Language Modeling Based Local Set Re-ranking using Manual Relevance Feedback

Manoj Kumar Chinnakotla and **Pushpak Bhattacharyya**
Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay
Mumbai, India
{manoj,pb}@cse.iitb.ac.in

Abstract

We present a novel approach to re-ranking documents using language modeling (LM) and manual relevance feedback (RF). The documents returned by an initial search algorithm, called the *Local Set*, is re-ranked based on manual relevance feedback using a ranking function modified to perform at the local set level. Instead of using the query independent collection model, which is too general, we use the query-specific local set, to model the background distribution. The resultant relevance model learns a more specific set of terms relevant to the query. We achieve better ranking performance than existing approaches that employ both LM and RF. We are guided by efficiency considerations and the need of new search paradigms like personalization, that require re-ranking of initial search results based on various criteria rather than launching a fresh search into the entire corpus.

1 Introduction

Relevance Feedback (RF) from the user is one of the important steps in Information Retrieval (Rocchio, 1971). It helps in bridging the gap between actual user information need and the query posed, which is typically short and ambiguous (on an average 2-3 words long). In RF, from the initial set of documents retrieved using the query, the user identifies a small number of documents that are relevant and supplies them to the IR system. These documents are then used to learn an updated model of relevance which is in-turn used to re-rank documents with improved precision and recall.

The Language Modeling (LM) approach to IR (Ponte and Croft, 1998; Lafferty and Zhai, 2001) models the language of each document D using a distribution over words $P(w|D)$. The query Q is assumed to be a sample from a *query relevance model* Θ_R - a distribution over words $P(w|\Theta_R)$ capturing the actual information need of the user. The documents are ranked in increasing order of divergence from the query relevance model. The divergence is measured using Kullback-Leibler (KL) Distance (Cover and Thomas, 1991). For any document D, the KL-Distance scoring function for ranking is given by:

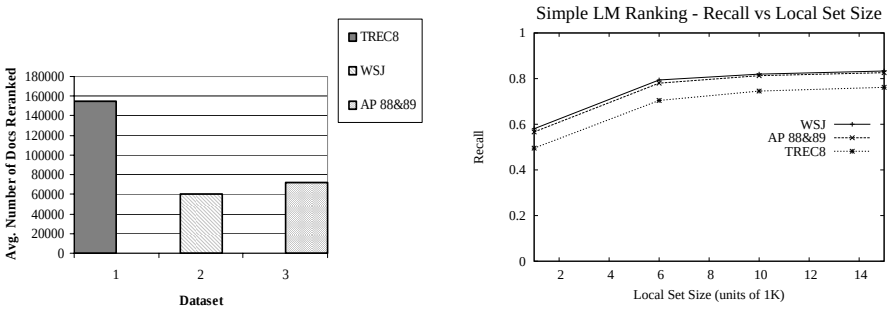
$$\begin{aligned} KL(\Theta_R, D) &= \sum_w P(w|\Theta_R)\, log\, \frac{P(w|\Theta_R)}{P(w|D)} \\ &\stackrel{rank}{\equiv} -\sum_w P(w|\Theta_R)\, log\, P(w|D) \end{aligned} \quad (1)$$

The above formulation casts the problem of retrieval into the problem of estimating distributions (query relevance model $P(w|\Theta_R)$ and document language model $P(w|D)$) from data and provides a framework for the application of various statistical estimation techniques.

Definition 1.1 (Local Set)**.** *Let $D_N = \{d_1, d_2, \ldots, d_N\}$ be the set of top N documents retrieved initially by some retrieval algorithm in response to a query Q. The set D_N is defined as the Local Set (LS) at rank N.*

Let $D_F = \{d_1^r, d_2^r, \ldots, d_k^r\}$ be the set of k documents selected by user from D_N and supplied as relevance feedback. Several approaches (Lafferty and Zhai, 2001; Zhai and Lafferty, 2001; Lavrenko and Croft, 2001) have been proposed for incorporating relevance feedback into the LM framework.

Proceedings of ICON 2009: 7th International Conference on Natural Language Processing
Macmillan Publishers, India. Also accessible from http://ltrc.iiit.ac.in/proceedings/ICON-2009



(a) Average number of documents re-ranked at the corpus level on three TREC datasets observed by us using Lemur Toolkit.

(b) Recall as a function of local set size for LM based Query Likelihood Ranking.

Figure 1: Some observations regarding Corpus Level Re-ranking Schemes and Initial Retrieval Algorithm on three standard TREC datasets.

In all these approaches, initially, the query relevance model Θ_R is based on the user query Q. Later, the relevance feedback D_F is used to learn an updated model query relevance model $\widehat{\Theta}_R$. The final query relevance model used for re-ranking is obtained by interpolating the updated model with the old query model using a parameter α.

$$\Theta_{Final} = \alpha \cdot \Theta_R + (1 - \alpha) \cdot \widehat{\Theta}_R \qquad (2)$$

Now, Θ_{Final} is used to re-rank the documents in the *entire corpus* using Equation 1 to output a new ranked list of results.

Traditional RF systems completely ignore the context, interests and activities of the user while re-ranking the corpus. Web Search has evolved from such straight jacketed "one size fits all" paradigms, where the same set of results are returned to users irrespective of their interests and backgrounds, to a more realistic paradigm of Personalized Web Search (Pitkow et al., 2002). In personalized web search, the results of a query are retrieved and customized based on the user profile like browsing history, e-mails, bookmarks *etc.,*. Pitkow *et. al* (Pitkow et al., 2002) propose two general strategies for personalization: *query augmentation* and *result re-ranking*. In query augmentation, the original query is augmented with terms to refine the search in the context of the user. In result re-ranking, the initial search results retrieved using the query are re-ranked assuming user profile, like desktop files, e-mails, browsing history *etc.,* as implicit relevance feedback. Recently, Teevan *et. al.* (Teevan et al., 2005) study personalization within a relevance feedback framework and report significant improvements over current search techniques. They locally re-rank search results at the client-end based on the user profile.

However, client-end re-ranking presents certain problems like unavailability of full text of documents (since multiple full text downloads are needed), corpus statistics *etc.,* Alternatively, the user profile could be maintained at the server end, requiring the user to logon during browsing to record the browsing history and context. This also gives some control over the personalization process to the user. *Google Personalized Search*[1] is an example of such scheme. However, the algorithm used for re-ranking search results is not known as it is proprietary.

The Personalized Search mechanisms require re-ranking the initial search results in the relevance feedback framework. Moreover, for the Web, which has grown to an enormous size, running into billions of pages, re-ranking at the corpus level is computationally expensive. Figure 1(a) shows our emperical observation regarding the average number of documents re-ranked at the corpus level using the Lemur Toolkit on three TREC datasets.

During the initial ranking phase, most standard ranking algorithms (like TF-IDF, LM Based Query Likelihood Ranking) succeed in fetching many relevant documents from the corpus albeit with a lesser precision. Figure 1(b) shows the recall at various LS sizes for LM Based Query Like-

[1] `http://www.google.com/psearch`

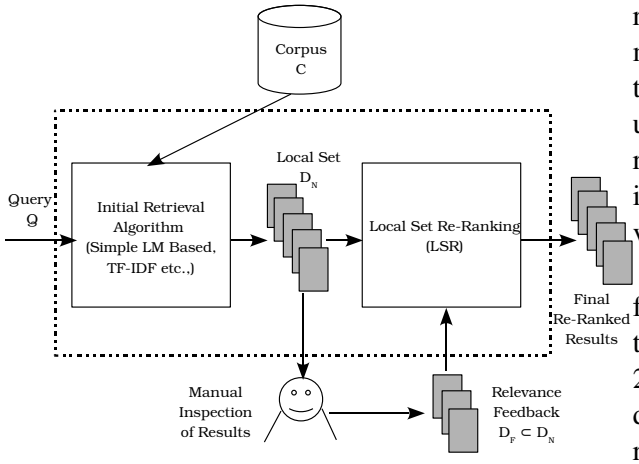


Figure 2: System Diagram of Local Set Re-ranking (LSR) Approach

lihood Ranking. It shows that the LS, at large ranks, usually contains a significant percentage of the total relevant documents.

Guided by the above considerations, instead of re-ranking the entire corpus, we re-rank the LS to arrive at an ordering that improves the average precision. Figure 2 shows the system diagram for our Local Set Re-ranking (LSR) scheme.

Definition 1.2 (Background Distribution)**.** *Let S be a subset of documents from the corpus C. A Background Distribution (BD) for S is a probability distribution $P(w|S_{BD})$ over words such that $P(w|S_{BD})$ represents the probability of finding the word w in any document randomly chosen from S.*

The collection distribution $P(w|C)$ is a BD for the entire collection C and is given by Equation 3 where $c(w;d)$ refers to the count of words in document d.

$$P(w|C) = \frac{\sum_{d \in C} c(w;d)}{\sum_{d \in C} |d|} \tag{3}$$

Re-ranking schemes operate based on identifying a set of terms which distinguish a relevant document from any random document in the given set. BD plays a key role in selection of such terms and thereby influences the document ranking. Generative models for feedback documents (Zhai and Lafferty, 2001) based on the LM approach, use the *query independent* collection distribution to model the background distribution of feedback documents. Due to this, they learn a query relevance model which is too general for re-ranking the LS. To make the query relevance model more specific in the context of re-ranking the LS, we modify the above generative model to use the BD of the LS which is *query specific*. As a result, we observe improvement over other existing re-ranking approaches based on manual relevance feedback.

The organization of the rest of the paper is as follows: In Section 2, we describe the generative mixture model formulation (Zhai and Lafferty, 2001) based on the LM framework for feedback documents and argue that it is too general for re-ranking the LS. In Section 3, we present our approach which modifies the above generative model to use a query specific Background Distribution. In Section 4, we present the experimental setup and results. Section 5 discusses the related work in this area. Section 6 concludes by highlighting some directions for future work.

2 Generative Mixture Model for Feedback Documents

In this section, we describe the generative mixture model for feedback documents that forms the basis for our work. It was proposed by Zhai and Lafferty (Zhai and Lafferty, 2001). The above mixture model is used to learn the query relevance model from the feedback documents.

The feedback document set consists of a combination of words - words that are relevant to the query, commonly occurring words from English and commonly occurring words from the domain of the corpus. For example, in a corpus of financial documents, other than the English function words (like *therefore*, *but*, *however etc.,*), a random document is also likely to contain terms commonly used in the domain like *currency*, *credit*, *debit*, *risk*, *stocks etc.,*

Let $P(w|C)$ be the *collection distribution* as defined in Equation 3. As explained earlier, the collection distribution serves as a BD for the entire corpus and represents the probability of finding the word in any random document in the corpus. Words which occur commonly across many documents in the corpus (function words in English, domain-specific common words) have higher probabilities in $P(w|C)$.

The generative mixture model for feedback documents is shown in Figure 3(a). Each feedback document is generated randomly by choosing the relevance distribution Θ_R with probability λ and

Number: 401 **Title:** foreign minorities, Germany **Description:** What language and cultural differences impede the integration of foreign minorities in Germany?

Table 1: Query 401 from TREC8 Collection.

picking a word w with probability $P(w|\Theta_R)$ or choosing the background collection distribution $P(w|C)$ with probability $1-\lambda$ and picking a word with probability $P(w|C)$. Hence, the observed distribution $P(w|D_F)$ is actually a mixture of two distributions - the query relevance distribution $P(w|\Theta_R)$ and the collection distribution $P(w|C)$ with mixing proportion λ. Given the feedback document set D_F, and mixing proportion λ, since the collection distribution $P(w|C)$ and feedback document distribution $P(w|D_F)$ is known, the query relevance model $P(w|\Theta_R)$ could be inferred using the EM algorithm (Dempster et al., 1977). The EM update steps are:

$$\text{E-Step: } t^{(n)}(w) = \frac{\lambda \cdot P^{(n)}(w|D_F)}{\lambda \cdot P^{(n)}(w|D_F) + (1-\lambda) \cdot P(w|C))} \quad (4)$$

$$\text{M-Step: } P^{(n+1)}(w|D_F) = \frac{\sum_{d \in D_F} c(w;d) \cdot t^{(n)}(w)}{\sum_{w \in W} \sum_{d \in D_F} c(w;d) \cdot t^{(n)}(w)} \quad (5)$$

Here, W refers to the vocabulary set of the entire corpus. From equations 4 and 5, we observe that terms which are more probable in the feedback set when compared to the background collection distribution $P(w|C)$ accumulate higher weight during successive iterations. Such terms help in distinguishing a relevant document from any random document in the corpus. Noise terms mentioned earlier, like function words in English and terms common to the entire domain of the corpus, receive a near zero weight since they are as likely to occur in the feedback document set as in the entire corpus and hence do not have any distinguishing power. The iterations are continued until convergence and the resultant distribution is the query relevance distribution Θ_R.

When compared to the entire corpus, the LS consists of a focused collection of documents retrieved in response to the initial query. For example, consider the following TREC8 query *"foreign minorities, Germany"* shown in Table 1. A synopsis of some of the relevant and irrelevant documents from the local set retrieved using basic TF-IDF ranking scheme is shown in Table 2. Terms like *"Germany"*, *"foreign"* and *"minority"* occur in many documents, irrespective of whether they are relevant or irrelevant. Hence, the above terms, are not useful in distinguishing a relevant document from any random document within the LS. However, the generative model with collection distribution as BD, assigns higher weightage to such terms since they occur more frequently in the relevant set of documents when compared to the collection. As a result, the query relevance model Θ_R becomes too general for re-ranking the LS.

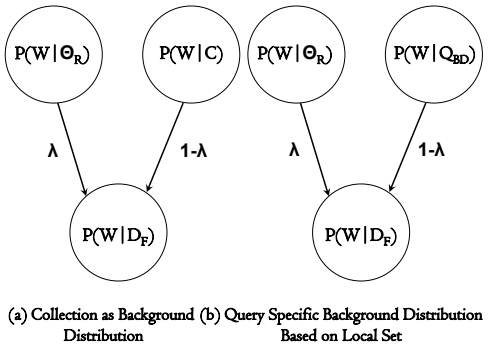


Figure 3: Two Different Generative Mixture Models for Feedback Documents.

3 Query Specific Background Distribution for LSR

In the last section, we described the generative mixture model for learning the query relevance model used to re-rank the corpus. Later, we argued that the query relevance model thus learnt is too general for re-ranking the local set. This is because of using the collection model as background distribution. In this section, we present our approach of learning a query specific background distribution using the LS.

For a given a query topic Q, there is a specific set of terms, different from what is commonly observed at the corpus level, that forms the BD. For example, consider the query mentioned in Table 1, terms like *"country"*, *"germany"*, *"foreign"*, *"german"* occur commonly in many retrieved documents. In this context, the LS could be viewed as a sample from the set of documents D_Q which discuss topics related to any of the keywords mentioned in the query Q. Hence, the word distribution in the LS could be used to approxi-

Rank: 2 (Relevant) *German* authorities are doing everything possible to protect the *foreigners* and other *minorities* in *Germany* and to prevent such incidents...

Rank: 35 (Relevant) *GERMAN* Chancellor Helmut Kohl ... yesterday agreed to set up a committee to discuss neo-Nazi attacks and citizenship rights for *Germany's* large Turkish *minority*...

Rank: 200 (Irrelevant) Mr Balladur said an enlarged Europe 'could not be federal', and warned some members of *Germany's* ruling Christian Democrat party..... representing four-fifths of the (Union's) population and wealth could be put in a *minority*...

Rank: 6000 (Irrelevant) This involves 'being partly hedged in our *foreign* currency exposure, ... many feel that *German* government bonds, traditionally a safe haven in times of market turmoil, offer relative stability.

Table 2: A few sample documents from the LS for query 401 in TREC8 Dataset.

	TREC8	**AP88&89**	**WSJ**
Disks	4,5	1,2	1,2
Number of Documents	556077	164597	173252
Number of Terms	151787183	40839529	42525024
Number of Unique Terms	721207	195942	173760
Average Document Length	272	248	245
Topics	401-450	101-150	151-200

Table 3: Details of TREC8, AP88&89 and WSJ Collections

	Rocchio	**MBF**	**LSR-QBG**
Based On	Vector Space Model	LM for IR	LM for IR
Ranking Function	Cosine Similarity	KL Divergence	KL Divergence
Re-Ranking Done At	Corpus Level	Corpus Level	LS Level
Background Distribution	-	Collection Model as BD	Query Specific BD from LS

Table 4: Summary of RF Approaches used as Baseline for Comparison

mate the background distribution in D_Q.

The generative mixture model for our approach is shown in Figure 3(b). Instead of using the collection model, we use the distribution of words in the local set, $P(w|Q_{BD})$, as background distribution.

$$P(w|Q_{BD}) = \frac{\sum_{d \in LS} c(w;d)}{\sum_{d \in LS} |d|} \tag{6}$$

In the formulation described in Section 2, we just replace the collection distribution with $P(w|Q_{BD})$ and the resultant EM update steps are:

$$\text{E-Step: } t^{(n)}(w) = \frac{\lambda \cdot P^{(n)}(w|D_F)}{\lambda \cdot P^{(n)}(w|D_F) + (1-\lambda) \cdot P(w|Q_{BD}))} \tag{7}$$

$$\text{M-Step: } P^{(n+1)}(w|D_F) = \frac{\sum_{d \in D_F} c(w;d) \cdot t^{(n)}(w)}{\sum_{w \in W} \sum_{d \in D_F} c(w;d) \cdot t^{(n)}(w)} \tag{8}$$

From equations 7 and 8, we can see that the terms which are more probable in the feedback set when compared to the local set distribution receive higher weightage. Such terms are useful in distinguishing a relevant document from any random document in the LS. Noise terms, specific to the LS, mentioned in the above example, like *"foreign"*, *"country"*, *"germany" etc.,* which occur commonly in the LS receive lesser weights. The query relevance distribution Θ_R thus learnt is specific to the LS.

In the next section, we present the results of our approach on three standard TREC datasets and compare it with other existing re-ranking approaches which use RF.

4 Experiments

In our experiments, we mimic the behavior of a user in a standard relevance feedback setting. Based on the query Q, we use LM based query likelihood ranking, to obtain the Local Set (LS). Like the user, we pick the top 10 "relevant" documents D_F from LS and submit it back to our system. These documents are used to re-rank the LS to produce a new ranked list of documents. In this work, we set the size of the local set to be 10K documents. However, as part of future work, we plan to develop algorithms for automatically choosing the local set size based on score distributions. We compare our approach - labeled as **LSR-QBG**, with two standard approaches - TF-IDF based Rocchio Feedback (RF) (Rocchio, 1971) and LM based Model Based Feedback (MBF) (Zhai and Lafferty, 2001). In MBF, we use the generative

Collection		Simple LM Ranking	LSR-QBG	Improvement
TREC8	Init Prec	0.2543	0.6614	+160.71%
	Avg Prec	0.0929	0.1700	+82.99%
	Recall	2353/4728	2838/4728	+20.61%
WSJ	Init Prec	0.3033	0.6402	+111.08%
	Avg Prec	0.1420	0.2220	+60.34%
	Recall	2278/3913	2595/3913	+56.34%
AP88&89	Init Prec	0.2340	0.5984	+155.73%
	Avg Prec	0.1002	0.2361	+135.63%
	Recall	2733/4805	3507/4805	+28.32%

Table 5: Our Approach (LSR-QBG) vs. Initial Ranking to show the effect of relevance feedback on improvement in ranking performance. LM based Query Likelihood was used as initial ranking algorithm to obtain the Local Set (LS). Init Prec and Avg Prec refer to Initial Precision and Average Precision evaluation metrics respectively.

Collection		Rocchio	MBF	LSR-QBG	Impr Over Rocchio	Impr Over MBF
TREC8	Init Prec	0.6406	0.6477	0.6614	+3.25%	+2.11%
	Avg Prec	0.1575	0.1612	0.1700	+7.93%	+5.46%
	Recall	2829/4728	2835/4728	2838/4728	+0.32%	+0.01%
WSJ	Init Prec	0.6231	0.5724	0.6402	+2.74%	+11.84%
	Avg Prec	0.2037	0.2168	0.2220	+8.98%	+2.40%
	Recall	2685/3913	2758/3913	2595/3913	-3.35%	-5.91%
AP88&89	Init Prec	0.6444	0.5591	0.5984	-7.14%	+7.03%
	Avg Prec	0.2323	0.2256	0.2361	+1.63%	+4.65%
	Recall	3527/4805	3501/4805	3507/4805	-0.57%	+0.17%

Table 6: Comparison of our approach (LSR-QBG) with other baseline RF approaches - TF-IDF based Rocchio Feedback (RF) and LM Based Model Based Feedback (MBF).

mixture model formulation described in Section 2. A summary of the baseline approaches is given in Table 4. We used the Lemur Toolkit (Ogilvie and Callan, 2001) for implementation. The standard Rocchio and MBF implementations in Lemur were used for experimentation. For each algorithm, the parameters of the individual approaches were varied to choose the run that yields the best performance for that algorithm.

4.1 Experimental Setting

We evaluated our approach on three standard TREC collections - TREC8, AP 88&89, WSJ. The details of the collections and the corresponding topics used for experiments are listed in Table 3. For all the topics, we only use the title field as the actual query since it is short and closely represents real-life queries. The indices were built after stemming (using *Porter* stemmer) and stopword removal. Dirichlet Smoothing (Zhai and Lafferty, 2004) was used to smooth the individual document language models and the local set model $P(w|Q_{BD})$. The Dirichlet prior was uniformly set to 2000 in all cases.

4.2 Evaluation

We use the following standard measures for evaluation (Yates and Neto, 2005): Initial Precision (Precision at 1), Average Precision, Recall and Precision-Recall curves. In all the runs, we consider the top 1000 documents for evaluation.

The final re-ranked list also includes the training documents provided by the user. Any improvement in ranking of training documents should not influence the evaluation measures as they have been already seen by the user. Hence, we use *Residual Ranking* (Ruthven and Lalmas, 2003), where before evaluation, the training documents provided by the user are removed from the ranked list. All the results reported have been computed based on Residual Ranking.

4.3 Results

Table 5 shows the results of the initial LM based query likelihood ranking and the results observed after re-ranking based on user feedback using our approach. Table 6 shows the results of comparison with Rocchio and Model Based Feedback (MBF). The corresponding Precision-Recall curves are shown in Figure 4.

4.4 Discussion

In Table 6, we observe that LSR-QBG, in most cases, performs better than MBF and Rocchio in Average and Initial Precision.

There is not much improvement in Recall and

Re-ranking Done On	Relevance Feedback	No Relevance Feedback
Corpus	Zhai and Lafferty *et. al.* (Zhai and Lafferty, 2001) Lafferty and Zhai (Lafferty and Zhai, 2001) Lavrenko *et. al.* (Lavrenko and Croft, 2001) Tao *et. al.* (Tao and Zhai, 2006) Ponte (Ponte, 1998)	Lavrenko *et. al.* (Lavrenko and Croft, 2001) Liu and Croft (Liu and Croft, 2004)
Local Set	LSR-QBG (Our Approach)	Oren Kurland *et. al.* (Kurland and Lee, 2005) Oren Kurland *et. al.* (Kurland and Lee, 2006) Michael Bendersky *et. al.* (Bendersky and Kurland, 2008)

Table 7: Our Approach (LSR-QBG) in the context of other Language Modeling based Re-ranking approaches.

in the case of WSJ, the improvement is negative. This could be explained as follows: Since LSR-QBG uses a more specific background distribution while learning the query relevance model, it selects a relatively specific set of terms for re-ranking as compared to MBF and Rocchio. Due to this, the recall drops as a specific term is likely to match lesser documents in the LS.

4.5 Effect of the Local Set Size

Figure 5 shows the effect of the LS size on Average Precision. In TREC8, the MAP reaches a maximum value at 10K and then decreases beyond it whereas in AP 88&89 and WSJ, the MAP increases and reaches a maximum and plateaus out. The local set size affects our approach in two ways: number of relevant documents considered during re-ranking step and the approximation of query specific background distribution.

Let D_R be the set of all relevant documents in the corpus for a query Q and let D_Q be the set of documents which contain any of the keywords mentioned in the query Q. For most of the queries, $D_R \subset D_Q$ and usually $|D_R| \ll |D_Q|$. Hence, the LS as a sample of D_Q is biased for smaller LS sizes since the proportion of relevant documents within the LS is high *i.e.,* more than what exists in D_Q. Hence, the Average Precision observed at smaller local set sizes is less. However, as the LS size increases, the background estimates become more reliable and relatively less biased. Hence, the performance improves. Also, greater number of relevant documents get included and promoted to higher ranks during re-ranking. After a certain size, the local set becomes too general like the corpus and then the performance of our scheme approaches that of MBF using collection model.

5 Related Work

Table 7 presents our work in the context of other language modeling based re-ranking approaches. Ponte (Ponte, 1998) uses a query expansion based approach to handle relevance feedback. He proposes a heuristic score, based on language modeling, to select the expansion terms.

Model based relevance feedback approaches (Lafferty and Zhai, 2001; Lavrenko and Croft, 2001; Zhai and Lafferty, 2001) offer a more principled way of incorporating relevance feedback into the LM framework. Lavrenko *et. al.* (Lavrenko and Croft, 2001) estimate the query relevance model using the query alone. Lafferty *et. al.* (Lafferty and Zhai, 2001) propose the query translation model. A document is ranked based on the probability that it will be translated into the query. The word to word translation probabilities are computed using a *random walk model* based on the word-document index of the entire corpus. Zhai *et. al.* (Zhai and Lafferty, 2001) propose two models for learning the query relevance model from the feedback documents. The first model, described in Section 2, is a generative mixture model for feedback documents with a mixing proportion fixed a-priori. The second model is based on choosing a query relevance model which minimizes the divergence with the distribution of feedback documents and simultaneously maximizes divergence with the collection distribution. Tao *et. al.* (Tao and Zhai, 2006) extend the generative model proposed by Zhai *et. al.* (Zhai and Lafferty, 2001) by considering a different mixture proportion λ for each document. A key feature that distinguishes our approach from all the above approaches is that after learning the query

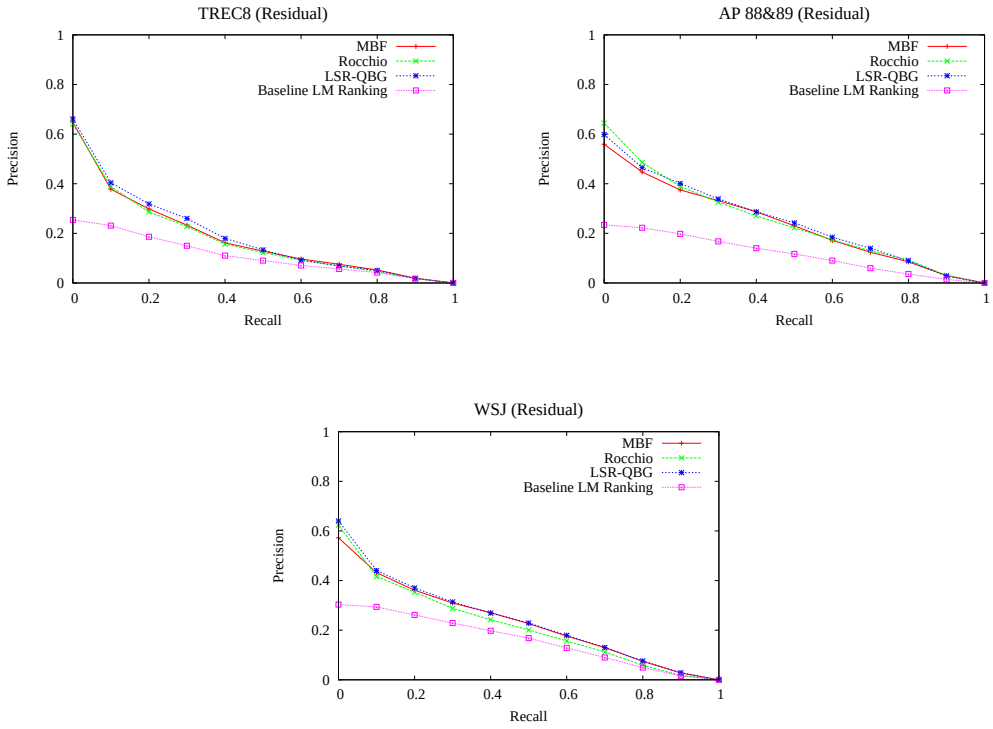


Figure 4: Precision-Recall Curves comparing Our Approach (LSR-QBG) with Rocchio and Model Based Feedback (MBF) across three standard TREC datasets. LM based Simple Query Likelihood Ranking which was used as initial ranking also shown.

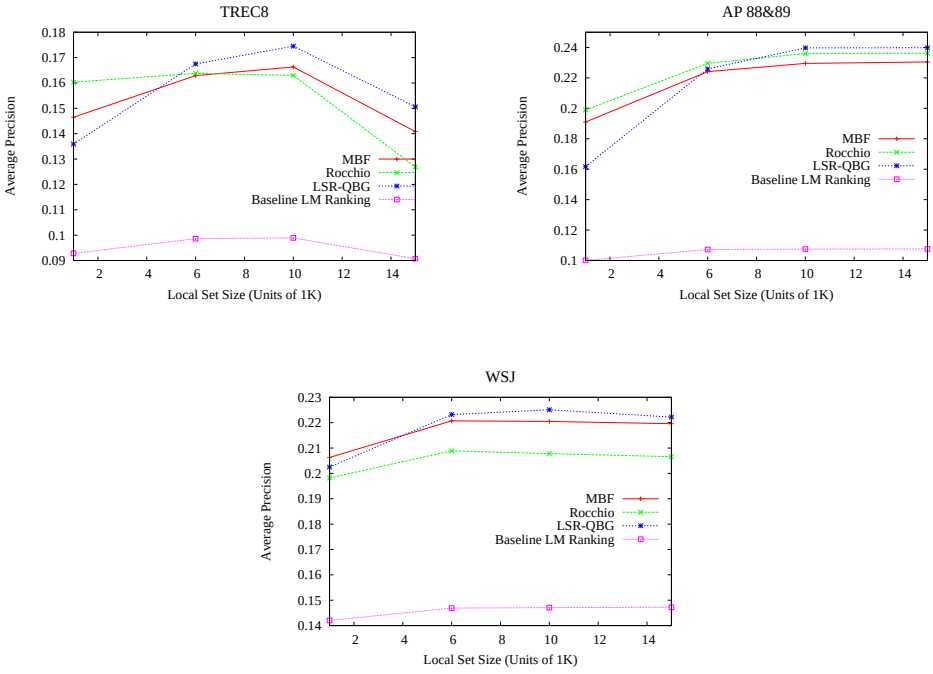


Figure 5: Effect of Local Set Size on Average Precision for three TREC collections.

relevance model based on relevance feedback, we do not re-rank the entire corpus. Instead, we re-rank the documents in the local set and modify the existing generative model to learn a query specific

query relevance model for this task.

There are existing approaches based on language modeling for locally re-ranking search results but do not use relevance feedback. Liu *et. al.* (Liu and Croft, 2004) use *query-specific clustering* for re-ranking documents. Recently, Oren Kurland *et. al.* (Kurland and Lee, 2005; Kurland and Lee, 2006) proposed two different approaches for structurally re-ranking the initial set of results (same as our Local Set) by exploiting inter-document asymmetric relationships. They use centrality measures like Pagerank and HITS to re-rank the initial result set. The relationships are induced using a language modeling based approach by considering the generation probabilities between documents. Bendersky *et. al.* (Bendersky and Kurland, 2008) use document passage graphs to re-rank the Local Set. In our current work we combine both RF and local re-ranking.

As mentioned earlier, the focus of the current work is Language Modeling based approaches since they offer a principled approach to deal with RF. However, outside of the Language Modeling domain, several approaches have been proposed for re-ranking search results locally for various applications. In the context of web, Krishna *et. al.* (Bharat, 2003) use interconnectivity based on hyperlinks to locally re-rank documents. In the context of Personalized Web Search, Teevan *et. al.* (Teevan et al., 2005) use implicit feedback on user interests and context to reorder the initial set of results at the client end. Kamps (Kamps, 2004) proposes a strategy to re-rank search results based on their distance to top-ranked documents.

6 Conclusion and Future Work

We have presented a novel Language Modeling based approach for re-ranking the documents obtained from initial retrieval using manual relevance feedback. Instead of using the query independent collection model, we used the background distribution of the LS to learn a *query specific* relevance model. We compared our approach with TF-IDF based Rocchio and Model based feedback approaches, which are existing approaches to re-rank based on RF. Experimental results on standard datasets show improvements in average precision and initial precision over both initial ranking and other baseline approaches for incorporating RF.

The local set size influences the number of relevant documents included during re-ranking and also the bias of the documents in the local set while using them as background distribution. Instead of fixing the local set size arbitrarily, in future work, we plan to extend the above approach to dynamically choose the local set size based on the score distributions of initial retrieval. We also plan to study the effect of initial retrieval on the performance by using other known ranking schemes like TF-IDF *etc.,* to fetch the LS.

References

[Bendersky and Kurland, 2008] Michael Bendersky and Oren Kurland. 2008. Re-ranking Search Results using Document-Passage Graphs. In *SIGIR '08:*, pages 853–854, New York, NY, USA. ACM.

[Bharat, 2003] Krishna Bharat. 2003. Ranking Search Results by Reranking the Results based on Local Inter-Connectivity. *U.S. Patent*, (6,526,440), February.

[Cover and Thomas, 1991] Thomas M. Cover and Joy A. Thomas. 1991. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA.

[Dempster et al., 1977] A. Dempster, N. Laird, and D. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39:1–38.

[Kamps, 2004] Jaap Kamps. 2004. Improving Retrieval Effectiveness by Reranking Documents Based on Controlled Vocabulary. In *ECIR*, pages 283–295.

[Kurland and Lee, 2005] Oren Kurland and Lillian Lee. 2005. PageRank without Hyperlinks: Structural Reranking using Links induced by Language Models. In *Proceedings of SIGIR*.

[Kurland and Lee, 2006] Oren Kurland and Lillian Lee. 2006. Respect my Authority! HITS without Hyperlinks, utilizing Cluster-based Language Models. In *Proceedings of SIGIR*.

[Lafferty and Zhai, 2001] John Lafferty and Chengxiang Zhai. 2001. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In *SIGIR '01:*, pages 111–119, New York, NY, USA. ACM Press.

[Lavrenko and Croft, 2001] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, New York, NY, USA. ACM Press.

[Liu and Croft, 2004] Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-based Retrieval using Language Models. In *SIGIR '04: Proceedings of the*

27th annual international ACM SIGIR conference on Research and development in information retrieval, pages 186–193. ACM Press.

[Ogilvie and Callan, 2001] Paul Ogilvie and James P. Callan. 2001. Experiments Using the Lemur Toolkit. In *TREC*.

[Pitkow et al., 2002] James Pitkow, Hinrich Schutze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, and Thomas Breuel. 2002. Personalized Search. *Commun. ACM*, 45(9):50–55.

[Ponte and Croft, 1998] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *SIGIR '98: Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 275–281.

[Ponte, 1998] Jay M. Ponte. 1998. A Language Modeling Approach to Information Retrieval. *Ph.D Thesis, University of Massachusetts at Amherst*.

[Rocchio, 1971] J. Rocchio. 1971. Relevance Feedback in Information Retrieval. *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323.

[Ruthven and Lalmas, 2003] I. Ruthven and M. Lalmas. 2003. A Survey on the use of Relevance Feedback for Information Access Systems. *Knowledge Engineering Review*, 18(1).

[Tao and Zhai, 2006] Tao Tao and ChengXiang Zhai. 2006. 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. ACM Press.

[Teevan et al., 2005] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. 2005. Personalizing Search via Automated Analysis of Interests and Activities. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456, New York, NY, USA. ACM Press.

[Yates and Neto, 2005] Ricardo Baeza Yates and Berthier Ribeiro Neto. 2005. *Modern Information Retrieval*. Pearson Education.

[Zhai and Lafferty, 2001] Chengxiang Zhai and John Lafferty. 2001. 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, New York, NY, USA. ACM Press.

[Zhai and Lafferty, 2004] Chengxiang Zhai and John Lafferty. 2004. A Study of Smoothing Methods for Language Models applied to Information Retrieval. *ACM Transactions on Information Systems*, 22(2):179–214.