Parsimonious Language Models for Information Retrieval

Djoerd Hiemstra
University of Twente
Enschede, The Netherlands
d.hiemstra@utwente.nl

Stephen Robertson
Mircrosoft Research
Cambridge, U.K.
ser@microsoft.com

Hugo Zaragoza
Mircrosoft Research
Cambridge, U.K.
hugoz@microsoft.com

ABSTRACT

We systematically investigate a new approach to estimating the parameters of language models for information retrieval, called *parsimonious* language models. Parsimonious language models explicitly address the relation between levels of language models that are typically used for smoothing. As such, they need fewer (non-zero) parameters to describe the data. We apply parsimonious models at three stages of the retrieval process: 1) at indexing time; 2) at search time; 3) at feedback time. Experimental results show that we are able to build models that are significantly smaller than standard models, but that still perform at least as well as the standard approaches.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Theory, Experimentation

Keywords

Information Retrieval, Language Models, Parameter Estimation

1. INTRODUCTION

In the period from their application to information retrieval in 1998 [5, 14, 18] until today, statistical language models have become a major research area in information retrieval. Language models are statistical models of language generation. They assign a probability to a piece of text. For instance, "this is really great" would be assigned a higher probability than "fish miss feeling grave", because the former words (or better, if n-grams are used, word sequences) occur much more frequently in English than the latter words. Automatic speech recognizers use these probabilities to improve recognition performance. For information retrieval, we build such models for each document. By this approach, the paper you are reading now would be assigned an exceptionally high probability for the word "parsimonious" indicating that it would be a good candidate for retrieval if the query contains this word.

Looking back at the past six years, we might attribute the success of language models to two characteristics: Firstly, language models do not need a *tf.idf* (or any other) term weighting algorithm. In fact, they provide an alternative explanation of why *tf.idf* term weighting is a good idea to begin with [5]. Secondly, basic language models for retrieval are easily combined with information from other models. For instance, language models have been successfully combined with statistical translation models [1] and as such been applied to cross-language retrieval [3, 4, 29]. Another example is the combination of a basic retrieval model with models of non-content information, e.g. the use of link information and url-type in web retrieval [8].

Language models have contributed to our understanding of information retrieval, but in many formulations they fail to model information retrieval's key notion explicitly: *relevance*. Relevance has always been taken as fundamental to information retrieval (see, e.g. [15, 22]). From the standpoint of retrieval theory, the presumption has been that relevance should be explicitly recognized in any formal model of retrieval. The probabilistic model of retrieval [20] does this very clearly, but the language model account of what retrieval is about is not that clear. Ponte and Croft [18] say the following about relevance:

> ... in our approach, we have shifted our emphasis from probability of relevance to probability of query production. We assume these are correlated but do not currently attempt to model that correlation explicitly.

Assuming that relevance and query generation are correlated is not wholly satisfying because in this way information from feedback – or any other information directly related to relevance – is not easily incorporated. Ponte's approach to relevance feedback [17], and the approach described by Miller et al. [14] are rather ad-hoc, and it is unclear how to apply them to e.g. cross-language retrieval. What is needed is an approach that includes relevance explicitly.

Recently, Lavrenko and Croft [10, 11] have suggested the use of *relevance models*: Models that capture the language use in the set of relevant documents. For every word in the vocabulary, their relevance model gives the probability of

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.

observing the word if we first randomly select a document from the set of relevant documents, and then pick a random word from it (see Section 2.3 for a more formal account of this approach). A similar approach is suggested by Lafferty and Zhai [9].

word	*probability*
the	0.0776
of	0.0386
and	0.0251
to	0.0244
in	0.0203
a	0.0198
amazon	0.0114
for	0.0109
that	0.0101
forest	0.0100
⋮	
assistence	0.0009
aleene	0.0008
macminn	0.0008
⋮	

Table 1: Example relevance model for TREC ad hoc topic 400: "amazon rain forest"

Table 1 shows an example relevance model estimated from some relevant documents for TREC ad-hoc topic 400 "amazon rain forest". It is interesting to note that the most probable words in relevant documents are function words "the", "of", and "and". Lavrenko and Croft [11] present an example Chinese relevance model with similar characteristics: Its most probable Chinese tokens are tokens for punctuation, possessive suffix, and "and". What happened? It is not a surprise that words or tokens that are most common in English or Chinese will also be most common in the relevant documents. However, it is unlikely that these words will contribute much to retrieval performance. Of course, function words can easily be omitted from the model by the use of a stop list, but we believe that this only hides a fundamental problem with relevance models and language models in general.

A little bit further down Table 1 we see words like "assistence", "aleene", and "macminn". These words are usually not very common in English, but they were relatively common in one of the example relevant documents. The word "assistence" is probably a spelling error, and the words "aleene", and "macminn" form the name of the reporter who wrote a relevant document. Including these words in the relevance model is a classical case of overtraining: It is unlikely that the words will contribute to retrieval performance, especially if the relevance model is trained – as is the case in the experiments described in Section 3.3 – on data from a different source than the source of the data on which the model will be used. It goes without saying that omitting such words from the model cannot be done with a stop list.

We believe that language models for information retrieval should not blindly model language use. Instead, they should model what language use *distinguishes* a relevant document from other documents. We hypothesize that the inclusion of words that are common in general English, or words that are highly specific to only one relevant document will *a*) degrade the performance of relevance models, and *b*) make them unnecessarily large. What we need is an approach that leads to *parsimonious* language models.

In this paper we describe a practical implementation of "parsimonious language modelling" that was recently suggested by Sparck-Jones et al. [24]. A parsimonious model optimizes its ability to predict language use but minimizes the total number of parameters needed to model the data. We use standard mixture models [1, 5, 14, 23] to build models of documents and models of search topics (i.e. relevance models). Mixture models consist of two or more basic language model components. It is standard to estimate one component without taking the other(s) into account. An interesting alternative estimation approach is suggested by Jin et al. [7], Zhai and Lafferty [30] and Zhang et al. [31] They estimate language models in a layered fashion by first defining the background language model, and then defining document models or relevance models using standard expectation maximization (EM-) estimation [2] taking the fixed background and a fixed mixture parameter into account. Models will have zero probability for terms that are already well explained by other components of the mixture, leading to models with fewer non-zero parameters. We call such models *parsimonious* models. EM-training of information retrieval language models is also used by Hofmann [6] to model documents by linear combinations of a small number of so-called aspect models. Hofmann's model is similar in nature to the image retrieval model used by Vasconcelos [26], which uses a mixture of a small number of Gaussians. Westerveld and De Vries [28] recently estimated such Gaussian mixture models parsimoniously as well.

In this paper we systematically explore the use of parsimonious models at a number of stages in the retrieval process, combining the work of several relevance model approaches [11, 30, 31]. Our approach bears some resemblance with early work on information retrieval by Luhn [12] who specifies two word frequency cut-offs, an upper and a lower to exclude non-significant words. The words exceeding the upper cut-off are considered to be common and those below the lower cut-off rare, and therefore not contributing significantly to the content of the document [19]. Unlike Luhn, we do not exclude rare words at indexing time (but rare words might be excluded if example relevant documents are available). Furthermore, we do not have simple frequency cut-offs. Similar approaches have recently been applied to back-off language models for automatic speech recognition: Stolcke [25] and Sankar et al. [21] describe an approach to reduce the language model size by pruning n-grams based on the relative entropy between the original model and the pruned model.

Parsimonious language models will be used in the KL-divergence retrieval framework (or cross-entropy retrieval framework) introduced by Lafferty and Zhai [9]. We will apply our models at three stages of the retrieval process: 1) at index time; 2) at request time, and 3) at search time (taking example relevant documents into account). The paper is organized as follows. Section 2 describes the technical details of our parsimonious language models approach. Section 3 describes the application of parsimonious models at the above mentioned three stages of the retrieval process. Section 4 presents additional experimental results. Finally, Section 5 presents conclusions, and a discussion of future work.

2. LANGUAGE MODELS AND THE NEED FOR PARSIMONY

For information retrieval, a language model is defined for each document. So, for each document D, its language model defines the probability $P(t_1, t_2, \cdots, t_n|D)$ of a sequence of n query terms $t_1, \cdots, t_n$, and the documents are ranked by that probability. The standard language modeling approach to information retrieval uses a mixture of the document model $P(t_i|D)$ with a general collection model $P(t_i|C)$ [1, 5, 9, 14, 16, 23]. The approach needs a parameter λ which is set empirically on some test collection, or alternatively estimated by the EM-algorithm on a test collection [14, 16].

$$P(t_1, \cdots, t_n|D) \;=\; \prod_{i=1}^{n} \big((1-\lambda)P(t_i|C) + \lambda P(t_i|D)\big) \qquad (1)$$

An interesting justification of the mixture of the general language model $P(t_i|C)$ and the document language model $P(t_i|D)$ is the following: The language that we use in writing a paper on a specialist subject is basically the same language we use when talking to our families, but it is modified by the particular context. Usually, the language model probabilities are defined as follows, with unrelated probabilities for the same event (unrelated because one can be defined without taking the other into account).

$$P(t_i|C) \;=\; \frac{cf(t_i)}{\sum_t cf(t)} \qquad (2)$$

$$P(t_i|D{=}d) \;=\; \frac{tf(t_i, d)}{\sum_t tf(t, d)} \qquad (3)$$

Here, $cf(t_i)$ is the collection frequency of the term, i.e. the number of occurrences of t_i in the collection; $tf(t_i, d)$ is the term frequency of the term in the document, i.e. the number of occurrences of t_i in the document d. So, a lot of words that are common in general English will be assigned a high probability in the document language model $P(t_i|D)$ as well. Like the example model presented in Table 1, $P(t_i|D)$ wastes some of its probability mass on modeling words from general English. To avoid this, we need an estimation method that rewards parsimony, that is, an algorithm that concentrates the probability mass on those terms that distinguish the document from the general model.

2.1 Parsimony at index time

What would an approach to parsimonious language models look like? It has to be noted that Equation 3 is the maximum likelihood estimate for the document model, that is, of all possible ways to define the document model Equation 3 is the one that maximizes the probability of observing the document. The maximum likelihood estimate does not lose any probability mass on terms that do not occur in the document. However, it is not very practical: If one query term (of many query terms) does not occur in the document, then Equation 3 will assign zero probability to the document. If we multiply the probability for each term given the document, then the document will not be retrieved at all. To avoid this so-called *sparse data problem* [13], we use a mixture of the document model $P(t|D)$ with a background model $P(t|C)$ as in Equation 1. The background model is non-zero for all terms in the collection.

So, we *have to* smooth with a background model. Given a background model $P(t|C)$ and a mixture parameter λ, Equation 3 is however no longer the maximum likelihood estimate of the documents. The estimate of $P(t|D)$ in Equation 1 that maximizes the probability of observing the document is given by the following iterative algorithm: Apply the E-step and the M-step for each term t until the estimates $P(t|D)$ do not change significantly anymore. The resulting $P(t|D)$ will concentrate the probability mass on even fewer terms than Equation 3.[1]

$$\text{E-step:} \quad e_t \;=\; tf(t, D) \cdot \frac{\lambda P(t|D)}{(1-\lambda)P(t|C) + \lambda P(t|D)}$$

$$\text{M-step:} \quad P(t|D) \;=\; \frac{e_t}{\sum_t e_t}, \text{ i.e., normalize the model}$$

The method is unsupervized, that is, it does not use information from queries or relevance judgements. It gives an answer to the the question: "How does the language of this document differ from that of the whole collection?" For $\lambda = 1$, the algorithm will produce the maximum likelihood estimate defined by Equation 3. Other values of λ will produce document models $P(t|D)$ for which some of the terms that actually occur in the document will be assigned zero probability. These terms are better explained by the general language model $P(t|C)$, that is, terms that occur about as frequent in the document as in the whole collection. The estimation method deals automatically with stopwords, but it should also remove words that are mentioned occasionally, for instance the word "thank" if a document contains acknowledgements but is not 'about thanking'.

2.2 Parsimony at request time

If we have a natural language statement of the user's information need, then we would expect such a statement – the request – to use some words of general English as well. By analogy of the previous case, we assume a request is generated from a mixture of a general model and a specific topic model. We can use the EM-algorithm above (replace document D by request R) to estimate a request model (or *relevance* model) $P(t|R)$ that describes the language use that distinguishes the request from general English.

How to apply the relevance model at search time? Lafferty and Zhai [9] suggest the use of Kullback-Leibler divergence between the relevance model and the document model as a possible score to rank the documents. This is equivalent with taking the cross-entropy as follows [11]:

$$H(R, D) \;=\; -\sum_t P(t|R) \cdot \log\big((1-\lambda)P(t|C) + \lambda P(t|D)\big) \quad (4)$$

The cross-entropy is small if the relevance model and the document model have similar probability distributions, so the ranking should be by increasing value of H. It is interesting to note that a maximum likelihood estimate of $P(t|R)$ (taking the probability of a term given the request as the frequency of occurrence in the request divided by the length of the request), will result in exactly the same ranking as the standard language modelling approach. The parsimonious estimate of $P(t|R)$ has fewer non-zero parameters than the maximum likelihood estimate, and therefore fewer terms that contribute to the sum of Equation 4.

[1]The document model can be estimated analytically as well, with complexity $m \log m$, where m is the number of unique terms in the document [32].

2.3 Parsimony at feedback time

Having a two-level mixture model, covering a whole-collection model and a single local model, is not enough when we have a set of relevant documents rather than a single request. Although we are considering documents that are all relevant to the same one request, this cannot be taken to imply, unrealistically, that these documents are identical, or are generated by a single model. We have to allow for the fact that requests as topic specifiers can be selective on the normally much richer topic content of documents, especially when the necessary variation in topic granularity is factored in. Even if the presumption is that (the whole of) a (highly) relevant document is *about* the request topic, this does not imply that the document only talks about the topic.

To model the fact that a relevant document might talk about other topics than the topic that makes it relevant, we actually need not a two-level but a three-level structure: 1) a whole-collection or generic model $P(t|C)$; 2) modified by a relevance model $P(t|R)$; 3) modified by an individual document model $P(t|D)$ [31]. The document model $P(t|D)$ includes the special aspects of a particular document that are specific to topics other than the topic of this particular user request, and that cannot be explained by the generic model. Given a relevant document D belonging to a set of relevant documents R, we assume that text $t_1, \cdots, t_l$ from this document is generated from the following mixture model.

$$P(t_1, \cdots, t_l|D) = \prod_{i=1}^{n} \big((1-\lambda-\mu)P(t_i|C) + \mu P(t_i|R) + \lambda P(t_i|D)\big) \quad (5)$$

If we have some example relevant documents, then we can train a parsimonious relevance model given a fixed background model $P(t|C)$, and two fixed mixture parameters λ and μ. Apply iteratively first the E-step for each term t in each relevant document D, and then the M-step until the estimates $P(t|R)$ do not change significantly anymore.

E-step:

$$r_{t,D} = \frac{tf(t,D) \cdot \mu P(t|R)}{(1-\lambda-\mu)P(t|C) + \mu P(t|R) + \lambda P(t|D)}$$

$$e_{t,D} = \frac{tf(t,D) \cdot \lambda P(t|D)}{(1-\lambda-\mu)P(t|C) + \mu P(t|R) + \lambda P(t|D)}$$

M-step:

$$P(t|R) = \sum_{D \in R} \frac{r_{t,D}}{\sum_t r_{t,D}}$$

$$P(t|D) = \frac{e_{t,D}}{\sum_t e_{t,D}}$$

The resulting $P(t|R)$ can them be used to rank unseen documents using the cross-entropy measure of Equation 4. The maximisation step is chosen such that a fixed value of $\mu = 1$ will result in a relevance model that is equal to Lavrenko's relevance model estimation [11], which is defined by:

$$P(t|R) = \sum_{D \in R} P(t|D)P(D|R), \text{ where } P(D|R) = \frac{1}{|R|} \quad (6)$$

The model assumes that once we pick a relevant document D, the probability of observing a word is independent of the set of relevant documents R. If we however assume independence between relevant documents as done by Zhai and Lafferty [30] then we end up with the following, alternative maximisation step for the relevance model.

M'-step:

$$P(t|R) = \frac{\sum_{D \in R} r_{t,D}}{\sum_{D \in R} \sum_t r_{t,D}} \quad (7)$$

If we additionally set $\lambda = 0$ then the algorithm results in their relevance model estimation algorithm.

2.4 Model Summary

We have identified mixture models that can be used at three stages in the retrieval process to infer parsimonious language models. Table 2 shows the models used in these three stages and the relation between them [24]. In each case in this table the base model is externally defined (independent of the object under consideration). The residual is whatever is needed to 'explain' the object in addition to the base model, leading to the full model in the right-hand column. If parsimony is applied, the residual is minimised: as much as possible is explained by the base model.

We assume that each document is generated by a collection or general language model GM, together with some unknown number of special topic models, which at indexing time we roll into a single residual model, DM(GM). The relevance hypothesis can now be expressed as follows. A request is generated from the general language model plus a specific topic (= relevance) model; the latter is RM(GM). If and only if a document is relevant to the request, this same RM will explain part of the document model. Thus for these documents we have a new residual (model for the unexplained parts), DM(RM,GM), and a new full model as shown in the table. The full relevant document models at 3a and 3b are the same, but the fitting procedure is different (see below).

We have assumed here that the GM applicable to the requests generally is the same one that applies to the documents, and which we would normally discover from the collection of documents rather than from any collection of requests. It is at least arguable that the general language model for requests should be different, possibly (depending on the user context) with no or different stopwords for example. This idea is not pursued in the present paper.

Now the procedure, in outline, is as follows. At indexing time we fit the level 1 GM to the collection and a level 2 DM(GM) to each document. In initial query formulation, we fit the RM(GM) to the only information that we have about the request, namely its original text. In [24], it was assumed that the search process would involve trying out the level 3 model on every document (shown as 3a in Table 2), against a null hypothesis which would be the level 2 model derived at indexing time. This comparison would again have to appeal to parsimony. However, in the present paper, this step is replaced by a comparison between RM(GM) and DM(GM), as suggested by Lavrenko and Croft [11].

At the feedback stage, we would like to re-estimate RM making use of the known relevant documents as well as the request. That is, we need to take just GM as given, for both the request and the relevant documents, and then simultaneously estimate as best we can both RM and the residual DMs – one for each document (This is shown as 3b in Table 2). The (revised) RM will form the basis for subsequent search; the DMs will be discarded, but must be there for the reason given above, that we do not want to assume that the RM consumes all of the topic-specific aspects of each document. So the question is, given the GM, what is the

Table 2: Levels of language models for document retrieval

	Object	Base model	Residual	Full model
1	Collection	–	GM	GM
2	Document, index time	GM	DM(GM)	GM+DM(GM)
	Request, query formulation time	GM	RM(GM)	GM+RM(GM)
3a	Relevant document, search time	GM+RM(GM)	DM(RM,GM)	GM+RM(GM)+DM(RM,GM)
3b	Relevant document, feedback time	GM	RM(GM)+DM(RM,GM)	GM+RM(GM)+DM(RM,GM)

best combination of RM and individual DMs to explain the request *and* each of the relevant documents? This notion of 'best' must again appeal to parsimony: that is, anything that the relevant documents have in common is best explained globally in the RM rather than locally in the DMs. (It is not clear that such a method makes sense if we have only one example of a relevant document.)

3. PARSIMONIOUS MODELS IN ACTION

In this section we describe a first test of our parsimonious estimation method on the TREC-7 ad hoc document test-collection. The collection consists of news data from the Los Angeles Times, the Financial Times, the Foreign Broadcast Information Service, and Federal Register covering the years 1989 to 1994. The experiments were set up as regular TREC tasks [27]. Sections 3.1 and 3.2 describe the experimental results of parsimonious document models and parsimonious request models on the TREC ad hoc task. Section 3.3 describes the experimental results of relevance models on the TREC routing task.

3.1 Parsimony at index time

To test the parsimonious models approach we build several indexes which were trained using different values of λ as described in Section 2.1. On each of these indexes we did searches for 50 short queries from TREC topics 351–400 (title-only, on average 2.5 query terms). We did not use a stop list; nor did we use a stemmer. Terms that are assigned a probability of less than 0.0001 after training were removed from the model, i.e., were put to zero.

Figure 1a shows how the number of postings in our index decreases with a decreasing value of λ. One posting corresponds to one term–document pair in our index. Figure 1b plots the retrieval performance in terms of mean average precision [27] for each value of λ of the parsimonious model, and for each value of λ on the standard language model (which uses all postings). The standard model degrades quickly if the value of λ decreases. However, the parsimonious model does not break down nearly as fast.

Optimum retrieval performance of the parsimonious model (0.181 average precision) is achieved at $\lambda = 0.1$. The index size consists of 86.8 million postings for this value, about 79 % of the number of postings used by the standard language model. Optimum performance of the standard language model (0.176 average precision) is achieved at $\lambda = 0.2$.

The parsimonious model achieves 0.124 average precision for $\lambda = 0.0002$ (69 % of the optimum), for which it only needs 14.8 million postings (7.5 % of the standard index size). This should indeed be attributed to the parsimonious

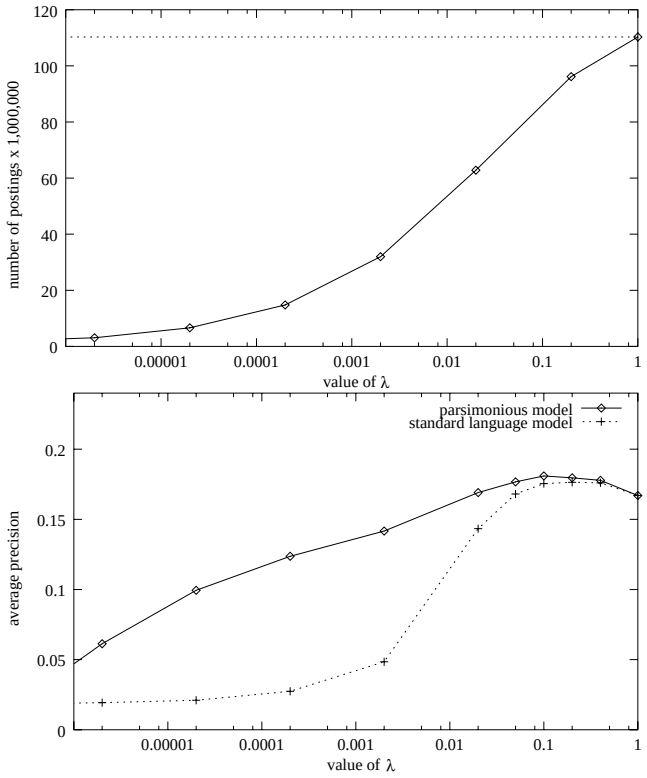

Figure 1: Document models: λ vs. a) number of postings b) average precision

estimation method, because the standard model on the full index has 0.027 average precision for the same value of λ (15 % of the optimum): It is possible to use a fraction of the index size and still get reasonable retrieval performance. Interestingly, the optimum performance of a standard language model is 0.177 (also at $\lambda = 0.2$) if we use a standard stoplist taken from [19]. The index size using the stoplist is 85.4 million postings, about the same size as the best performing parsimonious model.

3.2 Parsimony at request time

To test the performance of parsimonious model estimation at request time, we took the full TREC topic descriptions which contain about 38 words on average. We trained a relevance model for a number of different values of λ. Figure 2a shows the average number of query terms that has non-zero

probability in the relevance model for each value of λ. Figure 2b shows the average precision of using the relevance model on the collection. We used the standard document models with $\lambda = 0.2$.

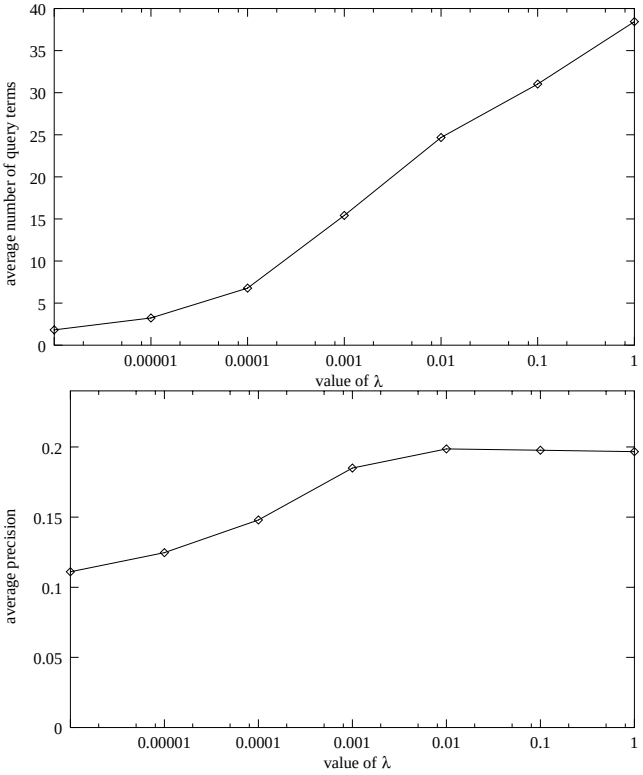

Figure 2: Query models: λ vs. 1) average number of query terms, 2) average precision

Changing the value of λ enables us to define relevance models of any size. The algorithm ensures that there is always at least 1 term left: The most distinguishing term. Sometimes, the algorithm decides that 2 or 3 terms are kept. If we would extend the graph to the left, the average number of query terms would never get below about 1.4 terms. Optimum performance of the model lies at $\lambda = 0.01$. For this value, the relevance model contains about 24 query terms on average, and the retrieval performance is 0.199 average precision.

3.3 Parsimony at feedback time

The performance of the three-level model described in Section 2.3 is tested in a routing experiment. The TREC routing task has a training phase and a testing phase. The Los Angeles Times data, which dates from 1989 and 1990, is used in the training phase. For each TREC topic, the relevant documents from the Los Angeles Times are used to build a relevance model. The relevance model is subsequently tested on the other data from the TREC collection, which dates from 1991 until 1994. The routing task measures the performance of systems with a relatively stable query and incoming documents. The relevant documents that are known at some point in time can be used to improve the routing query for subsequent documents.

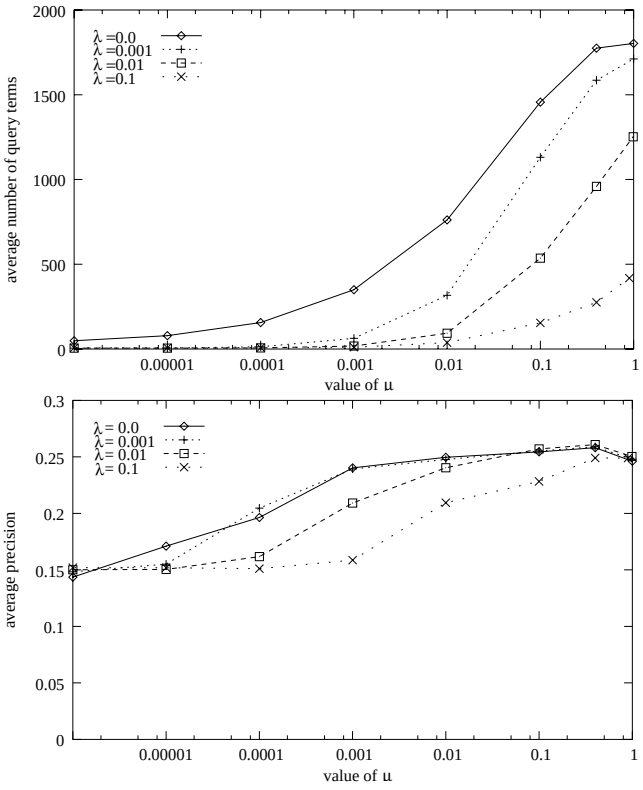

Figure 3: Relevance models: μ vs. 1) average number of query terms, 2) average precision

Figure 3a shows the average size of the relevance model for values of λ and μ. Tthe maximum size is reached when $\mu = 1$ (note that this implies that $\lambda = 0$). In this case, the algorithm results in Lavrenko's relevance model estimation [11] (see Equation 6).

The average precision graphs in Figure 3b show that performance of the full relevance model ($\mu = 1$) can be slightly improved by lowering the value of μ. The optimum lies around $\mu = 0.4$. Precision does not improve further when we introduce the single document models ($\lambda > 0$), although we are able to decrease the size of the relevance models without losing any precision for small values of λ, i.e. for $\lambda = 0.01$ and $\lambda = 0.001$.

4. ADDITIONAL EXPERIMENTAL RESULTS

Based on the the experiments on the TREC-7 collection described above, we used the optimal parameters on the topics 401–450 of the TREC-8 collection in two additional experiments. In the first experiment we compare parsimonious language models with standard models on the ad hoc search task. In the second experiment, we compare three relevance model estimation methods on the routing task.

Figure 4 shows recall-precision graphs of two language modelling approaches using the full topic description of TREC topics 401–450. The standard approach uses the full description (35 unique terms on average) on the full document index. The parsimonious approach uses the parsimonious

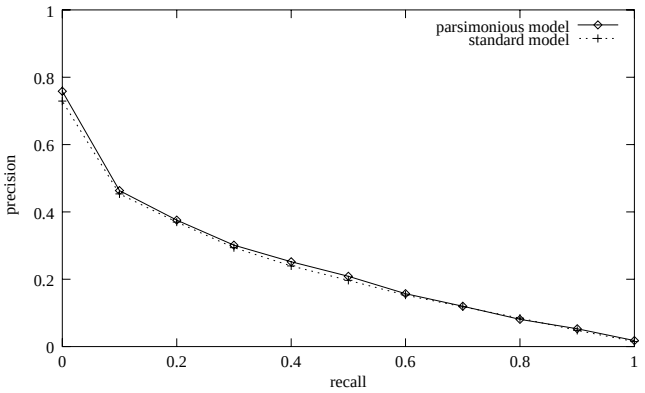

Figure 4: Recall-precision graphs of language models on the ad hoc task

request model (22 unique terms on average) on the parsimonious document index. The parsimonious model outperforms the standard model on all recall points, but the differences are very small (0.230 vs. 0.223 average precision) and both systems behave similarly. The difference in average precision is however significant at the 1 % level following the two-tailed pair-wise sign test (see e.g. [19]).

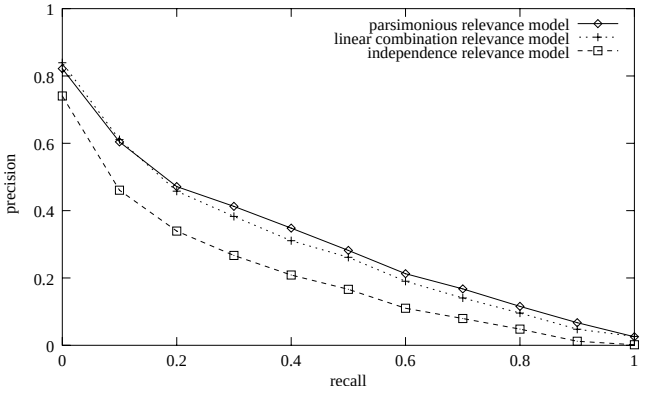

Figure 5: Recall precision graphs of relevance models on the routing task

Figure 5 shows recall-precision graphs of three relevance model approaches on the routing task described in Section 3.3, but now using TREC topics 401–450. The first method uses the parsimonious relevance model with $\lambda = 0.01$ and $\mu = 0.4$. The second method uses the same algorithm with $\lambda = 0$ and $\mu = 1$, which boils down to the method by Lavrenko [11] (which we called 'linear-combination relevance model'). The third method uses the alternative M-step of Equation 7 with $\lambda = 0$ and $\mu = 1$ which we called 'independence relevance model', i.e. the method by Zhai and Lafferty [30].

The independence relevance model performs significantly worse than the other two methods, which are not significantly different. However, the size of the parsimonious relevance models (959 terms on average) is significantly smaller than the size of the linear-combination relevance models (1803 terms on average).

5. CONCLUSION AND FUTURE WORK

We have systematically investigated parsimonious language modelling estimation at three stages of the retrieval process: at indexing time, at request time, and at feedback time. We showed that the approach enables system developers to build models of any size. We also showed that we are able to build models that are significantly smaller than standard models, and still perform as well as, or better than, standard approaches. A smaller model means that less storage overhead and less CPU time is needed. Future work should explore if the right level of parsimony can be determined automatically as suggested in [24].

Acknowledgements

We would like to thank Karen Sparck-Jones of the University of Cambridge and Richard Schwartz of BBN Technologies for helpful remarks and suggestions.

6. REFERENCES

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

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

[3] M. Federico and N. Bertoldi. 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)*, pages 167 – 174, 2002.

[4] D. Hiemstra and F.M.G. de Jong. Disambiguation strategies for cross-language information retrieval. In *Proceedings of the third European Conference on Research and Advanced Technology for Digital Libraries (ECDL'99)*, pages 274–293, 1999.

[5] D. Hiemstra and W. Kraaij. Twenty-One at TREC-7: Ad-hoc and cross-language track. In *Proceedings of the seventh Text Retrieval Conference (TREC-7)*, pages 227–238. NIST Special Publication 500-242, 1998.

[6] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 50–57, 1999.

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

[8] W. Kraaij, T. Westerveld, and D. Hiemstra. 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)*, pages 27–24, 2002.

[9] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization. In *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01)*, pages 111–119, 2001.

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

[11] V. Lavrenko and W.B. Croft. Relevance models in information retrieval. In W.B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 11–56. Kluwer Academic Publishers, 2003.

[12] H.P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165, 1958.

[13] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

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

[15] S. Mizzaro. Relevance: The whole story. *Journal of the American Society for Information Science*, 48(9):810–832, 1997.

[16] K. Ng. A maximum likelihood ratio information retrieval model. In *Proceedings of the eighth Text Retrieval Conference (TREC-8)*. NIST Special Publication 500-246, pages 483–492, 1999.

[17] J.M. Ponte. Language models for relevance feedback. In W.B. Croft, editor, *Advances in information retrieval : recent research from the Center for Intelligent Information Retrieval*, pages 73–95. Dordrecht: Kluwer, 2000.

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

[19] C.J. van Rijsbergen. *Information Retrieval, second edition*. Butterworths, 1979. (http://www.dcs.gla.ac.uk/Keith/Preface.html)

[20] S.E. Robertson and K. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[21] A. Sankar, V.R.R. Gadde, A. Stolcke, and F. Weng. Improved modeling and efficiency for automatic transcription of broadcast news. *Speech Communication*, 37:133–158, 2002.

[22] T. Saracevic. Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science*, 26:321–343, 1975.

[23] F. Song and W.B. Croft. A general language model for information retrieval. In *Proceedings of the 8th International Conference on Information and Knowledge Management (CIKM'99)*, pages 316–321, 1999.

[24] K. Sparck-Jones, S.E. Robertson, D. Hiemstra, and H. Zaragoza. Language modelling and relevance. In W.B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 57–71. Kluwer Academic Publishers, 2003.

[25] A. Stolcke. Entropy-based pruning of back-off language models. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pages 270–274, 1998.

[26] N. Vasconcelos. Bayesian Models for Visual Information Retrieval. Ph.D. thesis, *Massachusetts Institut of Technology*, 2000.

[27] E.M. Voorhees. Overview of TREC 2002. In *Proceedings of the 11th Text Retrieval Conference (TREC 2002)*, NIST Special Publication 500-251, pages 1–15, 2002.

[28] T. Westerveld and A.P. de Vries. Multimedia Retrieval using Multiple Examples. In *International Conference on Image and Video Retrieval (CIVR'04)*, 2004.

[29] J. Xu, R. Weischedel, and C. Nguyen. 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)*, pages 105–110, 2001.

[30] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th International Conference on Information and Knowledge Managemen (CIKM'01)*, pages 403–410, 2001.

[31] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pages 81–88, 2002.

[32] Y. Zhang, W. Xu, and J. Callan. Exact maximum likelihood estimation for word mixtures. In *Text Learning Workshop in International Conference on Machine Learning (ICML'02)*, 2002.