Fully Utilize Feedbacks: Language Model Based Relevance Feedback in Information Retrieval

Sheng-Long Lv[1], Zhi-Hong Deng[1,2], Hang Yu[1], Ning Gao[1], and Jia-Jian Jiang[1]

[1] Key Laboratory of Machine Perception (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
[2] The State Key Lab of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
sllv@pku.edu.cn, zhdeng@cis.pku.edu.cn,
{yh_030603,ninggao,jjj}@pku.edu.cn

Abstract. Relevance feedback algorithm is proposed to be an effective way to improve the precision of information retrieval. However, most researches about relevance feedback are based on vector space model, which can't be used in other more complicated and powerful models, such as language model and logic model. Meanwhile, other researches are conceptually restricted to the view of a query as a set of terms, and so cannot be naturally applied to more general case when the query is considered as a sequence of terms and the frequency information of a query tern is considered. In this paper, we mainly focuses on relevant feedback Algorithm based on language model. We use a mixture model to describe the process of generating document and use EM to solve model's parameters. Our research also employs semi-supervised learning to calculate collection model and proposes an effective way to obtain feedback from irrelevant documents to improve our algorithm.

Keywords: Relevance Feedback, Language model, Semi-supervised Learning, Irrelevant Document.

1 Introduction

Relevance feedback[9] is an interactive technique allowing users to evaluate whether a subset of documents, which usually are just returned by IR system, are relevant or not. Then the evaluation results are used to retrieve new documents. Relevance feedback is considered to be an effective way to improve performance of IR systems and has attracted much attention for a long time.

Unfortunately, most researches on relevance feedback were based on early introduced search models, such as vector space model and probabilistic model. Few work focused on implementing relevance feedback on relatively newly introduced models, for example, language model.

The language model was first introduced by Ponte and Croft. It leverages statistical methods and outperforms other search models in text retrieval area , making it an attractive text retrieval methodology. Although language modeling approach performs

J. Tang et al. (Eds.): ADMA 2011, Part I, LNAI 7120, pp. 395–405, 2011.

well in text retrieval task, its performance on feedback is not fully explored. In most existing work, relevance feedback was implemented in the same way as vector space model by simply adding extra terms to query. This method still views query as a set of keyword rather than a statistical model, which is more compatible with language modeling approach. In the work of Zhai and J. Lafferty, a model based approach to feedback was proposed. But it only incorporated labeled relevant documents and ignored other valuable feedbacks.

In this paper, we propose a language model based relevance feedback method on KL-divergence framework. Our method gets feedback from labeled relevant documents, labeled irrelevant documents and unlabeled documents. Thus, we conclude our contribution as following:

1. We propose an detailed method to implement relevance feedback on language model.
2. Using semi-supervised learning, our method incorporates unlabeled data to estimate model parameters.
3. We explore how to incorporate labeled irrelevant documents and propose an effective solution.

In this paper, relevance feedback is executed in the following flow:

1. Given a document collection and an original query, our IR system first implements KL-divergence without feedback and returns the most relevant document according to the query.
2. IR system gets feedback that whether last returned document is relevant
3. Incorporating feedbacks, IR system implements our method and returns the most relevant document which has not been returned.
4. Go to step 2 until all the documents is returned.

This flow is the same as the flow of Relevance Feedback track of INEX 2010, which simulates use case of an internet search engine. A user inputs his query and the search engine returns a list of results. The user scans results in descending order so he first views the most relevant result the list. The user evaluates its relevance and gives feedback to search engine. Then the search engine performs another search using feedback.

This paper is organized as follows: we introduce some related work in Section 2. We present our detailed feedback model in Section 3. In Section 4, we discuss the experiment results. Finally we conclude in Section 5.

2 Related Work

2.1 Language Modeling Approach

Language model[3] is first introduced by Ponte and Croft. It is often used in natural language processing and information retrieval. Language modeling approach builds a probabilistic language model θ_d for each document. Documents are ranked based on $P(Q|\theta_d)$, the probability of its model generating the query. Since the probabilistic language model is often smoothed, making it a fine-grained model, language modeling approach often outperforms other models[16].

In [4, 10], a probabilistic model θ_q is built for query Q. Documents are ranked by the similarity of document's language model and the query model, instead of the probability $P(Q|\theta_d)$. This extension makes language model better associated with relevance feedback because we can use feedbacks to update query model, under the same idea with query expansion. In this approach, KL divergence is used to measure the similarity of two models, as showed in the following:

$$D\ (Q||d) = \sum_w P(w|\theta_Q) * log \frac{P(w|\theta_Q)}{P(w|\theta_d)} \quad (1)$$

Where Q is a query and d is a document. θ_Q and θ_d are models we estimate.

Relative entropy is a non-negative value and the smaller $D\ (Q||d)$ is , the closer the two models are. When two models are all the same , their relative entropy is zero. Note that Relative entropy is not a legal distance function because it doesn't satisfy symmetry and triangle inequality.

2.2 Relevance Feedback Methods on Language Modeling Approach

Several papers[3,10,12,13,14,15] have focused on improving performance of language modeling approach by relevance feedback methods. In [3], the authors expanded query under the same way with Rocchio Algorithm. Terms appearing frequently in relevant documents but are relatively few in the whole collection were added to queries as a new keyword. Unfortunately, this method is limited by still viewing query as a set of keyword like VSM approach. It didn't perform all the potential of language model.

[10] and [12] focused on model-based relevance feedback methods. [10] proposed two schemes for building the query model based on a set of labeled relevant documents. The two schemes were based on regularized maximum likelihood criterion and minimizing the average KL-divergence between the query model and the relevance model. In [12], Victor Lavrenko. et al introduced a new method to construct a relevance model. It leveraged statistical methods to estimate P(t|Relevance) according to the terms' co-occurrence in documents. Both the two approaches improved performance of language modeling approach in their experiments. However, they only incorporated positive feedbacks but ignored labeled irrelevant documents and unlabeled documents.

There have been some works extending language modeling approach to obtain better results. Terms were usually considered independent in existing work. Dependence models extends language model by taking terms' relationship into consideration[15]. A cluster-based term selection algorithm was proposed in [14] for constructing refined query language model. [13] presented a learning approach to balance the original query and feedback information while most methods set this balance parameter to a fixed value.

3 Model Definition

In our approach, the fundamental task is to estimate query model and each document's model as most KL-divergence approach. The query model should involve

valuable feedback from labeled documents and global information from the whole collection.

3.1 Incorporating Labeled Relevant Documents

Intuitively, most relevant documents belong to the same topic. Thus, we assume there is a relevance model generates all the relevant document. To incorporate labeled relevant document, a generative relevance model θ_R should be estimated using feedbacks. Then our new query model is:

$$\theta_R' = \theta_Q + \theta_R \tag{2}$$

Where θ_Q is the original query model and θ_R is the relevance model.

Now our challenge is how to estimate relevance model θ_R. We assume each relevant document is a sample of the relevance model, then a natural way to estimate relevance model is to maximize the probability that relevance model generating all the relevant documents. The probability is:

$$P(D_r|\theta_R) = \prod_{d_i \in D_r} \prod_w P(w|\theta_R)^{c(w,d_i)} \tag{3}$$

Where D_r is the set of labeled relevant documents. $c(w, d_i)$ denotes the number of term w that appears in document d_i.

The equation 3 is sufficient if there is only relevant information in relevant documents. However, relevant documents often contain trivial parts. These parts neither do harm to nor contribute to the relevance of document. As a consequence, a more reasonable model would be a mixture model of relevance model and collection model. The collection model generates all the trivial part of the collection. This means that a term w can be generated by relevance model by probability $P(w|\theta_R)$, or can be generated by collection model by probability $P(w|\theta_C)$. We also assume a term in a document can only be generated by one model. So this problem can be viewed as a classification problem. Terms are classified into two classes: relevance and collection. We use I(θ,w) to denotes our classification result. It denotes the probability that term w is generated by model θ. Under this assumption the log-likelihood of labeled relevant documents is:

$$logL(\theta) = \sum_{d_i \in D_r} \sum_w c(w, d_i)(P_{w,R} + P_{w,C}) \tag{4}$$

$$P_{w,R} = I(\theta_R, w) log \tau_r P(w|\theta_R) \tag{5}$$

$$P_{w,C} = I(\theta_C, w) log \tau_c P(w|\theta_C) \tag{6}$$

Where τ_r denotes the probability that relevance model θ_R generates a term, and τ_c denotes the probability that collection model θ_C generates a term.

I(θ,w) and τ are maybe difficult to distinguish. I(θ,w) is our classification result as a hidden variable in the model and τ is model parameter which denotes the intrinsic feature of the model. And for each term I(θ,w) is different but for all the term τ is all the same.

How to maximize equation 4 will be presented in the next section.

3.2 Incorporating Unlabeled Documents

The basic model did not give a reasonable definition and interpretation of collection model θ_C. Under our definition of collection model, we should use all the trivial part of the collection to estimate collection model. However, it is difficult to identify which part is trivial in a document. Another way to solve this problem is to use all the unlabeled documents instead of trivial parts. This is because most parts of an irrelevant document are trivial parts and most of unlabeled documents are irrelevant. Thus, we should maximize the probability that collection model generates unlabeled document:

$$\mathrm{P}(D_u|\theta_C) = \prod_{d_i \in D_r} \prod_w P(w|\theta_C)^{c(w,d_i)} \tag{7}$$

Where D_u is the set of labeled relevant documents.

Thus, the new likelihood function is

$$logL(\theta) = \sum_{d_i \in D_r} \sum_w c(w,d_i)(P_{w,R} + P_{w,C}) + \sum_{d_j \in D_u} \sum_w c(w,d_j)\ P_{w,C} \tag{8}$$

$$P_{w,C} = log\tau_c P(w|\theta_C) \tag{9}$$

The newly added part in equation 8 is the log-likelihood of unlabeled documents. In the calculation of this part, we believe that all of the terms are generated by the collection model, so this is not a classification problem and I(θ,w) isn't used as a factor here.

We use EM (Expectation-Maximization) Algorithm[8] to maximize likelihood function and solve the parameter. Now the model parameters are $\tau_r, \tau_c, P(w|\theta_R)$ and $P(w|\theta_C)$ and the hidden parameters are I(θ_R,w) and I(θ_C,w). EM is an iterative algorithm. Each iteration consists of two steps: In E step hidden variables are estimated by the current model parameters' values; In M steps model parameters are re-estimated according the hidden values to maximize the likelihood function. After each iteration, the likelihood function is larger than the prior iteration. Meanwhile, as the gradual process of maximizing, EM algorithm will find a good local optimal solution in most cases.

In E step, hidden variable I(θ,w) is:

$$I(\theta_R, w) = \frac{\tau_r P(w|\theta_R)}{\tau_r P(w|\theta_R) + \tau_c P(w|\theta_C)} \tag{10}$$

$$I(\theta_c, w) = 1 - I(\theta_R, w) \tag{11}$$

In M step, using the value of I(θ_R,w) and I(θ_C,w) obtained in E step, we make the likelihood function maximum and estimate model parameters' values by employing Lagrange constrained optimization method. Here we have three constraints:

$$\tau_r + \tau_c = 0 \tag{12}$$

$$\sum_w P(w|\theta_R) = 1 \tag{13}$$

$$\sum_w P(w|\theta_C) = 1 \tag{14}$$

Considering these three constrains, we can get the optimal value of model parameters. The results are:

$$\tau_r = \frac{\sum_{d_i \in D_r} \sum_w c(w,d_i) I(\theta_R,w)}{\sum_{d_i \in D_r} \sum_w c(w,d_i) I(\theta_R,w) + \sum_{d_i \in D_r} \sum_w c(w,d_i) I(\theta_c,w) + \sum_{d_j \in D_u} \sum_w c(w,d_j) I(\theta_c,w)} \tag{15}$$

$$\tau_c = 1 - \tau_r \tag{16}$$

$$P(w|\theta_R) = \frac{\sum_{d_i \in D_r} c(w,d_i) I(\theta_R,w)}{\sum_w \sum_{d_i \in D_r} c(w,d_i) I(\theta_R,w)} \tag{17}$$

$$P(w|\theta_C) = \frac{\sum_{d_i \in D_r} c(w,d_i) I(\theta_c,w) + \sum_{d_j \in D_u} c(w,d_j) I(\theta_c,w)}{\sum_w (\sum_{d_i \in D_r} c(w,d_i) I(\theta_c,w) + \sum_{d_j \in D_u j} c(w,d_j) I(\theta_c,w))} \tag{18}$$

In this paper, the termination condition is:

$$\sum_w (\Delta P(w|\theta_R) + \Delta P(w|\theta_C)) < 10^{-4} \tag{19}$$

During calculation, we encountered the same problem as [10]. If we estimate these four parameters at the same time, τ_c will be very close to zero. Then our model degenerated to a unigram model. However, this problem didn't have bad impact on the final results. But we would like to have a non-zero τ_c to make our model more reasonable. So we set τ_c and τ_c to same constant during calculation.

3.3 Incorporating Labeled Irrelevant Documents

In our model, last returned document is always ranks first in all the documents which have not been returned. Then it is a more valuable feedback that if last returned document is labeled irrelevant, because last returned document is the most relevant document according to our estimation. That it is labeled irrelevant denotes our estimation can't perform properly, so our model needs modification.

Because last returned document is the most relevant document according to our estimation, so this irrelevant document must contain a number of highly relevant terms. In our model, the relevance of term is determined by the $P(w|\theta_R)$ indirectly, so this document must have some terms whose $P(w|\theta_R)$ are relatively high, making document model close to the query model. However, $P(w|\theta_R)$ is calculated from the set of labeled relevant documents and of very high credibility. Therefore, we believe that these terms whose $P(w|\theta_R)$ is high do no harm to the document's relevance. It is some of other terms that frequently appear in this document do harm to the document's relevance. And these words are of high irrelevance features so that the document that contains many relevant terms can be labeled irrelevant. In this paper, we call this kind of terms punishing terms. We extract punishing terms from irrelevant documents and punish the relevance of a document if it contains punishing terms.

For example, when a user enters the query "Nobel Prize" to find some introduction of Nobel Prize, a document about Ig Nobel Prize is labeled as irrelevant. This shows that the user wants to query the real Nobel Prize information. However, the document contains large numbers "Ig Nobel Prize" and the relevance of "Nobel" and "Prize" is very high. Traditional Relevance Feedback algorithm like Rocchio Algorithm would punish these three words at the same degree. In our model, because "Nobel" and

"Prize" is highly relevant, we mainly punish the relevance of “Ig”, which is fully in line with the user's search intent. This shows the advantages of our model.

When last returned document is labeled as irrelevant, we use the following method to extract the punishing terms:

1. For each term w in the document, calculate the value of $P(w|\theta_d)$-$P(w|\theta_R)$.
2. Rank terms in descending order according to the value.
3. Select the top m term as punishing terms.
4. Record $P(w|\theta_R)$ of every punishing terms.

We extracted m from each irrelevant documents and build the punishing model θ_P as follows:

$$P(w|\theta_P) = \frac{\sum_{d\in D_i} P(w|\theta_d)}{\sum_w P(w|\theta_P)} \tag{20}$$

Where D_i is the set of labeled irrelevant documents.

When last returned document is labeled as irrelevant, if there are still relevant documents that have not been returned , the feedback is sufficient to denotes that our model can’t recognize irrelevant features properly. When there is no relevant documents that have not been returned, returning a irrelevant document is inevitable. Extracting punishing word from that document may affect our model’s effectiveness. Therefore, we choose the first n irrelevant documents to extract punishing terms.

After the punishing model has been built, we use the relative entropy to compute the irrelevance of a document:

$$\mathrm{D}_P(\mathrm{d}||\mathrm{P}) = \sum_w P(w|\theta_P) * log\frac{P(w|\theta_P)}{P(w|\theta_d)} \tag{21}$$

3.4 Rank the Documents

In [11], Robertson proposed that documents can be sorted by the odds of their being observed in relevant class and irrelevant class, namely:

$$\frac{P(D|R)}{P(D|N)} \tag{22}$$

Where, R represents the document is relevant while N represents the document is not relevant.

In this paper, we follow Robertson's sorting methods, using the odds of document relevance score and document punishing score as the sorting criteria.

$$\frac{D_r(\mathrm{d}||\mathrm{Q})}{\mathrm{D}_P(\mathrm{d}||\mathrm{P})^{\alpha}} \tag{23}$$

Where α controls the weight of punishment.

According to previous research, experimental results are often better when irrelevant feedbacks have a smaller weight than relevant feedbacks. Thus, in our experiment, we set α = 0.5.

4 Experiment

4.1 Dataset and Competitors

We choose dataset of Relevance Feedback Track of INEX 2009 as our experiment dataset. This dataset was downloaded from Wikipedia at October, 2008 and labeled by YOGO (2008-w40-2). The size of dataset is 50.79GB. Documents in this dataset are all in XML format. Every document's relevance has been manually evaluated. And the evaluation is accurate to sentences, so we could know which sentence is relevant in a relevant document.

In our experiment, we randomly choose six topic to run our model. Following is each topic's information:

Table 1. Fig. 1. Information of topics

Topic ID	*Query*	*Document Count*	*Relevant Count*	*Relevant Percent*
2009023	Plays of Shakespeare Macbeth	751	86	0.1145
2009088	hatha yoga deity asana	754	13	0.0172
2009089	world wide web history	758	71	0.0937
2009095	Weka software	753	19	0.0252
2009109	circus acts skills	753	155	0.1955
2009115	virtual museums	759	61	0.0803
Sum		4528	405	0.0894

In our experiment, we compared our model to Rocchio Algorithm[6], KL-divergence approach. And we will analyze the impact of incorporating different set of documents.

Rocchio Algorithm has been proven stable and excellent as a relevance feedback approach in many experiments. Thus comparing with Rocchio Algorithm would give strong evidence of the performance of our model. Our model is based on KL-divergence, so comparing with KL-divergence will show how much the performance improve after acquiring feedbacks.

4.2 Experiment Results

In our experiment ,we treated each document as plain text by filtering all the tags in documents. And a document is considered relevant as long as it contains relevant sentences.

Our model vs KL-divergence

Our model is based on KL-divergence, so we run our model two times in this experiment. The first run has no feedbacks and the second run has feedbacks to update models. The following are experiment results:

Table 2. Results of our model vs. KL-divergence

	P@5	P@10	P@20	P@50
KL-divergence	0.57	0.50	0.51	0.35
Our Model	0.70	0.72	0.56	0.44

The table 2 shows our model signicantly improves the precison of KL-divergencem, especially at the low recall rate. The denotes our model is effective method for language modeling approach to incorporate feedbacks. Note that as the number of documents increases , the improvement of precison gets smaller.

VSM vs. KL-divergence

Before comparing our model and Rocchio Algorithm, we first run VSM approach and KL-divergence to obtain their performance. This is because our model and Rocchio Algorithm are based on different models. Our model is based on KL-divergence approach while Rocchio Algorithm is based on VSM approach. If there is a big difference between the effect of VSM approach and KL-divergence, our model and Rocchio Algorithm are not comparable. Comparison results are as follows:

Table 3. Results of VSM vs. KL-divergence

	P@5	P@10	P@20	P@50
VSM	0.51	0.52	0.39	0.33
KL-divergence	0.57	0.50	0.51	0.35

As can be seen from the table, KL-divergence is slightly better than VSM approach. This is because KL-divergence approach is based on statistical methods and relaxes restrictions of relevant documents on the query model, making it more suitable to apply to general case.

Our model vs. Rocchio Algorithm

In our experiments, we implemented Rocchio Algorithm as the following equation[7]:

$$Q' = \alpha Q_0 + \frac{\beta}{n_r}\sum_{d\in D_r} v_d - \frac{\gamma}{n_s}\sum_{d\in D_s} v_d \tag{24}$$

Our Rocchio Algorithm got feedbacks from both labeled relevant document and labeled irrelevant document.. According to previous experiment, we set a = 1, b = 0.75 and c = 0.5, Table 3 shows the experiment results.

Table 4. Results of our model vs. Rocchio

	P@5	P@10	P@20	P@50
Our Model	0.70	0.72	0.56	0.44
Rocchio	0.59	0.60	0.59	0.58

Table 4 shows that before 20 feedback documents our model is signicantly better than Rochhio algorithm. At 5 feedback documents, our model improves KL-divergece by 23% while Rocchio algorithm improves VSM appoach by 15%. And at this point, KL-divergence is better than VSM approach by 12%. We know that the better the results of a rmodel are, the more diffcult to improve it . At 10 feedback documents, our model improves KL-divergece by 44% while Rocchio algorithm improves VSM appoach by 15%.

However, at 50 feedback documents, our model isn't better than Rocchio Algorithm any longer. The reason for the decline is likely to be that our model gets a poor local optimal solution during the process of maximizing the expectation. This is because Rocchio algorithm uses the vector to update the query vector, that is, each of the relevant documents has the same weight to change query vector. Our model uses EM method iteratively update query model. After getting a large number of feedback documents, newly returned document is chosen according to similarity with query model, thus its document model changes query model little then, so it is easy to fall into local optimal solution.

5 Conclusion and Future Work

In this paper, we propose a relevance feedback model based on language modeling approach. Our model fully utilizes feedbacks including labeled relevant documents, labeled irrelevant documents and unlabeled documents to estimate a new query model. We employ EM Algorithm to solve model parameters, semi-supervised learning method for modeling the document collection and build punishing model to get valuable information from labeled irrelevant feedback documents.

Experiments show our model improves the precision of language modeling approach. Especially at the lower recall rate, the improvement is significant. However, at higher recall rate our improvement drops and finally the precision is lower than Rocchio Algorithm. But our model is still an effective approach for KL-divergence to incorporate feedbacks. Because users tend to access only the first few pages of search results when using internet search engine. The recall of first few pages is very low.

Acknowledgement. This work is partially supported by Project 61170091 supported by National Natural Science Foundation of China and Project 2009AA01Z136 supported by the National High Technology Research and Development Program of China (863 Program).

References

1. van Rijsbergen, C.J.: Information retrieval, 2nd edn. Butterworths (1979)
2. Crestani, F., Ruthven, I., Sanderson, M., van Rijsbergen, C.J.: The troubles with using a logical model of IR on a large collection of documents. In: Harman, D.K. (ed.) Proceedings of the Fourth Text Retrieval Conference (TREC-4), pp. 509–525. NIST special publication (1995)

3. Ponte, J., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the ACM SIGIR 1988, pp. 275–281 (1988)
4. Lafferty, J., Zhai, C.: Docunment language models, query models, and risk minimization for information retrieval. In: Proceedings of SIGIR 2001 (2001)
5. Lafferty, J., Zhai, C.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of SIGIR 2001 (2001)
6. Rocchio, J.J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) The SMART Retrieval System Experiments in Automatic Document Processing, ch. 14, pp. 313–323 (1971)
7. Ide, E., Salton, G.: Interactive search strategies and dynamic file organization in information retrieval. In: Salton, G. (ed.) The SMART Retrieval System - Experiments in Automatic Document Processing, ch. 18, pp. 373–393 (1971)
8. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of Royal Statist. Soc. B 39, 1–38 (1977)
9. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science 41(4), 288–297 (1990)
10. Zhai, C., Lafferty, J.: Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of CIKM 2001 (2001)
11. Robertson, S.E.: The Probability Ranking Principle in IR, pp. 281–286. Morgan Kaufmann Publishers, Inc., San Francisco (1997)
12. Lavrenko, V., Croft, B.: Relevance-based language models. In: Proceedings of SIGIR 2001 (2001)
13. Lv, Y., Zhai, C.: Adaptive relevance feedback in informationretrieval. In: Proceedings of CIKM 2009 (2009)
14. Tan, B., Velivelli, A., Fang, H., Zhai, C.: Termfeedback for information retrieval with languagemodels. In: Proceedings of SIGIR 2007 (2007)
15. Bai, J., Song, D., Bruza, P., Nie, J.-Y., Cao, G.: Queryexpansion using term relationships in language models forinformation retrieval. In: Proceedings of CIKM 2005 (2005)
16. Bennett, G., Scholer, F., Uitdenbogerd, A.: A comparative study of probabilistic andlanguage models for information retrieval. In: Nineteenth Australasian Database Conference (ADC). CRPIT, ACS, vol. 75, pp. 65–74 (2008)
17. Xu, Z., Akella, R.: A bayesian logisticregression model for active relevance feedback. In: Proceedings of SIGIR 2008 (2008)