2, NLWA_S1 and NLWA_S2. With the same similarity option, LWAs' weights are generally more smooth than NLWAs' weights. On the other hand, the similarity option S1 can give smoother weighting function than the S2 if we use the same weight allocation (WA) method.

4 Experiment Evaluation

4.1 Evaluation Configuration

Collections The evaluation involves topics 151-200 on WSJ87-92 (173,252 documents) and AP88-89 (164,597 documents) in TREC disks 1 and 2, as well as topics 601-700 on ROBUST 2004 (528,155 documents) in TREC disks 4 and 5. The *title* filed of the topics are used as queries. The documents involved are related to a variety of texts, e.g., newswire and journal articles. Lemur toolkit 4.7 [10] is used for indexing and retrieval. All collections are stemmed using the Porter stemmer and a standard stop words list is removed during the indexing.

Evaluation Set-up The first-round retrieval is carried out by a standard language model (LM), i.e., the query-likelihood (QL) model [14, 11]. LM is set as one of the baseline methods. The smoothing method for the document language model is the Dirichlet prior [14] with the fixed value 700. After the first-round retrieval, the top n ranked documents are selected as the pseudo relevance feedback (PRF) documents. Due to the limited space, only the results with respect to $n = 30$ PRF documents will be reported. Nevertheless, we have similar observations on other n (e.g., 50, 70, 90). Relevance Model (RM) in Equation 1,

is selected as the second baseline method, where the document prior is set as uniform. For all the involved query expansion (QE) models, The top 100 terms are selected for the QE terms.

Evaluation Procedure The aim is to test the query expansion performance of different weighting functions. Recall that different weighting functions in Equation 4 have different QE models. First, we evaluate the first-step of the proposed weight smoothing method, i.e., the smoothing topmost weights (STW) described in Section 3.1. Next, we compare different combinations of the weight allocation method (LWA or NLWA) and the similarity option (with query (S1) or without query (S1)).

We then compare the proposed approach with other three methods which can be used to adjust the document weights. They are: the score regulation (SR) approach [2–4], the graph-based smoothing framework [9], and the rank-related priors (RRP) [6]. The score regulation method is formulated as:

$$f^* = (I_n - \alpha D^{-1/2} W D^{-1/2})^{-1} y \tag{14}$$

where f^* is the optimal relevance score and y is the original relevance score (i.e., QL score). We use the normalized f^* (i.e., the sum is 1) as the weighting function for Equation 4. Under the smoothing framework [9], the DSDG method (i.e., smoothing relevance score with document graph) is formulated as:

$$s(q, d_u) = (1 - \lambda)\tilde{s}(q, d_u) + \lambda \sum_{v \in V} \frac{w(u, v)}{Deg(u)} s(q, d_v) \tag{15}$$

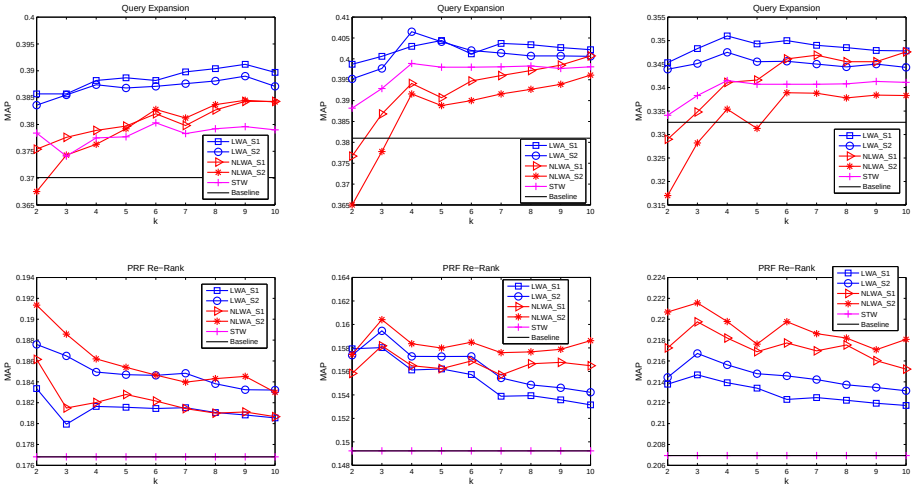
where $s(q, d_u)$ is the smoothed score for document d_u and $\tilde{s}(q, d_u)$ is d_u's original score. We use the normalized s as the corresponding weighting function for Equation 4. The rank-related priors [6] can be formulated as:

$$p(\theta_d) = \frac{1}{Z} \times \frac{\alpha + |d|}{\beta + Rank(d)} \tag{16}$$

where $|d|$ is the d's document length and $Rank(d)$ is d's rank. This prior $p(\theta_d)$ and the QL scores are integrated as the document weights in Equation 3.

Parameter Configuration For the proposed smoothing methods (i.e., STW and WAs), we tested different k in $[2,\ 10]$ with the increment 1. For the SR, we tuned three parameters: the α in $[0.1,\ 0.9]$ with the increment 0.1, the t^{-1} in $[0.1,\ 0.9]$ with the step 0.1, and the number of nearest neighbor kNN in $[5,\ 10]$ with the step 1. For the DSDG, we tuned two parameters: the λ in $[0.1,\ 0.9]$ with the increment 0.1 and the nearest neighbor kNN in $[5,\ 10]$ with the step 1. The iteration number is fixed to 3. Basically, the above parameter settings for both SR and DSDG are consistent with those in the original papers [2, 9]. The values of kNN are smaller than values in [2, 9], since we focus on the PRF task. As for the RRP, the α is set as 140 and the β is set as 50, where both values are the optimal values reported in [6].

Evaluation Metrics The Mean Average Precision (MAP), which reflects the overall ranking performance, is adopted as the primary evaluation metric. The Wilcoxon signed rank test is the measure of the statistical significance of the



(a) WSJ8792 (b) AP8889 (c) ROBUST2004

Fig. 1. The query expansion and re-ranking performance of different weight smoothing methods on three TREC collections. The baseline for the query expansion is the RM, while the baseline for the re-ranking is the PRF document rank by the LM. Note that only PRF documents are involved in the re-ranking performance evaluation.

improvement over baseline methods. For the original and the expanded query models, 1000 documents retrieved by the KL-divergence language model [10] are used for the performance evaluation.

4.2 Evaluation on Smoothing Topmost Weights

The aim is to test the performance of the smoothing topmost weights (STW) method described in Section 3.1. The results are reported in Figure 1, from which we can come up with the following observations. Firstly, the STW's performance increases before the k reaches a value ($k = 6$ on WSJ8792 and $k = 4$ on other collections), and then performance stabilizes. Secondly, for every k, STW outperforms the baseline (RM) on every collection. The best performance of STW is reported in Table 3, from which we can observe that STW outperforms RM by about 2.8%, 4.7%, and 2.7% on three collections, respectively.

4.3 Evaluation on Weight Allocation Methods

This set of experiments evaluates different combinations of the weight allocation method (LWA or NLWA) and the similarity measurement (S1 or S2) (see Section 3.2). The QE results of the four methods are summarized in Figure 1.

Generally, the results show that the smoother weighing functions generally give better results. Firstly, for every k, LWAs outperform the STW. This supports the effectiveness of the second step of the proposed smoothing framework. Secondly, in most cases, the LWA_S1, the most smooth method, gives the

MAP% (chg% over LM)	**WSJ8792**	**AP8889**	**ROBUST2004**
LM	31.25	30.43	29.15
RM	37.01 ($+18.4^{\alpha}$)	38.10 ($+25.2^{\alpha}$)	33.26 ($+14.1^{\alpha}$)
RRP	36.76 ($+17.6^{\alpha}$)	37.54 ($+23.4^{\alpha}$)	31.56 ($+8.2^{\alpha}$)
SR	38.51 ($+23.2^{\alpha\beta}$)	38.70 ($+27.1^{\alpha}$)	34.29 ($+17.6^{\alpha}$)
DSDG	38.26($+22.4^{\alpha}$)	39.44($+29.6^{\alpha\beta}$)	34.37($+17.9^{\alpha}$)
STW	38.03 ($+21.7^{\alpha}$)	39.89 ($+31.0^{\alpha\beta}$)	34.15($+17.1^{\alpha}$)
LWA	39.12 ($+25.2^{\alpha\beta}$)	40.44 ($+32.9^{\alpha\beta}$)	35.10 ($+20.4^{\alpha\beta}$)

Table 3. Overall query expansion results of different weight smoothing methods. Statistically significant improvements over LM and RM are marked with "α" and "β", respectively. Note that here LWA is actually the LWA_S1.

best performance, and LWAs generally outperform the NLWAs. Thirdly, each WA method with the S1 performs better than the WA with the S2. Specifically, the LWA_S1 outperforms the LWA_S2, and the NLWA_S1 outperforms the NLWA_S2. Finally, as for the parameter sensitivity of WAs , we can observe that NLWAs are more sensitive to different k values than LWAs.

Interesting observations can be made after we evaluate the re-ranking performance of the different WAs. The re-ranking performance is reported in the Figure 1, from which we can observe that all WAs outperform the baseline. This empirically demonstrates that WAs can improve the ranks of the truly relevant documents with lower initial weights (see the analysis in Section 3.2). In addition, we can see that the least smooth method (i.e. NLWA_S2) gives the best re-ranking performance. Moreover, the LWA_S2 performs much better than the LWA_S1, although the LWA_S1 is a smoother method. This indicates that the improved QE performance might be more due to the better smoothness of document weights than due to a better PRF rank list, which in turn emphasises the importance of the smoothness of document weights for the QE.

4.4 Comparison with Other Weight Smoothing Methods

Now, we compare the performance of LWA (i.e., LWA_S1) with those of other document weight revision methods, i.e., the score regulation (SR) [2, 3], the DSDG method [9] and the rank-related priors (RRP) [6]. The formulation and parameters configuration of SR, DSDG and PPR are described in Section 4.1. We report the best performance of the above three methods in Table 3.

Both SR and DSDG can outperform the RM, but not significantly on some collections. On the other hand, the LWA outperforms SR and DSDG, and improves the RM significantly on all three collections. It is probably because neither SR nor DSDG considers the document weight smoothness along the rank list. The main aim of SR and DSDG is to re-rank the documents. However, as discussed in the previous experiments (see Figure 1), a better PRF rank list may not guarantee a better QE performance.

For the RRP, we found that its performance (using $\alpha = 140$ and $\beta = 50$) is not so good. We think that this approach can help the RM become robust if a large number (e.g., 500) of PRF documents are involved, since it effectively depress the

weights of lower-ranked documents. However, if the number of PRF documents is relatively small (e.g., 30), we can observe that it can make the document weights less smoother, and hence possibly hurt QE retrieval performance.

4.5 Discussions

In the above experiments, we did not interpolate the expanded query model with the original query model, since we wanted to focus on the document weight smoothness issue. As observed from our experiments and also in [7], the QE performance is very sensitive to interpolation coefficient α. Actually, using a well-tuned α for the RM3 [4], the proposed weight smoothing method LWA can improve the RM3 by 4%-5% on the three TREC collections. However, the smoother weight smoothing methods (e.g., LWA) can not always have better QE performance. This also raises an important research question: how to define and control the weight smoothness degree for different queries? It is reasonable that different queries may need different degrees of the document weight smoothness for an optimal QE performance.

On the other hand, the score regulation method [2, 3] and the DSDG method under the graph-based smoothing framework [9] both target at re-ranking the documents. Recently, the portfolio theory has been adopted in [13] to derive an optimal document rank, by considering the document dependency into the probability ranking principle (PRP). However, for the query expansion task, as we have stressed before, a better PRF rank list may not guarantee a better QE performance. Therefore, how to further smooth the document weight after a good document rank has been obtained by a re-ranking method, becomes an important problem. We will investigate this issue in-depth in the future.

5 Conclusions and Future Work

We have proposed to study the document weight smoothness issue in query expansion (QE) based on PRF documents. We have also proposed a two-step document weight smoothing method, in which the first step is to smooth the sharply dropping weights along a small number of topmost-ranked documents, and the second step is to allocate the weights of the topmost-ranked documents to the lower-ranked ones, based on the inter-document similarity. Under the framework of the Relevance Model (RM), different document-weighting functions have been tested. The experiments on three TREC collections show that the smoother weighting functions derived by the proposed method have better QE performance. The proposed method, in particular the LWA, can significantly improve the RM's performance. Compared with other methods that can be used to revise the document weights, LWA also gives a better QE performance. We also would like to mention that LWA's good performance is because that it has

[4] The expanded query model by the RM1 can be interpolated by a original query model and then derive the RM3 [1]

better effect on weight smoothing than NLWA, although its re-ranking performance is not better than that of NLWA (see Figure 1). This also suggests the importance of the smoothness of document weights for the QE.

In the future, we will investigate how to adapt the smoothness degrees of document weights to individual queries, in order to obtain an optimal QE performance. We are also planning to derive methods to further smooth the document weights after a good document rank has been obtained by re-ranking methods. Furthermore, we will study the connection between the smoothness of document weights and the smoothness of document language models. Our goal is to build a formal and effective method for smoothing document weights not only under the RM framework but also for other QE models [8], to improve their performance.

Acknowledgments. We would like to thank Jun Wang, Leszek Kaliciak and anonymous reviewers for their constructive comments. This work is supported in part by the UK's EPSRC grant (No.: EP/F014708/2).

References

1. N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, D. Metzler, M. D. Smucker, T. Strohman, H. Turtle, and C. Wade. Umass at trec 2004: Novelty and hard. In *TREC '04*, 2004.
2. F. Diaz. Regularizing ad hoc retrieval scores. In *CIKM*, pages 672–679, 2005.
3. F. Diaz. Regularizing query-based retrieval scores. *Inf. Retr.*, 10(6):531–562, 2007.
4. F. Diaz. Improving relevance feedback in language modeling with score regularization. In *SIGIR*, pages 807–808, 2008.
5. V. Lavrenko and W. B. Croft. Relevance-based language models. In *SIGIR*, pages 120–127, 2001.
6. X. Li. A new robust relevance model in the language model framework. *Inf. Process. Manage.*, 44(3):991–1007, 2008.
7. Y. Lv and C. Zhai. Adaptive relevance feedback in information retrieval. In *CIKM*, pages 255–264, 2009.
8. Y. Lv and C. Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *CIKM*, pages 1895–1898, 2009.
9. Q. Mei, D. Zhang, and C. Zhai. A general optimization framework for smoothing language models on graph structures. In *SIGIR*, pages 611–618, 2008.
10. P. Ogilvie and J. Callan. Experiments using the lemur toolkit. In *TREC-10*, pages 103–108, 2002.
11. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281, 1998.
12. A. Tombros and C. J. van Rijsbergen. Query-sensitive similarity measures for information retrieval. *Knowl. Inf. Syst.*, 6(5), 2004.
13. J. Wang and J. Zhu. Portfolio theory of information retrieval. In *SIGIR*, pages 115–122, 2009.
14. C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*, pages 334–342, 2001.