Relevant Query Feedback in Statistical Language Modeling

Ramesh Nallapati, Bruce Croft and James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{nmramesh, croft, allan}@cs.umass.edu

ABSTRACT

In traditional relevance feedback, researchers have explored relevant document feedback, wherein, the query representation is updated based on a set of relevant documents returned by the user. In this work, we investigate relevant query feedback, in which we update a document's representation based on a set of relevant queries. We propose four statistical models to incorporate relevant query feedback.

To validate our models, we considered anchor text of incoming links to a given document as feedback queries and performed experiments on the home-page retrieval task of TREC 2001. Our results show that three of our four models outperform the query-likelihood baseline by at least 35% in MRR score on a test set.

Categories and Subject Descriptors

H.3.3 **[Information Search and Retrieval]**: Retrieval models - *language models*

General Terms

Algorithms

Key Words

relevance feedback, relevant document, relevant query

1. INTRODUCTION

Relevance feedback is a widely reported and largely successful technique in Information Retrieval. In traditional relevance feedback, the user's query is reformulated using a list of relevant of documents returned by the user. The main idea consists of selecting important terms from the relevant documents and enhancing the importance of these terms in the new query [1].

In the recent past, language modeling [11] has become very popular in IR owing to its sound theoretical basis and good empirical success. In the language modeling framework, one associates a unique probability distribution of words in the vocabulary, called the language model M_D, to each document D and estimates the relevance of the document to a given query Q by the probability of its generation from the document as shown below.

$$P(Q|M_D) = \prod_{w \in Q} P(w|M_D) = \prod_{w \in Q} (\lambda \hat{P}(w|D) + (1-\lambda)\hat{P}(w|C)) \quad (1)$$

In a recent paper [12], Robertson discussed a few potential problems of the language modeling framework with respect to the event spaces being modeled. Since the language model expresses the probability of a query given a document, the event space would consist of queries in relation to a particular document and these event spaces would be unique to each document. Under this interpretation, the query-likelihood scores of different documents for the same query would not be comparable because they come from different probability distributions in different event spaces. Robertson claimed that this would imply that the simple language model is not capable of supporting relevant document feedback for a given query. However, it would support relevant query feedback for a given document because the queries come from the same event space. We believe the theoretical issues raised by Robertson are still unresolved, but the discussion motivated us to investigate the problem of relevant query feedback in the framework of language modeling. We believe that apart from the theoretical motivation, the current investigation of relevant query feedback finds its utility in practical retrieval systems where users' feedback is available.

The problem of relevant query feedback, which is the flip side of relevant document feedback, consists of updating a document's representation given a set of queries relevant to the document. Although this is analogous to relevant document feedback, it is not quite the same. The entities that are fedback in the present context are very sparse while the document itself is richer in features.

In this work, we propose four statistical models for relevant query feedback in the language modeling framework. The reminder of this report is organized as follows. In section 2 we present a brief overview of the past work done in modeling relevant query feedback. In section 3, we describe the statistical models we built to incorporate relevant query feedback in the language modeling framework. We describe our experimental setup and present the results on the home-page finding task of TREC 2001 in section 4. We conclude the report with a brief discussion on future work in section 5.

2. RELATED WORK

Salton [14] discussed relevant query feedback in the context of dynamic document space modification. In [15, page 145], the idea is elegantly expressed as follows. "...when a number of documents retrieved in response to a given query are labeled by the user as relevant, it is possible to render these documents more easily retrievable in the future by making each item somewhat similar to

CIKM'03, November 3–8, 2003, New Orleans, Louisiana, USA.

the query used to retrieve them ...". Salton reported that the enhanced document representation thus obtained improved the recall and precision values up to 10% for future queries [15]. Empirical techniques that exploit term co-occurrences in query-relevant document pairs are described also in [5] and [17].

In probabilistic models, [7] looks at a learning network approach to IR that learns from queries. The work done by Berger and Lafferty [2] on applying translation model to IR can be viewed as an idea of exploring the correlation between documents and relevant queries. In the framework of language modeling, [9] discusses the similarity and difference of the language modeling approach and the classic probabilistic models, including the different possibilities for feedback.

The following section describes the four models we propose for relevant query feedback.

3. STATISTICAL MODELS

3.1 Mixture model

In this approach, we assume that a document's language model is a mixture of multiple component distributions where each component is associated with a prior probability of generation. Accordingly, the generative probability of a word w with respect to the document language model M_D is given by

$$P(w|M_D) = \sum_Z P(w|Z)P(Z|M_D) \quad (2)$$

where Z is a component distribution and $P(Z|M_D)$ is the component's prior. An underlying assumption here is that a word's generative probability is conditionally independent of the document model M_D given the component Z. A graphical representation of the model is shown in figure 1. Considering the document D, the

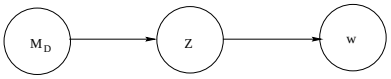

Figure 1: Mixture model

set of feedback queries F and the collection C as the components, the document's new language model becomes

$$P(w|M_D) = \alpha\hat{P}(w|D) + \beta\hat{P}(w|F) + (1-\alpha-\beta)\hat{P}(w|C) \quad (3)$$

The model now consists of two parameters α and β which are typically set by tuning for optimal performance on a training set.

3.2 Dependency model

In the Dependency model, we assume that both the document D as well as the set of feedback queries F depend on the words w that they consist of. The resulting Bayesian network is shown in a graphical representation in figure 2. We are interested in computing

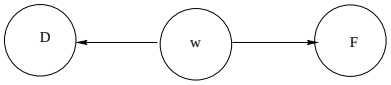

Figure 2: Dependency model

$P(w|D,F)$, the document language model given the evidence of the document content and the set of relevant queries. This can be evaluated from the Bayesian network as follows:

$$\begin{aligned} P(w|D,F) &= \frac{P(D,F|w)P(w)}{\sum_v P(D,F|v)P(v)} && (4)\\ &= \frac{P(D|w)P(F|w)P(w)}{\sum_v P(D|v)P(F|v)P(v)} && (5)\\ &= \frac{\frac{P(w|D)P(w|F)}{P(w)}}{\sum_v \frac{P(v|D)P(v|F)}{P(v)}} && (6)\end{aligned}$$

While steps 4 and 6 follow from Bayesian inversion, step 5 follows from the conditional independence of D and F with respect to w that follows from the definition of the Bayesian network in figure 2. We assume that the conditionals $P(w|D)$ and $P(w|F)$ are given by the smoothed unigram models of the document and relevant query set as shown below.

$$\begin{aligned} P(w|D) &= \alpha\hat{P}(w|D) + (1-\alpha)\hat{P}(w|C) && (7)\\ P(w|F) &= \beta\hat{P}(w|F) + (1-\beta)\hat{P}(w|C) && (8)\end{aligned}$$

The summation in equation 6 is over the entire vocabulary. The evaluation of this expression is computationally prohibitive as it involves evaluating the entire sum for each retrieved document. However, the expression can be greatly simplified by expanding it out as shown below.

$$\begin{aligned} \sum_v \frac{P(v|D)P(v|F)}{P(v)} &= \sum_{v\in D\cup F} \frac{P(v|D)P(v|F)}{P(v)} \\ &+ \sum_{v\in C\ominus(D\cup F)} \frac{P(v|D)P(v|F)}{P(v)} \quad (9)\end{aligned}$$

where $\ominus$ is the set difference operator. Next, we note that $\forall v \in C \ominus (D \cup F)$:

$$\begin{aligned} & \hat{P}(v|D) = 0 \quad \& \quad \hat{P}(v|F) = 0 \\ \Longrightarrow\ & \frac{P(v|D)P(v|F)}{P(v)} = (1-\alpha)(1-\beta)\hat{P}(v|C) \quad (10)\end{aligned}$$

In step 10, we used equations 7 and 8 and assumed that the prior probability of a word $P(v)$ is equal to its empirical distribution in the general English corpus $\hat{P}(v|C)$. Now, plugging 10 back in 9 and using the axiom that probabilities add up to unity, we get

$$\begin{aligned} \sum_v \frac{P(v|D)P(v|F)}{P(v)} &= \sum_{v\in D\cup F} \frac{P(v|D)P(v|F)}{P(v)} \\ &+ (1-\alpha)(1-\beta) \times (1 - \sum_{v\in D\cup F} \hat{P}(v|C)) \quad (11)\end{aligned}$$

Notice that the summation now is only over the vocabulary of D and F. Although the evaluation of the expression in equation 11 is still expensive, it is definitely more tractable than the original expression in equation 6. The dependency model too, consists of two parameters α and β which need to be tuned for optimal performance.

3.3 Density Allocation

In this model, we assume that the probability distribution of the document is a random vector variable $\mathbf{x}$, and there is a prior distribution $P_M(\mathbf{x})$ on this variable. Hence, in this model, generating a query Q involves sampling a distribution $\mathbf{x}$ from the prior $P_M(\mathbf{x})$ and then sampling the query terms independently from the distribution $P_{\mathbf{x}}(w)$ [8]. Accordingly, the generative probability of a query

with respect to the document model $P(Q|M)$ is given by

$$P(Q|M) = \int (\prod_{i=1}^{|Q|} P_{\mathbf{x}}(w_i)) P_M(\mathbf{x}) d\mathbf{x} \quad (12)$$

If we represent the smoothed unigram document language model of equation 1 as a vector $\mathbf{D}$, we can obtain the same model from Density Allocation by choosing a sharp prior $P_M(\mathbf{x})$ such as a Dirac function that is centered around $\mathbf{D}$ as follows:

$$P_M(\mathbf{x}) = \lim_{\epsilon \to 0} \frac{1}{\sqrt{4\pi\epsilon}} e^{\frac{\|\mathbf{x}-\mathbf{D}\|}{4\epsilon}} \quad (13)$$

In the presence of the evidence of feedback queries, we assume that the prior distribution is concentrated around two distributions, namely, the smoothed document model $\mathbf{D}$ and the smoothed feedback model $\mathbf{F}$ as defined in equations 7 and 8. The new Dirac prior is then given by

$$P_M(\mathbf{x}) = \lim_{\epsilon \to 0} \frac{1}{\sqrt{4\pi\epsilon}} e^{\frac{\|\mathbf{x}-\mathbf{D}\|+\|\mathbf{x}-\mathbf{F}\|}{4\epsilon}} \quad (14)$$

Performing the integration in equation 12 using this prior, we simply obtain

$$\begin{aligned} P(Q|M) &= \prod_{i=1}^{|Q|} P_{\mathbf{D}}(w_i) + \prod_{i=1}^{|Q|} P_{\mathbf{F}}(w_i) && (15) \\ &= \prod_{i=1}^{|Q|} (\alpha \hat{P}(w_i|D) + (1-\alpha)\hat{P}(w_i|C)) \\ &+ \prod_{i=1}^{|Q|} (\beta \hat{P}(w_i|F) + (1-\beta)\hat{P}(w_i|C)) && (16) \end{aligned}$$

Similar to the other two models discussed above, the Density Allocation model requires tuning of the parameters α and β.

3.4 Maximum Likelihood model

In this model, we leverage the evidence of relevant queries to optimize the smoothing parameter λ of the basic language model in equation 1. More formally, we want to find the value of λ that maximizes the likelihood of relevant set of queries F given the document model M_D. Mathematically, we can write:

$$\begin{aligned} \lambda_{opt}(D) &= \arg\max_{\lambda} P(F|M_D) \\ &= \arg\max_{\lambda} \prod_{i=1}^{|F|} P(w_i|M_D) \\ &= \arg\max_{\lambda} \prod_{i=1}^{|F|} (\lambda \hat{P}(w_i|D) + (1-\lambda)\hat{P}(w_i|C)) \end{aligned} \quad (17)$$

Since the domain of λ is restricted $(0 < \lambda < 1)$, it is quick and easy to find optimal λ through a simple binary search. In effect, we compute the optimum value of λ for each document and then use them in retrieval experiments. The computational effort in computing the best λ for each document could still be very expensive especially in collections that consist of millions of documents and hence we resort to some approximations which we will describe in the following section.

4. EXPERIMENTS AND RESULTS

An ideal experimental set up to test the performance of relevant query feedback would be to collect queries and relevance judgments from users for a long period of time and then evaluate the performance of the system on a new set of queries using enhanced document representations from relevant query feedback models. However, for the system to register any significant improvement in performance on new query sets, one would need a much larger number of queries and relevance judgments than are available in the present TREC collections. Although such resources may not be infeasible to procure in a commercial setting over a long period of time, we have found it impractical in our current research environment.

Hence, we have turned our attention to another valuable resource, the World Wide Web. In the web environment, researchers have considered links from one page to the other as a recommendation mechanism. Algorithms such as PageRank [10] and Kleinberg's HITS algorithm [6] have popularized this concept by estimating the authority of a web page by its link structure. In this work, we extend this concept one step further and consider the anchor text on the incoming links to a web document as relevant queries to the document. Since anchor text is a succinct description of the content of the document it is pointing to, we believe this is a reasonable assumption.

We have used *WT10G*, a 10 gigabyte subset of the world wide web from TREC 2001 as our test bed. We believe the relevant queries (anchor texts) available per document (an average of 13 words per document) are large enough in number to result in a substantial enhancement in the document representation using our relevant query feedback models. We have performed our experiments on the home-page finding task of TREC 2001 web track [3]. The task involves finding the home-page requested by the query. For example, when the query "Text Retrieval Conference" is issued, the system is expected to return the home-page of TREC, which is *http://trec.nist.gov*.

Participants in TREC 2001 [16, 13] used several features such as document content, document structure, anchor text, link structure, URL depth, etc. in this task. In the best performing system of University of Twente [16], the authors present a mixture model similar to the one described in section 3.1. Since we are primarily interested in statistical modeling of relevant query feedback, we will confine ourselves to document content and anchor text in our experiments. As such our results are not exactly comparable with those of the TREC 2001 participants.

There are 145 queries and corresponding relevance judgments in this collection. We used the first 75 as training queries and the remaining 70 as test queries. On both the training and test sets, we used the standard language model using the document content as our baseline. We tuned our models on the training set and determined the optimal parameter values and tested them on the test set of queries using the optimal parameter values. Note that the maximum likelihood model does not need a train-test split as the model is tuned on each document based on its feedback queries. However, we still evaluate the performance of the model on the training and test sets separately for fair comparison with other models. We used the *Lemur* toolkit [18] for all our experiments. Preprocessing steps in *Lemur* include pooling in all the anchor text on the links pointing to the document and constructing an index of feedback queries. The representations of the documents are updated based on all four models and retrieval experiments are performed.

We noticed that the dependency and maximum likelihood models are very expensive to perform experiments in a short period of time. Hence we made some simplifying assumptions in our experiments. Since the baseline smoothed unigram model and the maximum likelihood model will retrieve the same documents for a given query, we used the top 250 retrieved documents from the baseline and re-ranked them using the maximum-likelihood model.

	MRR	Top-10	Fail	Opt. parameters
Unigram	24.6	41.3	21.3	$\lambda = 0.9$
	28.6	**52.9**	**14.3**	
Mixture	47.5	68.0	8.0	$\alpha = 0.4, \beta = 0.5$
	46.4	**77.1**	**4.3**	
Dependency	40.4	60.0	21.3	$\alpha = 0.9, \beta = 0.9$
	54.4	**75.7**	**12.9**	
Density Alloc.	41.3	68.0	9.3	$\alpha = 0.9, \beta = 0.9$
	38.7	**71.4**	**8.6**	
Max L'hood	23.7	40.0	24.0	–
	27.9	**51.4**	**15.7**	

Figure 3: Results: bold faced numbers correspond to test set

Similarly, the mixture model and the dependency model retrieve the same documents based on the occurrence of query terms in document content and the feedback queries. Hence, we re-ranked the top 250 documents of mixture model using the dependency model. This results in much faster query processing and allows for more experimentation.

The evaluations are based on three non-independent measures: the Mean Reciprocal Rank (MRR), percentage of queries for which the relevant document is found in the top 10 retrieved documents (Top-10) and percertage of queries for which no relevant document is found in the top 100 retrieved documents (Fail). The best results from all four models and the baseline unigram model on the training and test sets are presented in figures 3. All numbers are in percentages.

We see that all the models except the maximum likelihood model improve performance on the baseline on all three evaluation measures. In particular, the mixture model seems to be the best on the training set with an improvement of 93.1% in *MRR*, 64.6% in *top-10* and a 62.4% drop in failure. However, the dependency model seems to be the best on the test set with an improvement of 90.5% in MRR as compared to an improvement of 62.2% of the mixture model. The maximum likelihood model, on the other hand performs worse than the baseline. Unlike the other models, the maximum likelihood model does not explicitly consider the words in the feedback queries as features in the model. The feedback queries are only implicitly used to update the model's smoothing parameter. We believe this could be a possible reason for the failure of the model.

5. CONCLUSIONS AND FUTURE WORK

In this work, we explored a non-traditional, document centric view of relevance feedback and built a few statistical language models to combine the features of the document's content with those of the relevant queries. We considered anchor text in the web environment as relevant queries and implemented our relevant query feedback models on the home-page finding task of TREC 2001. We have shown using our home-page finding experiments that three of the four models perform significantly better than the baseline.

As part of the future work, we hope to implement our system on the named-page finding task of TREC-2002 [4]. The task is very similar and we believe the results should be comparable. Additionally, we hope to do the 'actual' relevant query feedback experiments in the future by collecting a large collection of queries and relevance judgments.

Acknowledgments

The authors would like to thank Victor Lavrenko for his idea of Density Allocation and Fernando Diaz for his help with the *Lemur* tool kit. This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

6. REFERENCES

[1] Baeza-Yates, R. and Ribeiro-Neto, B., Modern Information Retrieval, *ACM Press*, 1999.

[2] Berger, A. and Lafferty, J., Information Retrieval as Statistical Translation, *SIGIR*, 222-229, 1999.

[3] Hawking, D. and Craswell, N., Overview of the TREC 2001 web track, *TREC proceedings*, 2001.

[4] Hawking, D. and Craswell, N., Overview of the TREC-2002 web track, *TREC proceedings*, 2002.

[5] Jackson, D. M., The construction of Retrieval Environments and Pseudoclassification based on External Relevance, *Information Storage and Retrieval*, vol. 6, no. 2, pp 187-219, 1970.

[6] Kleinberg, J. M., Authoritative sources in a hyperlinked environment, *Journal of the ACM*, vol. 46, no. 5, p604-632, 1999.

[7] Kwok, K.L., A Network Approach to Probabilistic Information Retrieval, *ACM TOIS*, 13:324-353, July 1995

[8] Lavrenko, V., Based on a presentation by Victor Lavrenko, *http://www.cs.umass.edu/ mlfriend/04-03-abstracts/lavrenko.htm*.

[9] Lafferty, J. and Zhai, C., Probabilistic relevance models based on document and query generation, *Language Modeling for Information Retrieval*, Kluwer International Series on Information Retrieval, Vol. 13, 2003.

[10] Page, L., Brin, S., Motwani, R. and Winograd, T., The PageRank Citation Ranking: Bringing Order to the Web, *Technical Report*, Stanford Digital Library Technologies Project, 1998.

[11] Ponte, J. and Croft, W. B., A language modeling approach to Information Retrieval, *ACM SIGIR*, pp. 275-281, 1998.

[12] Robertson, S. E., On Bayesian models and event spaces in Information Retrieval, *SIGIR Workshop on Mathematical/Formal models in Information retrieval*, 2002.

[13] Robertson, S.E., Walker, S. and Zaragoza, Microsoft Cambridge at TREC-10: Filtering and web tracks, *TREC Proceedings*, 2001.

[14] Salton, G., Dynamic Document Processing, *Communications of the ACM*, vol. 15, no.7, pp658-668, 1972.

[15] Salton, G. and McGill, M. J., *Introduction to Modern Information Retrieval*, chapter 4, McGraw-Hill, 1983.

[16] Westerveld, T., Kraaij, W. and Hiemstra, D., Retrieving Web Pages using Content, Links, URLs and Anchors, *Proceedings of the TREC Conference*, 2001.

[17] Yu, C. T. and Raghavan, V.V., A methodology for the construction of term classes, *Information Storage and Retrieval*, vol. 10, no. 7/8, p 243-251, 1974.

[18] The Lemur Toolkit for Language Modeling and Information Retrieval, http://www-2.cs.cmu.edu/ lemur/,