A Comparative Study of Methods for Estimating Query Language Models with Pseudo Feedback

Yuanhua Lv
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
ylv2@uiuc.edu

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
czhai@cs.uiuc.edu

ABSTRACT

We systematically compare five representative state-of-the-art methods for estimating query language models with pseudo feedback in ad hoc information retrieval, including two variants of the relevance language model, two variants of the mixture feedback model, and the divergence minimization estimation method. Our experiment results show that a variant of relevance model and a variant of the mixture model tend to outperform other methods. We further propose several heuristics that are intuitively related to the good retrieval performance of an estimation method, and show that the variations in how these heuristics are implemented in different methods provide a good explanation of many empirical observations.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms: Experimentation, algorithms

Keywords: Query language model, pseudo relevance feedback, language models, feedback heuristics

1. INTRODUCTION

As a new generation of probabilistic retrieval models, language modeling approaches to information retrieval (IR) [6, 9] have performed well empirically, with a significant amount of performance increase often coming from the use of pseudo/blind relevance feedback techniques for the estimation of query language models [8, 4, 7].

While many pseudo feedback techniques have been tried for estimating query language models, they have not been compared thoroughly. Previous studies [8, 4, 7] evaluate different methods using different query sets, document collections, and parameter setting, making it impossible to compare results across studies. As a result, we do not yet have a good understanding of the relative strength and weakness of different methods.

The purpose of this paper is to systematically compare these methods with the same experiment setting. We study five representative methods, including two variants of relevance model [4, 1], two variants of mixture model [8, 7], and a Rocchio-like method called divergence minimization model [8]. They are popular and representative methods that have already proven effective to improve retrieval accuracy, and thus a comparative study of them is very interesting.

CIKM'09, November 2–6, 2009, Hong Kong, China.

2. REPRESENTATIVE METHODS FOR ESTIMATING QUERY MODELS

Our basic retrieval model is the KL-divergence retrieval model [3], which scores a document D with respect to a query Q by computing the negative KL divergence between the query language model θ_Q and the document language model θ_D:

$$S(Q,D) = -D(\theta_Q||\theta_D) = -\sum_{w\in V} p(w|\theta_Q)\log\frac{p(w|\theta_Q)}{p(w|\theta_D)} \quad (1)$$

where V is the set of words in our vocabulary. Obviously accurate estimation of the query language model θ_Q plays a critical role in this language modeling approach

Without feedback information, query language models are often estimated by using the MLE method on the query text: $p(w|\theta_Q) = \frac{c(w,Q)}{|Q|}$, where $c(w,Q)$ is the count of word w in the query Q, and $|Q|$ is the total number of words in the query. However, through exploiting feedback information (e.g., assuming the top-ranked documents $F = \{D_1 \dots D_{|F|}\}$ are relevant), we can re-estimate a more accurate query language model $\theta_{Q'}$. We now review briefly several representative effective methods for query model estimation based on pseudo feedback techniques.

2.1 Relevance Model

In the first estimation method of relevance model (often called RM1) [4], the authors essentially use the query likelihood $p(Q|D)$ as the weight for document D and take an average of the probability of word w given by each document language model. Formally, let Θ represent the set of smoothed document models in the pseudo feedback collection F and $Q = \{q_1, q_2, \cdots, q_m\}$. The formula of RM1 is:

$$p_1(w|Q) \propto \sum_{\theta_D\in\Theta} p(w|\theta_D)p(\theta_D)\prod_{i=1}^{m} p(q_i|\theta_D) \quad (2)$$

In the second method (i.e., RM2), they compute the association between each word and the query using documents containing both query terms and the word as "bridges".

$$p_2(w|Q) \propto p(w)\prod_{i=1}^{m}\sum_{\theta_D\in\Theta} p(q_i|\theta_D)\frac{p(w|\theta_D)p(\theta_D)}{p(w)} \quad (3)$$

The Dirichlet smoothing method with a prior of μ_{fb} is used to smooth the language model of each pseudo-relevant document θ_D in both RM1 and RM2.

The relevance model $P(w|Q)$ can be interpolated with the original query model θ_Q to improve performance [1]. In this paper, we will only evaluate the following two interpolated versions of relevance model (called RM3 and RM4):

$$\text{RM3: } p(w|\theta'_Q) = (1-\alpha)p(w|\theta_Q) + \alpha p_1(w|Q) \quad (4)$$

$$\text{RM4: } p(w|\theta'_Q) = (1-\alpha)p(w|\theta_Q) + \alpha p_2(w|Q) \quad (5)$$

where α is a parameter to control the amount of feedback.

2.2 Divergence Minimization Model

The divergence minimization model (DMM) proposed in [8] assumes that the feedback model θ_F should be very close to the language model of every pseudo-relevant document but far away from the collection language model which can be regarded as an approximation of non-relevant language model. The following analytic solution is obtained by solving such an optimization problem:

$$p(w|\theta_F) \propto \exp\left[\frac{1}{1-\lambda}\left(\frac{1}{|F|}\sum_{i=1}^{|F|}\log p(w|\theta_i) - \lambda \log p(w|C)\right)\right] \quad (6)$$

where $p(w|\theta_i)$ is smoothed in the same way as the smoothing of document language model in the retrieval step. Finally, the query language model is updated by interpolating θ_F with the original query model θ_Q in the same way as in Equations 4 and 5 with a coefficient α.

2.3 Simple Mixture Model

In the simple mixture model (SMM) [8], the words in F are assumed to be drawn from two models: (1) background model $p(w|C)$ and (2) topic model $p(w|\theta_F)$. Thus the log-likelihood for the entire set of feedback documents is:

$$\log p(F|\theta_F) = \sum_{w \in V} c(w,F)\log((1-\lambda)p(w|\theta_F) + \lambda p(w|C)) \quad (7)$$

where $c(w,F)$ is the count of word w in the set of feedback documents F, and $\lambda \in [0,1]$ is the probability of choosing the background model $p(\cdot|C)$ to generate the word. The estimate of θ_F can be computed using the Expectation-Maximization (EM) algorithm to maximize the log-likelihood.

Similarly, θ_F is also interpolated with the original query model θ_Q to update the query model with a coefficient α.

2.4 Regularized Mixture Model

In the regularized mixture model (RMM) proposed in [7], each feedback document is allowed to have a potentially different amount of background words (i.e., λ_D). And the original query is combined with the feedback model through a conjugate (Dirichlet) prior on θ_F to respect the relevance of documents. The estimate of θ_F can be computed using the Maximum A Posteriori estimator and the EM algorithm.

Although the original RMM was proposed to eliminate the need for the interpolation parameter α in SMM, to make RMM comparable with other methods, we also introduce a comparable parameter α into RMM to indicate the amount of feedback. Parameter α is defined as: $\frac{r}{\mu'} = \frac{\alpha}{1-\alpha}$, where r and μ' are two parameters in RMM [7].

3. PERFORMANCE COMPARISON

We used several standard TREC data sets in our study, including AP88-89, TREC678, and WT2G. (see Table 1).

	AP88-89		WT2G	TREC678	
	Training	Testing	Testing	Training	Testing
label	AP1	AP2	WT2G	TREC6	TREC78
queries	51-100	101-150	401-450	301-350	351-400

Table 1: Document collections.

S.w.	Metric	MLE	RM3	RM4	DMM	SMM	RMM
Trained on AP1 and Tested on AP2							
w/	AvgPr	0.220	0.295	0.301	0.290	**0.304**	0.299
	Pr@10	0.386	0.408	0.418	**0.422**	0.400	0.398
	Recall	3074	3810	3892	3681	**3933**	3859
w/o	AvgPr	0.231	0.312	0.321	0.289	**0.324**	0.323
	Pr@10	0.398	0.436	**0.448**	0.424	0.432	0.446
	Recall	3154	3913	3908	3674	3921	**3927**
Trained on TREC6 and Tested on TREC78							
w/	AvgPr	0.217	0.249	0.242	0.235	**0.251**	0.243
	Pr@10	0.437	0.438	0.426	0.443	0.443	**0.451**
	Recall	5114	5805	5739	5476	**5821**	5625
w/o	AvgPr	0.217	0.251	0.243	0.235	**0.252**	0.249
	Pr@10	0.434	**0.454**	0.446	0.433	0.441	0.443
	Recall	5107	5799	5776	5500	**5896**	5833
Well-Tuned on WT2G							
w/	AvgPr	0.293	**0.338**	0.319	0.327	0.330	0.309
	Pr@10	0.450	**0.500**	0.470	0.494	0.496	0.458
	Recall	1830	1822	1806	1843	**1856**	1811
w/o	AvgPr	0.306	**0.344**	0.328	0.326	0.331	0.319
	Pr@10	0.456	**0.490**	**0.490**	0.476	0.476	0.482
	Recall	1870	1862	1879	1873	**1889**	1863

Table 2: Performance comparison.

We pre-processed documents and queries with two different strategies: in the first strategy, we only stemmed words with the Porter algorithm, which is indicated as "w/ s.w." (i.e., with stop words); in the second one, besides stemming, a total of 418 stop words from the standard InQuery stoplist were removed, which is labeled as "w/o s.w." (i.e., without stop words).

3.1 Feedback Effectiveness

We first compare the effectiveness of the five methods. We fix Dirichlet smoothing ($\mu = 1000$) for estimating the document language models. We also simply set the number of feedback documents to 10 and the number of terms in feedback model to 100, and the rest parameters are trained on the corresponding training data set (we would use the learned parameters in the rest of this paper if there is no extra specification). We summarize the results in Table 2. Overall we find SMM and RM3 most effective in our experiments. SMM is better on homogeneous data, e.g., AP2; while RM3 works more effectively on Web data, i.e., WT2G. For the recall, SMM dominates over all other methods obviously. DMM and RM4 do not work as well as other methods, although RM4 appears to be effective on homogeneous data which is consistent with the observation in [4]. Indeed we observe similar performance between RMM and SMM on most collections with several exceptions (i.e., on WT2G) where RMM worked clearly worse than SMM.

3.2 Robustness Analysis

In SMM and DMM, the parameter λ controls the influence of the collection language model. In RM3 and RM4, a parameter μ_{fb} plays a similar role. We set $\alpha = 0.5$ and examine how λ (or μ_{fb}) affects the average precision. We observe that the performance is quite sensitive to the parameter in DMM, but is relatively insensitive in SMM, RM3 and RM4, especially in RM3. It is interesting to see that RM3 often achieves a stable performance when we set $\mu_{fb} = 0$.

Recall that we interpolate the estimated feedback model θ_F with the original query model θ_Q. The interpolation is controlled by a coefficient α. Our experiment results show that the setting of α can affect the performance significantly for all the five methods. In another exploration [5], we have studied how to adapt this parameter to the characteristics of queries and feedback documents.

We further compare the robustness of different estimation methods w.r.t. the number of documents used for pseudo feedback in Figure 1. We notice that RM3 is much more robust than the other three methods. Yet it appears that all methods work well with 10 feedback documents.

4. EMPIRICAL ANALYSIS OF FEEDBACK HEURISTICS

To understand why some of these methods work better than others, we propose several heuristics that we would like every estimation method to satisfy, and analyze how each of the five estimation methods implements these heuristics.

IDF: *assigning more weights to discriminative terms. Explicit* IDF implementation exists in SMM, DMM and RMM, as they all use a collection language model to trim common terms from the feedback model, but RM3 and RM4 also have an *implicit* IDF effect due to the smoothing of document language models in the retrieval step [9].

TF: *favoring frequent terms in feedback documents.* The two variants of relevance model and two variants of mixture model employ an *arithmetic* mean to aggregate term frequency evidence from feedback documents, while DMM uses a *geometric* mean as shown in formula 6.

Document Weight: *respecting important feedback documents rather than taking all of them equally.*

Some methods discriminate feedback documents in terms of the amount of *relevance information*, e.g., RM3 uses the query likelihood score, while RM4 adopts an indirect query likelihood, which differs from RM3 in that RM4 sums over feedback documents by using the likelihood of each query word as the document weight and then aggregates the evidence from all the query words; RMM uses the original query as a prior to assign more weights to documents more relevant to the query: one interesting thing is that RMM combines document weight and mixture noise parameter together, controlled by a dynamic parameter λ_D.

Some methods favor *long documents*, e.g., SMM and RMM pools together terms from all the feedback documents, which is essentially using the *raw* document length as the document weight to combine document language models; DMM, RM3 and RM4 use Dirichlet smoothing method to estimate language models for feedback documents, which indeed assigns a weight $\frac{|D|}{|D|+\mu}$ to document D and thus tends to also prefer long documents.

We summarize in Table 3 how the five methods implement the above heuristics. We can see that different feedback methods implement heuristics quite differently. So we now turn to the following questions: (1) what could be the best implementation for each heuristic? and (2) is the bad retrieval performance due to the weakness of some heuristic implementation?

Heuristics	RM3	RM4	DMM	SMM	RMM
TF	am	am	gm	am	am
IDF	implicit	implicit	explicit	explicit	shared
Rel. score	QL	indirect	No	No	
Doc. length	Dir	Dir	Dir	raw	raw

Table 3: Heuristics analysis of different methods. "no" means no implementation; "am" and "gm" stand for arithmetic mean and geometric mean respectively; "QL" indicates query likelihood; "Dir" stands for Dirichlet smoothing.

4.1 Comparison of IDF Strategies

The mixture model and the divergence minimization provide two different strategies of implementing the IDF heuristic. To make the two strategies comparable, we design a new method Dir-SMM, which uses the same strategy as DMM to aggregate feedback documents, i.e., taking an average of the *smoothed* document language models, but uses the mixture model to factor out common terms. The results are reported in Table 4. It is interesting to see that Dir-SMM outperforms DMM consistently, which may suggest that the mixture model is better than the divergence minimization in terms of IDF effect.

Can the same IDF effect be achieved *implicitly* during the smoothing of document language models in the retrieval step? To seek the answer, we design another version of SMM, in which the parameter λ is set to 0. We find that the original SMM outperforms this new version of SMM stably, no matter how we tune the Dirichlet prior.

We observe similar performance between RMM and SMM on most collections, but RMM was much worse on WT2G. This may suggest that RMM, which uses the same parameter (i.e., λ_D) to control IDF effect and to capture document relevance score, loses to a simple mixture model.

4.2 Comparison of TF Strategies

To compare the effectiveness of the arithmetic mean and the geometric mean as TF strategies, we design two methods: Dir-SMM$_{\lambda=0}$ and DMM$_{\lambda=0}$. $\lambda = 0$ means that the IDF in the models is blocked. So the feedback models estimated by Dir-SMM$_{\lambda=0}$ and DMM$_{\lambda=0}$ are essentially $p_1(w|\theta_F) = \frac{1}{|F|}\sum_{i=1}^{|F|} p(w|\theta_i)$ and $p_2(w|\theta_F) = \left[\prod_{i=1}^{|F|} p(w|\theta_i)\right]^{\frac{1}{|F|}}$ respectively. We see from Table 4 that the former is clearly better than the latter, which may mean that the geometric mean is not a good TF measure for pseudo feedback.

4.3 Comparison of Doc Weight Strategies

Document length: We have two candidate strategies (i.e., Dirichlet smoothing and raw document length) for document weighting based on document length. The proposed Dir-SMM and the original SMM essentially use these two strategies respectively. In addition, we also introduce another member EDL-SMM into the comparison, which assigns equal weights to feedback documents. To fully exploit the strength of Dir-SMM, we also train the Dirichlet parameter which was fixed in the previous experiments. This enhanced version of Dir-SMM is labeled as Dir-SMM+. We compare SMM, Dir-SMM+ and EDL-SMM in Table 4. It is not surprising that Dir-SMM+ works overall the best; in fact, Dir-SMM+ takes SMM and EDL-SMM as its two extreme cases when setting $\mu = \infty$ and $\mu = 0$ respectively. However, one interesting observation is that, on collections with stopwords (i.e., w/ s.w.), EDL-SMM is better than or comparable to Dir-SMM+; while on other collections (i.e., w/o s.w.), SMM is better than or comparable to Dir-SMM+. Considering the complexity of Dir-SMM+ which has more

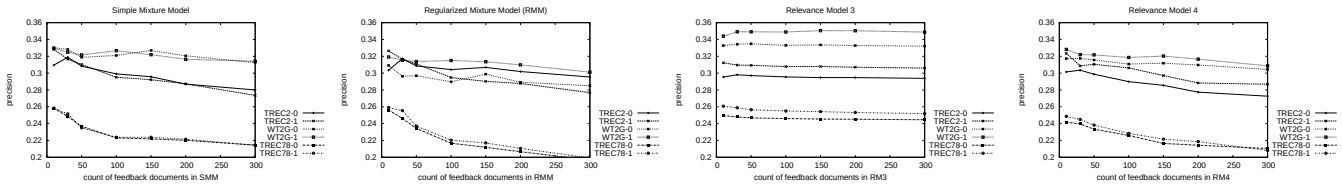

Figure 1: Sensitivity of MAP to the count of feedback documents in SMM (first), RMM (second), RM3 (third), and RM4 (fourth). Note that lines with label "-0" ("-1") mean that stopwords are kept (removed).

Training	Testing	S.w.	Dir-SMM	$DMM_{\lambda=0}$	Dir-$SMM_{\lambda=0}$	Dir-SMM+	EDL-SMM	QL-EDL-SMM
AP1	AP2	w/	0.308	0.254	0.267	0.308	0.310	0.302
		w/o	0.319	0.270	0.284	0.323	0.323	0.313
TREC6	TREC78	w/	0.256	0.225	0.235	0.256	0.255	0.253
		w/o	0.249	0.230	0.237	0.252	0.247	0.254

Table 4: MAP of methods with variations of feedback heuristics.

parameters to tune, a hybrid method which dynamically selects EDL-SMM or SMM to use could be a better strategy.

What are the reasons that document length can be exploited for document weight? Why does document length work differently on different type of collections? To answer these questions, we plot the document length and document ranking of the top-10 documents on AP1 data, where document length is normalized to sum to 1. We observe that document length is distributed almost randomly on AP1 w/ s.w., but on AP1 w/o s.w., document length tends to be large for highly-ranking documents. It could be one possible reason why document length works better on AP1 w/o s.w. than on AP1 w/ s.w. This phenomenon also confirms a finding in [2]: if all the query terms are discriminative words, the KL-divergence method will probably assign a higher score to a longer document, but if there are common terms in the query, longer documents are often overly-penalized and thus would not receive higher scores. Therefore, document length is more correlated to the relevance score on collections of which stopwords are removed.

Query likelihood: RM3 is most robust so far. According to our analysis, only RM3 uses the query likelihood score as the document weight, suggesting that the query likelihood could potentially lead to the robustness of RM3.

We next examine if the query likelihood score can also improve the retrieval precision. We design another family of mixture model QL-EDL-SMM, in which the weight of each feedback document model is exactly the query likelihood score (i.e., we do not use document length). We compare QL-EDL-SMM and EDL-SMM in Table 4. It is observed that although QL-EDL-SMM improves the MAP on TREC78 w/o s.w. slightly, it decreases the MAP on all other collections. It may suggest that the query likelihood score does not improve the retrieval precision, even though it appears to increase the robustness in taking different numbers of feedback documents.

5. CONCLUSIONS

Five methods for estimating query language models were evaluated, including RM3, RM4, DMM, SMM, and RMM. We found SMM and RM3 most effective in our experiments. Using SMM yielded effective retrieval performance in both precision and recall. RM3 performed similarly to SMM on precision but worse than SMM on recall measure. However, RM3 is more robust to the setting of feedback parameters. RM4 only appeared to be effective on homogeneous document collections. DMM had relatively poor performance and was quite sensitive to parameter setting. We found similar performance between RMM and SMM on most collections, but RMM was much worse on Web data (i.e., WT2G).

We further proposed several heuristics that are intuitively related to the good retrieval performance of an estimation method, and found that the implementation of proposed heuristics in different methods provided a good explanation of many empirical observations.

6. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their useful comments. We also thank Victor Lavrenko and Donald Metzler for valuable suggestions related to the relevance model. This material is based upon work supported by the National Science Foundation under Grant Numbers IIS-0347933, IIS-0713581, and IIS-0713571.

7. REFERENCES

[1] Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Donald Metzler, Mark D. Smucker, Trevor Strohman, Howard Turtle, and Courtney Wade. Umass at trec 2004: Novelty and hard. In *TREC '04*, 2004.

[2] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *SIGIR '04*, pages 49–56, 2004.

[3] John D. Lafferty and Chengxiang Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01*, pages 111–119, 2001.

[4] Victor Lavrenko and W. Bruce Croft. Relevance-based language models. In *SIGIR '01*, pages 120–127, 2001.

[5] Yuanhua Lv and ChengXiang Zhai. Adaptive Relevance Feedback in Information Retrieval. In *Proceedings of CIKM '09*, 2009.

[6] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281, 1998.

[7] Tao Tao and ChengXiang Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *SIGIR '06*, pages 162–169, 2006.

[8] ChengXiang Zhai and John D. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01*, pages 403–410, 2001.

[9] ChengXiang Zhai and John D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*, pages 334–342, 2001.