

# Robust Relevance-Based Language Models

Xiaoyan Li

Department of Computer Science, Mount Holyoke College

50 College Street, South Hadley, MA 01075, USA

Email: xli@mtholyoke.edu

## ABSTRACT

We propose a new robust relevance model that can be applied to both pseudo feedback and true relevance feedback in the language-modeling framework for document retrieval. There are three main differences between our new relevance model and the Lavrenko-Croft relevance model. The proposed model brings back the original query into the relevance model by treating it as a short, special document, in addition to a number of top ranked documents returned from the first round retrieval for pseudo feedback, or a number of relevant documents for true relevance feedback. Second, instead of using a uniform prior as in the original relevance model, documents are assigned with different priors according to their lengths (in terms) and ranks in the first round retrieval. Third, the probability of a term in the relevance model is further adjusted by its probability in a background language model. In both cases, we have compared the performance of our model to that of the two baselines: the original relevance model and a linear combination model. Our experimental results show that the proposed new model outperforms both of the two baselines in terms of mean average precision.

## KEY WORDS

Relevance models, language modeling, feedback, query expansion

## 1. Introduction

The language modeling framework to text retrieval was first introduced by Ponte and Croft in [9]. Many research activities related to this framework have been reported since then [3,4,5,6,7,8,10,11,13,17]. For example, query expansion techniques [5,7,8,11,13], feedback [3,5,7,8,11,13], parameter estimation methods [4], multi-word features [10], passage segmentations [17] and time constraints [6] have been proposed to the language modeling frameworks. Among them, query expansion with pseudo feedback can increase retrieval performance significantly [5,8,13]. It assumes a few top ranked documents retrieved with the original query to be relevant and uses them to generate a richer query model.

However, there are two major problems that are unsolved in query expansion techniques. First, the performance of a significant number of queries will decrease when query expansion techniques are applied. Second, existing query

expansion techniques are very sensitive to the number of documents used for pseudo feedback. Most approaches usually achieved the best performance when about 30 documents are used for pseudo feedback. As the number of feedback documents increases beyond 30, retrieval performance drops quickly.

Therefore, a more robust approach to query expansion in the language-modeling framework is needed. Based on the original relevance model approach by Lavrenko and Croft [5], we propose a new relevance-based language model that improve robustness, and can be applied to both pseudo feedback and true relevance feedback. There are three main mechanisms in the new relevance model to improve the robustness of a relevance-based language model: *treating the query as a special document*, *introducing document-rank-related priors*, and *discounting common words*.

We have carried out experiments for both pseudo feedback and true relevance feedback to compare the performance of our model to that of the two baselines: the original relevance model [5] and a linear combination model [1]. Queries on three data sets have been used: TREC title queries 101 to 200 on AP collections, queries 301 to 400 on a heterogeneous collection which includes all data from TREC disk 4 and 5, and queries 701 to 750 on the TREC terabyte collection. In all of the three sets of experiments, the proposed new model outperforms both of the two baselines. Furthermore, the new approach is less sensitive to the number of pseudo feedback documents than the two baseline models, and it requires fewer relevant documents to achieve good performance with true relevance feedback.

The rest of the paper is structured as follows. In Section 2, we briefly introduce relevance-based language models and then a simple variation of relevance models. Our method of constructing a new robust relevance model and a theoretic justification are described in Section 3. Section 4 provides experimental results in comparing the new relevance model to two baselines based on experimental results with TREC queries. An analysis of the components of the new relevance model is given in Section 5. Finally, Section 6 summarizes the paper with conclusions and future work.

## 2. Baseline Approaches

Our new relevance model is based on the relevance-based language model proposed by Lavrenko and Croft [5]. Therefore, before we introduce the new robust relevance

model, we will briefly describe the relevance-based language model, referred as “original relevance model” in the rest of this paper. Then, a simple variation of relevance models [1] that linearly combines the query and the relevance model is described.

### 2.1 Original Relevance Model

The relevance-based language model was proposed by Lavrenko and Croft [5] in 2001. It is a model-based query expansion approach in the language-modeling framework [8]. A relevance model is a distribution of words in the relevant class for a query. Both the query and its relevant documents are treated as random samples from an underlying relevance model  $R$ . The main challenge for a relevance-based language model is how to estimate its relevance model with no relevant documents available but only queries. Once the relevance model is estimated, the KL-divergence between the relevance model (of a query and its relevant documents) and the language model of a document can be used to rank the document. Documents with smaller divergence are considered more relevant thus have higher ranks. Equations (1) and (2) are the formulas [5] used for approximating a relevance model for a query:

$$P_o(w | R) \approx \frac{P(w, q_1 \dots q_k)}{P(q_1 \dots q_k)} \quad (1)$$

$$P(w, q_1 \dots q_k) = \sum_{D \in M} P(D) P(w | D) \prod_{i=1}^k P(q_i | D) \quad (2)$$

where  $P_o(w | R)$  stands for this original relevance model of the query and its relevant documents, in which  $P(w, q_1 \dots q_k)$  stands for the total probability of observing the word  $w$  together with query words  $q_1 \dots q_k$ . A number of top ranked documents (say  $N$ ) returned with a query likelihood language model are used to estimate the relevance model. In Equation (2)  $M$  is the set of the  $N$  top ranked documents used for estimating the relevance model for a query (together with its relevant documents).  $P(D)$  is the prior probability to select the corresponding document language model  $D$  for generating the total probability in Equation (2). In the original relevance model approach, a uniform distribution was used for the prior.

### 2.2 Linear Combination Relevance Model

The original relevance model does not work well on every query, though on average it significantly outperforms the basic query likelihood language model [16]. The performance of some queries may be hurt badly by using the relevance model, when compared to using the query solely in the query likelihood language model. For such a query, putting the original query back into the relevance model may help. A simple approach to bring the original query back into its relevance model is to linearly combine the query with the relevance model, as in Equation (3), which was used by Abdul-Jaleel et al [1] in their work for the 2004 TREC HARD Track:

$$P_{lc}(w | R) = \lambda P(w | Q) + (1 - \lambda) P_o(w | R) \quad (3)$$

In Equation (3),  $P_{lc}(w | R)$  stands for the relevance model obtained by linearly combining the query with the original relevance model.  $P(w | Q)$  stands for the original query model that may be calculated by the maximum likelihood estimation in the experiments, which will be described in Section 4.  $P_o(w | R)$  stands for the original relevance model described in Equation (1). The weighting parameter  $\lambda$  is used for linearly combining the query and the relevance model. The best value of  $\lambda$  learned with the training data is 0.05, which will be used in our experiments reported in Section 4.

## 3. The New Relevance Model

Based on the original relevance model approach, we propose a new relevance model to further improve retrieval performance and robustness. Three significant changes have been made to the original relevance model in order to estimate a more accurate relevance model for a query: *treating the original query as a special document*, *introducing rank-related prior*, and *discounting common words*. We will give a theoretical justification of the three changes made in the new model in Section 3.1 and detail each of the three improvements in Section 3.2.

### 3.1 Theoretical justifications of the new model

In the original relevance model, queries and relevant documents are treated random samples from an underlying relevance model  $R$  as shown in Figure 1.

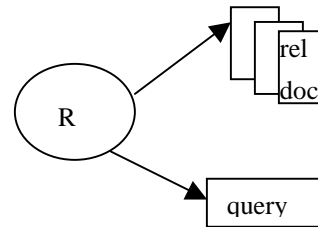


Figure 1: Sampling process in the original relevance model

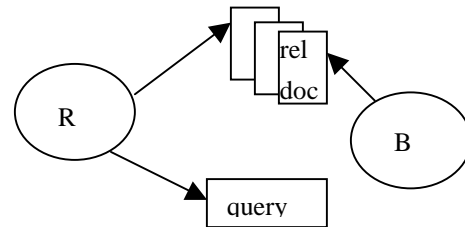


Figure 2: Sampling process in the new relevance model

In the new relevance model, Queries are random samples from the underlying relevance model  $R$  and relevant documents are sampled from both the underlying relevance model  $R$  and a background language model  $B$  as shown in Figure 2. The original relevance model assumes that the sampling process could be different for queries and documents, thus only relevant documents (top ranked documents) are used for approximating the relevance model

R. In the new relevance model, we assume that the way that query words are sampled from R is the same as the way that topic words in relevant documents are sampled. Therefore, a query could be treated as a short document and be considered in the new relevance model. Non-topic words in relevant documents are sampled from the background language model B. To approximate a more accurate relevance model, a *common-word-discounting* component is incorporated into the new relevance model, which will reduce the influence of non-topic words in the process of constructing the relevance model R. Furthermore, top-ranked documents are not necessarily true relevant documents. However, documents with higher ranks are more likely to be relevant to a query thus can play a more important role than documents with lower ranks in approximating the relevance model. The *rank-relate-priors* component in the new relevance model is designed to address this issue.

### 3.2 Three changes in the new relevance model

#### 3.2.1 Query as special document

First, the proposed model brings back the original query into the relevance model by treating it as a short, special document, instead of using a simple linear combination as in [1] (Section 2.2). The total probability of observing the word  $w$  together with query words  $q_1 \dots q_k$  becomes:

$$P_{new}(w, q_1 \dots q_k) = \sum_{D \in S} P(D) P(w | D) \prod_{i=1}^k P(q_i | D) \quad (4)$$

Note that, unlike the set M including only top N documents' models in equation (2) for the original relevance model, the set S in equation (4) includes both the query model and the document models for the top N documents. The new model attempts to include the query model for a relevance model approximation so that it may lead to higher performance, especially for the queries whose performance decreased with the original relevance model. A title query usually consists of a couple of key words. However, the length of a query is much smaller than the average length of its relevant documents. Therefore, it is reasonable to assign a relatively small prior to the query and larger priors to relevant documents or top ranked documents for estimating the relevance model for the query. This was implemented by using document length related priors for  $P(D)$  into Equation (4). This is further explained in the following sub-section.

#### 3.2.2 Rank-related priors

The second change is to assign different priors to the top N documents and the query (which is as a special document) according to the ranks of document, using equation (5):

$$P(D) = \frac{1}{Z_1} * \frac{\alpha + |D|}{\beta + Rank(D)} \quad (5)$$

$$Z_1 = \sum_{D \in S} \frac{\alpha + |D|}{\beta + Rank(D)} \quad (6)$$

In the above equations,  $|D|$  denotes the length of document

$D$  or the length of the query – the special document. *Rank(D)* denotes the rank of document  $D$  in the ranked list of documents returned by using the basic query likelihood language model. The rank of the query is set to 0 so that it has the highest rank among all the documents used for relevance model approximation.  $Z_1$  is the normalization factor that makes the sum of the priors to 1 (in Equation (6)). Parameters  $\alpha$  and  $\beta$  are used to control how much a document's length and its rank affect the prior of the document, respectively. If both  $\alpha$  and  $\beta$  are assigned very large values, then the priors will obey a uniform distribution, which is the same as that in the original relevance model approach. The change of the priors was inspired by two pieces of work. One is Li and Croft's time-based language model [5], in which the uniform priors were replaced by an exponential distribution to favor recent documents. The other one is Wessel et al's work [12] for entry page finding. In their work, a fixed prior probability was learned for each category of pages. In our experiments, the parameters  $\alpha$  and  $\beta$  were tuned on the query set used as training data. It turned out that the best performance were achieved on the training queries when  $\alpha$  took the value around 140 and  $\beta$  took the value around 50.

#### 3.2.3 Common word discounting

The last change to the original relevance models is to discount the probabilities of words that are common in the whole collection. In the framework of the original relevance models, relevant documents are samples of the underlying relevance model. In the new relevance models, words in relevant documents can be grouped into two classes: *topical words* and *non-topical words*. Here we introduce a *background language model* in our approach for this purpose. We assume that topical words are sampled from the underlying relevance model and non-topical words are sampled from the background language model. Therefore, discounting the probabilities of words that are common in the whole collection will help to estimate a more accurate relevance model. The new relevance model is described by the following equations:

$$P_{new}(w | R) = \frac{1}{Z_2} \frac{P_{new}(w, q_1 \dots q_k)}{\gamma + P(w | C)} \quad (7)$$

$$Z_2 = \sum_{w \in V} \frac{P_{new}(w, q_1 \dots q_k)}{\gamma + P(w | C)} \quad (8)$$

$P_{new}(w | R)$  denotes the probability of word  $w$  in the new relevance model.  $P(w | C)$  denotes the probability of word  $w$  in the background language model.  $\gamma$  is the parameter for discounting the probability of a word in the new relevance model by its probability in the background language model.  $Z_2$  is the normalization factor that makes the sum of the probabilities of words in the new relevance model to 1 (Equation (8)). The best value of  $\gamma$  learned with the training queries is 0.02.

Note that the first change (*query as a special document* – Equation (4)) has been incorporated in Equation (7), and

the second change (*rank-related priors* - Equation (6)) has been incorporated in Equation (4) when the new total probability is calculated. Therefore Equation (7) integrates all the three changes. In the following two sections, we will first present our experimental results of the overall performance of the new relevance model versus the original relevance model and the linear combination model. Then we will perform a component analysis in order to obtain a better understanding of the roles of each of the three changes (components).

## 4. Experiments And Results

We have carried out three sets of experiments with four TREC query sets on three data collections. We applied the new relevance model to document retrieval with both true relevance feedback and pseudo feedback. In the case of true relevance feedback, all relevant documents were assigned a same value for *Rank (D)* in Equation (5), since all the documents are supposed to be equally relevant. Therefore, ranking does not affect the priors when true relevant documents are used for relevance model approximation in Equation (5). However weighting over the lengths of the documents is considered. In the case of pseudo feedback, both the ranks and the lengths of the relevant documents are used in the prior calculation.

We compared the new robust relevance model with two baselines. One is the original relevance model (Section 2.1) and the other is the linear combination model (Section 2.2), which linearly combines the query model with the original relevance model. In all experiments, we used the query likelihood language model [9] to retrieve top-ranked documents for feedback. All experiments were performed with the Lemur toolkit [14]. The Krovetz stemmer [18] was used for stemming and the standard stopword list was used to remove *about* 420 common terms.

### 4.1 Data

The two baseline approaches are the original relevance model and the linear combination model. Both were described in Section 2. We used four query sets from three document collections in our experiments. One query set as the training data and the other two query sets as the testing data to evaluate the proposed model and two baseline models: (1). Queries 151 to 200 on AP 88 and AP89 collection. This was also used in [5]. This data set was used as training data. The parameters in both the two baseline relevance models and the new relevance models were tuned on this query set. (2). Queries 101 to 150 on the Associated Press data set (AP88 and AP89). This was also used in [5,11,13]. Therefore, we can consider Tao and Zhai’s two-stage mixture model for pseudo feedback [11] as another baseline and compare our new relevance model with it on this query set. (3). Queries 301 to 400 on a heterogeneous collection TREC45 that includes all data from TREC disk 4 and disk 5. (4). Queries 701 to 750 on a sub-collection of the TREC Terabyte data set. To construct the subset, the

top-ranked 10,000 documents for each of the 50 queries that were retrieved using the basic query likelihood language model were selected. The subset has 466,724 unique web documents and is about 2% of the entire terabyte collection [5].

The statistics of the AP88&89 collection, the TREC45 collection, and the subset of terabyte collection are shown in Table 1. The average length of the documents in the TREC45 collection is 318, which is about 25% longer than the average length (254) of the news articles in the AP collection. The average frequency of terms in the TREC45 collection is about 18% more than that in the AP collection. The average length of the web documents in the terabyte collection is 2054, which is about ten times longer than the average length (254) of the news articles in the AP collection. The average frequency of terms in the subset of the terabyte collection is 30% more than that in the AP collection. It is obvious that the three collections are very different, though the explicit impact of these factors to query expansion needs further study. In Section 4.2 and 4.3, similar results were obtained with the testing query sets even though they were carried out on the two very different collections.

**Table 1. Statistical comparison of the two document collections**

Collection Statistics	AP88&AP89	TREC45	Terabyte (GOV2)
# of documents	164,597	561,445	466,724
# of terms	41,827,813	178,893,105	958,740,730
# of unique terms	204,469	741,630	3,637,433
Length of documents	254	318	2,054
Frequency of terms	205	241	264

### 4.2 Pseudo Feedback

In the case of pseudo feedback for retrieval, a few of the top-ranked documents were assumed relevant thus were used to estimate the relevance model for a query. Figure 3 gives the performance of the proposed model, together with that of the original relevance model and linear combination model with pseudo feedback on the training set (queries 151 to 200 on the AP collection). Figure 4 Figure 5 and Figure 6 compare the performance of the three models on three testing sets: TREC queries 101 to 150 on AP collection, 301 to 400 on the TREC45 collection, and queries 701 to 750 on a sub-collection of the TREC Terabyte data set.

#### 4.2.1 Experimental Results

There are two main conclusions that can be drawn based on the experimental results on the four query sets, given in Figure 3, Figure 4, Figure 5 and Figure 6, respectively. (1). The new relevance model consistently outperformed the original relevance model and the linear combination model no matter how many documents were used for feedback. This can be clearly seen from the four graphs. (2). The new relevance model is less sensitive to the number of feedback documents than the two baselines.

In the Figures 3, 4, 5 and 6, both the new approach and the baselines achieved the best performance around the area

where about 30 or 50 documents were used for feedback. However, for the AP collection and the TREC45 collection (Figures 3, 4 and 5), as the number of feedback documents increases, the performance of the original relevance model and the linear combination model dropped more quickly than the performance of the proposed new relevance model. On the subset of the TREC Terabyte collection, the performance of our new relevant model keep high when above 30 documents were used for feedback. As the number of feedback documents increases, the performance of the original relevance model dropped significantly. The drop of the performance of the linear combination model is not as significant, but its performance is always lower than that of the new relevance model.

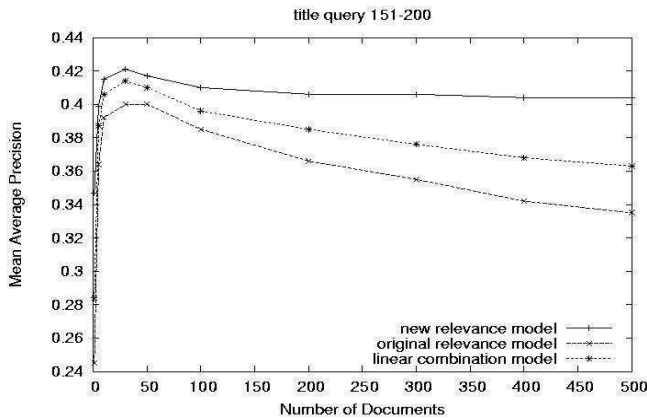


Figure 3. Comparison between the new relevance model, the original relevance model and the linear combination model with pseudo feedback on the training set (query set 151-200)

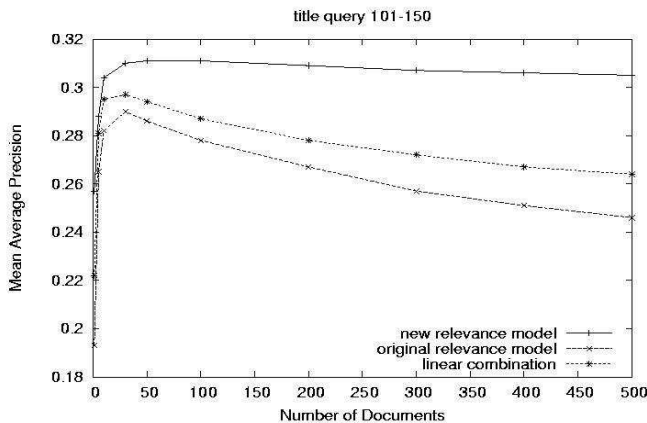


Figure 4. Comparison between the new relevance model, the original relevance model and the linear combination model with pseudo feedback on testing query set 101-150

Our model is also more robust than the results reported in [11] with the TREC 101-150 queries on AP88-89 collection. The sensitivity to the number of feedback documents of a two-stage mixture model was studied in [11]. Based on the results reported, the two-stage mixture model achieved the best performance around 30 feedback documents with the queries from 101 to 150 on AP88-89 collection. As the number of feedback documents increased to 500, the average precision dropped about 12%. Our model achieved the best performance around 50 feedback

documents but only dropped less than 5% when top 500 documents were considered for relevance model approximation with the same query set on the same collection.

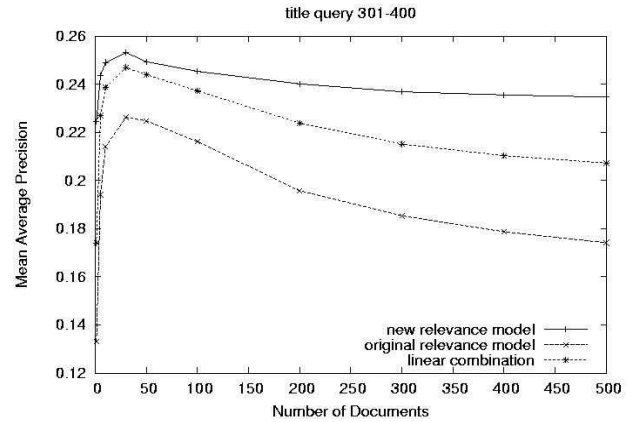


Figure 5. Comparison between the new relevance model, the original relevance model and the linear combination model with pseudo feedback with testing query set 301-400

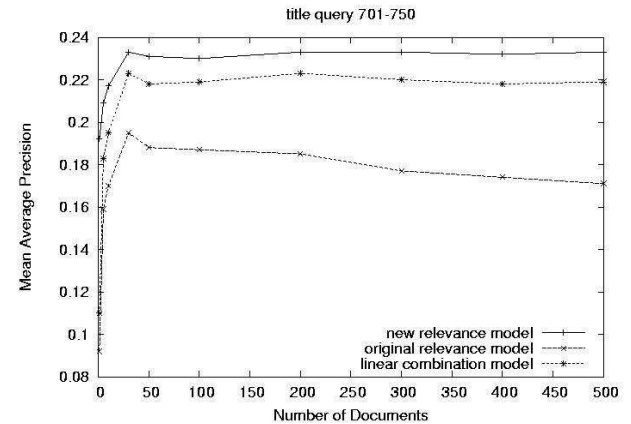


Figure 6. Comparison between the new relevance model, the original relevance model and the linear combination model with pseudo feedback with query set 701-750 on a subset of the TREC Terabyte Track collection.

#### 4.2.2 Discussion

Relevance models can be viewed as a way of query expansion in the sense that they introduce more words into the query representation. Query expansion techniques are not guaranteed to work on every query though they usually can achieve better performance than using the query when measuring the mean average precision on a set of queries. The performance of some queries may be hurt using query expansion techniques while some queries can get significant improvements. Table 2 showed how many queries were affected significantly by using the new relevance model and two baseline models. In the table,  $N_i$  denotes the number of queries whose performance increased by 40% in terms of average precision compared to the performance of the query likelihood language model.  $N_d$  denotes the number of queries whose performance decreased by 40%. We have the following observations based on our experiments.

First, there were more queries whose performance

increased significantly but fewer queries whose performance was hurt badly using the new relevance model than using the original relevance model and the linear combination model. This is obvious in Table 2 in that Ni is the almost always the highest for the new relevance model than the other two baselines, whereas Nd is always the lowest among the three models

Second, for queries 101 to 150 and 151 to 200 on the AP collection, there are more queries whose performance was improved significantly than the queries whose performance was hurt badly, with all the three models. This is also true for queries 301 to 400 on the TREC45 collection, with an exception for the original relevance model. There are 28 queries whose performance was significantly increased but with the performance of 30 queries decreased.

**Table 2. Query-based comparisons of relevance models to query likelihood language models (Orig.: the original relevance model, LC: linear combination model, New: the new relevance model)**

\ Query Method	101-150		151-200		301-400		701-750	
	Ni	Nd	Ni	Nd	Ni	Nd	Ni	Nd
Orig.	20	12	20	9	28	30	3	23
LC	17	7	24	3	25	16	7	21
New	21	7	25	4	33	17	7	19

However, this is not true on the subset of the Terabyte collection. For queries 701 to 750 in Table 2, the performance of a large number of queries decreased significantly. Based on our preliminary experimental results, all three relevance models implemented in this paper did not improve retrieval performance with the queries 701 to 750 on the TREC terabyte collection. This observation is similar to the findings by the groups who applied relevance models or query expansion techniques in the TREC Terabyte Track. Nevertheless, even with this query set, the new robust relevance model performed the best among the three.

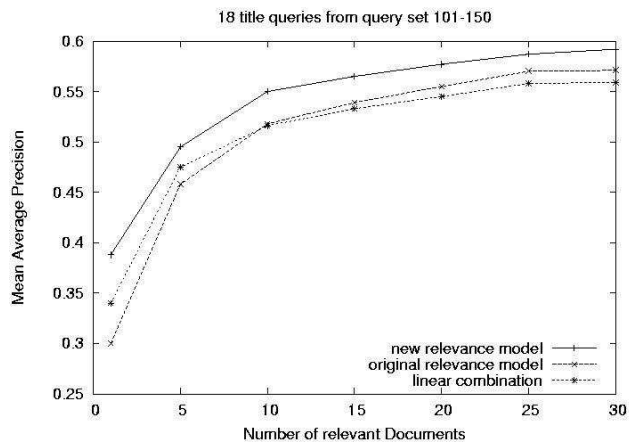
Third, compared to queries 101 to 200, for queries 301 to 400 on the TREC45 collection there are more queries whose performance was decrease significantly and fewer queries whose performance was improved significantly.

The main reason for the different behaviour between queries 101 to 200 on the AP collection and queries 301 to 400 on the TREC45 collection is that the TREC45 collection is composed of news articles from many different resources. Some of the documents are very long and may span multiple topics. When long, cross-topic documents are used for feedback, words related to other topics in the documents will play a negative role in constructing the relevance models for a query, therefore drive the estimated relevance models to drift away from the true relevance model of the query. This explanation is also verified with our experiments with data from the TREC Terabyte track. The Terabyte collection is more diverse than the TREC45 collection. It has many noisy web pages as well as long

documentations spanning multiple topics. Passage retrieval was reported effective on collections that have long cross-topic documents [17]. Therefore, a future extension of the new relevance model is to incorporate passage retrieval for a consistent retrieval performance for queries over heterogeneous collections.

### 4.3 True Relevance Feedback

In the case of true relevance feedback, a number of known relevant documents were used to estimate the relevance model for a query. Figure 7 shows the average performance of the new relevance model, the original relevance model and the linear combination model with true relevance feedback on 18 queries selected from the TREC queries 101 to 150. The purpose of the experiments for true relevance feedback is to study how the three models behave when more relevant documents are given for relevance model approximation. The criterion in selecting the queries was: each of the 18 queries used in this experiment have at least 30 relevant documents within the 200 top-ranked documents from the first round retrieval. Note that this test does not include queries from the training set (queries 151-200).



**Figure 7. Comparison between the new relevance model, the original relevance model and the linear combination model with true relevance feedback on 18 queries chosen from testing queries 101-150**

Three main conclusions can be drawn based on the experimental results given in Figure 7: (1). As the number of feedback relevant documents increased, the new relevance model consistently outperformed the two baselines in terms of mean average precision. (2). The new relevance model achieved even better performance than the two baselines when using fewer relevant documents. The new relevant model achieved about 0.57 of mean average precision when 15 relevant documents used for feedback. But the original relevance model can only achieved the same performance (0.57) and the linear combination model achieved 0.56, respectively, with as many as 30 relevant documents used each for feedback. (3). The linear combination model outperformed the original model only when a small number of relevant documents were used for estimating relevance models. However, as more feedback relevant documents were used, the performance of the

original relevance model was closer to and even better than the performance of linear combination model.

## 5. Component Analysis

In the new relevance model, there are three new components added to the original relevance model: *treating the query as a special document*, *introducing document-ranking-related priors*, and *discounting common words*. To separate the contribution of the three components, we have carried out a set of experiments to breakdown the performance of the new relevance model on both the training data set (queries 151-200) and a testing data set (queries 101-150).

Our first step was to study the contribution of treating queries as special documents by removing query from the set  $S$  in Equation (4). Therefore, only top ranked documents were used for relevance model approximation. The curves labeled by “no query” in Figure 8 and 9 stand for the performance of experiments without the *query as special document* component. Compared to the performance of the new relevance model with all three components, the performance on average dropped about 2.5% for queries 151 to 200 and 1.8% for queries 101 to 150, respectively.

Our second step was to explore the role of the *rank-related priors* component in the new relevance model. In Figure 8 and 9, the curves labeled by “no ranking” mean that document ranking will not be used in adapting document priors, which was implemented by assigning a very large value to  $\beta$  in Equation (5). Compared to the performance of the new relevance model with all the three components, the performance without *rank-related priors* component got about the same performance with 50 feedback documents used for query 101 to 150, and with 30 feedback documents used for query 151 to 200. But the performance without rank-related priors dropped more with more documents for feedback on both query sets, up to as large as about 12% when the number of documents is 500.

Our last step of component analysis was to study the role of the *common word discounting* component. We removed the *common word discounting* component from the framework of the new relevance model but kept the other two components. The performance is represented by the curves by “no word discounting” in Figure 8 and 9. Compared to the performance of the new relevance model with all three components, the performance on average dropped about 5% on for queries 101 to 150 and about 3% for query 151 to 200.

Figures 8 and 9 compared the contribution of each component on both the training query set 151-200 and the test query set 101-150. From the above discussions, we can draw several conclusions: (1). Replacing the uniform priors in the original relevance model with document-ranking-related priors given in Equation (5) makes the model less sensitive to the number of pseudo feedback documents. (2). Both considering query as a special document and discounting common word probabilities can improve the

performance in terms of mean average precision. (3). Most of the performance gain of the new relevance model on average is caused by the word discounting component, but more performance gain is caused by rank-related priors when more documents are used for feedback.

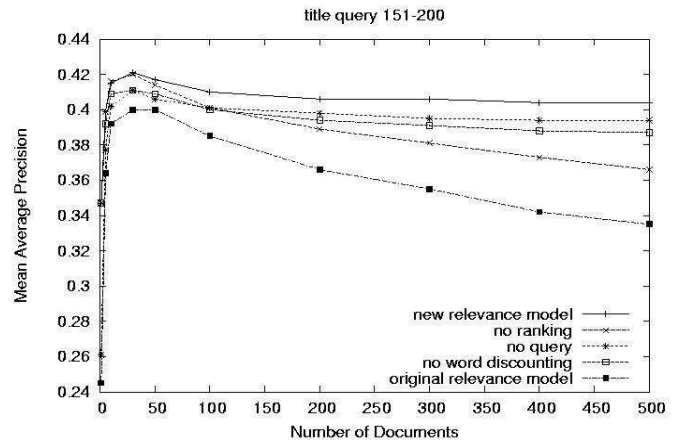


Figure 8. New relevance model component analysis on training query set 151-200 (No query: query is not considered for relevance model approximation; no ranking: document rank is not considered; no word discounting: doesn't discount probabilities of the words that are common in the collection)

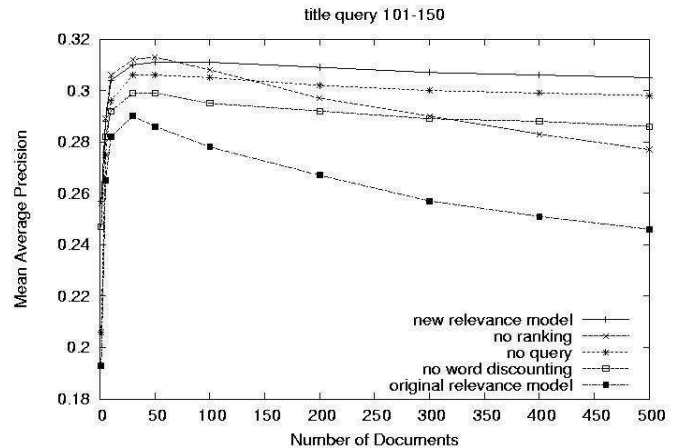


Figure 9. New relevance model component analysis on testing query set 101-150 (No query: query is not considered for relevance model approximation; no ranking: document rank is not considered; no word discounting: doesn't discount probabilities of the words that are common in the collection)

## 6. Conclusions And Future Work

In this paper, a new robust relevance model has been proposed. It was applied to both pseudo feedback and true relevance feedback in the language-modeling framework. The main contributions of this work are: (1) Studying features that will influence the performance of retrieval based on well-designed experiments. The features include key words from original queries, relevance ranks of documents from the first round retrieval and common words in the background data collection. (2) Seamlessly incorporating the features into the original relevance-based language model to improve the performance and robustness of the model. Three changes are bringing back the original

title query as a special document, introducing document-ranking-related priors, and discounting common words

Three main conclusions have been drawn from the experimental results queries on three data collections: queries 101 to 150 and 151 to 200 on the AP88&89 collection, queries 301 to 400 on the TREC45 collection for the TREC ad-hoc retrieval task, and queries 701 to 750 on a sub-collection of the TREC Terabyte data set. First, the new model outperforms both the original relevance model and the linear combination model in terms of mean average precision on document retrieval with both pseudo relevance feedback and true relevance feedback. Second, all three models achieved their best performance when about 30~50 top ranked documents were used for relevance model approximation, but our new model is more robust in the sense that it is less sensitive to the number of documents considered for pseudo feedback than the two baseline models compared. Therefore, the new relevance model can benefit from a large number of feedback documents while the performance drops quickly with the original relevance model and the linear combination model as the number of feedback documents increases. Third, in case of true relevance feedback, the new relevance model achieves a better performance with less relevant documents. This property is very important and desirable because relevance judgments are expensive and usually very hard to obtain.

We note here that although the new relevance model outperforms the original relevance model, there are still some queries, whose retrieval performance in fact decreased when using pseudo feedback. Future work will focus on query-based relevance models that allow parameters in the new relevance models to have different values for different queries. A possible way is to incorporate selective query extension techniques, such as Cronen-Townsend et al's work in [2], into the new relevance model. Queries may be first grouped into two classes. Queries belonging to the first class are likely to have better performance with query expansion techniques and queries belonging to the second class are likely to decrease performance with query expansion techniques. Therefore, the new relevance model may learn different parameter values for the two different classes of queries. As another future work, new approaches to query expansion techniques need to be developed for retrieval on heterogeneous collections (e.g., the Terabyte collection), which may include web documents, blogs, emails as well as news articles. In this case, incorporating passage retrieval and features like metadata into relevance models may be helpful.

## 7. Acknowledgments

This work was supported in part by CIIR at UMass and by DARPA under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are the author and do not

necessarily reflect those of the sponsors.

## 8. References

- [1] N. Abdul-Jaleel, et al, UMass at TREC2004. In *the Thirteen Text Retrieval Conference (TREC 2004) Notebook*, 2004.
- [2] S. Cronen-Townsend, Y. Zhou, and W.B. Croft, A framework for selective query expansion. In *Proceedings of Thirteenth International Conference on Information and Knowledge Management*, 2004.
- [3] D. Hiemstra. *Using language models for information retrieval*. PhD thesis, University of Twente, 2001.
- [4] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *24th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, 2001.
- [5] V. Lavrenko and W. Bruce Croft. Relevance-based Language Models. In *24th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, 2001.
- [6] X. Li and W. B. Croft. Time-based Language Models. In *Proc. 12th International Conference on Information and Knowledge Management*, 2003.
- [7] D. H. Miller, T. Leek, and R. Schwartz. A hidden Markov model information retrieval system. In *22nd ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, 1999.
- [8] J. Ponte. *A Language modeling approach to information retrieval*. PhD thesis, UMass-Amherst, 1998.
- [9] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *21st ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, 1998.
- [10] F. Song and B. Croft. A general language model for information retrieval. In *22nd ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 279-280, 1999.
- [11] T. Tao and C. Zhai. A two-stage mixture model for pseudo feedback. In *27th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04)*, 2004.
- [12] K. Wessel, W. Thijs and H. Djoerd, The importance of prior probabilities for entry page search. In *25th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'02)*, 2002.
- [13] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proc. Tenth International Conference on Information and Knowledge Management*, 2001.
- [14] <http://www-2.cs.cmu.edu/~lemur/3.1/doc.html>
- [15] C. Clarke, N. Craswell, and I. Soboroff, Overview of the TREC 2004 terabyte track. In *the Thirteen Text Retrieval Conference (TREC 2004) Notebook*, 2004
- [16] V. Lavrenko and W. Bruce Croft. Relevance models in information retrieval. In *Language modeling for information retrieval*, W. Bruce Croft and John Lafferty (eds.), Kluwer Academic Publishers, 2003, pages 11-56.
- [17] X. Liu and W. Bruce Croft, Passage retrieval based on language models. In *Proc. 11th International Conference on Information and Knowledge Management*, 2002.
- [18] R. Krovetz, 1993: Viewing morphology as an inference process. In *Proc. 16th ACM SIGIR Conference*, Pittsburgh, June 27-July 1, 1993: 191-202.