Information Retrieval as Statistical Translation

Adam Berger
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
aberger@cs.cmu.edu

John Lafferty
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
lafferty@cs.cmu.edu

January 12, 1999

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. On a series of experiments, a prototype translation-based retrieval system significantly outperforms conventional retrieval techniques. This prototype system is but a skin-deep implementation of the full translation-based approach, and as such only begins to tap the full potential of translation-based retrieval.

Keywords: information retrieval, statistical machine translation, language models, source-channel model, hidden alignments, EM algorithm, document-query models.

Word count: 4,903 words, excluding figures, tables, and formulas.

1. Introduction

When a user 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 and retrieval process, models that can be implemented in a quite straightforward way and that exhibit promising empirical behavior.

Viewing document retrieval as a problem in translation may seem rather fanciful. But in fact, from this perspective, the task of retrieval is just a matter of inverting the translation: given a query, find the document(s) in the collection most likely to translate to the query. Moreover, the translation perspective offers another compelling argument in its favor: one can dispense with many of the *ad hoc* garlands common and critical to modern high-performance retrieval systems, such as query expansion and term weighting, since they are built in to the translation framework.

We begin by detailing the conceptual model of information retrieval. In formulating a query to a retrieval system, a user begins with an information need. We view this need as an "ideal document"—a perfect fit for the user, but almost certainly not present in the retrieval system's collection of documents. From this ideal document, the user selects a group of identifying terms. In the context of traditional IR, one could view this group of terms as akin to an expanded query. The user then formulates a query from this group of terms by removing duplicates 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 $\mathbf{d}_\Im$.
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—from an ideal document $\mathbf{d}_\Im$ to a query $\mathbf{q}$—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.

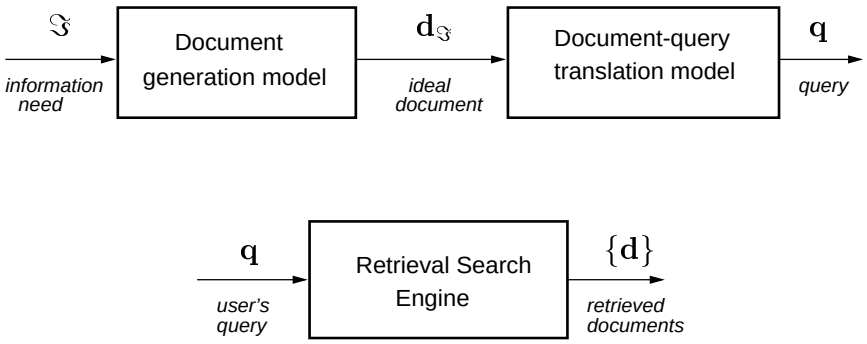

Figure 1. *Model of query generation (above) and retrieval (below)*

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.*

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 $\rho(\mathbf{d}; \mathbf{q})$ of a document to a query as

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

Equation (2) highlights the decomposition of relevance into two terms: first, a query-dependent term measuring the relevance 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 preferences. Though in this work we take the prior term to be uniform over all documents, ultimately the prior will be very important for improved performance, and for allowing the system to adapt 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.

* 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 $\mathbf{d}$.

Section 4 contains a detailed formulation of a simple $p(\mathbf{q} \mid \mathbf{d})$ model, but here we will briefly outline the strategy for constructing the model. The strategy requires a corpus of $(\mathbf{d}, \mathbf{q})$ pairs, with each pair consisting of a query and a document relevant to the query. Section 5 describes how, given this data, one can construct a translation model $p(\mathbf{q} \mid \mathbf{d})$, which assigns a probability to the event that $\mathbf{q}$ is a distillation of (a translation of) $\mathbf{d}$. Given such a model and a new query $\mathbf{q}'$, assigning relevance judgments is a matter of computing $p(\mathbf{q}' \mid \mathbf{d})$ for each $\mathbf{d} \in \mathcal{C}$.

Ultimately, 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 find and rank the documents it returns to the user.

The rest of the paper will proceed as follows. Section 2 outlines some antecedents to this work. Section 3 describes in greater detail the relationship between information retrieval and statistical translation. Section 4 introduces a specific model of translation from documents to queries. Section 5 explains how to construct such a model automatically from a collection of documents. Section 6 presents results of several different experiments on a widely-used benchmark in information retrieval. These results demonstrate the competitiveness of translation-based retrieval, compared to some popular retrieval techniques such as *tfidf*. We conclude in Section 7 with some shortcomings of this approach and directions for future research.

2. Previous Work

In modeling how a document is distilled into a query, our approach bears some similarity to a longstanding concern in the field of library science: finding a concise group of index terms that characterize a document. Towards this end, Bookstein and Swanson [1] observed, in proposing their 2-Poisson model, that the appearance of content terms in a document follows one of two distributions: `pope`, for instance, follows a Poisson distribution with a higher mean in documents about the Pope, and a distribution with a lower mean in other documents. So term frequency information could be useful for an algorithm whose responsibility is to automatically compile a list of index terms for documents. Though an important step in both indexing and retrieval, this model is imperfect (as the authors concede) in that there exist shades of "aboutness"—not all documents are either about the Pope or only peripherally about the Pope. Moreover, the family of n-Poisson models have to date achieved only limited empirical success [10].

In the same paper, Bookstein and Swanson also propose a mathematical model to ac-

count for the fact that over a heterogeneous collection of documents, content terms (like `laser`) tend to occur in clusters, while non-content terms (like `obtain`) typically occur more uniformly. In a more detailed study of this phenomenon, Robertson and Sparck Jones [11] used statistical techniques to analyze the distribution of search terms, with an eye towards increasing the performance of document retrieval systems by automatically assigning different "relevancy" weights to the terms in a query. Refinements of this work by many parties over the past few decades have led to the popular *tfidf* weighting scheme [13], a central ingredient in the search engines of several multi-billion dollar companies.

Robertson and Sparck Jones used these relevance weights as part of a probabilistic approach to retrieval: given a query and a set of documents, estimate for each document the probability that the document is relevant to the query. This retrieval framework—predicting whether the document is relevant to the query—is a popular one, appearing in other probabilistic approaches to IR as well [6]. In contrast, the approach we take is to evade the notion of "relevance" altogether, replacing it with an information-theoretic perspective that asks whether the document could have given rise to the query through a process of distillation. While the parameters of *tfidf* and Okapi systems [12] can be manually adjusted to improve performace, our approach is to construct automatically trainable models that can be fit from data and adapted to a user's interaction with the system.

A quite different probabilistic approach to retrieval from the relevancy-prediction strategy appears in the `INQUERY` system, under development at the University of Massachusetts [14]. As with the approach presented herein, the `INQUERY` system distinguishes between a user's (hidden) information need and his query. `INQUERY` uses a Bayesian network to predict whether the retrieved documents satisfy the information need of the user, as expressed in the query.

Perhaps closest in spirit to the present work is the language modeling approach introduced recently by Ponte and Croft [10, 9]. To each document in the collection, they associate a probability distribution $p(\cdot \mid \mathbf{d})$ over terms; they call this distribution a "language model" to accord with terminology developed in the field of speech recognition. 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 demonstrated to be effective. However, their framework does not appear general enough to handle the important issues of *synonymy* and *polysemy*: multiple terms sharing similar meanings and the same term having multiple meanings. To indicate the difficulties that synonomy causes, imagine a document which contains none of the query terms, and yet is still relevant to the query. A document on the history of the Vatican, for instance, may not contain the term `pontiff`, but is nonetheless relevant to a query consisting of just that single term. A more general framework appears necessary to handle such phenomena gracefully.

3. Statistical Translation

Besides its intellectual pedigree in information retrieval, the roots of the present work lie in the field of statistical machine translation (MT). Automatic translation by computer was first contemplated by Warren Weaver when modern computers were in their infancy [15], but received serious attention only in the past decade. In particular, the form of the mathematical model that will appear in Section 3 bears a resemblance to one introduced in [5] in the context of machine translation.

The central problem of statistical MT is to build a system that automatically learns how to translate text, by inspecting a large corpus of sentences in one language along with their translations into another language. The *Candide* system, an experimental project at IBM Research in the early 1990s, employed for this purpose the recorded proceedings of the Canadian Parliament, which happen to be maintained in both English and French.

Language translation has a convenient restatement in information-theoretic terms. We illustrate with the example of a French-to-English translation system that has just been presented with a French sentence to translate. Imagine that the person who generated this French sentence was in fact thinking of an English sentence, which was somehow "corrupted" (between the user's conceiving of it, and presenting it to the translation system) into the given French sentence. The goal of the translation system is to recover the original English sentence. In performing this inversion, the translation system has at its disposal not just the input French sentence, but also a model of how English sentences are "corrupted" into French sentences and a model of well-formed English sentences, both learned automatically from the bilingual corpus.

The reader may recognize this framework as analogous to how we have earlier posed the document retrieval problem. The original English sentence is akin to the ideal document representing the user's information need; the French sentence is like the query. One important difference between the two settings is that the search for the optimal input to the channel is, in the case of retrieval, restricted to documents in the collection, whereas a translation system considers (at least in principle) all English sentences.

Of course, the source-channel framework is only a restatement of the problem, not a solution. Still missing is the parametric form of the channel model, a method for learning the parameters of this model from a corpus, and a way to apply this model to the task of translating an input French sentence. It would take us too far afield to offer any detail on these steps, and we refer the reader to Brown *et. al* [3, 4] for details on all three issues. Instead we briefly describe the most primitive of the IBM statistical translation models. This model contains a *translation probability* $t(f \mid e)$ for each English word e translating to each French word f. The probability that an English sentence $\mathbf{e} = \{e_1, e_2, \ldots\}$ translates to a French sentence $\mathbf{f} = \{f_1, f_2 \ldots\}$ is calculated as

$$p(\mathbf{f} \mid \mathbf{e}) = \gamma \sum_{\mathbf{a}} \prod_{j=1}^{m} t(f_j \mid e_{a_j})$$

where γ is a normalizing factor. The hidden variable in this model is the *alignment*

a between the French and English words: $a_j = k$ means that the kth English word translates to the jth French word.

Brown *et. al* [4] propose a series of increasingly complex and powerful statistical models of translation, of which this model is the first, and appropriately called *Model 1*.

4. A Simple Model of Document-Query Translation

In this section we introduce a simple model of how documents are distilled or "translated" into queries. 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`.

To represent this process in terms of a statistical model, we make the assumption, clearly invalid, that the analyst generates a list of words 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 a distribution $l(w \mid \mathbf{d})$ that we call the *document language model*. Next, he translates w into the word or phrase q according to a *translation model*, with parameters $t(q \mid w)$. 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

$$\begin{aligned} p(\mathbf{q} \mid \mathbf{d}) &= \sum_n \phi(n \mid \mathbf{d}) \sum_{n_1>0} \cdots \sum_{n_m>0} \binom{n}{n_1 \ldots n_m} \prod_{i=1}^{m} p(q_i \mid \mathbf{d})^{n_i} \\ &= \sum_n \phi(n \mid \mathbf{d}) \sum_{n_1>0} \cdots \sum_{n_m>0} \binom{n}{n_1 \ldots n_m} \prod_{i=1}^{m} \left(\sum_w l(w \mid \mathbf{d})\, t(q_i \mid w) \right)^{n_i}. \end{aligned}$$

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 more friendly appearance:

$$p(\mathbf{q} = q_1, \ldots, q_m \mid \mathbf{d}) = e^{-\lambda(\mathbf{d})} \prod_{i=1}^{m} \left(e^{\lambda(\mathbf{d})\, p(q_i \mid \mathbf{d})} - 1 \right) . \tag{3}$$

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.

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.

Both our Model 1 for document-to-query translation and IBM Model 1 for English-French natural language translation contain only two basic types of parameters: string length probabilities, and word translation probabilities. However, the form of the models is qualitatively and mathematically different. Document-query translation requires a *distillation* of the document, while natural language translation will tolerate little being thrown away.

In fact, our Model 1 was inspired by another IBM statistical translation model, but one that was designed for modeling a bilingual dictionary [5]. In this model, one imagines a dictionary entry being constructed by a lexicographer collecting translations of an English word or phrase $\mathbf{e}$. After gathering a number of independent translations, the lexicographer removes duplicates to form a dictionary entry for $\mathbf{e}$. Our basic model and training algorithm is an adaptation of this scheme.

4.1. The simplest case: word-for-word translation

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

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

Under this model, the query terms are chosen simply according to their frequency of occurrence in the document (or some smoothed version of that frequency). As a further simplification, let us fix 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 is a constant times the product of the language model probabilities:

$$p(\mathbf{q} = q_1, \ldots, q_m \mid \mathbf{d}) \approx e^{-\lambda} \lambda^m \prod_{i=1}^{m} l(q_i \mid \mathbf{d}) \tag{4}$$

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

5. Building the System

We now describe an implementation of Model 1 as described in the previous section, and its application to TREC data. The key ingredient in Model 1 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. Ideally, we should have a collection of query/document pairs to learn from, obtained by human relevance judgments. But a collection of such data of the size needed to estimate parameters for general queries is difficult to come by.

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 training the Model 1 translation probabilities on 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 had chosen. 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 collection, we weighted and ranked its words $w \in \mathbf{d}$ according to their

mutual information statistic

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

Here $p(w \mid \mathbf{d})$ is the probability of the word in the document, and $p(w \mid \mathcal{D})$ is the probability of the word in the collection at large. (While the quantity $I(w, \mathbf{d})$ is typically positive, we could encounter negative values and so need to scale these statistics appropriately.) We then drew $n \sim Poisson(\lambda)$ random samples from the document according to this distribution.

Given this synthetic corpus of documents and queries, we fit the translation probabilities of Model 1 to it using the EM algorithm [7], run only for three iterations as a simple means to avoid overfitting. Space limitations prevent us from explaining the details of the training process, but some of these can be found in the papers [4, 5] which describe similar models. A sample of the resulting translation probabilities, when trained on the *Associated Press* (AP) portion of the TREC volume 3 corpus, are 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 particular 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 crucial. We used a simple linear mixture of the background unigram model and the EM-trained translation model:

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

The weight was empirically set to $\alpha = 0.05$ by optimizing performance on a different dataset: a portion of the 1998 TREC *Spoken Document Retrieval* (SDR) data. The models for the baseline language modeling approach, or Model 0, were also smoothed using linear interpolation:

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

This mixture 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 those that were determined by unsupervised EM training of the translation probabilities, was carried out on the TREC volume 3 data that we ran our evaluation on.

q	$t(q \mid w)$
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	$t(q \mid w)$
defend	0.676
trial	0.015
case	0.012
court	0.011
charge	0.011
judge	0.010
attorney	0.009
convict	0.007
prosecutor	0.006
accuse	0.006

$w =$ defend

q	$t(q \mid w)$
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

q	$t(q \mid w)$
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	$t(q \mid w)$
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	$t(q \mid w)$
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	$t(q \mid w)$
ibm	0.674
computer	0.042
machine	0.022
analyst	0.012
software	0.011
workstation	0.007
stock	0.006
system	0.006
business	0.005
market	0.005

$w =$ ibm

q	$t(q \mid w)$
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	$t(q \mid w)$
whittaker	0.535
climber	0.048
everest	0.032
climb	0.023
expedition	0.018
garbage	0.015
chinese	0.015
peace	0.015
cooper	0.013
1963	0.012

$w =$ whittaker

Figure 2. *Sample translation probabilities*

6. Experimental Results on TREC Data

In this section we summarize the quantitative results of using Model 1, as described in the previous two sections, to rank documents for a set of queries. Though not a real-time computation, calculating the relevance score $\rho(\mathbf{d};\mathbf{q})$ for each query occurs quickly enough on modern-day equipment to obviate the need for a "fast match" to throw out obviously irrelevant documents.

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 *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 relatively small collection of about 2800 transcripts of news broadcasts.

Precision-recall curves for the AP and SJMN data, generated from the output of the TREC evaluation software, appear in Figure 3. The baseline curve in this plot is the result of using Model 0 to score the documents, using only word-for-word translations. In this approach, documents receive high relevance scores by containing terms appearing in the query. Model 1 improves the average precision over this baseline by 15.9% on the AP data, and by 10.5% on the SJMN data. The R-precision improves by 11.0% on the AP data and by 6.5% on the SJMN data. The actual numbers are tabulated in Table 1.

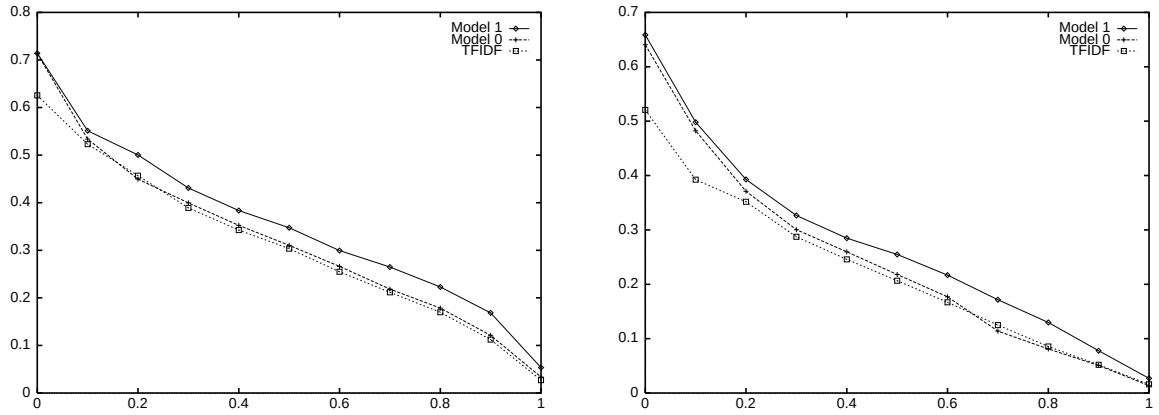

Figure 3. *Precision-recall curves on TREC data. The plots compares the performance of Model 1 to the baseline Model 0 on AP data (left) and SJMN data (right) when ranking documents for queries formulated from the concept fields for topics 51–100.*

These plots also show the performance of the *tfidf* measure using Robertson's *tf* score, as described by Ponte in [9]. Although Ponte and Croft report an appreciable improvement of 8.7% in average precision and 6.2% in R-precision on these same queries [10, 9], evaluated however on the *entire* volume 3 collection, we see only a small difference of

3.5% and 0.3% on the AP portion of the data. We expect that this can be attributed to our simple method of linear interpolation with a fixed weight, and underscores the need for smoothing when working with these language models.

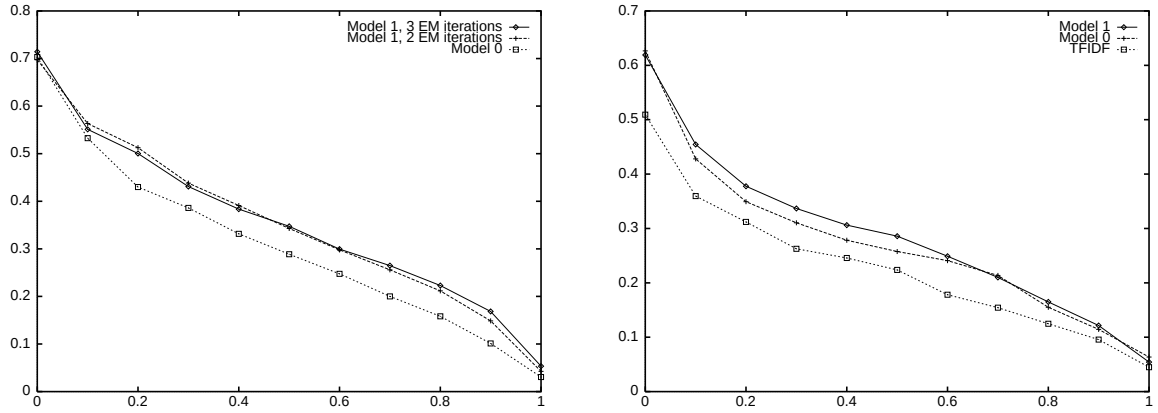

Figure 4. *Precision-recall curves on the AP portion of the TREC data. The left plot shows the discrepancy between two and three EM iterations of training Model 1. The right plot shows the performance of Model 0 and Model 1 on the short (average 2.8 words/query) queries obtained from the title field of topics 51–100.*

As discussed earlier, overfitting during EM training is a concern because of the manner in which the synthetic queries are generated. In Figure 4 we show the performance of Model 1 on the AP data when the probabilities are trained for two and three 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 Figure 4 reveals, the precision-recall curves are qualitatively different, showing a degradation in performance in the high-precision range. Overall, Model 1 achieves a smaller improvement of 5.2% in average precision and 2.7% in R-precision on these short queries; the numbers are tabulated in Table 2.

In Figure 5 two precision-recall plots for the SDR task are given. The left plot shows the improvement of Model 1 over the model having only the trivial word-for-word translation probabilities. There is an improvement of 17.2% in average precision and 12.9% in R-precision. The 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 (4) is indeed a good approximation. These same data are tabulated in Table 2.

AP	Model 0	Model 1	%Δ
Relevant:	5845	5845	—
Rel.ret.:	5845	5845	0.0
Precision:			
at 0.00	0.7031	0.6974	-0.8
at 0.10	0.5327	0.5634	+5.8
at 0.20	0.4299	0.5127	+19.3
at 0.30	0.3861	0.4381	+13.5
at 0.40	0.3316	0.3911	+17.9
at 0.50	0.2887	0.3429	+18.8
at 0.60	0.2475	0.2980	+20.4
at 0.70	0.2001	0.2559	+27.9
at 0.80	0.1582	0.2117	+33.8
at 0.90	0.1012	0.1486	+46.8
at 1.00	0.0306	0.0426	+39.2
Avg.:	0.2937	0.3405	+15.9
Precision at:			
5 docs:	0.5404	0.5404	-0.0
10 docs:	0.5000	0.5170	+3.4
15 docs:	0.4596	0.4922	+7.1
20 docs:	0.4447	0.4723	+6.2
30 docs:	0.4121	0.4362	+5.8
100 docs:	0.2960	0.3330	+12.5
200 docs:	0.2283	0.2600	+13.9
500 docs:	0.1398	0.1569	+12.2
1000 docs:	0.0866	0.0959	+10.7
R-Precision:	0.3153	0.3499	+11.0

SJMN	Model 0	Model 1	%Δ
Relevant:	2322	2322	—
Rel.ret.:	2322	2322	0.0
Precision:			
at 0.00	0.6319	0.6586	+4.2
at 0.10	0.4891	0.4980	+1.8
at 0.20	0.3730	0.3928	+5.3
at 0.30	0.3023	0.3264	+8.0
at 0.40	0.2653	0.2848	+7.4
at 0.50	0.2271	0.2548	+12.2
at 0.60	0.1915	0.2169	+13.3
at 0.70	0.1244	0.1715	+37.9
at 0.80	0.0894	0.1297	+45.1
at 0.90	0.0569	0.0779	+36.9
at 1.00	0.0152	0.0270	+77.6
Avg.:	0.2349	0.2595	+10.5
Precision at:			
5 docs:	0.4750	0.4583	-3.6
10 docs:	0.3979	0.4021	+1.1
15 docs:	0.3528	0.3708	+5.1
20 docs:	0.3250	0.3385	+4.2
30 docs:	0.2903	0.3042	+4.8
100 docs:	0.1767	0.1948	+10.2
200 docs:	0.1203	0.1327	+10.3
500 docs:	0.0622	0.0687	+10.5
1000 docs:	0.0354	0.0386	+9.0
R-Precision:	0.2577	0.2745	+6.5

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

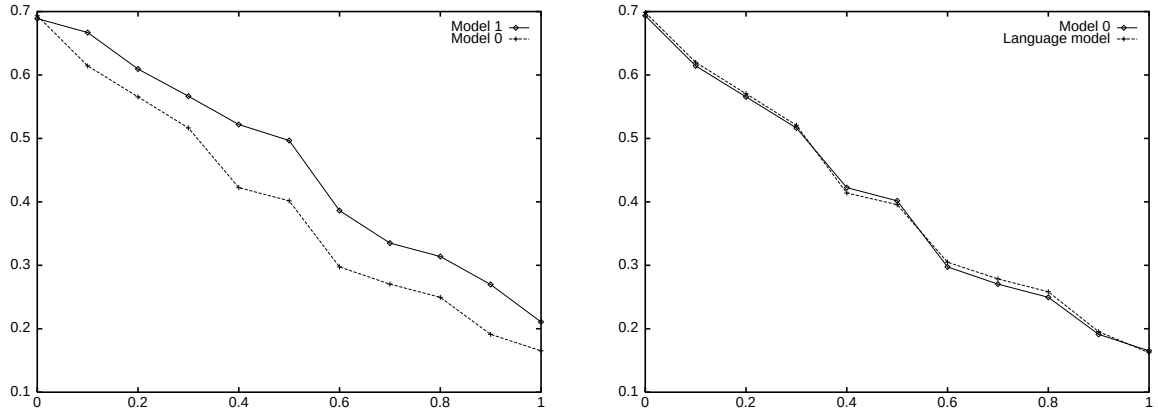

Figure 5. *The left plot compares Model 1 to Model 0 on the SDR data. The right plot compares the same language model scored according to Model 0 and using the product* $\prod_{i=1}^{m} l(q_i \mid \mathbf{d})$ *demonstrating that the approximation in equation (4) is very good.*

SDR	Model 0	Model 1	%Δ
Relevant:	390	390	—
Rel.ret.:	390	390	0.0
Precision:			
at 0.00	0.6934	0.6889	-0.7
at 0.10	0.6144	0.6670	+8.6
at 0.20	0.5656	0.6094	+7.7
at 0.30	0.5165	0.5666	+9.7
at 0.40	0.4225	0.5220	+23.6
at 0.50	0.4016	0.4967	+23.7
at 0.60	0.2975	0.3863	+29.8
at 0.70	0.2704	0.3351	+23.9
at 0.80	0.2495	0.3139	+25.8
at 0.90	0.1912	0.2697	+41.1
at 1.00	0.1652	0.2107	+27.5
Avg.:	0.3805	0.4458	+17.2
Precision at:			
5 docs:	0.4348	0.5217	+20.0
10 docs:	0.3783	0.4130	+9.2
15 docs:	0.3188	0.3565	+11.8
20 docs:	0.2761	0.3152	+14.2
30 docs:	0.2406	0.2638	+9.6
100 docs:	0.1030	0.1217	+18.2
200 docs:	0.0600	0.0700	+16.7
500 docs:	0.0303	0.0315	+4.0
1000 docs:	0.0159	0.0167	+5.0
R-Precision:	0.3753	0.4237	+12.9

SDR	Model 0	LM	%Δ
Relevant:	390	390	—
Rel.ret.:	390	390	0.0
Precision:			
at 0.00	0.6934	0.6985	+0.7
at 0.10	0.6144	0.6201	+0.9
at 0.20	0.5656	0.5705	+0.9
at 0.30	0.5165	0.5208	+0.8
at 0.40	0.4225	0.4140	-2.1
at 0.50	0.4016	0.3956	-1.5
at 0.60	0.2975	0.3049	+2.5
at 0.70	0.2704	0.2785	+3.0
at 0.80	0.2495	0.2582	+3.5
at 0.90	0.1912	0.1954	+2.2
at 1.00	0.1652	0.1622	-1.8
Avg.:	0.3805	0.3838	+0.9
Precision at:			
5 docs:	0.4348	0.4435	+2.0
10 docs:	0.3783	0.3870	+2.3
15 docs:	0.3188	0.3188	-0.0
20 docs:	0.2761	0.2848	+3.2
30 docs:	0.2406	0.2391	-0.6
100 docs:	0.1030	0.1035	+0.5
200 docs:	0.0600	0.0598	-0.3
500 docs:	0.0303	0.0303	-0.0
1000 docs:	0.0159	0.0159	-0.0
R-Precision:	0.3753	0.3880	+3.4

Table 2. *Precision-recall data for SDR portion of TREC data corresponding to Figure 5.*

7. Discussion

The view of a user's interaction with an information retrieval system as translation of an information need into a query is a natural one. 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 simple statistical translation models can be built and used to advantage.

The lessons of statistical methods in speech recognition, natural language processing, and machine translation suggest the potential benefit of applying a source-channel paradigm to retrieval. Far more than a simple application of Bayes' law, there are compelling reasons why the ritual of turning a search problem around to predict the input should be rewarding. When designing a statistical model for language processing tasks, often

the most natural route is to build a *generative* model which builds 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 fashion. Time and again, researchers have found that predicting what is already known (*i.e.*, the query) from competing hypotheses is easier than directly predicting all of the hypotheses (*i.e.*, the documents).

Our simple Model 1 only begins to tap the potential of the approach. More complex models of the query generation process, 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 a word *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 to the model. The use of a *null word* in the document for generating spurious or content-free terms in the query could be useful (consider, for example, a query $\mathbf{q}$ = `Find all of the documents...`). The use of *distortion probabilities* could be important for discounting relevant words that appear towards the end of a document, and rewarding those that appear at the beginning. Many other natural extensions to the model are possible.

The source-channel approach is also compelling for various extensions to a basic system. Consider, for example, a multilingual IR system where a user issues queries in a language T to retrieve documents in a source language S. In the document-query translation model framework that we propose, the full context of the document can potentially be used for disambiguating, resolving word senses, and obtaining a better mapping from the document to the query. This is quite different from the most direct approach that 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. This simply underscores how the notions of term weighting and query expansion are built in to the statistical translation approach.

Finally, while our methods have been presented in the spirit of "bag of words" techniques that can scale to handle massive data sets by making strong independence assumptions, the basic approach may be just as important for "higher-order" IR tasks, such as fact extraction and question answering from a database ($\mathbf{q}$ = `How many stomachs does a cow have?`). For such systems we should be working with stochastic grammars and logical forms, and using them to translate selected phrases in a document into phrases in the query.

8. Conclusions

We have presented an approach to information retrieval that exploits ideas and methods of statistical machine translation. After outlining the approach, we presented a simple, motivated model of the document-query translation process. With the EM algorithm, the parameters of this model 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 methods such as the vector-space approach, but without explicit query expansion and term weighting schemes, which are inherent in the translation approach itself.

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). Slides available at `http://www.research.digital.com/SRC/personal/monika/-focs-talk-m78/ppframe.htm`

[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 (1993a). "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 (1993b). "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.