21 language models @ TREC:

A language modeling approach to the Text Retrieval Conference*

Djoerd Hiemstra
University of Twente

Wessel Kraaij
TNO-TPD

Abstract

In this paper we present the language modeling approach to information retrieval as a toolbox to systematically combine information from different sources. Four TREC subtasks (Ad Hoc, Entry Page, Adaptive Filtering and Cross-language) are used to illustrate the application of language models to different information retrieval problems.

1 Introduction

The EU-funded project "Twenty-One" started in 1996 originally with the objective to build a prototype cross-language information retrieval system. This led to a number of fruitful TREC participations, in which we evaluated the use of a probabilistic modeling approach known as *language modeling*. This chapter describes the Twenty-One language modeling experiments on a variety of TREC tasks.

The term "language models" originates from probabilistic models of language generation developed for automatic speech recognition systems in the early 1980's (see e.g. Rabiner 1990). Language models assign a probability to a piece of text. For instance, "how are you today" would be assigned a higher probability than "cow barks moo soufflé", because the words in the former phrase (or word pairs or word triples if so-called n-grams are used) occur much more frequently in English than the words in the latter phrase. Automatic speech recognizers use language model probabilities to improve recognition performance. Language models were applied to information retrieval by a number of research groups in the late 1990's (Ponte and Croft 1998; Hiemstra and Kraaij 1999; Miller et al. 1999; Berger and Lafferty 1999; Ng 2000). For information retrieval, language models are built for each document. By following this approach, the language model of the book you are reading now would assign an exceptionally high probability to the word "TREC" indicating that this book would be a good candidate for retrieval if the query contains this word.

1.1 Probabilistic models and IR: an overview

Interestingly, probabilistic modeling has been around in information retrieval for much longer than the late 1990's, or even the 1980's, and – in a way – the language modeling approach builds directly on many of the ideas of the more traditional probabilistic models for information retrieval.

*Published as Chapter 16 of E.M. Voorhees and D. Harman (eds.), TREC: Experimentation and Evaluation in Information Retrieval, MIT Press, 2005

It is fair to say that the approach to information retrieval presented in this chapter was originally introduced by Maron and Kuhns (1960). In a time when manual indexing was still guiding the field, they suggested that an indexer, which runs through the various index terms q that possibly apply to a document D, might assign a probability $P(q|D)$ to a term given a document instead of making a yes/no decision. Using Bayes' rule and a document prior $P(D)$, they then suggest to rank the documents by the probability that the document is relevant $P(D|q)$. Maron and Kuhns described how $P(D)$ could be inferred automatically, but they were not really looking for automatic ways to infer $P(q|D)$, or if they were, they did not know how the probabilities $P(q|D)$ could be defined.

Van Rijsbergen (1986) introduced an idea quite similar to that of Maron and Kuhns: modeling information retrieval as documents "implying" the query terms with some probability $P(q|D)$. Again, the definition of the probabilities $P(q|D)$ was not easily found, hampering its application to practical retrieval problems. In lack of such a definition, the INQUERY system (Turtle and Croft 1992) used some ad-hoc combination of *tf.idf* weights to define the probabilities $P(q|D)$.

But there are alternatives to *tf.idf* weighting. The well-known probabilistic model by Robertson and Sparck-Jones (1976) is built around the probability of relevance. The model can be seen as a discriminative model, i.e. it tries to separate the relevant documents from the non-relevant documents by following the well-known "naive Bayes" assumption (Duda and Hart 1973): The terms in the document (usually restricted by some query terms) are conditionally independent given relevance (or non-relevance). We might look at this as a mechanism to generate an unseen relevant, or non-relevant, document. A substantial set of relevant documents is needed however to estimate the probabilities for a single query, which makes it hard to apply the model to practical retrieval situations like the TREC Ad hoc task.

Another interesting probabilistic modeling approach is suggested by Bookstein and Swanson (1974) and Harter (1975). They assume that documents are created by a random stream of term occurrences. For each term, the collection can be divided into two subsets, where one subset treats a subject represented by a term to a greater extent than the other. The number of term occurrences *tf* may then be modeled by a mixture of two Poisson distributions, one for each subset. Unfortunately, as with relevance, it is unknown to which subset each document belongs, making it hard to apply the model to practical situations. The "two Poisson" model did however inspire the Okapi BM25 weighting algorithm (Robertson and Walker 1994).

Knowing what we know now from the language modeling approach, and looking back at the history of probabilistic modeling for information retrieval, we might observe that we are actually using many of the early ideas. We will use different TREC tasks to illustrate different aspects of the language modeling approach. Where appropriate, we will refer to the classics of probabilistic modeling for information retrieval.

1.2 A language model for every task

Different tracks in TREC call for different approaches to information retrieval. Some tasks, like the Ad Hoc topic search task, might already be served quite well by a basic retrieval approach, but many other TREC tasks call for including some special "non-content" information. Cross-language information retrieval obviously needs to deal with some form of automatic translation, Adaptive Filtering needs to deal with the user's feedback on the selected documents; and the Web Entry Page search task might benefit from e.g. counting the number of in-links to a document.

This chapter will introduce a language modeling approach for four TREC subtasks. Every model is built around the basic query generation language model, but each model has its own little twist. In Section 2, we introduce the basic model and show how document priors can improve performance on the TREC Ad Hoc task. Section 3 elaborates on the use of document priors by applying them to the TREC Web Entry Page task. Section 4 extends the model by including a statistical translation model for application to the TREC Cross-language retrieval task. Finally, Section 5 presents a relevance feedback method, which is applied to the TREC Adaptive Filtering task.

2 The basic language model and the TREC Ad Hoc task

We believe that many of the early probabilistic models failed as general models for diverse retrieval tasks, because they failed to answer a few quite fundamental questions about the use of probabilistic models in general: What justifies the use of probabilistic models? What probability mechanisms are involved? And, how do these mechanisms fit the reality of information retrieval?

The use of probability theory might be justified by modeling the process of a user formulating a query Q while he/she has a relevant document D in mind. Imagine picking a word at random from this page by pointing at the page with closed eyes. Such a process would define a probability $P(Q|D)$ which might be used as Van Rijsbergen's (1986) "logical implication". Is this really how users formulate queries? A pragmatic answer to that question would be: If such a model achieves good performance on a real retrieval task, then the model fits reality well. Test collections, like those developed in the TREC conferences can be used to measure a model's performance on realistic retrieval tasks in a controlled setting.

Actually, the answer to the above question is "no", such a model does not work very well in practice because of the so-called sparse data problem (Manning and Schütze 1999). The mechanism above suggests that terms that do not occur in a document are assigned zero probability, but the fact that a term is never observed in a document does not mean this term is never entered in a query for which the document is relevant. The reality of information retrieval is that users are not very good in formulating queries. Many query terms do not seem to come from relevant documents at all, they seem to come from some general vocabulary. These might be words like "Find documents about" (which are often found in TREC topic descriptions, see Harman 2005), but it might be any other query term that seems plausible but that does not contribute to retrieval performance.

We will call query terms that presumably were generated from the relevant document the *important* terms, and the terms that presumably were generated from the user's general vocabulary the *unimportant* terms. Given a document collection C and a relevant document D, the process of generating a query term q_i might be modeled by a mixture of two probability measures: $P(q_i|D)$ for the important terms, and $P(q_i|C)$ for the unimportant terms. Of course, just from looking at the query it is unknown which terms are the important terms and which are the unimportant terms. Therefore, the mixture parameter λ defines the unknown probability of term importance. Equation 1 defines our basic language model if we assume that each term is generated independently from previous terms given the relevant document.

$$P(q_1, q_2, \cdots, q_n|D) \;=\; \prod_{i=1}^{n} \Big(\lambda P(q_i|D) + (1-\lambda) P(q_i|C) \Big) \tag{1}$$

The basic language model addresses both the sparse data problem – all terms are generated with a non-zero probability – and the fact that queries consist of content words and query jargon (Zhai and Lafferty 2001). Interestingly, like Maron and Kuhns (1960), we take the document / query (term) implication $P(Q|D)$ as the basis of our model. Like Harter (1975), we assume an explicit probability mechanism that defines a mixture model.

Ideally, we would like to train the probability of an unimportant term on a large corpus of queries. In practice however, we will use the document collection C to define these probabilities, hence the notation $P(q_i|C)$. Whenever we use the TREC topic descriptions, a small number of words like "Find documents about" will be removed from the query to compensate for the lack of a query corpus. We use Bayes' rule as shown in Equation 2 to define the posterior probability of the document D being relevant given the query $Q = q_1, \cdots, q_n$.

$$P(D|q_1, q_2, \cdots, q_n) \;=\; \frac{P(q_1, q_2, \cdots, q_n|D)P(D)}{P(q_1, q_2, \cdots, q_n)} \tag{2}$$

Note that the denominator on the right hand side does not depend on the document. It might therefore be ignored when a document ranking is needed. The prior $P(D)$ however, should only be ignored if we assume a uniform prior, that is, if we assume that all documents are equally likely to be relevant in absence of a query. Some non-content information, like the source of a document, its age, etc. might contain some hints on whether it is likely to be relevant or not. Robertson and Walker (1994) and Singhal et al. (1995) argued that on the Ad Hoc task, the length of a document already contains some clues. The longer the document, the more likely it is to be relevant.

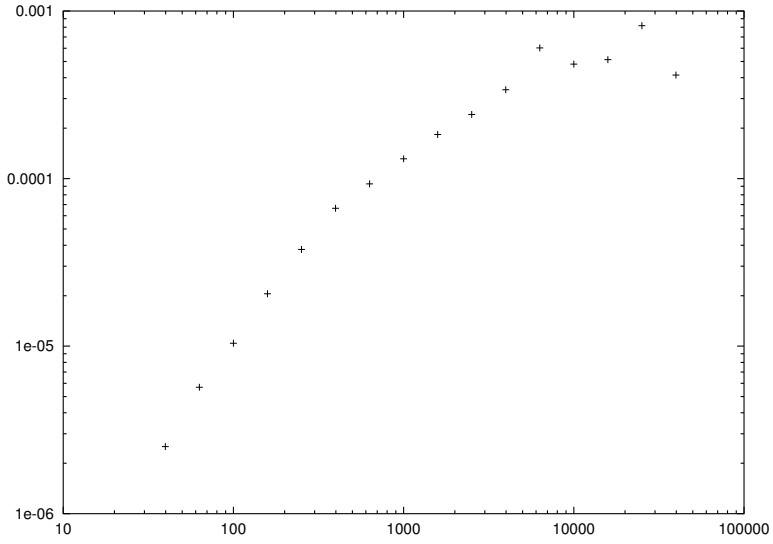

Figure 1: Prior probability of relevance $P(D)$ given document length on the Ad Hoc task

Figure 1 shows the probability of relevance given the document length for the TREC Ad Hoc task. We divided the document length, which varies from documents containing only one or two words to documents containing over 10,000 words, into 16 bins on a log scale. Each point on the plot marks the probability of relevance of the documents in one of these bins. The 16 bins and the corresponding probabilities define a discrete probability measure $P(D)$

which takes one of 16 different values based on the bin in which D falls. As such, it can be used directly in Equation 2. Alternatively, looking at the plot, one could make the general modeling assumption that the a-priori probability of relevance is taken as a linear function of the document length, so:

$$P_{\text{doclen}}(D) = c \cdot \mathit{doclen}(D) \tag{3}$$

where $\mathit{doclen}(D)$ is the total number of words in document D, and c is a constant that can be ignored in the ranking formula.

runname	description	avg. prec.
tno7cbm25	BM25 weighting	0.232
tno7tw3	language model	0.241
tno7tw4	language model with doclen prior	0.251

Table 1: Results of TREC-7 Ad Hoc runs

Table 1 lists the results of TREC-7 Ad Hoc experiments using title, description and narrative of the topics[1] (Harman 2005b), one run using the BM25 algorithm (implemented as in Singhal et al. 1995), and one using the language modeling algorithm with a document length prior. As mentioned above, the language model has an unknown parameter λ that defines the mixture of local and global frequency information. We used $\lambda = 0.15$ based on experiments on the TREC-6 Ad Hoc collection (Hiemstra and Kraaij 1999).

3 Prior probabilities and the TREC Entry Page task

An application where the prior information component of the basic model (cf. Equation 2) is even more important is the TREC Entry Page task, which was run as a new task of the Web track of TREC-10 in 2001. Earlier issues of the Web track had already targeted the issue of integrating information about link-structure with traditional IR models for Ad Hoc retrieval. Since these attempts had shown no significant benefit for link based approaches and it was realized that links played an important role in commercial search engines like Google (Brin and Page 1998), a special task was created investigating the important subclass of searching for the entry page of an organization. This decision readjusted TREC's focus to "real-life" search tasks, which were no longer limited to the classical information seeking queries modeled by the initial Ad Hoc and Routing tasks.

For an elaborate description of the Entry Page task we refer to Hawking (2005) and Kraaij, Westerveld, and Hiemstra (2002). The basic idea is that each organization has an entry page on the web, functioning as a portal to its information. Entry page search differs in several aspects from Ad Hoc search: i) there is only one (sometimes a few) entry page(s) for a particular organization, so high precision is important, ii) web data is different from news data, the main difference being link structure. We could thus formulate the challenge of the Entry Page task as follows: integrate knowledge about external properties and context of a document with our basic model in order to improve high precision.

We will show that the generative probabilistic approach we have presented in Section 2 can easily accommodate information derived from these knowledge sources. In Formula 4, l refers to the event a user likes a document, given a certain task. In the context of entry page search, a user is interested in an entry page as specified by the query Q. Formula 4

[1] The runs were redone based on the official Twenty-One experiments.

decomposes the posterior probability that a document is being liked given the query and a specific document by applying Bayes' rule.

$$P(l|D,Q) \;=\; \frac{P(Q|l,D)P(l|D)}{P(Q|D)} \tag{4}$$

We include l here to relate the language modeling approach to the Robertson/Sparck-Jones model (Lafferty and Zhai 2003), and to show that estimating priors is not really different from estimating the probability of relevance (i.e., the probability that the user 'likes' the document) as done by Robertson and Sparck-Jones (1976). Since our aim is to rank documents by their posterior probability, we can apply any convenient order preserving transformation. It is customary to work with the log-odds of being liked instead of the pure probability, since it is difficult to estimate the normalizing probability $P(Q|D) = P(Q,l|D) + P(Q,\bar{l}|D)$. We further approximate the probability of the query given a document which is not relevant by generating it from the background collection.

$$\log\frac{P(l|D,Q)}{P(\bar{l}|D,Q)} \;=\; \log\frac{P(Q|l,D)}{P(Q|\bar{l},D)} + \log\frac{P(l|D)}{P(\bar{l}|D)} \;=\; \log\frac{P(Q|l,D)}{P(Q|C)} + \log\frac{P(l|D)}{P(\bar{l}|D)} \tag{5}$$

Assuming term independence and applying smoothing by linear interpolation with a background model leads to:

$$\log\frac{P(l|D,Q)}{P(\bar{l}|D,Q)} \;=\; \sum_{q_i\in Q}\big(\log\frac{\lambda P(q_i|l,D)+(1-\lambda)P(q_i|C)}{P(q_i|C)}\big) + \log\frac{P(l|D)}{P(\bar{l}|D)} \tag{6}$$

This model can be interpreted as a Bayesian update process. The prior log-odds of being liked is initially purely determined by the document properties itself and subsequently updated with the additional knowledge of the likelihood of the query given the fact that the document is liked versus the likelihood of the query given a background model. We will see that unlike the Ad Hoc task, prior knowledge is of utmost importance for entry page search.

We investigated three properties of web pages in order to provide an initial estimate of the prior probability of a web page: document length, the number of documents pointing to the document via a hyperlink (inlinks) and the form of the URL. It is well known that longer documents have a higher probability of relevance for Ad Hoc search, but it is not clear whether long documents have a higher probability of being an entry page. The conjecture that a high number of incoming links indicates that the page pointed to is an entry page is already much more intuitive. Finally, most web users, even with very little search experience know that entry pages usually have short URL's. There are probably some very simple explanations for this fact i) entry pages benefit from being short, since they can be memorized more easily ii) since information on the web is often stored in a hierarchical file system, entry pages are usually located in the top or at least high in the directory hierarchy.

We measured the informativeness of each of the three features in the following way: for each feature x_i, we divided the document set in disjunct classes v_{ij} and directly estimated $P(l|x_i = v_{ij})$ on a training set. This set consisted of 100 training topics and corresponding entry pages of the TREC-2001 Web track Entry Page task. The probability of being an entry page given the information that a document belongs to a certain bin is defined by:

$$P(l|x_i = v_{ij}) = \frac{c(EP, v_{ij})}{c(v_{ij})}, \tag{7}$$

where $c(v_{ij})$ is the cardinality of class v_{ij} and $c(EP, v_{ij})$ is the number of entry pages in class v_{ij}. The training set is small, but probably sufficient to estimate probabilities for a small number of classes. Size and number of classes are chosen such that a class contains at least 5 home pages of the training set, while trying to maximize the number of classes in order to reduce variance. The goal of this procedure is to define a partitioning of the data set, by means of feature restrictions, which correlates well with being an entry page.

3.1 Document Length

For the document length feature, we created classes by quantization into 16 bins on a log scale. Section 2 showed that document length priors are useful in an Ad Hoc search task. Here we investigate whether the length of a document is also a useful indicator of the probability that a document is an entry page. Figure 2 shows a plot of the probability of relevance versus page length, calculated on the training data provided for the Entry Page task of TREC-2001's Web track. Note that the probability of relevance is also plotted on a log scale; therefore bins with zero probability of relevance do not appear.

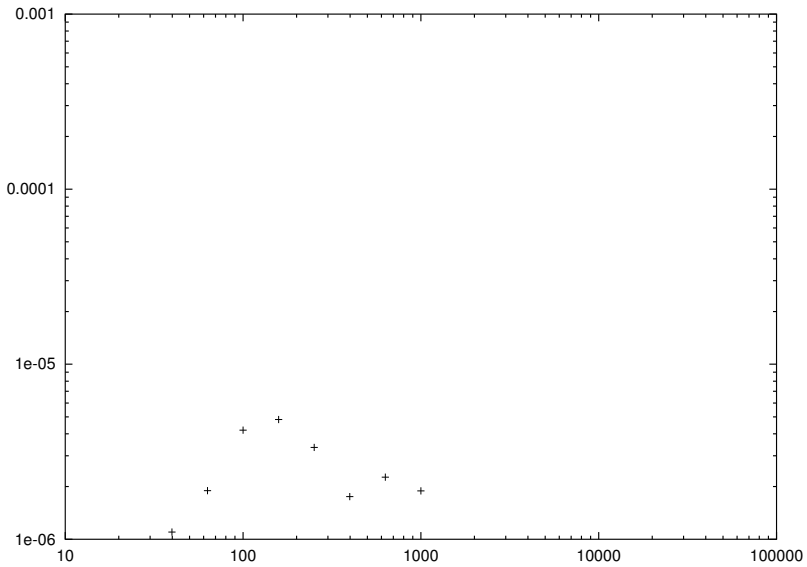

Figure 2: Prior probability of relevance given document length on the Entry Page task ($P(\textit{entry page}|\textit{doclen})$)

Indeed document length can predict the relevance of a page, since the distribution is not uniform. Pages with a medium length (60-1000 words) have a higher probability, with a maximum around 100-200 words. However, the differences are much less marked than for Ad Hoc search.

3.2 Number of inlinks

For the inlinks feature, we created classes by quantization into 9 bins on a log scale. The number of inlinks is a much better predictor of being an entry page as is shown in Figure 3 (which is based on 18 bins). The prior could probably also be modeled as a linear function.

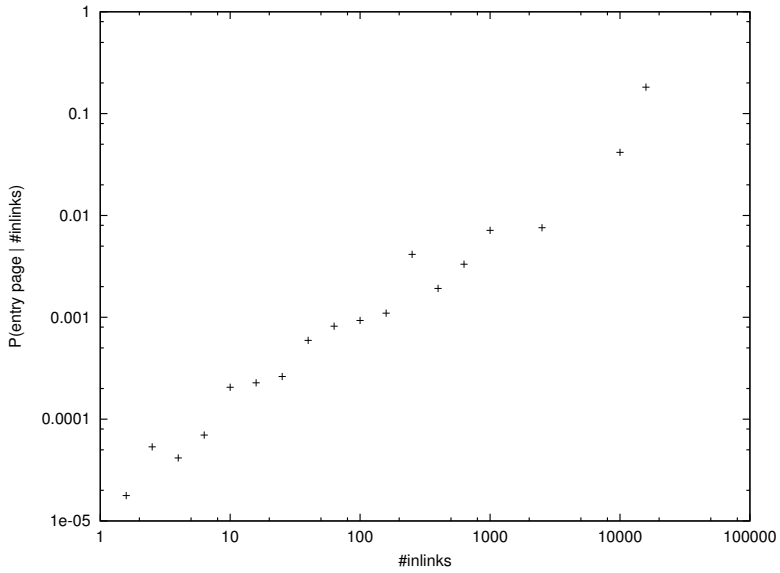

Figure 3: Prior probability of relevance given number of inlinks on the Entry Page task ($P(\textit{entry page}|\#\textit{inlinks})$)

3.3 URL depth

For the URL attribute of the web pages we defined four classes in the following way:

root: a domain name, optionally followed by 'index.html'
(e.g. `http://trec.nist.gov`)

subroot: a domain name, followed by a single directory, optionally followed by 'index.html'
(e.g. `http://trec.nist.gov/publications/`)

path: a domain name, followed by an arbitrarily deep path, but not ending in a file name other than 'index.html'
(e.g. `http://trec.nist.gov/publications/trec8/system-descriptions/`)

file: anything ending in a filename other than 'index.html'
(e.g. `http://trec.nist.gov/resources.html`)

The resulting probabilities for the different URL-types are listed in Table 2. Note that the prior probabilities differ several orders of magnitude; a root page has an almost 2000 times larger probability of being an entry page than any other page.

Document type	$P(EP)$
root	$6.44 \cdot 10^{-3}$
subroot	$3.95 \cdot 10^{-4}$
path	$9.55 \cdot 10^{-5}$
file	$3.85 \cdot 10^{-6}$

Table 2: Prior probabilities for different URL-types, estimated on the training data

3.4 Generalizing to a combination of features

We have identified several non-content features that seem promising in discriminating between relevant and non-relevant web pages. To estimate the prior on a combination of these features, we investigated the following. Let's assume each document D is described by a number of features $x_1, x_2, \ldots, x_n$, where x_1 represents URL depth, x_2 represents the number of inlinks, etc. The problem can be formalized as follows, we want to estimate:

$$\log \frac{P(l|D)}{P(\bar{l}|D)} \;=\; \log \frac{P(l|x_1, x_2, \ldots, x_n)}{P(\bar{l}|x_1, x_2, \ldots, x_n)} \tag{8}$$

A combined inlinks/URL-depth prior would already need $9 \times 4 = 36$ model parameters. Because the training set of 100 entry pages is too small to reliably estimate this many parameters, we assume conditional independence of the features given relevance, giving:

$$\log \frac{P(l|D)}{P(\bar{l}|D)} \;=\; \sum_{i=1}^{n} \log \frac{P(x_i|l)}{P(x_i|\bar{l})} + \log \frac{P(l)}{P(\bar{l})} \tag{9}$$

Some readers might recognize the above definition as the probabilistic model of information retrieval by Robertson and Sparck-Jones (1976). An important difference with the Robertson/Sparck-Jones model is the fact that the features are *not* the index terms, but some non-content information about the document. As a consequence, we are able to use relevance information over 100 entry page queries of our training set, and reliably estimate the probability of relevance. In practice, the training set did not contain labeled *non* entry pages, which made it difficult to estimate $P(x_j|\bar{l})$. Instead we used $P(x_j)$ which might be a good approximation as there are many more web pages than entry pages in the training data.

Instead of assuming conditional independence between inlinks and URL-depth we might also merge some of the 36 classes. We only partitioned the root URL class by number of inlinks, since most entry pages have root URL's, so a further division based on inlinks can still yield reasonably reliable parameter estimates. Table 3 shows the statistics of the 12 classes.

Document type t_i	#entry pages		#WT10g		$P(EP)$
root with 0-1 inlinks	11	(10.1%)	1484	(0.0%)	0.0074
root with 2-4 inlinks	14	(12.9%)	3431	(0.2%)	0.0041
root with 5-9 inlinks	14	(12.9%)	2446	(0.1%)	0.0057
root with 10-19 inlinks	8	(7.4%)	1404	(0.0%)	0.0057
root with 20-49 inlinks	12	(11.1%)	1110	(0.0%)	0.011
root with 50-99 inlinks	5	(4.6%)	412	(0.0%)	0.012
root with 100-199 inlinks	6	(5.5%)	205	(0.0%)	0.029
root with 200-999 inlinks	5	(4.6%)	175	(0.0%)	0.028
root with 1000+ inlinks	4	(3.7%)	38	(0.0%)	0.11
subroot	15	(13.9%)	37959	(2.2%)	0.00043
path	8	(7.4%)	83734	(4.9%)	0.00010
file	6	(5.6%)	1557719	(92.0%)	0.000038

Table 3: Distribution of entry pages and WT10g over different document types

Table 4 shows the mean reciprocal rank (MRR) for runs on the TREC-2001 test collection in combination with different priors. Both inlinks and URL-depth help to increase search effectiveness, especially the URL prior is highly effective. The combination of the inlinks prior and the URL prior based on a conditional independence assumption shows somewhat lower performance than the run based on only the URL. This might indicate that the independence assumption does not hold. The set-up based on 12 disjoint classes as defined in Table 3 yielded a MRR of 0.7832, which is a small improvement with respect to the run based on just the content and URL depth information. We think results could be even more improved, with a larger training set and a more principled way to define classes, e.g. using methods proposed by (Dougherty et al. 1995).

Ranking method	MRR	Description
$P(Q\|D)$	0.3375	no prior
$P(Q\|D)P_{inlinks}(D)$	0.5064	inlinks prior (analytical)
$P(Q\|D)P_{URL}(D)$	0.7705	URL-depth prior (4 bins)
$P(Q\|D)P_{inlinks+URL}(D)$	0.7504	combined inlinks/URL-depth prior assuming conditional independence
$P(Q\|D)P_{inlinks+URL}(D)$	0.7832	combined inlinks/URL-depth prior using direct estimation (12 classes)

Table 4: Results for different priors

4 The translation model and the TREC Cross-language task

Our work on Cross Language Information Retrieval (CLIR) in TREC and the closely related CLEF evaluation is another good example that the basic language model can be easily extended for a new task. The CLIR problem deals with the retrieval situation where the query is formulated in a different language than the documents (Harman, Braschler and Sheridan 2005). A simple approach is to use a Machine Translation (MT) system to translate either the query or all of the documents, such that the problem is reduced to a monolingual problem. There are several caveats and disadvantages to this approach. Firstly, full MT is not available for all language pairs, secondly there are reasons to believe that an MT based approach is not optimal, since it provides just one translation. The fact that multiple translations could be helpful to find relevant documents (we assume a query translation approach, since it is more efficient) is very similar to a monolingual situation, where the searcher tries to enhance recall by providing synonyms for salient terms. Professional searchers use faceted queries (Pirkola et al. 1999) for this purpose, where alternatives for each concept are specified as a disjunction and these disjunctions themselves are connected by a conjunction operator. The idea to use some kind of Boolean structure for CLIR was first proposed by Hull (1997). For TREC-7 and TREC-8, we designed a probabilistic version of this idea, by realizing that a conjunction can be modeled by summing over probabilities of translation alternatives

We reformulate our basic model here to show the derivation of the extended model. Suppose we have an English document D_E, and a French query Q_F consisting of n different words

(types) f_i, each occurring $tf_q(f_i)$ times in Q_F, then we can reformulate Formula 1 as follows:

$$\log P(Q_F|D_E) = \sum_{i=1}^{n} tf_q(f_i) \log(\lambda P(f_i|D_E) + (1-\lambda)P(f_i)) \tag{10}$$

Estimation of $P(f_i|D_E)$ i.e. a word in the source language given a document in the target language is of course more difficult than its monolingual counterpart. By introducing a variable for a word in the target language, we can reduce $P(f_i|D_E)$ to two simpler estimation problems:

$$P(f_i|D_E) \;=\; \sum_{j=1}^{N_E} P(f_i, e_j|D_E) \;=\; \sum_{j=1}^{N_E} P(f_i|e_j, D_E)P(e_j|D_E) \;\approx\; \sum_{j=1}^{N_E} P(f_i|e_j)P(e_j|D_E) \tag{11}$$

where N_E is the size of the English target vocabulary. The approximation $P(f_i|e_j, D_E) \approx P(f_i|e_j)$ assumes that translation is independent of the document context. $P(f_i)$ can either be directly estimated on a corpus in the source language, but also – by the same derivation – on a corpus in the target language. The latter has the advantage that both estimates stem from corpora from the same size and domain, which makes the estimates better comparable. After substitution, Formula 10 can be rewritten as:

$$\log P(Q_F|D_E) = \sum_{i=1}^{n} tf_q(f_i) \log \sum_{j=1}^{N_E} \big[P(f_i|e_j)(\lambda P(e_j|D_E) + (1-\lambda)P(e_j|C_E))\big] \tag{12}$$

where n is the number of different terms in the query. Although often referred to as query translation, we think that this is actually a model for document (model) translation, since the language model representing the document is first "mapped" to the source language, before the actual matching process takes place. The approach is different though from what is usually referred to as document translation in CLIR, since in that case a document model is estimated on a translated document, instead of translating the document model. We will refer to this model as "document model translation" (**dmt**).

An alternative approach is to match in the target language and to use the reverse translation model $P(e_j|f_i)$. We actually first normalized the basic ranking Formula 1 by taking the geometric mean of $P(Q|D)$ yielding:

$$\begin{aligned} \log P(Q|D)^{\frac{1}{ql}} \;=\; \log \frac{P(Q|D)}{ql} \;&=\; \sum_{i=1}^{n} \frac{tf_q(q_i)}{\sum_{k=1}^{n} tf_q(q_k)} \log(\lambda P(q_i|D) + (1-\lambda)P(q_i|C)) \\ &=\; \sum_{i=1}^{n} P(q_i|Q) \log(\lambda P(q_i|D) + (1-\lambda)P(q_i|C)) \end{aligned} \tag{13}$$

Note that the query length ql is defined as follows: $ql = \sum_{k=1}^{n} tf_q(q_k)$. We can restate this as a CLIR model where the event space is defined over the vocabulary in the target language:

$$\log P(Q_F|D_E)^{\frac{1}{ql}} \;=\; \sum_{i=1}^{n} P(e_i|Q_F) \log(\lambda P(e_i|D_E) + (1-\lambda)P(e_i|C_E)) \tag{14}$$

In this case, $P(e_i|Q_F)$ can be estimated with the aid of a reverse translation model and a derivation similar to (11):

$$\log P(Q_F|D_E)^{\frac{1}{ql}} \;=\; \sum_{i=1}^{n} \sum_{j=1}^{N_E} P(e_j|f_i)P(f_i|Q_F) \log(\lambda P(e_j|D_E) + (1-\lambda)P(e_j|C_E)) \tag{15}$$

In formula (15) the query is first mapped to a probability distribution in the target language, by assuming word by word context insensitive translation. We will refer to this model as "query model translation" (**qmt**). Since $P(e_j|f_i)$ is zero for all English words e_j which are not translations of a query term f_i, the model is just as efficient as the **dmt** model.

4.1 Related work

A similar model has been developed for the Chinese and Arabic track by BBN (Xu et al. 2001). Our model is also quite similar to the IR model proposed by Berger and Lafferty (1999), who view monolingual IR as a translation process. These models have a different approach to smoothing, since $P(f_i)$ is estimated on a source language corpus. Another related approach is the use of structured queries, as advocated by Pirkola (1998). Here, translations of a term form an equivalence class by using INQUERY's synonym operator. This approach is similar to a special instantiation of the translation probability matrix, namely the case where $P(f_i|e_j) = 1$ for each translation of a source term f_i. We will refer to this model by **syn**.

Our model assumes context independent word-by-word translation, which is clearly too simplistic to reflect real-world translation problems. Recently, new language model based approaches for CLIR have been proposed, which start from weaker assumptions. In particular, immediate context is taken into account by using a bigram model (Federico and Bertoldi 2002) or document aligned corpora are exploited to estimate cross-lingual relevance models (Lavrenko et al. 2002). However, the gain in effectiveness of these models is relatively small and their efficiency is unfortunately much lower.

4.2 Comparison of different CLIR model variants

We will illustrate the relative performance of the models on the CLEF 2000 dataset, for 40 English queries on the French subcollection (Le Monde 87,191 docs) and 40 French queries on the English subcollection (LA times, 113,005 docs). Documents and queries were lemmatized using the Xelda morphological toolkit from Xerox Grenoble.

We will compare the three discussed systems (**qmt**, **dmt**, **syn**, complemented with monolingual runs, a run based on query model translation with equal probabilities **qmt-eq**, a run where we took just the best translation in a query model translation setting (**qmt-bt**) and a run where the queries were translated using the web based MT service Babelfish **MT**.

We estimated the translation models $P(f_i|e_j)$ and $P(e_j|f_i)$ on a parallel web corpus constructed at RALI (Nie et al. 1999; Kraaij et al. 2003). The translation models were pruned by taking the 100,000 best translation relations according to an entropy criterion.

As an illustration, we present the French translation of the word "drugs" taken from query `C003` about drug policy, tuned for several CLIR models in Table 5. It is clear that the `dmt` has a query expansion potential. However, it expands both the medical and the narcotic sense. We will see that the dmt model is able to take advantage of this query expansion effect, even if the expansion set is noisy.

Table 6 lists the results for the different CLIR models. The bottom of the table shows a few statistics about the translation models: **#fw** is the average number of translations in the forward translation model (source language to target language), which is used for all the **qmt**-based runs. **#rev** is the average number of translations of the model used for the **dmt** run. The **%missed** statistic refers to the percentage of query terms, for which no translations were found.

run id	translation
MT	drogues
qmt	<drogue,0.44, medicament,0.36; consommation, 0.06; relier, 0.01; consommer, 0.02; drug; 0.01; usage, 0.01; toxicomanie, 0.01; substance, 0.01; antidrogue, 0.01; utilisation, 0.01; lier, 0.01; therapeutique, 0.01; actif, 0.01; pharmaceutique, 0.01>
qmt-eq	<drogue,0.06, medicament,0.06; consommation, 0.06; relier, 0.06; consommer, 0.06; drug; 0.06; usage, 0.06; toxicomanie, 0.06; substance, 0.06; antidrogue, 0.06; utilisation, 0.06; lier, 0.06; therapeutique, 0.06; actif, 0.06; pharmaceutique, 0.06>
qmt-bt	<drogue,1.0>
dmt	<médicament,0.79; drogue,1.0; toxicomane,0.23; drug,1.0; alcoolisme,0.24; drugs,0.70; stupéfiant,0.34; antidrogue,1.0; médicamenteux,0.36; droguer,1.0; pharmacorésistance,0.47; pharmacothérapie,0.25; assurance-médicaments,0.33; relargage,0.53; pharmacorésistants,0.28; anti-inflammatoire,0.17; surdose 0.28; stéroïdiens,0.35, drogué 0.61; pharmacodépendance,0.27 narcotrafiquants,0.57; anticancéreux,0.22; escherichia,0.14; pharmacovigilance,0.49; selby,0.16; homelessness,0.14; bounce,0.23; anti-drogues,0.14; antidiarrhéique,0.12; imodium,0.12; surprescription,0.10>
syn	<drogue; medicament; consommation; relier; consommer; drug; usage; toxicomanie; substance; antidrogue; utilisation; lier; therapeutique; actif; pharmaceutique>

Table 5: Example translations of the word 'drugs'. The numbers in the top part of the table are the translation probabilities $P(f_i|e_j)$ and the numbers in the bottom part of the table are the reverse translation probabilities $P(e_j|f_i)$.

There are several effects that can be observed from this table. First of all, translations based on a noisy parallel web corpus outperform the high quality lexicons used by Babelfish. We think that this is due to the fact that our model is able to exploit multiple translations. Secondly, we see that both the **qmt** and **dmt** models are very well able to deal with many translations: the translation models provide around 10 translations per term on average. It is clear that a lot of those "translations" are probably highly related terms, so one could argue that we actually do some form of query expansion on a parallel corpus. Comparing the several variant **qmt** runs show that using the translation weighting is very important. If we replace the corpus based estimates by a uniform probability $(1/n)$, the retrieval effectiveness is significantly reduced. Using the best translation only is even better, which also bears evidence that weighting translations is crucial. The importance of weighting translations (or proper embedding of translation into the model) is also illustrated in a direct comparison of the (unweighted) synonym run following Pirkola (**syn**) and the run based on document model translation. The approach based on unweighted synonyms is clearly not able to handle the noisy translations from the web corpus in a robust way.

Finally, there is no single best CLIR model for the two CLIR tasks: for the EN-FR task the **dmt** model has the best retrieval effectiveness and for the FR-EN model it is the qmt model. This actually correlates well with the percentage of terms for which a translation is found in the respective models. In other words the $P(e|f)$ model is better than the $P(f|e)$ model. Since the models were trained on exactly the same sentenced aligned dataset, using the same techniques this asymmetry is surprising. Perhaps it could be due to the differences

run id	EN-FR		FR-EN	
mono	0.4489	(100.0%)	0.4323	(100.0%)
MT	0.3141	(69.9%)	0.3908	(90.4%)
qmt	0.3525	(78.5%)	0.4207	(97.3%)
qmt-eq	0.2698	(60.1%)	0.3777	(87.4%)
qmt-bt	0.3336	(74.3%)	0.3834	(88.7%)
dmt	0.3732	(83.1%)	0.3693	(85.4%)
syn	0.2445	(54.5%)	0.2352	(54.4%)
%missed fw	11.43		13.42	
#fw	8.82		8.39	
%missed rev	10.74		15.45	
#rev	12.16		14.59	

Table 6: Mean Average Precision and translation statistics (best 100k parameters)

in verbosity of the French and English language. A French translation of an English text is approximately 10% longer. This means that it is more difficult to align an English word to a French word than a French word to an English word in a sentence aligned corpus.

5 Relevance feedback and the TREC Adaptive Filtering task

The TREC Adaptive Filtering task evaluates systems that actively disseminate personalized information to the user. A filtering system receives a constant stream of news, e.g. from USENET, and alerts the user only if a news item matches the user's profile. The user is able to control the system by giving feedback, either *yes*, I like this item, or *no*, I do not like this item (Robertson and Hull 2005).

In this section, our special interest lies in the development of a relevance feedback algorithm for the language modeling approach. In principle, reasoning about relevance feedback for a query generation language model is problematic, although some rather ad-hoc solutions have been proposed by Miller et al. (1999) and Ponte (2000). The problem is the following: We assumed in Section 2 that queries have been generated from one (and only one) relevant document. So, it is easy to reason about multiple queries (one might argue that we reasoned about multiple queries when we used the translation models in Section 4), but it is not as easy to reason about multiple relevant documents (Sparck-Jones, Robertson, Hiemstra, and Zaragoza 2003). A possible solution is to 'reverse' the language model by assuming that documents are generated by a profile or a 'relevance model' as done by Jin et al. (1999) and Spitters and Kraaij (2000). The query generation model and the document generation model might be combined as well to model two-staged retrieval (or pseudo relevance feedback) as suggested by (Lavrenko and Croft 2001).

In this section we present a relevance feedback approach for the query generation language models by introducing a term-specific smoothing parameter λ_i for each term q_i in the query. Term-specific smoothing models some characteristics of practical retrieval systems that are often left 'outside' the retrieval model, like stop words and mandatory terms (Hiemstra 2002): From Equation 16, it is easy to verify that if $\lambda_i = 0$, then the term does not affect the ranking (like a stop word), and that if $\lambda_i = 1$, then the term is mandatory: All documents that do

not contain the term are assigned zero probability.

$$P(q_1, q_2, \cdots, q_n|D) \;=\; \prod_{i=1}^{n}\Big(\lambda_i P(q_i|D) + (1-\lambda_i)P(q_i|C)\Big) \tag{16}$$

What is the probability mechanism behind such a model? Query generation is much like coin tossing. For each query term, imagine one first throws a coin. If the coin comes up heads, then we take the general model. If it comes up tails then we take the relevant document's model. We might think of a mechanism for which there is a different (unfair) coin for each query term. Each document that is relevant for a query gives some independent evidence on which coin is used on each draw.

E-step: $$m_i = \sum_{j=1}^{r} \frac{\lambda_i^{(p)} \cdot P(q_i|D_j)}{\lambda_i^{(p)} P(q_i|D_j) + (1-\lambda_i^{(p)})P(q_i|C)}$$

M-step: $$\lambda_i^{(p+1)} = \frac{m_i + 1.5}{r + 3}$$

Figure 4: relevance feedback algorithm: EM-algorithm

The EM-algorithm (Dempster, Laird, and Rubin 1977) of Figure 4 iteratively maximizes the probability of the query $q_1, q_2, \cdots, q_n$ given r independently observed relevant documents $D_1, D_2, \cdots, D_r$. Before the iteration process starts, the importance weights λ_i are initialized to their default values $\lambda_i^{(0)}$, where i is the position in the query. Each iteration p estimates a new $\lambda_i^{(p+1)}$ by first doing the E-step and then the M-step until the value of the weight does not change significantly anymore. We added a little constant, equivalent to three documents, to the M-step, because a small number of relevant documents should not radically change the initial weights.

Initially, when no information on relevant documents is available, each term in the profile will get the same importance weight $\lambda_i = 0.5$. So, initially we assume that the profile is best explained if on average half of the profile terms is sampled from relevant documents and the other half is sampled from the general model. If a relevant document is available, it might be possible to explain the profile better. The EM-algorithm for re-estimation of importance weights λ_i will make sure that terms that occur often in the relevant documents that are selected so far, get a high importance weight λ_i. Profile terms that do not occur (often) in the relevant documents are more likely to be sampled from the background model and get a low importance weight λ_i.

Six strategies were tried on the TREC-8 Adaptive Filtering tasks, three optimized for LF_1 and three optimized for LF_2

$$\begin{array}{lcll} \mathrm{LF}_1 & = & 3\,r \;-\; 2\,(n-r) & r : \text{number of relevant documents selected} \\ \mathrm{LF}_2 & = & 3\,r \;-\; (n-r) & n : \text{number of documents selected} \\ \mathrm{LF}_3 & = & 2\,r \;-\; (n-r) & R : \text{total number of relevant documents} \end{array}$$

The utility measures LF_1, LF_2 and LF_3 assign a value or cost to each document, based on whether it is relevant or not. The measures do not use the total number of relevant documents R, representing the fact that users are not especially interested in recall, as long as they do

not get too much irrelevant items. The first measure represents a user for which a relevant selected document has a value of 3, and a non-relevant selected document has a cost of 2. This user needs to see at least 2 relevant documents in each 5 selected. So, the system should select the document if its probability of relevance is greater than $2/5 = 0.4$. The second measure represents a user whose costs of reading a non-relevant document are twice as low. Two versions of the prototype system will be tested, one optimized for LF_1 and one optimized for LF_2. The systems are evaluated by the measures for which they are optimized. The higher the utility score of a system for a user profile, the better the system is performing. (The LF_3 measure was used in TREC-9 and 10). For both utility functions the same three experiments were done.

1. A baseline run that only uses the initial threshold setting and threshold adaptation routines;
2. the same run as 1, but with relevance weighting of profile terms;
3. the same run as 1, but using a very high initial threshold.

The high initial threshold experiments were done to check whether a very conservative threshold algorithm could possibly be more beneficial than a query reweighting technique. The threshold adaption algorithm is described by Kraaij et al. (2000).

run	LF_1	LF_2	precision	recall
LF_1 optimized	-9.30	4.86	0.242	0.240
LF_1 optimized; profile reweighting	-7.28	7.10	0.243	0.251
LF_1 optimized; high initial threshold	-1.20	2.46	0.216	0.105
LF_2 optimized	-12.96	4.80	0.232	0.254
LF_2 optimized; profile reweighting	-9.12	6.60	0.237	0.254
LF_2 optimized; high initial threshold	-5.54	1.34	0.199	0.127

Table 7: Adaptive Filtering results averaged over topics

Table 7 lists the evaluation results of the runs using four evaluation measures: LF_1, LF_2, precision and recall averaged over topics. The utility scores reported are averaged over the 50 test profiles. Precision and recall were averaged over the profiles by assigning 0 % recall to topics with no relevant documents and assigning 0 % precision to topics with empty retrieved sets.

Both baseline runs show a consistent improvement in the average utility, average precision and average recall when applying the relevance feedback algorithm. Interestingly, relevance feedback has a different impact on the two systems. It causes improved recall for the LF_1 system and improved precision for the LF_2 system: The LF_1 system selected 5 % more documents after query reweighting, but the LF_2 system selected 8 % fewer documents. Note however that it is better to ignore the LF_1 altogether, because it did not beat the base line of not selecting any document at all (which would result in zero utility).

6 Conclusions and future work

In this chapter we approached the TREC Ad Hoc, Entry Page, Cross-language and Adaptive Filtering tasks by using language models for information retrieval. Each TREC task illustrates

a different aspect of the language modeling approach. The Ad Hoc task illustrates the need for a basic retrieval model; the Entry Page tasks illustrates the possibility to integrate non-content information with the basic model; the Cross-language task illustrates the use of structured queries; and finally, the Adaptive Filtering task illustrates the possibility to optimize the basic model using relevance feedback.

Looking back at more than 40 years of probabilistic modeling for information retrieval, it is interesting to see that many of the ideas that we presented in this paper under the term "language models" have been out there for at least 25 years now. Maron and Kuhns (1960) presented the basis for the models presented in this chapter: Adding a linear combination of two probability models, as in Harter's 2-Poisson model, is enough to make their model work. Robertson's probability of relevance estimation might be seen as the basis for estimating document priors.

But maybe there are different ways to give the language modeling approach a place in information retrieval history. There are still major challenges for the language modeling approach to information retrieval: For instance, how to include document structure, like author name, title, year, etc. into the model; or how to model multiple relevant documents generating one query. The evaluation of new models and ideas will be of the utmost importance, and evaluation conferences like TREC are invaluable for the progress of the field.

Acknowledgements

Our first participations in TREC were partly funded by the EU project "Twenty-One" on cross-language information retrieval (`http://twentyone.tpd.tno.nl`). Later participations work were funded by the DRUID project (`http://dis.tpd.tno.nl/druid`). Many people helped us during the TREC experiments that form the basis of this paper, we especially want to thank Rudie Ekkelenkamp, Renée Pohlmann and Thijs Westerveld. Thanks also to Richard Schwartz of BBN Technologies for helpful comments and discussions about generative language models. Furthermore we would like to thank Michel Simard (RALI, Université de Montréal) for helping with the construction of aligned corpora and building translation models. We also thank George Foster and Jian-Yun Nie (also RALI) for general discussions about the application of statistical translation models to cross-language retrieval.

References

Berger, A. and J. Lafferty (1999). Information retrieval as statistical translation. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pp. 222–229. ACM Press.

Bookstein, A. and D. Swanson (1974). Probabilistic models for automatic indexing. *Journal of the American Society for Information Science 25*(5), 313–318.

Brin, S. and L. Page (1998). The anatomy or a large-scale hypertextual Web search engine. *Computers and ISDN Systems 30*.

Dempster, A., N. Laird, and D. Rubin (1977). Maximum likelihood from incomplete data via the em-algorithm plus discussions on the paper. *Journal of the Royal Statistical Society 39*(B), 1–38.

Dougherty, J., R. Kohavi, and M. Sahami (1995). Supervised and Unsupervised Discretization of Continuous features. In *Proceedings of the twelfth International Conference on Machine Learning*, pp. 194–202. Morgan Kaufmann.

Duda, R. and P. Hart (1973). *Pattern classification and scene analysis*. Wiley-interscience.

Federico, M. and N. Bertoldi (2002). Statistical cross-language information retrieval using n-best query translations. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pp. 167–174. ACM Press.

Harman, D. (2005). TREC Test Collections. In (Voorhees and Harman 2005).

Harman, D. (2005b). The Ad Hoc Task. In (Voorhees and Harman 2005).

Harman, D., M. Braschler and P. Sheridan (2005). Cross-language Retrieval. In (Voorhees and Harman 2005).

Harter, S. (1975). An algorithm for probabilistic indexing. *Journal of the American Society for Information Science 26*(4), 280–289.

Hawking, D. (2005). Web Retrieval. In (Voorhees and Harman 2005).

Hiemstra, D. (2002). Term-specific smoothing for the language modeling approach to information retrieval: The importance of a query term. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pp. 35–41. ACM Press.

Hiemstra, D. and W. Kraaij (1999). Twenty-One at TREC-7: Ad Hoc and Cross-language track. In *Proceedings of the seventh Text Retrieval Conference (TREC-7)*, pp. 227–238. NIST Special Publication 500-242.

Hull, D. (1997). Using structured queries for disambiguation in cross-language information retrieval. In *AAAI Symposium on Cross-Language Text and Speech Retrieval*. American Association for Artificial Intelligence, pp. 84–98. (http://raven.umd.edu/dlrg/filter/sss/papers/)

Jin, H., R. Schwartz, S. Sista, and F. Walls (1999). Topic tracking for radio, tv broadcast, and newswire. In *Proceedings of the 2nd Topic Detection and Tracking Workshop (TDT-2)*. (http://www.nist.gov/speech/publications/darpa99/)

Kraaij, W., J.Y. Nie and M. Simard (2003). Embedding Web-based Statistical Translation Models in Cross-Language Information Retrieval. *Computational Linguistics 29*(3).

Kraaij, W., R. Pohlmann, and D. Hiemstra (2000). Twenty-One at TREC-8: using language technology for information retrieval. In *Proceedings of the eighth Text Retrieval Conference (TREC-8)*, pp. 285–300. NIST Special Publication 500-246.

Kraaij, W., T. Westerveld, and D. Hiemstra (2002). The importance of prior probabilities for entry page search. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pp. 27–34. ACM Press.

Lafferty, J. and C. Zhai (2003). Probabilistic relevance models based on document and query generation. In *Language Modeling for Information Retrieval*, pp. 1–10. Kluwer Academic Publishers.

Lavrenko, V., M. Choquette, and W. Croft (2002). Cross-lingual relevance models. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pp. 175–182. ACM Press.

Lavrenko, V. and W. Croft (2001). Relevance-based language models. In *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01)*, pp. 120–128. ACM Press.

Manning, C. and H. Schütze (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Maron, M. and J. Kuhns (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the Association for Computing Machinery 7*, 216–244.

Miller, D., T. Leek, and R. Schwartz (1999). A hidden Markov model information retrieval system. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pp. 214–221. ACM Press.

Ng, K. (2000). A maximum likelihood ratio information retrieval model. In *Proceedings of the eighth Text Retrieval Conference (TREC-8)*, pp. 483–492. NIST Special Publication 500-249.

Nie, J.Y., M. Simard, P. Isabelle, and R. Durand (1999). Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pp. 74–81. ACM Press.

Pirkola, A. (1998). The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pp. 55–63. ACM Press.

Pirkola, A., H. Keskustalo, and K. Järvelin (1999). The effects of conjunction, facet structure and dictionary combinations in concept-based cross-language retrieval. *Information Retrieval 1*(3), 217–250.

Ponte, J. (2000). Language models for relevance feedback. In W. Croft (Ed.), *Advances in Information Retrieval*, pp. 73–95. Kluwer.

Ponte, J. and W. Croft (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pp. 275–281. ACM Press.

Rabiner, L. (1990). A tutorial on hidden Markov models and selected applications in speech recognition. In A. Waibel and K. Lee (Eds.), *Readings in speech recognition*, pp. 267–296. Morgan Kaufmann.

Rijsbergen, C.J., van (1986). A non-classical logic for information retrieval. *The Computer Journal 29*(6), 481–485.

Robertson, S.E. and D. Hull (2005). Filtering and Routing. In (Voorhees and Harman 2005).

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

Robertson, S. and S. Walker (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th ACM Conference on Research and Development in Information Retrieval (SIGIR'94)*, pp. 232–241. ACM Press.

Singhal, A., G. Salton, M. Mitra, and C. Buckley (1995). Document length normalization. Technical Report TR95-1529, Cornell University.

Sparck-Jones, K., S. Robertson, D. Hiemstra, and H. Zaragoza (2003). Language modeling and relevance. In *Language Modeling for Information Retrieval*, pp. 57–72. Kluwer Academic Publishers.

Spitters, M. and W. Kraaij (2000). A language modeling approach to topic tracking. In *Proceedings of the third Topic Detection and Tracking Workshop (TDT-3).* (http://www.nist.gov/speech/tests/tdt/tdt2000/Papers-n-slides/)

Turtle, H. and W. Croft (1992). A comparison of text retrieval models. *The Computer Journal 35*(3), 279–290.

Voorhees, E.M. and D. Harman. (2005). (editors) TREC Test Collections. *TREC: Experimentation and Evaluation in Information Retrieval*. MIT Press.

Xu, J., R. Weischedel, and C. Nguyen (2001). Evaluating a probabilistic model for cross-lingual information retrieval. In *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01)*, pp. 105–110. ACM Press.

Zhai, C. and J. Lafferty (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01)*, pp. 334-342. ACM Press.