Information Retrieval as Statistical Translation

Adam Berger and **John Lafferty**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
<aberger,lafferty>@cs.cmu.edu

Abstract

We propose a new probabilistic approach to information retrieval based upon the ideas and methods of statistical machine translation. The central ingredient in this approach is a statistical model of how a user might distill or "translate" a given document into a query. To assess the relevance of a document to a user's query, we estimate the probability that the query would have been generated as a translation of the document, and factor in the user's general preferences in the form of a prior distribution over documents. We propose a simple, well motivated model of the document-to-query translation process, and describe an algorithm for learning the parameters of this model in an unsupervised manner from a collection of documents. As we show, one can view this approach as a generalization and justification of the "language modeling" strategy recently proposed by Ponte and Croft. In a series of experiments on TREC data, a simple translation-based retrieval system performs well in comparison to conventional retrieval techniques. This prototype system only begins to tap the full potential of translation-based retrieval.

1 Introduction

When a person formulates a query to a retrieval system, what he is really doing is distilling an information need into a succinct query. In this work, we take the view that this distillation is a form of translation from one language to another: from documents, which contain the normal superfluence of textual fat and connective tissue such as prepositions, commas and so forth, to queries, comprised of just the skeletal index terms that characterize the document.

We take this view not because it is an accurate model of how a user decides what to ask of an information retrieval system, but because it turns out to be a useful expedient. By thinking about retrieval in this way, we can formulate tractable mathematical models of the query generation process, models that naturally account for many of the features that are critical to modern high performance retrieval systems, such as term weighting and query expansion. Moreover, statistical translation models for information retrieval can be implemented in a quite straightforward way, and our experiments with these models demonstrate very promising empirical behavior.

We begin by detailing a conceptual model of information retrieval. In formulating a query to a retrieval system, we imagine that a user begins with an information need. This information need is then represented as a fragment of an "ideal document"—a portion of the type of document that the user hopes to receive from the system. The user then translates or "distills" this ideal document fragment into a succinct query, selecting key terms and replacing some terms with related terms: replacing `pontiff` with `pope`, for instance.

Summarizing the model of query generation,

1. The user has an information need $\Im$.
2. From this need, he generates an ideal document fragment $\mathbf{d}_{\Im}$.
3. He selects a set of key terms from $\mathbf{d}_{\Im}$, and generates a query $\mathbf{q}$ from this set.

One can view this imaginary process of query formulation as a corruption of the ideal document. In this setting, the task of a retrieval system is to find those documents most similar to $\mathbf{d}_{\Im}$. In other words, retrieval is the task of finding, among the documents comprising the collection, likely preimages of the user's query. Figure 1 depicts this model of retrieval in a block diagram.

We have drawn Figure 1 in a way that suggests an information-theoretic perspective. One can view the information need $\Im$ as a signal that gets corrupted as the user $\mathcal{U}$ distills it into a query $\mathbf{q}$. That is, the query-formulation process represents a noisy channel, corrupting the information need just as a telephone cable corrupts the data transmitted by a modem. Given $\mathbf{q}$ and a model of the channel—how an information need gets corrupted into a query—the retrieval system's task is to identify those documents $\mathbf{d}$ that best satisfy the information need of the user.

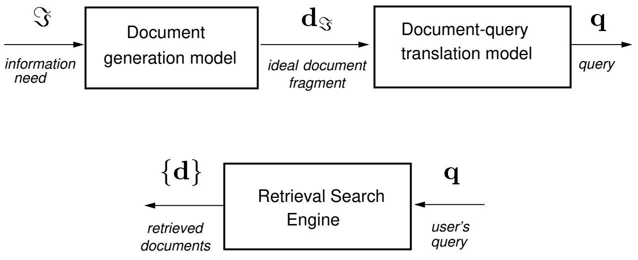

Figure 1. *Model of query generation and retrieval*

More precisely, the retrieval system's task is to find the *a posteriori* most likely documents given the query; that is, those $\mathbf{d}$ for which $p(\mathbf{d} \mid \mathbf{q}, \mathcal{U})$ is highest. By Bayes' law,

$$p(\mathbf{d} \mid \mathbf{q}, \mathcal{U}) = \frac{p(\mathbf{q} \mid \mathbf{d}, \mathcal{U})\, p(\mathbf{d} \mid \mathcal{U})}{p(\mathbf{q} \mid \mathcal{U})}. \tag{1}$$

Since the denominator $p(\mathbf{q} \mid \mathcal{U})$ is fixed for a given query and user, we can ignore it for the purpose of ranking documents, and define the relevance $\rho_{\mathbf{q}}(\mathbf{d})$ of a document to a query as

$$\rho_{\mathbf{q}}(\mathbf{d}) = \underbrace{p(\mathbf{q} \mid \mathbf{d}, \mathcal{U})}_{\text{query-dependent}} \quad \underbrace{p(\mathbf{d} \mid \mathcal{U})}_{\text{query-independent}}. \tag{2}$$

Equation (2) highlights the decomposition of relevance into two terms: first, a query-dependent term measuring the proximity of $\mathbf{d}$ to $\mathbf{q}$, and second, a query-independent or "prior" term, measuring the quality of the document according to the user's general preferences and information needs. Though in this work we take the prior term to be uniform over all documents, we imagine that in real-world retrieval systems the prior will be crucial for improved performance, and for adapting to the user's needs and interests. At the very least, the document prior can be used to discount short documents, or perhaps documents in a foreign language.

High-performance document retrieval systems must be sophisticated enough to handle polysemy and synonymy—to know, for instance, that `pontiff` and `pope` are related terms. The field of statistical translation concerns itself with how to mine large text databases to automatically discover such semantic relations. Brown *et al.* [3, 4] showed, for instance, how a system can "learn" to associate French terms with their English translations, given only a collection of bilingual French/English sentences. We shall demonstrate how, in a similar fashion, an IR system can, from a collection of documents, automatically learn which terms are related, and exploit these relations to better rank documents by relevance to a query.

The rest of the paper proceeds as follows. Section 2 outlines some antecedents to this work, and describes in greater detail the relationship between information retrieval and statistical translation. Section 3 introduces two specific models of translation from documents to queries. Section 4 explains how to estimate the parameters of such models automatically from a collection of documents. Section 5 presents results of several experiments on widely-used benchmarks in information retrieval. These experimental results demonstrate the competitiveness of translation-based retrieval, compared to standard *tfidf* vector space techniques. We conclude in Section 6 with some further discussion of this approach and directions for future research.

2 Background and Related Work

There is a large literature on probabilistic approaches to information retrieval, and we will not attempt to survey it here. Instead, we focus on the language modeling approach introduced recently by Ponte and Croft [10, 9], which is closest in spirit to the present work. To each document in the collection, this approach associates a probability distribution $p(\cdot \mid \mathbf{d})$ over terms; using the terminology developed in the field of speech recognition, Ponte and Croft call this distribution a "language model." The probability of a term t given a document is related to the frequency of t in the document. The probability of a query $\mathbf{q} = q_1, q_2, \ldots q_m$ is just the product of the individual term probabilities, $p(\mathbf{q} \mid \mathbf{d}) = \prod_i p(q_i \mid \mathbf{d})$. The relevance of a document $\mathbf{d}$ to a query $\mathbf{q}$ is presumed to be monotonically related to $p(\mathbf{q} \mid \mathbf{d})$.

The language modeling approach represents a novel and theoretically motivated approach to retrieval, which Ponte and Croft demonstrate to be effective. However, this framework does not allow the capability to model different forms or styles of queries, nor does it directly address the important issues of *synonymy* and *polysemy*: multiple terms sharing similar meanings and the same term having multiple meanings. In formulating a statistical translation-based approach to IR, we aim to develop a general statistical framework for handling these issues.

The inspiration and foundation for the present work comes from statistical machine translation—an area of research that until now has existed independently of information retrieval. It would take us too far afield to present an overview of the motivation and methods of statistical translation. Instead, we refer the reader to two articles [3, 4] outlining the IBM work that first developed the idea of statistical machine translation. We will, however, briefly describe the general form of the IBM statistical translation models, illustrated for the case of translating from French into English. These models are comprised of *translation probabilities* $t(f \mid e)$ for each English word e translating to each French word f, *fertility probabilities* to model the number of French

We use the convention that boldface roman letters refer to collections of words such as documents or queries, while italic roman letters refer to individual terms. Thus $p(q \mid \mathbf{d})$ refers to the probability of generating a *single* query word from an entire document $\mathbf{d}$.

words that an English word can generate, and *distortion probabilities* for introducing a dependence on relative word order in the two languages. These parameters are used to form a model $p(\mathbf{f},\mathbf{a}\,|\,\mathbf{e})$ for generating a French sentence $\mathbf{f} = \{f_1, f_2 \ldots, f_m\}$ from an English sentence $\mathbf{e} = \{e_1, e_2, \ldots, e_n\}$ in terms of a hidden *alignment* $\mathbf{a}$ between the words in the two sentences.

Brown *et al.* propose a series of increasingly complex and powerful statistical models of translation, the parameters of which are estimated by a bootstrapping procedure. We refer to [4] for a detailed presentation of these models, including the mathematical details of training them using the EM algorithm. In the next section we discuss analogous models for query generation.

3 Two Models of Document-Query Translation

Suppose that an information analyst is given a news article and asked to quickly generate a list of a few words to serve as a rough summary of the article's topic. As the analyst rapidly skims the story, he encounters a collection of words and phrases. Many of these are rejected as irrelevant, but his eyes rest on certain key terms as he decides how to render them in the summary. For example, when presented with an article about Pope John Paul II's visit to Cuba in 1998, the analyst decides that the words `pontiff` and `vatican` can simply be represented by the word `pope`, and that `cuba`, `castro` and `island` can be collectively referred to as `cuba`.

In this section we present two statistical models of this query formation process, making specific independence assumptions to derive computationally and statistically efficient algorithms. While our simple query generation models are mathematically similar to those used for statistical translation of natural language, the duties of the models are qualitatively different in the two settings. Document-query translation requires a *distillation* of the document, while translation of natural language will tolerate little being thrown away.

3.1. Model 1: A mixture model

We first consider the simplest of the IBM translation models for the document-to-query mapping. This model produces a query according to the following generative procedure. First we choose a length m for the query, according to the distribution $\psi(m\,|\,\mathbf{d})$. Then, for each position $j \in [1 \ldots m]$ in the query, we choose a position i in the document from which to generate q_j, and generate the query word by "translating" d_i according to the translation model $t(\cdot\,|\,d_i)$. We include in position zero of the document an artificial "null word," written `<null>`. The purpose of the null word is to generate spurious or content-free terms in the query (consider, for example, a query $\mathbf{q} =$ `Find all of the documents`...).

Let's now denote the length of the document by $|\,\mathbf{d}\,| = n$. The probability $p(\mathbf{q}\,|\,\mathbf{d})$ is then the sum over all possible alignments, given by

$$p(\mathbf{q}\,|\,\mathbf{d}) = \frac{\psi(m\,|\,\mathbf{d})}{(n+1)^m} \sum_{a_1=0}^{n} \cdots \sum_{a_m=0}^{n} \prod_{j=1}^{m} t(q_j\,|\,d_{a_j}). \qquad (3)$$

Just as the most primitive version of IBM's translation model takes no account of the subtler aspects of language translation, including the way word order tends to differ across languages, so our basic IR translation approach is but an impressionistic model of the relation between queries and documents relevant to them. Since IBM called their most basic scheme *Model 1*, we shall do the same for this rudimentary retrieval model.

A little algebraic manipulation shows that the probability of generating query $\mathbf{q}$ according to Model 1 can be rewritten as

$$p(\mathbf{q}\,|\,\mathbf{d}) = \psi(m\,|\,\mathbf{d}) \prod_{j=1}^{m} \left(\frac{n}{n+1} p(q_j\,|\,\mathbf{d}) + \frac{1}{n+1} t(w\,|\,\texttt{<null>}) \right)$$

where

$$p(q_j\,|\,\mathbf{d}) = \sum_{w} t(q_j\,|\,w)\, l(w\,|\,\mathbf{d})\,,$$

with the *document language model* $l(w\,|\,\mathbf{d})$ given by relative counts. Thus, we see that the query terms are generated using a *mixture model*—the document language model provides the mixing weights for the *translation model*, which has parameters $t(q\,|\,w)$. An alternative view (and terminology) for this model is to describe it as a Hidden Markov Model, where the states correspond to the words in the vocabulary, and the transition probabilities between states are proportional to the word frequencies.

The simplest version of Model 1, which we will distinguish as *Model 0*, is the one for which each word w can be translated only as itself; that is, the translation probabilities are "diagonal":

$$t(q\,|\,w) = \begin{cases} 1 & \text{if } q = w \\ 0 & \text{otherwise}. \end{cases}$$

In this case, the query generation model is given by

$$p(q\,|\,\mathbf{d}) = \frac{n}{n+1} l(q\,|\,\mathbf{d}) + \frac{1}{n+1} t(w\,|\,\texttt{<null>})\,,$$

a linear interpolation of the document language model and the background model associated with the null word.

3.2. Model 1′: A binomial model

Our imaginary information analyst, when asked to generate a brief list of descriptive terms for a document, is unlikely to list multiple occurrences of the same word. To account for this assumption in terms of a statistical model, we assume that a list of words is generated by making several independent translations of the document $\mathbf{d}$ into a single query term q, in the following manner. First, the analyst chooses a word w at random from the document. He chooses this word according to the document language model $l(w \mid \mathbf{d})$. Next, he translates w into the word or phrase q according to the translation model $t(q \mid w)$. Thus, the probability of choosing q as a representative of the document $\mathbf{d}$ is

$$p(q \mid \mathbf{d}) = \sum_{w \,\in\, \mathbf{d}} l(w \mid \mathbf{d})\, t(q \mid w).$$

We assume that the analyst repeats this process n times, where n is chosen according to the *sample size model* $\phi(n \mid \mathbf{d})$, and that the resulting list of words is filtered to remove duplicates before it is presented as the summary, or query, $\mathbf{q} = q_1, q_2, \ldots q_m$.

In order to calculate the probability that a particular query $\mathbf{q}$ is generated in this way, we need to sum over all sample sizes n, and consider that each of the terms q_i may have been generated multiple times. Thus, the process described above assigns to $\mathbf{q}$ a total probability

$$p(\mathbf{q} \mid \mathbf{d}) = \sum_n \phi(n \mid \mathbf{d}) \sum_{n_1 > 0} \cdots \sum_{n_m > 0} \binom{n}{n_1 \cdots n_m} \prod_{i=1}^{m} p(q_i \mid \mathbf{d})^{n_i}$$

In spite of its intimidating appearance, this expression can be calculated efficiently using simple combinatorial identities and dynamic programming techniques. Instead of pursuing this path, we will assume that the number of samples n is chosen according to a Poisson distribution with mean $\lambda(\mathbf{d})$:

$$\phi(n \mid \mathbf{d}) = e^{-\lambda(\mathbf{d})} \frac{\lambda(\mathbf{d})^n}{n!}\,.$$

Under this assumption, the above sum takes on a much friendlier appearance:

$$p(\mathbf{q} \mid \mathbf{d}) = e^{-\lambda(\mathbf{d})} \prod_{i=1}^{m} \left(e^{\lambda(\mathbf{d})\, p(q_i \mid \mathbf{d})} - 1 \right). \qquad (4)$$

This formula shows that the probability of the query is given as a product of terms. Yet the query term translations are *not* independent, due to the process of filtering out the generated list to remove duplicates. We refer to the model expressed in equation (4) as *Model 1′*.

Model 1′ has an interpretation in terms of binomial random variables. Suppose that a word w does *not* belong to the query with probability $\beta_w = e^{-\lambda(\mathbf{d}) p(w \mid \mathbf{d})}$. Then Model 1′ amounts to flipping independent β_w-biased coins to determine which set of words comprise the query. That is, we can reexpress the probability $p(\mathbf{q} \mid \mathbf{d})$ of equation (4) as

$$p(\mathbf{q} \mid \mathbf{d}) = \prod_{w \,\in\, \mathbf{q}} (1 - \beta_w) \prod_{w \,\notin\, \mathbf{q}} \beta_w\,.$$

Our use of this model was inspired by another IBM statistical translation model, one that was designed for modeling a bilingual dictionary [5].

Model 1′ also has an interpretation in the degenerate case of diagonal translation probabilities. To see this, let us make a further simplification by fixing the average number of samples to be a constant λ independent of the document $\mathbf{d}$, and suppose that the expected number of times a query word is drawn is less than one, so that $\max_i \lambda l(q_i \mid \mathbf{d}) < 1$. Then to first order, the probability assigned to the query according to Model 1′ is a constant times the product of the language model probabilities:

$$p(\mathbf{q} = q_1, \ldots, q_m \mid \mathbf{d}) \approx e^{-\lambda}\, \lambda^m \prod_{i=1}^{m} l(q_i \mid \mathbf{d})\,. \qquad (5)$$

Since the mean λ is fixed for all documents, the document that maximizes the righthand side of the above expression is that which maximizes $\prod_{i=1}^{m} l(q_i \mid \mathbf{d})$. This is precisely the value assigned to the query in what Ponte and Croft (1998) call the "language modeling approach."

4 Building a Translation-Based IR System

We now describe an implementation of the models described in the previous section, and their application to TREC data. The key ingredient in these models is the collection of translation probabilities $t(q \mid w)$. But how are we to obtain these probabilities? The statistical translation strategy is to learn these probabilities from an aligned bilingual corpus of translated sentences, using the likelihood criterion. Ideally, we should have a collection of query/document pairs to learn from, obtained by human relevance judgments. But we know of no publicly-available collection of data sufficiently large to estimate parameters for general queries.

4.1. Synthetic training data

Lacking a large corpus of queries and their associated relevant documents, we decided to tease out the semantic relationships among words by generating *synthetic queries* for a large collection of documents and estimating the translation probabilities from this synthetic data. To explain the rationale for this scheme, we return to our fictitious information analyst, and recall that when presented with a document $\mathbf{d}$, he will tend to select terms that are suggestive of the content of the document. Suppose now that he himself selects an arbitrary document $\mathbf{d}$ from a database $\mathcal{D}$, and asks us to guess, based only upon his summary $\mathbf{q}$, which document he chose. The amount by which we are able to do better, on average, than randomly guessing a document from $\mathcal{D}$ is the *mutual information* $I(D; Q) = H(D) - H(D \mid Q)$ between the random variables representing his choice of document D and query Q. Here $H(D)$ is the entropy in the analyst's choice of document, and $H(D \mid Q)$ is the conditional entropy of the document given the query. If he is playing this game cooperatively, he will generate queries for which this mutual information is large.

With this game in mind, we took a collection of TREC documents $\mathcal{D}$, and for each document $\mathbf{d}$ in the

q	$t(q\,|\,w)$
solzhenitsyn	0.319
citizenship	0.049
exile	0.044
archipelago	0.030
alexander	0.025
soviet	0.023
union	0.018
komsomolskaya	0.017
treason	0.015
vishnevskaya	0.015

$w =$ `solzhenitsyn`

q	$t(q\,|\,w)$
carcinogen	0.667
cancer	0.032
scientific	0.024
science	0.014
environment	0.013
chemical	0.012
exposure	0.012
pesticide	0.010
agent	0.009
protect	0.008

$w =$ `carcinogen`

q	$t(q\,|\,w)$
zubin_mehta	0.248
zubin	0.139
mehta	0.134
philharmonic	0.103
orchestra	0.046
music	0.036
bernstein	0.029
york	0.026
end	0.018
sir	0.016

$w =$ `zubin`

q	$t(q\,|\,w)$
pontiff	0.502
pope	0.169
paul	0.065
john	0.035
vatican	0.033
ii	0.028
visit	0.017
papal	0.010
church	0.005
flight	0.004

$w =$ `pontiff`

q	$t(q\,|\,w)$
everest	0.439
climb	0.057
climber	0.045
whittaker	0.039
expedition	0.036
float	0.024
mountain	0.024
summit	0.021
highest	0.018
reach	0.015

$w =$ `everest`

q	$t(q\,|\,w)$
wildlife	0.705
fish	0.038
acre	0.012
species	0.010
forest	0.010
environment	0.009
habitat	0.008
endangered	0.007
protected	0.007
bird	0.007

$w =$ `wildlife`

Figure 2. *Sample translation probabilities after EM training on synthetic data.*

collection, computed the mutual information statistic $I(w, \mathbf{d})$ for each of its words according to

$$I(w, \mathbf{d}) = p(w, \mathbf{d}) \log \frac{p(w \,|\, \mathbf{d})}{p(w \,|\, \mathcal{D})}.$$

Here $p(w \,|\, \mathbf{d})$ is the probability of the word in the document, and $p(w \,|\, \mathcal{D})$ is the probability of the word in the collection at large. By scaling these $I(w, \mathbf{d})$ values appropriately, we constructed an artificial cumulative distribution function over words in each document. We then drew $m \sim \phi(\cdot \,|\, \mathbf{d})$ random samples from the document according to this distribution, forming a query $\mathbf{q} = q_1, \ldots, q_m$, and several such queries were generated for each document.

4.2. EM training

The resulting corpus $\{(\mathbf{d}, \mathbf{q})\}$ of documents and synthetic queries was used to fit the translation probabilities of Models 1 and 1′ with the EM algorithm [7], run for only three iterations as a means to avoid overfitting. Space limitations prevent us from explaining the details of the training process, but enough detail to enable implementation can be found in the papers [4, 5], which describe very similar models. A sample of the resulting translation probabilities, when trained on the *Associated Press* (AP) portion of the TREC volume 3 corpus, is shown in Figure 2. In this figure, a document word is shown together with the ten most probable query words that it will translate to according to the model. The probabilities in these tables are among the 47,065,200 translation probabilities that were trained for our 132,625 word vocabulary. They were estimated from a corpus obtained by generating five synthetic mutual information queries for each of the 78,325 documents in the collection.

For statistical models of this form, *smoothing* or interpolating the parameters away from their maximum likelihood estimates is important. We used a linear interpolation of the background unigram model and the EM-trained translation model:

$$\begin{aligned} p_\alpha(q \,|\, \mathbf{d}) &= \alpha\, p(q \,|\, \mathcal{D}) + (1-\alpha)\, p(q \,|\, \mathbf{d}) \\ &= \alpha\, p(q \,|\, \mathcal{D}) + (1-\alpha) \sum_{w \in \mathbf{d}} l(w \,|\, \mathbf{d})\, t(q \,|\, w). \end{aligned}$$

The weight was empirically set to $\alpha = 0.05$ on heldout data. The models for the baseline language modeling approach, or Model 0, were also smoothed using linear interpolation:

$$l_\gamma(w \,|\, \mathbf{d}) = \gamma\, p(w \,|\, \mathcal{D}) + (1-\gamma)\, l(w \,|\, \mathbf{d}).$$

This interpolation weight was simply fixed at $\gamma = 0.1$. The Poisson parameter for the sample size distribution was fixed at $\lambda = 15$, independent of the document. No adjustment of any parameters, other than those determined by unsupervised EM training of the translation probabilities, was carried out on the TREC volume 3 data that we ran our evaluation on.

5 Experimental Results on TREC Data

In this section we summarize the quantitative results of using the models described in the previous two sections to rank documents for a set of queries. Though

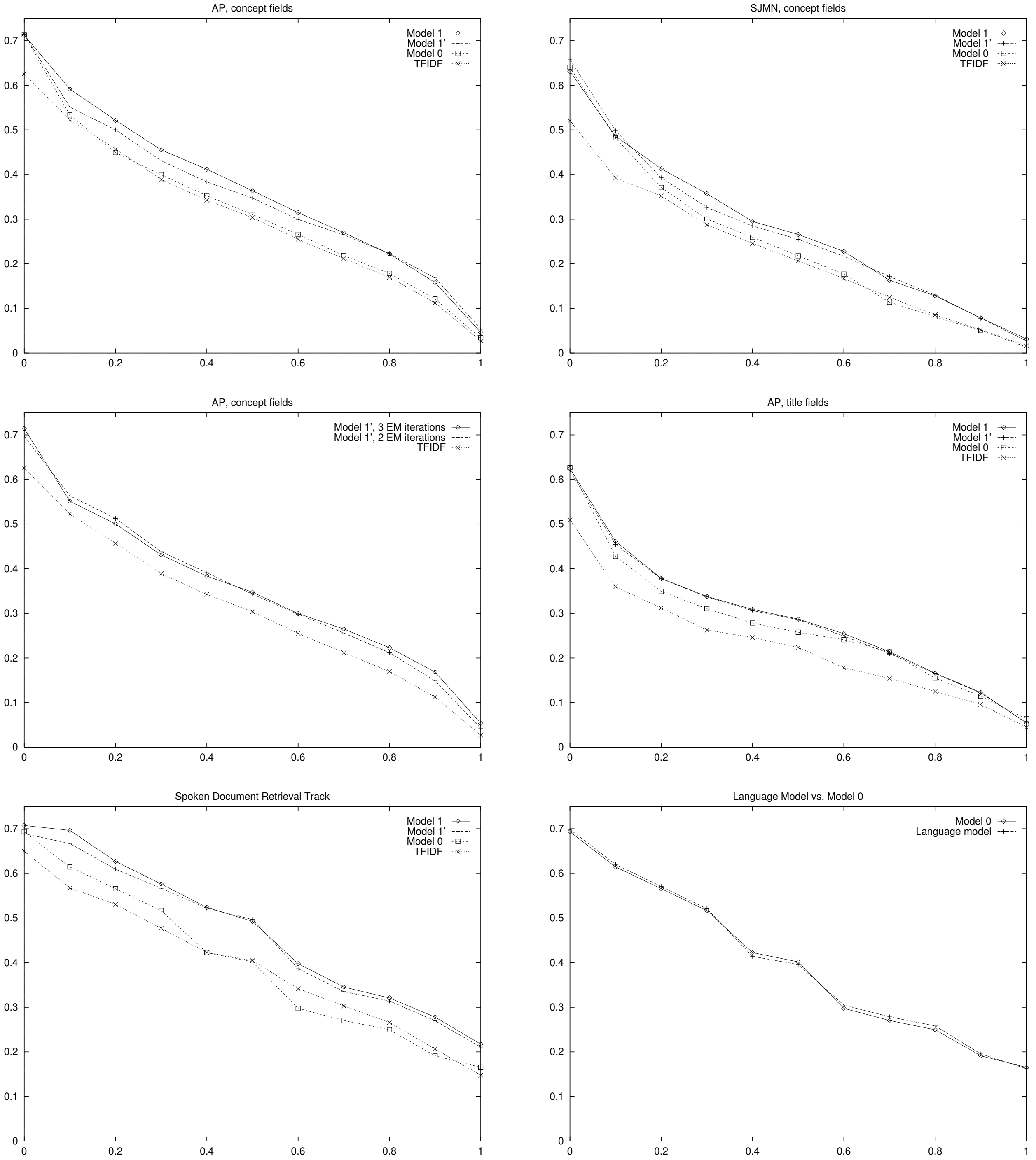

Figure 3. *Precision-recall curves on TREC data. The two plots in the top row compare the performance of Models 1 and 1′ to the baseline tfidf and Model 0 performance on AP data (left) and SJMN data (right) when ranking documents for queries formulated from the concept fields for topics 51–100. The middle row shows the discrepancy between two and three EM iterations of training for Model 1′ (left) and the performance of models on the short (average 2.8 words/query) queries obtained from the title field of topics 51–100 (right). The bottom row compares Models 1 and 1′ to Model 0 and tfidf on the SDR data (left) and the same language model scored according to Model 0 and using the product* $\prod_{i=1}^{m} l(q_i \mid \mathbf{d})$*, demonstrating that the approximation in equation (5) is very good (right).*

not a real-time computation, calculating the relevance score $\rho_{\mathbf{q}}(\mathbf{d})$ for each query can be done quickly enough to obviate the need for a "fast match" to throw out documents that are clearly irrelevant.

Our experiments fall into three categories. First, we report on a series of experiments carried out on the AP portion of the TREC data from volume 3 for both the concept and title fields in topics 51–100. The concept field queries comprise a set of roughly 20 keywords, while the title fields are much more succinct—typically not more than four words. Next, we tabulate a corresponding series of experiments carried out on the 90,250 document *San Jose Mercury News* (SJMN) portion of the data, also evaluated on topics 51–100. Finally, we present results on the *Spoken Document Retrieval* (SDR) track of the 1998 TREC evaluation, a small collection of 2,866 broadcast news transcripts. All of the data is preprocessed by converting to upper case, stemming, and filtering with a list of 571 stopwords from the SMART system.

Precision-recall curves for the AP and SJMN data, generated from the output of the TREC evaluation software, appear in Figure 3. The baseline curves in these plots show the performance of the *tfidf* measure using Robertson's *tf* score, as described by Ponte in [9]. They also show the result of using Model 0 to score the documents, using only word-for-word translations. In this approach, documents receive high relevance scores if they contain terms appearing in the query. Model 1 improves the average precision over the *tfidf* baseline by 19.4% on the AP data, and by 27.3% on the SJMN data. The R-precision improves by 10.0% on the AP data and by 22.8% on the SJMN data. The actual numbers for AP are collected in Table 1.

As discussed earlier, overfitting during EM training is a concern. Figure 3 shows the performance of Model 1′ on the AP data when the probabilities are trained for two and three EM iterations. To study the effects of query length on the models, we also scored the documents for the title fields of topics 51–100, where the average query length is only 2.8 words. As the middle right plot of Figure 3 reveals, the precision-recall curves are qualitatively different, showing a degradation in performance in the high-precision range. Overall, Model 1 achieves an improvement over the *tfidf* baseline of 30.2% in average precision and 17.8% in R-precision on these short queries. The marginal improvement of Model 1 over Model 0 is smaller here—6.3% in average precision and 4.9% in R-precision.

Two precision-recall plots for the SDR task are given. The bottom left plot of Figure 3 shows the improvement of Model 1 over the baseline. There is an improvement of 22.2% in average precision and 18.4% in R-precision. The bottom right plot compares Model 0′ to the result of ranking documents according to the probability $\prod_{i=1}^{m} l(q_i \mid \mathbf{d})$ for the same language model, as in Ponte and Croft's method. There is very little difference in the results, showing that equation (5) is indeed a good approximation.

	tfidf	Model 1	%Δ
Relevant:	5845	5845	—
Rel.ret.:	5845	5845	—
Precision:			
at 0.00	0.6257	0.7125	+13.9
at 0.10	0.5231	0.5916	+13.1
at 0.20	0.4569	0.5217	+14.2
at 0.30	0.3890	0.4554	+17.1
at 0.40	0.3425	0.4119	+20.3
at 0.50	0.3035	0.3636	+19.8
at 0.60	0.2549	0.3148	+23.5
at 0.70	0.2117	0.2698	+27.4
at 0.80	0.1698	0.2221	+30.8
at 0.90	0.1123	0.1580	+40.7
at 1.00	0.0271	0.0462	+70.5
Avg.:	0.2993	0.3575	+19.4
Precision at:			
5 docs:	0.4809	0.5574	+15.9
10 docs:	0.4702	0.5170	+10.0
15 docs:	0.4326	0.5135	+18.7
20 docs:	0.4213	0.4851	+15.1
30 docs:	0.3894	0.4539	+16.6
100 docs:	0.2960	0.3419	+15.5
200 docs:	0.2350	0.2653	+12.9
500 docs:	0.1466	0.1610	+9.8
1000 docs:	0.0899	0.0980	+9.0
R-Precision:	0.3254	0.3578	+10.0

Table 1. *Model 1 compared to the baseline system for queries constructed from the concept fields. These numbers correspond to the upper left plot in Figure 3.*

6 Discussion

Viewing a user's interaction with an information retrieval system as translation of an information need into a query is natural. Exactly this formulation is made in a recent overview of issues in information science presented to the theoretical computer science community [2]. In this paper we have attempted to lay the groundwork for building practical IR systems that exploit this view, by demonstrating how statistical translation models can be built and used to advantage.

When designing a statistical model for language processing tasks, often the most natural route is to apply a *generative* model which builds up the output step-by-step. Yet to be effective, such models need to liberally distribute probability mass over a huge space of possible outcomes. This probability can be difficult to control, making an accurate *direct* model of the distribution of interest difficult to construct. The source-channel perspective suggests a different approach: turn the search problem around to predict the input. Far more than a simple application of Bayes' law, there are compelling reasons why reformulating the problem in this way should be rewarding. In speech recognition, natural language processing, and machine translation, researchers have time and again found that predicting what is already known (*i.e.*, the query) from competing hypotheses can be more effective than directly predicting all of the hypotheses.

Our simple models only begin to tap the potential of the translation-based approach. More powerful models of the query generation process, even along the lines of IBM's more complex models of translation, should offer performance gains. For example, one of the fundamental notions of statistical translation is the idea of fertility, where a source word can generate zero or more words in the target sentence. While there appears to be no good reason why a word selected from the document should generate more than a single query term per trial, we should allow for *infertility* probabilities, where a word generates no terms at all. The use of stop word lists mitigates but does not eliminate the need for this improvement. The use of distortion probabilities could be important for discounting relevant words that appear toward the end of a document, and rewarding those that appear at the beginning. Many other natural extensions to the model are possible.

Our discussion has made the usual "bag of words" assumption about documents, ignoring word order for the sake of simplicity and computational ease. But the relative ordering of words is informative in almost all applications, and crucial in some. The sense of a word is often revealed by nearby words, and so by heeding contextual clues, one might hope to obtain a more accurate translation from document words to query words.

The retrieval-as-translation approach applies quite naturally to the problem of multilingual retrieval, where a user issues queries in a language T to retrieve documents in a source language S. The most direct approach to multilingual IR expands the query $\mathbf{q}_T$ to $\mathbf{q}'_T$ and then translates directly into a source language query $\mathbf{q}'_S$, which is then issued to the database. The retrieval-as-translation approach would learn translation models which capture how words in S translate to words in T. The source-channel framework should also be well-suited to "higher-order" IR tasks, such as fact extraction and question answering from a database.

7 Conclusions

We have presented an approach to information retrieval that exploits ideas and methods of statistical machine translation. After outlining the approach, we presented two simple, motivated models of the document-query translation process. With the EM algorithm, the parameters of these models can be trained in an unsupervised fashion from a collection of documents. Experiments on TREC datasets demonstrate that even these simple methods can yield substantial improvements over standard baseline vector space methods, but without explicit query expansion and term weighting schemes, which are inherent in the translation approach itself.

Acknowledgements

We would like to thank Jamie Callan, Jaime Carbonell, Ramesh Gopinath, Yiming Yang, and the anonymous reviewers for helpful comments on this work. This research was supported in part by NSF KDI grant IIS-9873009, an IBM University Partnership Award, an IBM Cooperative Fellowship, and a grant from Justsystem Corporation.

References

[1] A. Bookstein and D. Swanson (1974). "Probabilistic models for automatic indexing," *Journal of the American Society for Information Science*, **25**, pp. 312–318.

[2] A. Broder and M. Henzinger (1998). "Information retrieval on the web: Tools and algorithmic issues," Invited tutorial at Foundations of Computer Science (FOCS).

[3] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer, and P. Roossin (1990). "A statistical approach to machine translation," *Computational Linguistics*, **16**(2), pp. 79–85.

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

[5] P. Brown, S. Della Pietra, V. Della Pietra, M. Goldsmith, J. Hajic, R. Mercer, and S. Mohanty (1993). "But dictionaries are data too," In *Proceedings of the ARPA Human Language Technology Workshop*, Plainsborough, New Jersey.

[6] W. B. Croft and D.J. Harper (1979). "Using probabilistic models of document retrieval without relevance information," *Journal of Documentation*, **35**, pp. 285–295.

[7] A. Dempster, N. Laird, and D. Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society,* **39**(B), pp. 1–38.

[8] W. Gale and K. Church (1991). "Identifying word correspondences in parallel texts," in *Fourth DARPA Workshop on Speech and Natural Language*, Morgan Kaufmann Publishers, pp. 152–157.

[9] J. Ponte (1998). *A language modeling approach to information retrieval.* Ph.D. thesis, University of Massachusetts at Amherst.

[10] J. Ponte and W. B. Croft (1998). "A language modeling approach to information retrieval," *Proceedings of the ACM SIGIR*, pp. 275–281.

[11] S.E. Robertson and K. Sparck Jones (1976). "Relevance weighting of search terms," *Journal of the American Society for Information Science*, **27**, pp. 129–146.

[12] S. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau (1992). "Okapi at TREC," In *Proceedings of the first Text REtrieval Conference (TREC-1)*, Gaithersburg, Maryland.

[13] G. Salton and C. Buckley (1988). "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, **24**, pp. 513–523.

[14] H. Turtle and W. B. Croft (1991). "Efficient probabilistic inference for text retrieval," *Proceedings of RIAO 3.*

[15] W. Weaver (1955). "Translation (1949)," In *Machine Translation of Languages*, MIT Press.