Robust Query-specific Pseudo Feedback Document Selection for Query Expansion

Qiang Huang, Dawei Song, Stefan Rüger

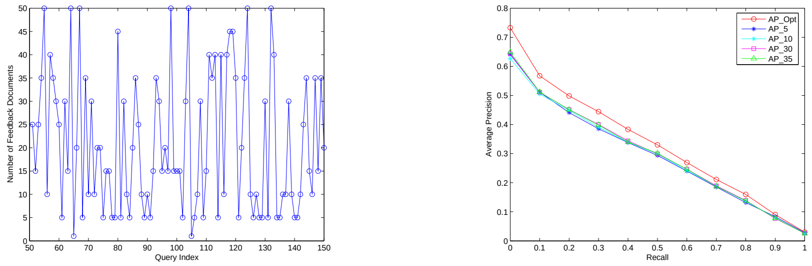
Knowledge Media Institute,
The Open University, UK
{q.huang,d.song,s.rueger}@open.ac.uk

Abstract. In document retrieval using pseudo relevance feedback, after initial ranking, a fixed number of top-ranked documents are selected as feedback to build a new expansion query model. However, very little attention has been paid to an intuitive but critical fact that the retrieval performance for different queries is sensitive to the selection of different numbers of feedback documents. In this paper, we explore two approaches to incorporate the factor of query-specific feedback document selection in an automatic way. The first is to determine the "optimal" number of feedback documents with respect to a query by adopting the clarity score and cumulative gain. The other approach is that, instead of capturing the optimal number, we hope to weaken the effect of the numbers of feedback document, i.e., to improve the robustness of the pseudo relevance feedback process, by a mixture model. Our experimental results show that both approaches improve the overall retrieval performance.

1 Introduction

To document retrieval, the pseudo relevance feedback tries to build an expanded query language model using the top-selected documents according to the initial retrieval results. Naturally, the top-ranked documents are assumed to be relevant to the user's query. In the process of building an expanded query model, traditional methods tend to select a fixed number ($\leqslant 50$, typically) of top-ranked documents as feedback, regardless of different queries. However, an intuitive but critical fact has long been ignored: the retrieval performance for a specific query is often sensitive to the selected number of feedback documents.

Figure 1(a) and 1(b) show the effects of different numbers, {5,10,15,20,25,30, 35,40,45,50}, of feedback documents by testing TREC query topics 51-150 (only title field) on collection AP88-90. Figure 1(a) shows the manually identified "optimal" (i.e., best performing) number of documents for each query, which is obviously not a constant value for different queries. A comparison of the retrieval performances between the expanded query language model using the query-specific optimal numbers of feedback documents (based on Figure 1(a)) versus other four expanded query language models using a fixed number of top-N ($N \in (5, 10, 30, 35)$) documents to all the queries. It turns out that the former can generate a large improvement in average precision over the others. Following



(a) The optimal number of feedback documents to each query

(b) Precision-recall curves (TREC topics 51-150 on Collection AP8890)

Fig. 1. The optimal number of feedback documents

this preliminary experiments, the question we are concerned about is: Is there an automatic method of selecting the feedback documents with respect to individual query?

There can be three directions towards finding a solution to the problem. The first is to build a model by finding the truly relevant documents in the top-ranked documents [8, 9] using a support vector machine (SVM) based semi-supervised method with the user's help. The second direction is to directly capture the optimal number of documents with respect to each query [1, 3, 11]. Some methods, such as computing a *clarity score* using Kullback-Leibler (KL) divergence [3] and using the maximum clarity score as the model-selection criterion [11]. The third direction is to build a mixture model combining several expanded query language models to weaken the effect of pseudo feedback document selection [6, 2].

In summary, all the aforesaid attempts try to address the problem of document/model selection for generating a new query model. However, in order to build an optimal expanded query language model, a fully automatic method, for either pursuing a single optimal model or combining multiple models, still remains an open and attractive topic. In this paper, we explore novel approaches incorporating the factor of query-specific feedback document selection in a fully automatic way, and apply the existing *clarity score* (CS) and present two new approaches respectively based on *discount cumulative gain* (DCG) and *mixture model* (MM) for the document retrieval.

2 Determination of the Query-specific Optimal Model

2.1 Clarity Score (CS)

In general, if the collection is large enough, it is often assumed that the distribution of words in the document collection is uniform. The model with uniform distribution is generally considered as the worst model for document retrieval because the importance of words to query can not be distinguished from each other. The clarity score defined here is the KL divergence of the expanded query language model M to the collection model M_{coll}, as shown in Equation 1.

$$clarity\ score = \sum_{w \in V} P(w|M) \log_2 \frac{P(w|M)}{P(w|M_{coll})} \tag{1}$$

where V is the word vocabulary for the collection. The smaller distance between the two models is assumed to imply a poor retrieval performance for the query. Based on this assumption, the clarity score can be used to predict the retrieval performance of an expanded query language model. The pseudo code below describes the application of the clarity score for selecting the optimal model from several query language models $M_i, (1 \leq i \leq m)$.

```
$for\ \ i\ \ =\ \ 1\ :\ m,$
    $CS_i = \sum_{w \in V}\ \ P(w|M_i) \log_2 \frac{P(w|M_i)}{P(w|M_{coll})}$
$end$
$M^* = max_{1 \leq i \leq m}\ \ CS_i$
```

The model corresponding to the maximum clarity score is chosen as the optimal model. The clarity method has a clear advantage that it does not require doing the actual retrieval. However, it can not guarantee that the selected model is the truly best performing one. On one hand, the words in the collection model may not distribute uniformly. On the other hand, even if the collection model had the uniform distribution, the larger divergence between a query language model and the collection model does not necessarily mean the query language model closer to the best model we expect.

2.2 Discount Cumulative Gain (DCG)

Compared with the *clarity score* measure, *discount cumulative gain* (DCG) is a more complex approach to measure the possible highly relevant documents. Unlike the binary measure, by which queries are judged relevant or irrelevant with regard to the query, the cumulative gain generally uses multiple graded relevance judgments [10, 4, 5, 7]. The cumulative gain based measure was summarized into two points: (1) highly relevant documents are more valuable than marginally relevant documents, (2) the lower the ranked position of a relevant document (of any relevant level), the less valuable it is for the user. The details are referred to [4]. In this paper, we apply the DCG to predicting the retrieval performance of a model. So we hope to select an "optimal" model by comparing the cumulative gains of each query language model. The cumulative gain is computed as below:

Collection: Given a query, collect the top 100 documents ranked after the initial retrieval.

Re-ranking: Re-rank the 100 documents based on 10 expanded query language models which is built by using $\{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$ top-ranked documents, respectively. Simultaneously, 10 rank lists of the 100 documents are respectively obtained as well.

Identification: Compute the summation of the order of a document in the 10 rank lists, so there are 100 values of the summation corresponding to the 100 documents. Select 16 documents as "pseudo" highly relevant documents whose summation values are smaller.

Label: Label the 16 selected documents (16 is an experience value) with four grades of ranking (also called gain value in [7]), namely $R = [4, 4, 4, 4, 3, 3, 3, 3, 2, 2, 2, 2, 1, 1, 1, 1]$

Computation: Compute the cumulative gain:

$$DCG_{M_i} = \sum_{j=1}^{16} \frac{2^{label(j)}-1}{\log_2(j+1)}$$

where $label(j)$ is the gain value associated with the label of the document at the j^{th} position of the ranked list. $log_2(j + 1)$ is a discounting function that reduces document's gain value as its rank increases [7].

In the process of computation, the relevance levels can be mapped to numerical values, with 4 corresponding to the highest level of relevance and 1 corresponding to the lowest level of relevance. The difference in gain values assigned to highly relevant and relevant documents changes the score of cumulative gain. The method of computing cumulative gain is almost same as that used in [7], in which a normalized discount cumulative gain (NDCG) averaged over the queries is used to evaluate the performance of the multiple nested ranker algorithm. In addition, the computation also means that the re-ranking is needed over all expanded query language models. The similar method using re-ranking over multiple models for model selection can also be found in [11], but our method only runs on the top 100 documents ranked by the initial retrieval rather than searching the whole collection of documents with each query model, as done in [11].

2.3 Mixture Models (MM)

The above two methods based on the *CS* and *DCG* aim to find the "optimal" model in the multiple models. In this section, we attempt to build a mixture model by combining all query language models rather than only selecting one. The application of mixture models is to bind all N models whatever the value of N is to a target model, aiming to smooth the effects from different models [6]. In the process of building a mixture model, the key step is to estimate the mixture weight of each model, as shown in Equation 2:

$$M_{opt} = \sum_{j} \lambda_j M_j \tag{2}$$

where $\sum_j \lambda_j = 1$. In [12, 6], an approach based on Kullback-Leibler (KL) distance was used to optimize the weights for mixture models. Here we briefly describe the optimization procedure, and the details can be found in [6].

$$D = \sum_{i} T(w_i) log \frac{T(w_i)}{M_{opt}(w_i)} \tag{3}$$

Table 1. Test collection and test topics.

Collection	Contents	# of docs	Size	Queries (topics)	# of Queries
AP88-90	Associated Press	242,918	0.73Gb	51-150	99
WSJ87-92	Wall Street Journal	173,252	0.51Gb	1-200	200
SJM91	San Jose Mercury News	90,257	0.29Gb	51-150	94

In equation 3, KL distance is computed between the target model T and the mixture model M_{opt}. In [6], a similar optimization was adopted as below:

$$H_\lambda(T|M_j) = -\sum_w T_w \log \sum_j (\lambda_j / \sum_j \lambda_j) M_{j,w} \tag{4}$$

In order to find the maximum of Equation 4, a derivation on λ_k is taken, and the derivation is set to be zero.

$$\frac{\partial H_\lambda}{\partial \lambda_j} = -\sum_w \frac{T_w M_{j,w}}{\sum_j \lambda_j M_{j,w}} + \frac{1}{\sum_j \lambda_j} = 0 \tag{5}$$

Suppose λ_k^n is the mixing weight of element k after n iterations of the algorithm. Then at the next iteration the weight should become:

$$\lambda_k^{n+1} \longleftarrow \sum_w \frac{T_w M_{j,w} \lambda_k^n}{\sum_j (\lambda_j^n / \sum_j \lambda_j^n) M_{j,w}} \tag{6}$$

Here, the optimization of the weight to each model is to make the mixture model best approximate the target model, so the selection of the target model is actually key to the final results that the mixture models can achieve.

In [6], Lavrenko used the mixture model to weaken the effect of selecting the number of feedback documents. Here, we exploit this idea in two different ways. Firstly, we select the original words distribution on the top 50 documents as the target model instead of a known relevant document as used in [6]. The reason is that [6] needs a relevant document, which is generally selected manually, to build the target model. Secondly, the model built by using the top 50 documents could be the worst model compared with less documents being used because there are more irrelevant information being included. If the performance of MM based on the top 50 documents is good, then it could mean less documents used will generate better result.

We have presented three approaches to deal with the problem caused by selecting the number of feedback documents. The first two approaches, respectively based on *CS* and *DCG*, try to select the “optimal” model. The *MM* aims to smooth this factor. In the next section, we will test their performances with two TREC topic sets on three TREC collections.

3 Data and Experiments

The experiments are run by testing two query topics (only using the title field) on three standard TREC data sets, whose statistic are summarized in Table 1.

Table 2. Results (Average Precisions) of different models

	The number of feedback documents										
	5	10	15	20	25	30	35	40	45	50	Optimal
AP8890	0.2829	0.2852	0.2863	0.2867	0.2886	0.2893	0.2888	0.2888	0.2862	0.2859	**0.3228**
SJM	0.2309	0.2303	0.2346	0.2339	0.2325	0.2335	0.2350	0.2342	0.2356	0.2346	**0.2727**
WSJ8792	0.3026	0.3065	0.3037	0.3026	0.3039	0.3042	0.3028	0.3031	0.3023	0.3021	**0.3356**

Table 3. The average precisions obtained by using three different approaches

	worst	best	Clarity score	Change over	Change over
	model	model		worst model(%)	best model(%)
AP8890	0.2829	0.2893	0.2863	1.2	-1
SJM	0.2303	0.2356	0.2328	1.1	-1.2
WSJ8792	0.3021	0.3065	0.3028	0.2	-1.2
	worst	best	Cumulative Gain	Change over	Change over
	model	model		worst model(%)	best model(%)
AP8890	0.2829	0.2893	0.2872	1.5	-0.7
SJM	0.2303	0.2356	0.2356	2.3	0
WSJ8792	0.3021	0.3065	0.3031	0.3	-1.4
	worst	best	Mixture Model	Change over	Change over
	model	model		worst model(%)	best model(%)
AP8890	0.2829	0.2893	0.2889	2.1	-0.1
SJM	0.2303	0.2356	0.2402	4.3	1.9
WSJ8792	0.3021	0.3065	0.3087	2.1	0.7

In our system, each expanded query language model is built by using Jelinek-Mercer linear interpolation between a query language model and the collection model, in which the query language model is modeled using maximum likelihood with the top-N, ($N \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$) documents. The expansion is generated by running the Lemur toolkit. In this paper, we build 10 baseline expanded query language models $M_i, (1 \leq i \leq 10)$. For each model, the top 100 words are selected according to their distribution $P(w|M_i)$ to form the expanded query. The linear combination coefficient is set to be 0.9 and μ is set to be 1000 for the retrieval process.

In Table 2, the average precision obtained by using 10 baseline expansion models are listed in the order of increasing number of feedback documents used. At the most right-hand side, the optimal average precisions are listed, which are obtained by manually selecting the optimal model to each query. Naturally, the optimal performance is much better than the baseline expansion models generated by applying a fixed number of feedback documents to all queries, and can be considered as the upper bound of the retrieval performance. To show the different characteristics of the proposed automatic approaches, in the rest of this section, we use three performance measures, i.e. average precision, average precision @30 docs and robustness via query-by-query comparison.

Table 3 shows the retrieval performances using the three approaches for the different collections, The "worst expansion model" and "best expansion model" respectively represent the model with the lowest and highest average precision among the 10 baseline expansion models as shown in Table 2. All three approaches give higher average precision than the "worst expansion model". The average precisions of the *CS* and *DCG* are slightly lower than the "best expansion model". The *CS* gives the lowest performance for all the three collections.

Table 4. The precisions @ 30 docs using three different approaches

	Best Expansion Model	Clarity Score	Cumulative Gain	Mixture Model
AP8890	0.4451	0.4411	0.4453	0.4455
SJM	0.2720	0.2626	0.2761	0.2762
WSJ8792	0.4002	0.3996	0.4062	0.4033

Table 5. Robust analysis to the retrieval performance on three collections

		vs. best expansion model			vs. worst expansion model		
		Better	Neutral	Worse	Better	Neutral	Worse
AP 88-90	Clarity Score	41	6	52	43	6	50
	Cumulative Gain	48	6	45	52	5	42
	Mixture Model	50	3	46	57	2	40
WSJ 87-92	Clarity Score	43	3	48	45	2	47
	Cumulative Gain	48	2	44	49	1	44
	Mixture Model	47	0	47	48	0	46
SJM 91	Clarity Score	96	6	98	98	5	97
	Cumulative Gain	101	2	97	110	2	88
	Mixture Model	99	3	98	110	2	88

The *MM* generates better results and even outperforms the "best expansion model" on two collections. As we discussed in Section 2.1, the *CS* simply measures the distance between a model and the collection model and it seems to fail in selecting the appropriate number of feedback document. On the other hand, the *MM* tries to combine the information from multiple models, which can help weaken the effect of the model selection.

In addition, we list the precisions @ 30 docs, where the *DCG* and *MM* perform better than the *CS*, and also outperform the best expansion model. This could be because the *DCG* takes into account the ranking of the relevant documents and MM combines the useful information from different models, and also smooth them by weighting scheme to weaken the effect of "noisy" information.

A robustness analysis is shown in Table 5. The baselines are the best expansion model and the worst expansion model with a fixed-number of feedback documents. We perform a comparison of the mean average precisions between each of the three methods and the two baseline models query by query. Here, the terms *better/neutral/worse* in Table 5 stand for the numbers of queries for which our approach gives a *better/neutral/worse* than the two baselines, respectively. We can observe the robustness of using *CS* is a little lower than the other two approaches. Furthermore, compared with the *CS*, both *DCG* and *MM* show more robust performance improvement. *DCG* improves the most number of queries' performance but hurts the least number of queries, thus is the most robust.

4 Conclusion

In this paper, we present three approaches to automatically determine the query-specific optimal number of pseudo feedback documents for query expansion. The *CS* and *DCG* are used to look for an optimal value to the number of feedback documents, and *MM* to reduce the effect of selecting the optimal number. The *MM* can combine the multiple expansion models instead of trying to capture the best one. Its advantage is that it not only makes use of more useful information, but also smooths "noisy" information. It is verified by our experimental results: the *MM* shows better effectiveness (average precision and precision @30) than the other two. Using *DCG* also shows promising result, especially in the query by query robustness analysis. Both *DCG* and *MM* outperform the *CS* in terms of both effectiveness and robustness. There is still a big gap between the performance of our proposed approaches and the upper bound average precision generated by the manually selected optimal model (as shown in Table 2). This means there is a plenty of room for further performance improvement. In the future, we will not only take into account the effect of selecting documents, but also terms as well, which are kept constant in our experiments.

Acknowledgement

This work is funded by the UK's Engineering Physical Sciences Research Council (EPSRC), grant number EP/E002145/1

References

1. K. Collins-Thompson and J. Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the SIGIR'07*, pages 303–310, 2007.
2. W. B. Croft. Combining approaches in information retrieval. In W. B. Croft, editor, *Advances in Information Retrieval: Recent Research from the CIIR*, pages 1–36, Boston, 2000. Academic Publishers.
3. S. Cronen-Townsend, Y. Zhou, and W. B. Croft. A language modeling framework for selective query expansion. Technical report, CIIR, 2004.
4. K. Jarvelin and J. Kekalainen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of SIGIR'00*, pages 41–48, 2000.
5. K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of ir techniques. *ACM Transaction on Information Systems*, 20:422–446, 2002.
6. V. Lavrenko. Optimal mixture models in ir. In *ECIR 2002, LNCS 2291*, pages 193–212, Berlin, 2002. Springer-Verlag.
7. I. Matveeva, C. Burges, T. Burkard, A. Laucius, and L. Wong. High accuracy retrieval with multiple nested ranker. In *Proceedings of SIGIR'06*, 2006.
8. M. Okabe, K. Umemura, and S. Yamada. Query expansion with the minimum relevance judgments. In *AIRS-2005*, pages 31–42, Korea, 2005.
9. T. Onoda, H. Murata, and S. Yamada. One class classification methods based non-relevance feedback document retrieval. In *The International Workshop on Intelligent Web Interaction*, pages 389–392, Hong Kong, 2006.
10. E. Voohees. Evaluating by highly relevant documents. In *Proceedings of SIGIR'01*, pages 74–82, 2001.
11. M. Winaver, O. Kurland, and C. Domshlak. Towards robust query expansion: Model selection in the language modeling framework (poster). In *Proceedings of SIGIR'07*, Amsterdam, 2007.
12. J. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt. Topic tracking in a news stream. In *Proceedings of DARPA Broadcast News Workshop*, pages 133–136, 1999.