Relevance-Based Language Models

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

We explore the relation between classical probabilistic models of information retrieval and the emerging language modeling approaches. It has long been recognized that the primary obstacle to effective performance of classical models is the need to estimate a *relevance model*: probabilities of words in the relevant class. We propose a novel technique for estimating these probabilities using the query alone. We demonstrate that our technique can produce highly accurate relevance models, addressing important notions of synonymy and polysemy. Our experiments show relevance models outperforming baseline language modeling systems on TREC retrieval and TDT tracking tasks. The main contribution of this work is an effective formal method for estimating a relevance model with no training data.

1. INTRODUCTION

Recent developments in the conceptual view of Information Retrieval marked a departure from the traditional models of relevance and the emergence of *language modeling* frameworks, introduced by Ponte and Croft [16]. A number of following publications [5, 14, 21, 10], adopted a similar framework, introducing refinements in parameter estimation, the use of multi-word features and expansion techniques. A common theme in these models is that they abandon explicit models of relevance, instead attempting to model the query generation process. These approaches model query generation as random sampling from one of the document models. During the retrieval process, the documents are ranked by the probability that a query would be observed as a random sample from the respective document model.

Earlier work on probabilistic models of information retrieval [19, 18, 17, 22] took a conceptually different approach. Researchers explicitly attempted to model word occurrences in relevant and non-relevant classes of documents, and used their models to classify the document into the more likely class. For retrieval purposes, the documents were ranked by the probability that they belong to the relevant class.

SIGIR'01, September 9-12, 2001, New Orleans, Louisiana, USA.

The difference between the two approaches is not superficial, but is in fact quite important. Traditional probabilistic frameworks allow for arbitrarily rich and complex query models (e.g. [22]), whereas the recent language modeling approaches treat the query as a fixed sample of text with little room for modeling. Many popular techniques in Information Retrieval, such as relevance feedback and automatic query expansion have a very intuitive interpretation in the traditional probabilistic models, but are conceptually difficult to integrate into the language modeling framework of [16] because they involve augmenting the sample (as in [15]) rather than adjusting the model. Furthermore, explicit models of relevance are better suited to other information organization tasks, such as summarization [6], question answering [4], topic detection and tracking (TDT) [25, 12] and text segmentation [3].

The primary obstacle to constructing effective models of relevance is the lack of training data. In a typical retrieval environment we are given a query, a large collection of documents and no indication of which documents might be relevant. In this paper we introduce a technique for constructing a relevance model from the query alone and compare resulting query-based relevance models to language models constructed with training data. We also evaluate the retrieval effectiveness of query-based relevance models on two distinct tasks: the *ad-hoc* retrieval task of TREC [1], and the *tracking* task of TDT [2]. The remainder of the paper is structured as follows. We briefly survey the classical relevance-based probabilistic models as well as the recent language modeling approaches in section 2. Our method for formally constructing and applying relevance models is detailed in section 3. Section 4 provides a thorough evaluation of query-based relevance models in terms of cross-entropy, retrieval effectiveness and tracking accuracy.

2. RELATED WORK

There are two directions of information retrieval research that provide a theoretical foundation for our model: the now classic work on probabilistic models of relevance, and the recent developments in language modeling techniques for IR. To the former we owe the concept of a relevance model: a language model representative of a class of relevant documents. To the latter we credit the methods of modeling and specific estimation techniques. In this section we give a brief survey of several developments in both of these directions, highlighting interesting connections between the two.

2.1 Classical Probabilistic Models

Underlying most research on probabilistic models of Information Retrieval is the *probability ranking principle*, detailed by Robertson in [19], which suggests ranking the documents D by the odds of their being observed in the relevant class: $P(D|R)/P(D|N)$.

Here R represents the class of documents relevant to user's query, and N is the class of non-relevant documents. At this point we can make an important connection to the recent language modeling approaches: according to Hiemstra [11], R and N can be identified with generative language models, where $P(w|R)$ and $P(w|N)$ define probabilities of observing a word w in relevant and non-relevant document sets respectively.

The estimation of $P(D|R)$ differs in various models. The Binary Independence Model [17, 23] treats each document as a binary vector over the vocabulary space, ignoring word frequencies. In language modeling terms, this means that R and N can be viewed as *multiple-Bernoulli* language models:

$$P(D|R) = \prod_{w \in D} P(w|R) \prod_{w \notin D} (1 - P(w|N)) \qquad (1)$$

Here $P(w|R)$ are the probabilities of the word w being present in a document sampled from the relevant class. These probabilities are estimated using heuristic techniques in the absence of relevance information. $P(D|N)$ is calculated in a similar manner.

The 2-Poisson Model [18] goes a step further in modeling term frequencies in documents according to a mixture of two Poisson distributions. This implies that R could be viewed as a multiple-Poisson language model. The Inference Network Model [22] treats R as a feed-forward Bayesian belief network model, with parameters estimated in a heuristic manner.

Estimation differences aside, a common feature of the classical probabilistic methods in IR is their explicit notion of a class of relevant documents R, and their attempt to estimate word probabilities $P(w|R)$ in this class.

2.2 Language Modeling Approaches

Recent work in conceptual models of Information Retrieval shifted away from explicit models of relevance and focused on viewing documents themselves as models and queries as strings of text randomly sampled from these models. Most of these approaches rank the documents in the collection by the probability that a query Q would be observed during repeated random sampling from the document model M_D: $P(Q|M_D)$.

The calculation of the above probability differs significantly from model to model. Ponte and Croft [15, 16], treat the query Q as a binary vector over the entire vocabulary, leading to a *multiple-Bernoulli* view of model M_D:

$$P(Q|M_D) = \prod_{w \in Q} P(w|M_D) \prod_{w \notin Q} (1 - P(w|M_D)) \qquad (2)$$

Note the similarity of this model to the Binary Independence Model in equation (1). The major difference comes in estimation of individual word probabilities. Ponte and Croft used an intricately smoothed non-parametric estimate for $P(w|M_D)$. Accurate estimation of the model is possible because we have an example document D, which we can use to construct its model M_D. Recall that this step is precisely the stumbling block of traditional probabilistic models: it is hard to estimate the relevance model R with no training data.

Miller et al. [14], Song and Croft [21], and Hiemstra [10] choose to treat the query Q as a *sequence* of independent words, rather than a binary vector, leading to a *multinomial* view of model M_D:

$$P(Q|M_D) = \prod_{w} P(w|M_D)^{q_w} \qquad (3)$$

Here q_w is the number of times the word w occurs in the query (this was restricted to 0 or 1 in the model of Ponte and Croft [16]). The probabilities $P(w|M_D)$ are taken to be smoothed relative frequencies of words from the document D itself. In section 3 we will use the same query representation, but in a completely different formalism.

Berger and Lafferty [5] view the query Q as a potential *translation* of the document D, and use powerful estimation techniques detailed in [7] and synthetic training data to compute $P(Q|M_D)$. A major drawback to a wide acceptance of the translation model is its requirement for training data and the complexity of parameter estimation. It is worth noting that the translation model of Berger and Lafferty naturally includes a rudimentary expansion component when they consider non-diagonal word-for-word translation probabilities. Our model performs a similar expansion, with a distinction that word probabilities are conditioned on a set of words and not on individual words as in [5].

A common theme in these approaches is their shift away from trying to model relevance and towards careful estimation of the sampling probabilities. A notable exception is the recent work on supervised topic models for Topic Detection and Tracking (TDT) tasks. Researchers [12, 25, 26] used language modeling techniques to construct highly effective topic models from a small number of training stories. In section 4.3 we show that we can construct accurate topic models from just the topic title and no training stories.

3. A FORMAL RELEVANCE MODEL

We use the term **relevance model** to refer to a mechanism that determines the probability $P(w|R)$ of observing a word w in the documents relevant to a particular information need.

One of the main obstacles to effective performance of the classical probabilistic models has been precisely the challenge of estimating the relevance model. 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 absence of training data, researchers used heuristic estimates for $P(w|R)$, leading to models that are difficult to interpret. 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.

The main contribution of this paper is a theoretically justified way of estimating the relevance model when no training data is available. Section 3.1 briefly reviews how we could use the probabilities $P(w|R)$ in Information Retrieval and Topic Tracking tasks. Section 3.2 proposes two approaches for estimating the relevance model with no training data. Section 3.3 provides the details of estimation and smoothing.

3.1 Ranking with a Relevance Model

Suppose we have an accurate model of relevance R, and would like to rank a set of documents to be presented to a user. The famous *probability ranking principle*, advocated by Robertson in [19], asserts that optimal [1] performance will be achieved if the documents

[1] with respect to a number of widely accepted measures of IR per-

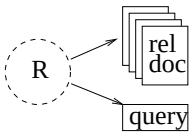

Figure 1: Queries and relevant documents are random samples from an underlying relevance model R. Note: the sampling process could be different for queries and documents.

are ranked by the posterior probability that they belong to the *relevant* class R. Robertson [19] 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 [17, 23], we can rank the documents by:

$$\frac{P(D|R)}{P(D|N)} \sim \prod_{w \in D} \frac{P(w|R)}{P(w|N)} \tag{4}$$

Here we made an implicit assumption that a document is represented as a sequence of words, similar to [12, 25, 26]. This contrasts the more traditional vector representation used in [17, 19, 18] and subsequently [16].

3.2 Approximating a Relevance Model

This section describes a way of constructing the probability distribution $P(w|R)$ when no labeled training data is available. 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* [8] the counts. However, in a typical retrieval setting we are given 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 for every information need there exists an underlying relevance model R, which assigns the probabilities $P(w|R)$ to the word occurrence in the relevant documents. The relevance model also assigns probabilities $P(Q|R)$ to the various queries that might be issued by the user for that specific information need. Figure 1 shows the relationship graphically: we assume that the relevant documents are random samples from the distribution $P(w|R)$. The query Q is also a sample from R. However, we would like to stress that the sampling process that generates the queries does not have to be the same as the process that generates the words in the relevant documents. In other words, the probability of a one-word query "w" under R need not be the same as the probability of observing w in a random relevant document.

Our assumptions about query generation are similar to the assumptions made in [16, 21, 14, 5], 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.

formance, including average precision

Let $Q = q_1 \ldots q_k$. Suppose we play the following game. We have an unknown process R, a black box, from which we can repeatedly sample words. After sampling k times we observe the words $q_1 \ldots q_k$. 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 $q_1 \ldots q_k$, so our best bet is to relate the probability of w to the conditional probability of observing w given that we just observed $q_1 \ldots q_k$:

$$P(w|R) \approx P(w|q_1 \ldots q_k) \tag{5}$$

By definition, we can express the conditional probability in terms of the joint probability of observing w with the query words $q_1 \ldots q_k$:

$$P(w|R) \approx \frac{P(w, q_1 \ldots q_k)}{P(q_1 \ldots q_k)} \tag{6}$$

The challenge now lies in estimating the joint probability of observing the word w together with the query words $q_1 \ldots q_k$. We advocate two methods for estimating this probability. The first method is conceptually simpler, and assumes that w was sampled in the same way as the query words. The second method assumes that w and the query words were sampled using two different mechanisms. The two methods differ in the independence assumptions that are being made, and we try to highlight this in the derivations.

3.2.1 Method 1: i.i.d. sampling

Let's assume that the query words $q_1 \ldots q_k$ and the words w in relevant documents are sampled identically and independently from a unigram distribution M_R. Let $\mathcal{M}$ represent some finite universe of unigram distributions from which we could sample. The sampling process proceeds as follows: we pick a distribution $M \in \mathcal{M}$ with probability $P(M)$, and sample from it $k+1$ times. Then the total probability of observing w together with $q_1 \ldots q_k$ is:

$$P(w, q_1 \ldots q_k) = \sum_{M \in \mathcal{M}} P(M) P(w, q_1 \ldots q_k | M) \tag{7}$$

Because we assumed that w and all q_i are sampled independently and identically to each other, we can express their joint probability as the product of the marginals:

$$P(w, q_1 \ldots q_k | M) = P(w|M) \prod_{i=1}^{k} P(q_i|M) \tag{8}$$

When we substitute equation (8) into equation (7), we get the following final estimate for the joint probability of w and $q_1 \ldots q_k$:

$$P(w, q_1 \ldots q_k) = \sum_{M \in \mathcal{M}} P(M) P(w|M) \prod_{i=1}^{k} P(q_i|M) \tag{9}$$

Note that in the course of this derivation we made a very strong independence assumption: in equation (8) we assumed that w and $q_1 \ldots q_k$ are mutually independent once we pick a distribution M. A graphical diagram of the dependencies between the variables involved in the derivation is shown on the left side of Figure (2).

3.2.2 Method 2: conditional sampling

Now let's consider a different approach to sampling. We fix a value of w according to some prior $P(w)$. Then perform the following process k times: pick a distribution $M_i \in \mathcal{M}$ according to $P(M_i|w)$, the sample the query word q_i from M_i with probability

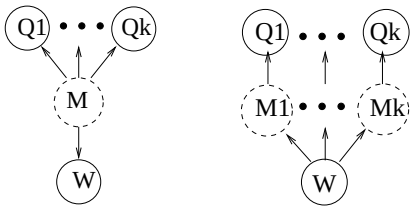

Figure 2: Dependence networks for two methods of estimating the joint probability $P(w, q_1 \ldots q_k)$. Left: i.i.d. sampling (method 1). Right: conditional sampling (method 2).

$P(q_i|M_i)$. A graphical diagram of the sampling process is given on the right side of Figure (2).

The effect of this sampling strategy is that we assume the query words $q_1 \ldots q_k$ to be independent of each other, but we keep their dependence on w:

$$P(w, q_1 \ldots q_k) = P(w) \prod_{i=1}^{k} P(q_i|w) \tag{10}$$

To estimate the conditional probabilities $P(q_i|w)$ we compute the expectation over the universe $\mathcal{M}$ of our unigram models.

$$P(q_i|w) = \sum_{M_i \in \mathcal{M}} P(M_i|w) P(q_i|M_i) \tag{11}$$

Note that we made an additional assumption that q_i is independent of w once we picked a distribution M_i. When we substitute the result of equation (11) into equation (10), we get the following final estimate for the joint probability of w and $q_1 \ldots q_k$:

$$P(w, q_1 \ldots q_k) = P(w) \prod_{i=1}^{k} \sum_{M_i \in \mathcal{M}} P(M_i|w) P(q_i|M_i) \tag{12}$$

3.2.3 *Comparison of the two methods*

It is informative to contrast the assumptions made in the two methods we proposed for estimating the joint $P(w, q_1 \ldots q_k)$. By Bayes' rule, Method 1 can also be viewed as sampling of $q_1 \ldots q_k$ conditioned on w. However, for Method 1, this would add an additional constraint that all query words q_i are sampled from the same distribution M, whereas in Method 2 we are free to pick a separate M_i for every q_i. We believe that Method 1 makes a stronger mutual independence assumption in equation (8), compared to a series of pairwise independence assumptions made by Method 2 in equation (11). In empirical evaluations, we found that Method 2 is less sensitive to the choice of our universe of distributions $\mathcal{M}$. Method 2 also performs slightly better in terms of retrieval effectiveness and tracking errors.

Given its robustness and superior performance, we use Method 2 for estimating joint probabilities in all the following experiments. Refer to Table 1 for a sample of probabilities generated by our relevance model on the TDT2 dataset.

3.3 Final Estimation Details

This section provides the final estimation details for our relevance model. The estimation choices in this section are either dictated by the axioms of probability theory, or represent techniques that are well-known and accepted by the language modeling community. To ensure proper additivity of our model, we set the query prior $P(q_1 \ldots q_k)$ in equation (6) to be:

$$P(q_1 \ldots q_k) = \sum_{w} P(w, q_1 \ldots q_k) \tag{13}$$

where the summation is performed over all words w in the vocabulary, and $P(w, q_1 \ldots q_k)$ is computed according to equation (12). Similarly, we set the word prior $P(w)$ to be:

$$P(w) = \sum_{M \in \mathcal{M}} P(w|M) P(M) \tag{14}$$

For reasons having to do with both effectiveness and efficiency, we limit our universe $\mathcal{M}$ of possible unigram distributions to contain only the document models (in the sense used by Song [21]). We further restrict $\mathcal{M}$ to only contain 50 models M_D corresponding to the top-ranked documents retrieved by the query $q_1 \ldots q_k$. For retrieval, we use a baseline language modeling approach to IR similar to [21]. We use a linear interpolation technique [14] to smooth our maximum likelihood document models with the background model of English:

$$P(w|M_D) = \lambda \frac{tf(w, D)}{\sum_v tf(v, D)} + (1 - \lambda) P(w|G) \tag{15}$$

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).

For Method 1 we arbitrarily choose to use unigram distribution priors $P(M)$. For Method 2, we choose to estimate the conditional probabilities of picking a distribution M_i based on w as follows:

$$P(M_i|w) = P(w|M_i) P(w) / P(M_i) \tag{16}$$

Here $P(w|M_i)$ is calculated according to equation (15), $P(w)$ is computed as in equation (14), and $P(M_i)$ is kept uniform over all the distributions in $\mathcal{M}$.

3.4 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 $q_1 \ldots q_k$. In a nutshell, we approximate the probability of observing the word w in the relevant set by the probability of co-occurrence between w and the query. $P(w|R)$ is what we ultimately want to estimate. We argue that $P(w|q_1 \ldots q_k)$ is a good approximation in the absence of any training data. We present two formal derivations of this probability of co-occurrence. We use widely accepted estimation techniques in section 3.3.

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.

"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.

4. EXPERIMENTAL RESULTS

We now turn our attention to evaluating the effectiveness of our method for constructing a model of relevance from query alone. We present three types of evaluation. First, we measure the cross-entropy of our model with the "true" model of relevance. Then we show that a ranking method presented in section 3.1 with our relevance model outperforms the baseline retrieval systems on TREC data. Finally, we demonstrate that our model of relevance can be successfully used for topic tracking in the context of TDT [2].

The experiments in sections 4.1 and 4.3 are carried out on the TDT2 dataset with 96 LDC-judged topics. The relevance judgments are exhaustive for all 96 topics in TDT2: every document is manually judged to be relevant or not relevant to every one of the 96 selected topics. The TDT2 dataset contains roughly 63,000 broadcast news and newswire stories, spanning 6 consecutive months in 1998. The topics are centered around a specific event, and so are much more focused than typical TREC topics. See [9] for a detailed description of the TDT corpora and topics. For queries, we used the short 2-3 word titles assigned by the LDC annotators.

The results in section 4.2 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 all cases 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].

4.1 Model cross-entropy

Availability of relevance judgments allows us to create a very good approximation to the "true" relevance model R – one that is constructed directly from all the relevant documents in the collection. We can then measure the cross-entropy between this "true" model R and the approximation we construct in Section 3.2. Cross-entropy is an information-theoretic measure of distance between two distributions, measured in bits. Minimizing cross-entropy in some cases leads to improved performance.

Figure 3 shows the average cross-entropy of the Relevance Model against the "true" relevance model R. For comparison, we show both estimation methods (1) and (2). Cross-entropy is plotted as a function of n, the number of top-ranked document models we include in our universe $\mathcal{M}$. Both models are smoothed in the same manner, as described in section 3.3. Both models exhibit lowest cross-entropy around 50 documents, however the model estimated with Method 2 achieves lower absolute cross-entropy, and also deteriorates slower as we add more and more document models to our universe $\mathcal{M}$. This suggests that using Method 2 is considerably more robust than Method 1. The dotted horizontal line marks the cross-entropy of the model constructed from a single relevant document. Note that our relevance model achieves a lower cross-entropy. In section 4.3 we show that this difference translates to our relevance model outperforming a typical TDT tracking system.

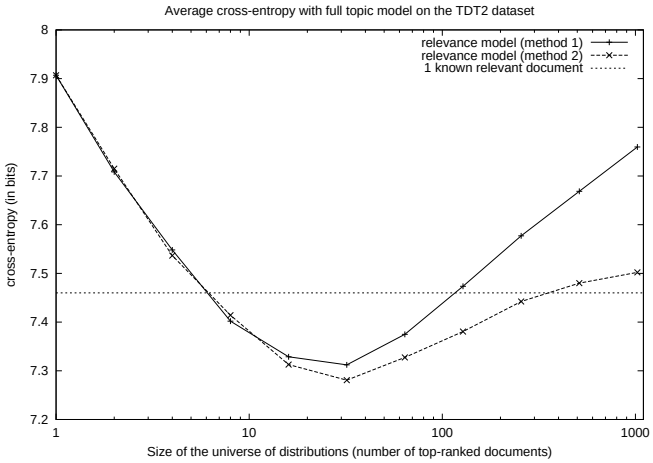

Figure 3: Estimation using Method 2 results in lower cross-entropy with the true topic model. Note that cross-entropy is lower than what can be achieved by simply smoothing one training example.

4.2 TREC ad-hoc retrieval

Our next set of experiments evaluates the performance of relevance models on the TREC ad-hoc retrieval task. We evaluate performance on two query sets: TREC title queries 101-150 and 151-200. As our baseline we take the performance of the language modeling approach similar to [21], where we rank documents by their probability of generating the query (equation 3). Performance of the heuristic $tf.idf$[2] approach is very similar to the baseline language model in our experiments, and so is not shown to avoid excessive clutter. Figure 4 illustrates the comparisons and Table 2 highlights

[2]We used OKAPI [20] *tf* formula with InQuery [1] *idf* formula.

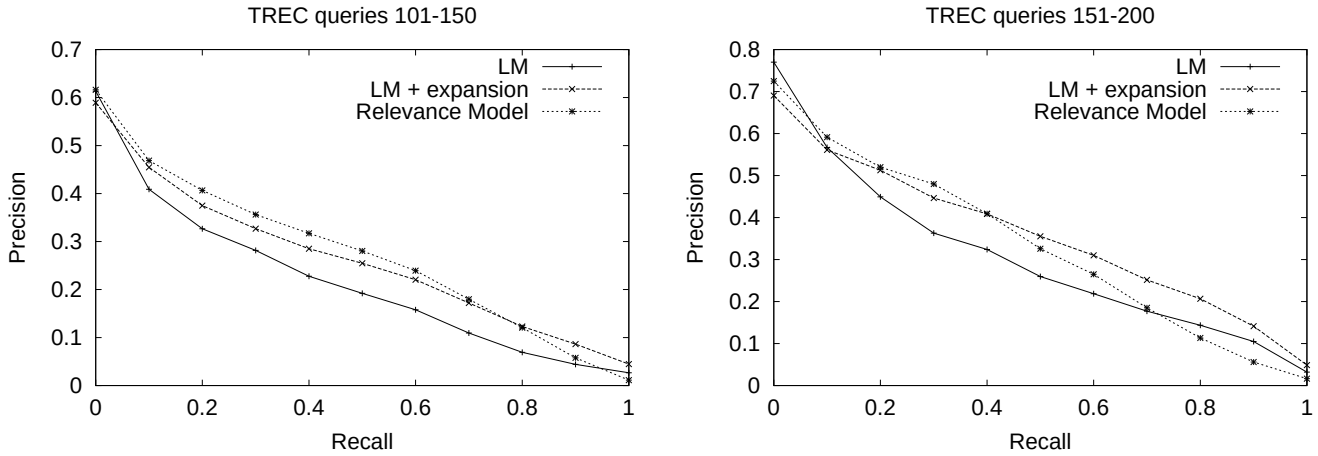

Figure 4: Retrieval performance of relevance models on the AP dataset with TREC title queries. Relevance model outperforms the baseline language modeling approach (LM). On average, relevance model performs roughly as well as a language model augmented with query expansion.

TREC queries 101-150 (title)

	LM	Rel.M	%chg	Wilc.
Rel	4805	4805		
Rret	2981	3733	+25.23	0.0156*
0.00	0.6132	0.6161	+0.5	0.4524
0.10	0.4090	0.4686	+14.6	0.0651
0.20	0.3267	0.4066	+24.5	0.0042*
0.30	0.2815	0.3562	+26.6	0.0035*
0.40	0.2277	0.3171	+39.3	0.0001*
0.50	0.1922	0.2803	+45.8	0.0000*
0.60	0.1579	0.2393	+51.6	0.0001*
0.70	0.1094	0.1799	+64.5	0.0027*
0.80	0.0693	0.1205	+74.0	0.0411*
0.90	0.0441	0.0578	+30.8	0.3576
1.00	0.0267	0.0113	-57.7	0.0372*
Avg	0.2021	0.2617	+29.50	0.0017*
5	0.3840	0.4240	+10.4	0.1707
10	0.3760	0.3940	+4.8	0.3001
15	0.3480	0.3880	+11.5	0.1112
20	0.3260	0.3810	+16.9	0.0610
30	0.3007	0.3520	+17.1	0.0214*
100	0.2104	0.2652	+26.0	0.0013*
200	0.1527	0.1971	+29.1	0.0013*
500	0.0954	0.1171	+22.7	0.0217*
1000	0.0596	0.0747	+25.2	0.0156*
RPr	0.2546	0.2935	+15.27	0.0056*

TREC queries 151-200 (title)

	LM	Rel.M	%chg	Wilc.
Rel	4933	4933		
Rret	3288	3222	-2.01	0.0367*
0.00	0.7699	0.7248	-5.9	0.1697
0.10	0.5669	0.5913	+4.3	0.1524
0.20	0.4494	0.5201	+15.7	0.0084*
0.30	0.3628	0.4797	+32.2	0.0005*
0.40	0.3239	0.4090	+26.3	0.0096*
0.50	0.2596	0.3258	+25.5	0.0256*
0.60	0.2187	0.2649	+21.1	0.0400*
0.70	0.1772	0.1852	+4.5	0.2976
0.80	0.1436	0.1134	-21.0	0.1327
0.90	0.1048	0.0561	-46.5	0.0172*
1.00	0.0319	0.0165	-48.2	0.0571
Avg	0.2878	0.3182	+10.55	0.0971
5	0.5400	0.5200	-3.7	0.2552
10	0.4880	0.4980	+2.0	0.3288
15	0.4640	0.4907	+5.7	0.1891
20	0.4430	0.4690	+5.9	0.0890
30	0.4000	0.4287	+7.2	0.0522
100	0.2532	0.2832	+11.8	0.1288
200	0.1835	0.1992	+8.6	0.3631
500	0.1086	0.1097	+1.1	0.7538
1000	0.0658	0.0644	-2.0	0.0367*
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.

the differences. 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.

We also compare our relevance model to an unsupervised query expansion technique, proposed by Ponte [15] as an augmentation of his original model. The expansion process adds 5 best words from the 5 top retrieved documents to the query (the numbers we selected to give the highest possible average precision). The words are ranked by the following score:

$$s(w) = \sum_{D} log \frac{P(w|M_D)}{P(w|G)}$$

as suggested in [15]. The results are also shown in Figure 4. We observe that the relevance model outperforms the expanded language model on the first query set, but performs worse on the second set. It is worth noting, however, that the relevance model always performs better than the expanded language model at low recall. This makes relevance models an attractive choice in high-precision applications.

4.3 TDT topic tracking

Our final set of experiments addresses a modification of the TDT topic tracking task. In the TDT tracking task, a system is given a small number Nt of stories known to be relevant to some topic, and required to "track" the topic, i.e. report to the user all subsequent stories that discuss the same topic. The task bears a lot of similarities to the TREC filtering task, with two distinctions: (i) there is no initial query, and (ii) the testing set is different for every topic. The tracking task in TDT is evaluated using Detection Error Tradeoff (DET) curves [13] (see figure 5). A DET curve exposes the tradeoff between two types of classification errors: Misses and False Alarms. DET curves are a modification of ROC curves popular in classification literature.

We wanted to test whether a relevance model constructed with no training examples could perform as well as a state-of-the-art tracking system that uses relevance information. To make the comparison easier, we ran the tracking system on a modified task, where the test set was the entire TDT2 corpus, including the training stories. This may artificially boost the performance of supervised systems, but does not improve the query-based relevance model, since it does not use training examples. Both systems used a length-normalized version of the probability ranking principle [19].

The results are presented in Figure 5. The tracking performance we show is typical for TDT tracking systems with 1 to 4 training examples [2, 12, 25, 26]. Surprisingly, a relevance model that uses no training information outperforms the TDT tracking system at $Nt = 1$, and performs almost as well as tracking with $Nt = 2$. Both the relevance model and tracking with $Nt = 2$ achieve a 10% miss rate with a false alarm rate of 1%. The reason may be that the topic titles, which we used as queries, are excellent summaries of each topic. However, in our experiments the language modeling approach of [21, 16] with the title used as a query resulted in very poor tracking performance. Adding query expansion improved performance slightly, but not enough to match the tracking system with $Nt = 1$.

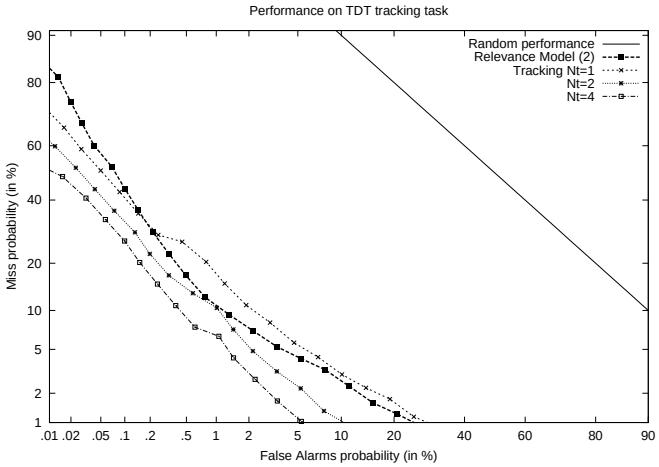

Figure 5: Comparison of Relevance Model to supervised topic models. Relevance Model achieves better performance than a supervised model with one training example.

5. DISCUSSION

There are a number of ways to interpret the technique we proposed in section 3.2. From a traditional IR perspective, our method is a massive query expansion technique. When compared to other query expansion techniques [15, 24], our method is attractive because it does not require careful tuning of parameters. For example, the query expansion method in [15] is quite sensitive to the number of top documents used for expansion and to the number of words added to the query. In contrast to that, our method completely replaces the original query with a distribution over the entire vocabulary. Our technique is also relatively insensitive to the number of documents used in "expansion".

From a language modeling point of view, our method provides an elegant way of estimating a language model when we have no training examples except a very short (2-3 word) description. Our technique could prove useful in a number of applications where it is desirable to bias a general language model towards a specific topic. When compared with more elaborate language modeling approaches (e.g. [5]), our model is simple to implement and does not require any training data.

There are a number of interesting directions for further investigation of relevance models. The good performance of unsupervised relevance model on the TDT tracking task makes it a worthwhile alternative to current tracking systems. We plan to extend the model to take advantage of training data, which is available to tracking systems. Another direction for exploration is the use of different smoothing techniques. In preliminary experiments we were able to achieve higher performance by using a different type of smoothing on the document models.

6. CONCLUSIONS

We discussed a model of retrieval that bridges a gap between the classical probabilistic models of information retrieval, and the emerging language modeling approaches. We suggested why classical models with their explicit notion of relevance may potentially be more attractive than models that limit queries to being a sample of text. We highlighted the major difficulty faced by a researcher in classical framework: the need to estimate a relevance model with

no training data, and 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. We also demonstrated that unsupervised Relevance Models can be competitive with supervised topic models in TDT, outperforming a state-of-the-art tracking system at $Nt = 1$.

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.

7. 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, in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912, and in part by the Advanced Research and Development Activity in Information Technology (ARDA) under its Statistical Language Modeling for Information Retrieval Research Program, contract number MDA904-00-C-2106. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

8. 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] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. In *Machine Learning, vol.34*, pages 1–34, 1999.

[4] A. Berger, R. Caruana, D.Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: Statistical approaches to answer-finding. In *Proceedings of SIGIR*, pages 192–199, 2000.

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

[6] A. Berger and V. Mittal. OCELOT: a system for summarizing web pages. In *Proceedings of SIGIR*, pages 144–151, 2000.

[7] P. Brown, S. D. Pietra, V. D. Pietra, and R. Mercer. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics, 19(2)*, pages 263–311, 1993.

[8] 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.

[9] C. Cieri, D.Graff, M.Liberman, N.Martey, and S.Strassel. The TDT-2 text and speech corpus. In *Proceedings of the DARPA Broadcast News Workshop, pp 57-60*, 1999.

[10] D. Hiemstra. Using language models for information retrieval. In *PhD Thesis, University of Twente*, 2001.

[11] 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.

[12] H. Jin, R. Schwartz, S. Sista, and F. Walls. Topic tracking for radio, TV broadcast and newswire. In *Proceedings of DARPA Broadcast News Workshop, pp 199-204*, 1999.

[13] A. Martin, G. Doddington, T. Kamm, and M. Ordowski. The DET curve in assessment of detection task performance. In *EuroSpeech*, pages 1895–1898, 1997.

[14] 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.

[15] J. Ponte. *A Language Modeling Approach to Information Retrieval*. PhD thesis, Dept. of Computer Science, University of Massachusetts, Amherst, 1998.

[16] 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.

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

[18] 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.

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

[20] 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.

[21] 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.

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

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

[24] J. Xu and W. B. Croft. Improving the effectiveness of informational retrieval with local context analysis. In *ACM TOIS, vol. 18, no. 1*, pages 79–112, January 2000.

[25] J. Yamron, I. Carp, L. Gillick, S.Lowe, and P. van Mulbregt. Topic tracking in a news stream. In *Proceedings of DARPA Broadcast News Workshop, pp 133-136*, 1999.

[26] J. Yamron, S. Knecht, and P. van Mulbregt. Dragon's tracking and detection systems for the TDT2000 evaluation. In *Proceedings of Topic Detection and Tracking Workshop, pp 75-80*, 2000.