Relevance-Based Language Models: Estimation and Analysis

Victor Lavrenko and W. Bruce Croft
Center for Intelligent Information Retrieval
Department of Comupter Science
University of Massachusetts, Amherst, MA 01003
{lavrenko,croft}@cs.umass.edu

Abstract

It has long been recognized that the primary obstacle to effective performance of classical models is the need to estimate a relevance model with no training data. We propose a novel technique for estimating such models using the query alone. We demonstrate that our technique can produce highly accurate relevance models. Our experiments show relevance models outperforming baseline language modeling systems on TREC retrieval. The main contribution of this work is an effective formal method for estimating a relevance model with no training data.

1 INTRODUCTION

Classical probabilistic models of Information Retrieval [8, 9, 10, 13] are concerned with estimating probability of relevance. The famous *probability ranking principle*, advocated by Robertson in [10], asserts that optimal [1] performance will be achieved if the documents are ranked by the posterior probability that they belong to the *relevant* class R. Robertson [10] also shows that it is equivalent to rank the documents by the odds of their being observed in the relevant class: $P(D|R)/P(D|N)$. If we make a common word independence assumption [8, 14], we can rank the documents by:

$$\frac{P(D|R)}{P(D|N)} \sim \prod_{w \in D} \frac{P(w|R)}{P(w|N)} \qquad (1)$$

One of the main obstacles to effective performance of the classical probabilistic models has been the difficulty of estimating word probabilities in relevant documents: $P(w|R)$. Estimating $P(w|R)$ in a typical retrieval environment is difficult because we have no training data: we are given a query, a large collection of documents and no indication of which documents might be relevant. Faced with the absense of training data, researchers used heuristic estimates for $P(w|R)$. Note that estimating $P(w|N)$ is easier, since we have plenty of training data: for typical queries, almost every document in the collection is non-relevant.

From a language-modeling perspective, Hiemstra [5] suggests that R and N can be thought of as generative language models; $P(w|R)$ and $P(w|N)$ define probabilities of observing a word w in relevant and non-relevant document sets respectively. The main contribution of this paper is a theoretically justified way of estimating $P(w|R)$ when no training data is available.

2 RELEVANCE MODEL

This section describes a way of constructing the probability distribution $P(w|R)$ with no labeled training data. Recall $P(w|R)$ is the relative frequency with which we expect to see the word w during repeated independent random sampling of words from all of the relevant documents. If we had available training data in the form of relevance judgments, estimating $P(w|R)$ could be as simple as counting the number of occurrences of w in the relevant documents and appropriately *smoothing* [4] the counts. However, in a typical retrieval environment we have no training data for R.

We are given a large collection of documents and a user query Q. We do not know which documents comprise the relevant set, but we do know that they are somehow related to Q. We formalize this relationship as follows: we assume that there exists an underlying ***relevance model*** R, which is the source of both the query Q and the documents relevant to Q. Figure 1 shows the relationship graphically: we assume that the query Q and all the documents relevant to Q are random samples from

[1] with respect to a number of widely accepted measures of IR performance, including average precision

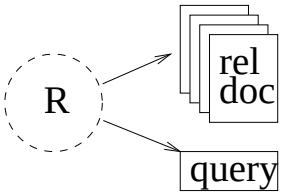

Figure 1: Underlying relevance model R is the source of user query and documents which are relevant to that query.

the relevance model R. This is similar to the assumptions made in [7, 12, 6, 3], but there is a crucial distinction: we don't assume that the query is a sample from any specific document model, instead we assume that both the query and the documents are samples from an unknown relevance model R. In the remainder of this section we show how we can leverage the fact that Q is a random sample from R to learn the parameters of R.

Let $Q = q_1...q_k$. Suppose we play the following game (see Figure 2 for an illustration). We have an unknown model R, from which we can repeatedly sample words. After sampling k times we observe the words $q_1...q_k$. What is the probability that the next word we pull out of R will be w? The only information we have is that we just observed $q_1...q_k$ from R, so our best estimate is:

$$P(w|R) \approx P(w|Q = q_1...q_k) \tag{2}$$

Equation (2) can be thought of as a probability of "translating" the query Q into the word w, similarly to [3]. Note that we are "translating" a set of words into a single word. By definition of conditional probability:

$$P(w|Q) = \frac{P(w,Q)}{P(Q)} \tag{3}$$

In order to estimate the joint probability $P(w,Q)$, we will make the following simplifying assumption. We assume that observations $q_1...q_k$ are independent of each other given w, leading to the following estimate:

$$P(w,Q) = P(w) \prod_{q_i \in Q} P(q_i|w) \tag{4}$$

To calculate the conditional probability $P(q_i|w)$, we compute the expectation over the possible models M that could have generated both q_i and w:

$$P(q_i|w) = \sum_M P(q_i|M)P(M|w) \tag{5}$$

Now we can substitute equation (5) into equation (4) we get the following estimate for the probability of observing w after seeing a sequence $q_1...q_k$:

$$P(w|Q) = \frac{P(w)}{P(Q)} \prod_{q_i \in Q} \left(\sum_M P(q_i|M)P(M|w) \right) \tag{6}$$

Equation (6) is our final estimate for the relevance model. Technical details are provided in the following section. Refer to Table 1 for sample probabilities given by our relevance model for topic titles in the TDT2 dataset.

2.1 Estimation Details

This section provides the final estimation details for our relevance model. To ensure proper additivity of our model, we set the query prior to be:

$$P(Q) = \sum_v P(v,Q) \tag{7}$$

where $P(v,Q)$ is computed according to equation (4). Similarly, we set the word prior to be:

$$P(w) = \sum_M P(w|M)P(M) \tag{8}$$

For efficiency purposes, in equations (5), (6) and (8), we limit the set of models to be the models of documents retrieved by a query using an approach similar to [12]. We restrict the retrieved set to contain 50 to 100 top-ranked documents. We use a linear interpolation technique [6] to smooth our maximum likelihood document models with the global model:

$$P(w|M_D) = \lambda \frac{tf(w,D)}{\sum_v tf(v,D)} + (1-\lambda)P(w|G) \tag{9}$$

Here $tf(w,D)$ is the number of occurrences of w in D, and $P(w|G)$ is just the collection frequency of w divided by the total number of tokens in the collection. We experimented with a number of ways for setting the smoothing parameter λ and finally settled on a simple constant $\lambda = 0.6$ over all words and documents. Our experiments show little variation in performance with λ anywhere between 0.4 and 0.8. The same linear interpolation scheme is used to smooth the final probabilities from equation (6).

2.2 Independence Assumptions

It is important to realize equation (6) is not the only valid formula for estimating the conditional probability $P(w|Q)$. In our derivation we made two independence assumptions: equation (4) assumes that query words $q_1...q_k$ are independent once we fix w, and equation (5) assumes that w is pairwise-independent from every q_i

"Monica Lewinsky Case"		"Israeli Palestinian Raids"		"Rats in Space"		"John Glenn"		"Unabomber"	
$P(w\|Q)$	w	$P(w\|Q)$	w	$P(w\|Q)$	w	$P(w\|Q)$	w	$P(w\|Q)$	w
0.041	lewinsky	0.077	palestinian	0.062	rat	0.032	glenn	0.046	kaczynski
0.038	monica	0.055	israel	0.030	space	0.030	space	0.046	unabomber
0.027	jury	0.034	jerusalem	0.020	shuttle	0.026	john	0.019	ted
0.026	grand	0.033	protest	0.018	columbia	0.016	senate	0.017	judge
0.019	confidant	0.027	raid	0.014	brain	0.015	shuttle	0.016	trial
0.016	talk	0.012	find	0.012	mission	0.011	seventy	0.013	say
0.015	case	0.011	clash	0.012	two	0.011	america	0.012	theodore
0.014	president	0.010	bank	0.011	seven	0.011	old	0.012	today
0.013	clinton	0.010	west	0.010	system	0.010	october	0.011	decide
0.010	starr	0.010	troop	0.010	nervous	0.010	say	0.011	guilty

Table 1: Sample probabilities from the query-based relevance models on the TDT2 dataset and TDT2 topics.

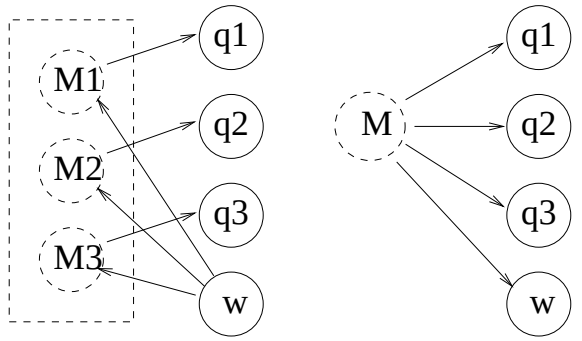

Figure 2: Dependence networks for two ways of estimating $P(w|Q)$. Left: model implied by equation (6). Right: an alternative model, equation (10).

once we fix a generating model M (refer to left side of Figure 2 for illustration).

If we made different independence assumptions, we could arrive at a different formula for $P(w|Q)$. For example, if we assume that w and $q_1 ... q_k$ are all mutually independent once we fix a generating model M, we arrive at the following estimate:

$$P(w, Q) = \sum_M \left(P(M) P(w|M) \prod_{q \in Q} P(q|M) \right) \quad (10)$$

The corresponding dependence diagram is shown on the right side of Figure 2. Note that *mutual* independence assumption leading to equation (10) is a much stronger assumption than *pairwise* independence assumption of equation (6). We tested our model with both equations (6) and (10). Our initial experiments showed that equation (6) consistently results in more robust performance.

2.3 A Brief Summary of the Model

We presented a novel technique for estimating probabilities of words in the unknown set of documents relevant to the query Q. In a nutshell, we approximate the probability of observing the word w in the relevant set by the probability of co-occurrence of w and the query. $P(w|R)$ is what we ultimately want to estimate. We argue that $P(w|Q)$ is a good approximation, and equations (3) through (6) present a formal derivation of this probability of co-occurrence. We use widely accepted estimation techniques in section 2.1.

The main contribution of this work is a formal probabilistic approach to estimating $P(w|R)$, which has been done in a heuristic fashion by previous researchers.

3 EXPERIMENTAL RESULTS

We now turn our attention to evaluating the effectiveness of our method for constructing a model of relevance from query alone. We show that the *probability ranking principle* [10] with our relevance model outperforms the baseline retrieval systems on TREC data. The results in section are obtained on the AP subset of TREC volumes 1 and 2, against two sets of TREC title queries: 101-150 and 151-200. The AP dataset contains over 164,000 Associated Press newswire stories. The relevance assessments are not exhaustive, they are created by pooled evaluations of top-scoring documents from previous TREC runs. In both the documents and the queries were stemmed using a dictionary-augmented stemmer, and a total of 418 stopwords from the standard *InQuery* stoplist were removed [1].

As our baseline we take the performance of the language modeling approach similar to [12], where we rank documents by their probability of generating the query.

Table 2 highlights the improvements. We observe that for both query sets the relevance model provides an improvement over the baseline. Average precision is improved by 29% on the first query set and by 10% on the second. The improvements are statistically significant at several levels of recall according to a one-sided Wilcoxon test (indicated by the stars in Table 2). We also observe a noticeable increase in R-Precision for both query sets.

4 CONCLUSIONS

In this paper we addressed the major challenge faced by a researcher in a classical probabilistic framework of IR: the need to estimate a relevance model with no training data. We proposed a novel technique for estimating such models. Our method produces accurate models of relevance, which leads to significantly improved retrieval performance, when applied in the context of classical probabilistic models with modern estimation techniques. The model outperforms baseline language modeling approaches and on average performs as well as their expanded versions.

The main contribution of our work is a formal probabilistic approach to estimating a relevance model with no training data. This has been done in a heuristic fashion in the past, and may have stifled the performance of classical probabilistic approaches. The experiments show that with our estimate of the relevance model, classical probabilistic models of retrieval outperform state-of-the-art heuristic and language modeling approaches.

5 ACKNOWLEDGMENTS

This material is based on work supported in part by the Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623, and in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

References

[1] J. Allan, J. Callan, F. Feng, and D. Malin. INQUERY and TREC-8. In D. Harman, editor, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, 1999.

[2] J. Allan, R.Papka, and V.Lavrenko. On-line new event detection and tracking. In *Proceedings of ACM SIGIR, pp 37-45*, 1998.

[3] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings on the 22nd annual international ACM SIGIR conference*, pages 222–229, 1999.

[4] S. F. Chen and J. T. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the ACL*, 1996.

[5] D. Hiemstra and A. de Vries. Relating the new language models of information retrieval to the traditional retrieval models. In *CTIT Technical Report TR-CTIT-00-09*, 2000.

[6] D. Miller, T. Leek, and R. Schwartz. A hidden markov model information retrieval system. In *Proceedings on the 22nd annual international ACM SIGIR conference*, pages 214–221, 1999.

[7] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings on the 21st annual international ACM SIGIR conference*, pages 275–281, 1998.

[8] S. Robertson and K. S. Jones. Relevance weighting of search terms. In *Journal of the American Society for Information Science, vol.27*, 1977.

[9] S. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference*, pages 232–241, 1996.

[10] S. E. Robertson. *The Probability Ranking Principle in IR*, pages 281–286. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1997.

[11] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. OKAPI at TREC-3. In D. Harman, editor, *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*, 1996.

[12] F. Song and W. B. Croft. A general language model for information retrieval. In *Proceedings on the 22nd annual international ACM SIGIR conference*, pages 279–280, 1999.

[13] H. Turtle and W. B. Croft. Efficient probabilistic inference for text retrieval. In *Proceedings of RIAO 3*, pages 644–651, 1991.

[14] C. J. van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33:106–119, 1977.

TREC queries 101-150 (title)					TREC queries 151-200 (title)				
	LM	Rel.M	%chg	Wilc.		LM	Rel.M	%chg	Wilc.
Rel	4805	4805			Rel	4933	4933		
Rret	2981	3733	+25.23	0.0156*	Rret	3288	3222	-2.01	0.0367*
0.00	0.6132	0.6161	+0.5	0.4524	0.00	0.7699	0.7248	-5.9	0.1697
0.10	0.4090	0.4686	+14.6	0.0651	0.10	0.5669	0.5913	+4.3	0.1524
0.20	0.3267	0.4066	+24.5	0.0042*	0.20	0.4494	0.5201	+15.7	0.0084*
0.30	0.2815	0.3562	+26.6	0.0035*	0.30	0.3628	0.4797	+32.2	0.0005*
0.40	0.2277	0.3171	+39.3	0.0001*	0.40	0.3239	0.4090	+26.3	0.0096*
0.50	0.1922	0.2803	+45.8	0.0000*	0.50	0.2596	0.3258	+25.5	0.0256*
0.60	0.1579	0.2393	+51.6	0.0001*	0.60	0.2187	0.2649	+21.1	0.0400*
0.70	0.1094	0.1799	+64.5	0.0027*	0.70	0.1772	0.1852	+4.5	0.2976
0.80	0.0693	0.1205	+74.0	0.0411*	0.80	0.1436	0.1134	-21.0	0.1327
0.90	0.0441	0.0578	+30.8	0.3576	0.90	0.1048	0.0561	-46.5	0.0172*
1.00	0.0267	0.0113	-57.7	0.0372*	1.00	0.0319	0.0165	-48.2	0.0571
Avg	0.2021	0.2617	+29.50	0.0017*	Avg	0.2878	0.3182	+10.55	0.0971
5	0.3840	0.4240	+10.4	0.1707	5	0.5400	0.5200	-3.7	0.2552
10	0.3760	0.3940	+4.8	0.3001	10	0.4880	0.4980	+2.0	0.3288
15	0.3480	0.3880	+11.5	0.1112	15	0.4640	0.4907	+5.7	0.1891
20	0.3260	0.3810	+16.9	0.0610	20	0.4430	0.4690	+5.9	0.0890
30	0.3007	0.3520	+17.1	0.0214*	30	0.4000	0.4287	+7.2	0.0522
100	0.2104	0.2652	+26.0	0.0013*	100	0.2532	0.2832	+11.8	0.1288
200	0.1527	0.1971	+29.1	0.0013*	200	0.1835	0.1992	+8.6	0.3631
500	0.0954	0.1171	+22.7	0.0217*	500	0.1086	0.1097	+1.1	0.7538
1000	0.0596	0.0747	+25.2	0.0156*	1000	0.0658	0.0644	-2.0	0.0367*
RPr	0.2546	0.2935	+15.27	0.0056*	RPr	0.3212	0.3519	+9.56	0.0638

Table 2: Comparison of Relevance Model (Rel.M) to the Language Modeling (LM) on the AP subset of TREC. Stars indicate statistically significant differences in performance with a 95% confidence according to the Wilcoxon test.