A Comparative Study of Probabilistic and Language Models for Information Retrieval

Graham Bennett **Falk Scholer** **Alexandra Uitdenbogerd**

School of Computer Science and Information Technology
RMIT University, GPO Box 2476V, Melbourne 301, Australia
Email: {gbennett,fscholer,alu}@cs.rmit.edu.au

Abstract

Language models for information retrieval have received much attention in recent years, with many claims being made about their performance. However, previous studies evaluating the language modelling approach for information retrieval used different query sets and heterogeneous collections, which make reported results difficult to compare. This research is a broad-based study that evaluates language models against a variety of search tasks — topic finding, named-page finding and topic distillation. The standard Text REtrieval Conference (TREC) methodology is used to compare language models to the probabilistic Okapi BM25 system. Using consistent parameter choices, we compare results of different language models on three different search tasks, multiple query sets and three different text collections. For *ad hoc* retrieval, the Dirichlet smoothing method was found to be significantly better than Okapi BM25, but for named-page finding Okapi BM25 was more effective than the language modelling methods. Optimal smoothing parameters for each method were found to be dependent on the collection and the query set. For longer queries, the language modelling approaches required more aggressive smoothing but they were found to be more effective than with shorter queries. The choice of smoothing method was also found to have a significant effect on the performance of language models for information retrieval.

Keywords: Information retrieval, language models, probabilistic models, smoothing.

1 Introduction

Searching for documents via textual queries has, thanks to the Internet boom, become a common practice for a large proportion of society. However, not all searches have the same goals. In some cases, the user wants to find out about a topic (topic-finding or *ad hoc* retrieval); in others they have a specific web page or document in mind (named-page finding). These different types of task are likely to require different retrieval techniques in order to produce the best answers for the user.

Search engines are a type of information retrieval (IR) system. To retrieve a ranked, or sorted, list of documents in response to the user's search request, an IR system must use evidence of similarity between the query and each document. A range of different models have been proposed for text retrieval. The most widely used include vector-space models [Salton, 1971], probabilistic models [Sparck Jones et al., 2000], and more recently language models [Ponte and Croft, 1998].

Since language models became popular for use in information retrieval in the late 90s, many variant models have been proposed. However, reported evaluations of the language modelling approach for *ad hoc* search tasks use different query sets and collections. Thus, even when standard collections such as those from the TREC conferences are used, the reported results are difficult to compare. In particular, it is not clear whether language models can offer consistent performance gains over other state-of-the-art retrieval approaches, across a range of queries, collections, and search tasks.

Our research investigates whether language modelling can offer significant performance improvements over a strong probabilistic baseline method across a range of search tasks. We investigate the following research questions:

1. Are differences in retrieval performance between language models and probabilistic models repeatable for different topic sets and document collections?
2. Does the choice of smoothing method have a significant effect on the performance of language models for *ad hoc* retrieval?
3. Does the type of search task (topic finding, named-page finding, and topic distillation) influence the relative performance of language modelling approaches compared to other models?
4. Does query length influence the relative performance between different types of retrieval models?

The remainder of this paper is structured as follows: in Section 2 we present background material on information retrieval systems, and provide details of the probabilistic and language modelling approaches that we evaluate in this work. Our experimental framework, including topics, collections, and search tasks, is described in Section 3. The results from our experiments are presented in Section 4, and conclusions are discussed in Section 5.

2 Text Information Retrieval Models

A range of different models have been proposed in the information retrieval literature, based upon different notions of what it means for a document to be relevant to a query. While some models, such as the Boolean model, have been important historically, the most common form of *ad hoc* information retrieval today is ranked retrieval. In this scenario,

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the Nineteenth Australasian Database Conference (ADC2008), Wollongong, Australia, January 2008. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 75, Alan Fekete and Xuemin Lin, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

used for example in Web search, a query is treated as an unordered set of keywords (also known as a "bag of words" query). Using statistics about how terms are distributed in documents and across the collection as a whole, the IR system calculates a similarity measure between the query and each document, and returns a list of documents ordered by decreasing similarity score to the user. Different models calculate the similarity between queries and documents in different ways. In this work we constrain our attention to two highly popular and successful families of models: probabilistic and language models.

2.1 Probabilistic Models

Probabilistic models of information retrieval aim to evaluate, for each query-document pair, the probability that the document is relevant to the query. While a range of alternative probabilistic models have been proposed for information retrieval, we focus our attention on the highly successful Okapi similarity function. This model is based on a binary independence model developed by Sparck Jones and Robertson [1976], with extensions based on a 2-Poisson model to capture term frequencies [Robertson and Walker, 1994]. Due to space constraints we omit a full derivation of the Okapi model; the reader is referred to Sparck Jones et al. [2000] for a thorough presentation. The Okapi BM25 similarity function is:

$$BM25 = \sum_{t \in Q} \log \frac{(N - n_t)}{n_t} \cdot \frac{(k_1 + 1) f_{d,t}}{K + f_{d,t}} \cdot \frac{(k_3 + 1) f_{q,t}}{k_3 + f_{q,t}} \tag{1}$$

where: Q is a query; N is the number of documents in the collection; n_t is the total number of documents that contain term t; $f_{d,t}$ is the number of occurrences of term t in document d (that is, the term frequency of term t in the current document); $f_{q,t}$ is the number of occurrences of term t in the query Q (i.e. the query term frequency); $K = k_1 \cdot \left((1-b) + \frac{b \cdot dl_d}{avl}\right)$; dl_d is the number of terms in document d; avl is the average document length; and k_1, k_3, and b are tuning parameters. Broadly speaking, the first term of the BM25 function captures the inverse document frequency, while the second and third terms capture the within-document and within-query term frequencies, respectively.

Although it is possible to set the tuning parameters to values that give optimum performance on particular collections, we use a consistent set of values in our experiments. We use values recommended by Robertson and Walker [1999], which have been found to be effective in many different retrieval environments: $k_1 = 1.2$; $b = 0.75$; $k_3 = 1000$.

2.2 Language Models

Statistical language models estimate the distribution of words in an input language. In the context of information retrieval, a document is generally viewed as a sample from an underlying language model. That is, the document is only one possible version of the information that is being conveyed by the author; terms in the collection are generated with specific probabilities. Documents are ranked by the likelihood that each document language model could have generated the user's query terms.

Many variations on the language modelling approach to information retrieval have been proposed in the literature, including multiple Bernoulli models [Ponte and Croft, 1998], multinomial models [Song and Croft, 1999, Hiemstra and Kraaij, 1998], and relevance models [Lavrenko and Croft, 2001]. Despite differences in the model implementations, the underlying process can be broadly viewed as consisting of three main steps: first, a language model is estimated for each document in the collection; second, the system calculates the probability that we would observe the sequence of query terms if we sampled terms at random from each document language model; and, finally, the documents are ranked in order of these probabilities.

Under the *query-likelihood approach*, language models for IR try to estimate for each document the probability that the query Q was generated by the underlying language model, M_D. If it is assumed that terms occur independently, then the probability becomes the product of the individual query terms given the document model:

$$P(Q|M_D) = \prod_{t \in Q} P(t|M_D) \tag{2}$$

In information retrieval, it is common to use unigram models, where terms do not depend on their context. (While more sophisticated models could be expected to improve performance, work using higher order models has not been able to demonstrate consistent gains for IR, while such models are much more complex to estimate [Lavrenko, 2003].)

It therefore remains to estimate the probability of individual query terms. The document under consideration, D, is a sample from the language model, M_D. The maximum likelihood estimate of an individual query term is therefore given by:

$$\hat{P}(t|M_D) = \frac{f_{d,t}}{|D|} \tag{3}$$

where $f_{d,t}$ is the within-document frequency of term t in document d, and $|D|$ is the total number of terms in the document.

We note here that if a query term does not occur in the document, then the maximum likelihood estimate for that term is zero, giving an overall similarity score of zero for the query and the document. However, it is not sensible to rule out a document just because a single query term is missing. Therefore language models make use of *smoothing* to balance probability mass between occurrences of terms in documents, and those terms not found in the documents. We discuss the key approaches for smoothing language models in in Section 2.3.

An alternative approach to ranking documents in the language modelling framework is *model comparison*. Here, both the query and each document are modelled using a multinomial unigram language model, as above. The documents in the collection are then ranked according to divergence of the two probability distributions, measured by the Kullback-Leibler divergence of the query language model M_Q and the document language model M_D [Zhai and Lafferty, 2001b]:

$$H(M_Q||M_D) = \sum_{t \in Q} P(t|M_Q) \log \frac{P(t|M_Q)}{P(t|M_D)} \tag{4}$$

We use model comparison in our experiments below.

2.3 Smoothing

Smoothing is an important feature of language models: it balances term probabilities by discounting the probability of terms seen in the document, and adjusting low or zero probabilities upwards for other terms. This circumvents the zero frequency problem, where query terms that do not occur in a document would otherwise lead to an overall query likelihood of zero. Typically, smoothing approaches combine document term frequencies with the frequency of the term in the collection as a whole. To investigate the relative performance of language models, we compare retrieval results under four major smoothing approaches.

Jelinek-Mercer

Jelinek-Mercer smoothing [Jelinek and Mercer, 1980] combines the relative frequency of a query term in the document D with the relative frequency of the term in the collection as a whole. The maximum likelihood estimate is moved uniformly toward the collection model probability $P(t|M_C)$:

$$P(t|M_D) = (1-\lambda)\frac{f_{d,t}}{|D|} + \lambda P(t|M_C) \tag{5}$$

The value of λ is query- and collection- dependent. A value of $\lambda \approx 0.1$ is suitable for short queries, and larger values (e.g. $\lambda = 0.7$) are more suitable for longer queries [Zhai, 2006].

Dirichlet (Bayesian) Smoothing

Dirichlet smoothing makes smoothing dependent on the document size; because longer documents allow us to estimate the language model more accurately, they are more likely to require less smoothing. If we use the multinomial distribution to represent a language model, the conjugate prior of this distribution is the Dirichlet distribution [Zhai, 2002]. This gives:

$$P(t|M_D) = \frac{f_{d,t} + \mu P(t|M_C)}{|D| + \mu} \tag{6}$$

As μ gets smaller, the contribution from the collection model becomes smaller also, and more emphasis is given to the relative term weighting. According to Zhai [2002], the optimal prior value for μ is around 2,000.

Absolute Discounting

Ney et al. [1994] describe a smoothing method where all non-zero counts are discounted by subtracting a constant δ from the counts of each term. The probability mass acquired from the present terms is distributed over unseen events uniformly. Zhai [2006] formulates absolute discounting in information retrieval as:

$$P(t|M_D) = \frac{\max(f_{d,t} - \delta, 0) + \delta\,|D|_u\, p(t|M_C)}{|D|} \tag{7}$$

where $|D|_u$ is the number of unique words in the document D, and $0 < \delta < 1$.

Two-Stage Smoothing

A further type of smoothing that has been proposed for information retrieval using language models is a two-stage strategy Zhai [2002]. First, the system smoothes the document language model using the Dirichlet prior; secondly, the system mixes the document language model with a 'query background' model using Jelinek-Mercer smoothing. In this approach it is assumed that, in the absence of data to estimate the query background model, $P(t|M_C)$ is a reasonable approximation. The smoothing function is therefore:

$$P(t|M_D) = (1-\lambda)\frac{f_{d,t} + \mu P(t|M_C)}{|D| + \mu} + \lambda P(t|M_C) \tag{8}$$

where μ is the Dirichlet prior parameter, and λ is the Jelinek-Mercer parameter. Zhai and Lafferty [2004] indicate that the parameters μ and λ can be estimated automatically in the two-stage smoothing approach.

3 Experimental Setting

In this section, we describe our experiments to investigate the effectiveness of the probabilistic and language modelling approaches to *ad hoc* text retrieval.

3.1 Search Tasks

Users of information retrieval systems aim to fulfill an information need. While such needs can vary widely, they can be broadly categorised according to the user's objective and the nature of the search task [Broder, 2002]. *Informational* searches are those where the user searches the collection for documents that are topically relevant. The user endeavours to learn 'about' something — such tasks are also called *topic finding*. *Named-page finding* is a navigational search task, where the user's objective is to find a specific named resource. *Topic distillation* represents the user's requirement to view a short list of key site entry-pages, such as those that might form a Web browser bookmark file on a topic.

These types of search tasks have different objectives, and require different resources in response to a query. To investigate the relative advantages and disadvantages of retrieval models, we therefore investigate performance for each of these types of search tasks. Further details of search tasks are discussed in Section 4.

3.2 Retrieval Models

In our experiments we investigate the performance of a range of IR models: Okapi BM25, Jelinek-Mercer smoothing, Dirichlet smoothing, Absolute Discounting smoothing, and Two-Stage smoothing. The models, and associated parameter settings, are summarised in Table 1.

For our retrieval experiments we used Lemur, a publicly available information retrieval system developed jointly by the University of Massachusetts Amherst and Carnegie Mellon University (`http://www.lemurproject.org`). The software can index large collections of documents, and the toolkit implements a wide variety of information retrieval models. The toolkit supports the construction of unigram language models for documents and queries, and implements retrieval systems based on Okapi BM25 term weighting, the vector space model, and simple language model approaches. We used version 4.3.2 of the Lemur software.

3.3 Parameter Choices

All of the retrieval models considered in our work require the setting of parameters. In this section we discuss the parameter choices.

Okapi BM25: The Okapi BM25 similarity measure includes three tuning parameters: k1, b and k3. k1 and b are usually set to 1.2 and 0.75 respectively, although smaller values for b may work effectively for

Table 1: IR Models used in experiments.

IR Model	Label	IR Model Description	Parameters Used
Okapi BM25	Okapi	Probabilistic Model — Okapi BM25 retrieval function	$k1 = 1.2$, $b = 0.75$, $k3 = 1000$
LM with Jelinek-Mercer smoothing	JM	Unigram language model based on Kullback-Leibler divergence, using Jelinek-Mercer smoothing	$\lambda = 0.7$, where λ is the collection language model weight for JM interpolation
LM with Bayesian (Dirichlet) Smoothing	Dir	Unigram language model based on Kullback-Leibler divergence, using Dirichlet prior smoothing	$\mu = 2,000$, where μ is the Dirichlet prior parameter
LM with Absolute Discounting Smoothing	Dis	Unigram language model based on Kullback-Leibler divergence, using absolute discounting smoothing	$\delta = 0.7$, where δ is the delta discounting constant
LM with Two-Stage Smoothing	Two-Stage	Unigram Language Model, smoothed by a combination of Dirichlet prior and Jelinek Mercer smoothing, Kullback-Leibler divergence language model based retrieval method	This method estimates the parameters automatically

some queries [Robertson and Walker, 1999]. For parameter k3, Robertson and Walker [1999] state: "...in long queries k3 is often set to 7 or 1000 (effectively infinite.)" They do not describe a different setting for shorter queries.

We note that these parameters are query- and collection-dependent. That is, if relevance information was available in advance, it would be possible to tune the Okapi BM25 function to further increase MAP or other information retrieval metrics. However, this is not possible in real retrieval scenarios. We therefore restrict our investigation to the recommended parameter settings.

Language Modelling Smoothing Parameters: Retrieval performance is very sensitive to choice of smoothing parameters [Zhai and Lafferty, 2001a]. Smoothing is the most critical component of the language modelling approach, because the IR system must accurately estimate document and query model parameters for effective retrieval performance.

Jelinek-Mercer smoothing relies on a parameter λ to interpolate the maximum likelihood probabilities with the expected term frequency in the collection. When λ is small, less smoothing is applied and there is more emphasis on relative term weights. Such a small value of λ represents a conjunctive interpretation of the query terms [Zhai and Lafferty, 2004]. In our experiments we consistently used a value of $\lambda = 0.7$, and conducted separate sensitivity tests to confirm that this value is reasonable.

For Dirichlet smoothing, previous work shows that $\mu = 2000$ gives good performance in nearly all cases [Zhai and Lafferty, 2004]. It has also been demonstrated that, for absolute discounting of term probabilities, smoothing using a constant $\delta = 0.7$ is near best for both long and short queries, and across various test collections.

We remark that these parameter values give good performance across a range of collections and user requests, but that no parameter setting is guaranteed to be optimal for all queries and collections. Both Okapi BM25 and the language modelling approaches rely on parameters which are determined empirically. We did not attempt to optimise parameter choices using methods such as Expectation-Maximisation, leave-one-out or cross-validation using a training corpus.

We use the same set of smoothing parameters — using values recommended in the IR literature, as explained above — for each language modelling method. This enables comparison across collections and tasks.

3.4 Test Collections

In our experiments, we use a range of TREC test collections, topics and relevance judgements. The collections represent different types of data, such as newswire and Web data.

Newswire Data: We use the *newswire* collection used at TREC-8 for the *ad hoc* retrieval task. The data consists of newswire text from Financial Times Limited, the Los Angeles Times, the Foreign Broadcast Information Service, and the Federal Register. A set of 50 search topics (numbers 401–450) are used in our experiments for topic finding searches on newswire data.

Web Data (WT10g): The WT10g collection is a 10 gigabyte snapshot of a 1997 Web crawl. It is a subset of the larger VLC2 collection [Bailey et al., 2003]. This engineered subset contained many features representative of real Web data, including inter-server links.

The topic finding queries used with the collection were sampled from actual Web search engine logs, and formed the 'title' part of the TREC topic. The description and narrative were added later.

To study the effectiveness of language modelling techniques for different types of search requests, we also use named-page queries in our experiments. Such searches are not typical for collections consisting of newswire text, but are common on the Web. We aim to investigate how the type of task influences the relative performance of language modelling approaches as compared with Okapi BM25.

The topic sets we use for this collection are:

- TREC-9 (2000) Web queries 451–500, taken from search engine query logs (including word misspellings)
- TREC-10 (2001) Web queries 501–550, taken from search engine query logs (no word misspellings)
- TREC-10 (2001) Named-Page queries NP1–145, to evaluate the task of homepage finding

Web Data (.GOV): The .GOV TREC test collection [Craswell and Hawking, 2002] is a set of 1.25 million Web documents from the .gov (U.S. government Internet) domain in early 2002. As well as HTML, the collection includes the extracted text from Adobe

```
<num> Number:  403
<title> osteoporosis
<desc> Description:
Find information on the effects of the dietary intakes
of potassium, magnesium and fruits and vegetables as
determinants of bone mineral density in elderly men
and women thus preventing osteoporosis (bone decay).
<narr> Narrative:
A relevant document may include one or more of the
dietary intakes in the prevention of osteoporosis.
Any discussion of the disturbance of nutrition and
mineral metabolism that results in a decrease in bone
mass is also relevant.
```

Figure 1: A sample topic from TREC 9.

Acrobat, Microsoft Word, Postscript and plain text files.

As with the WT10g collection, we have a set of named-page finding requests for the .GOV collection. We also use a set of requests that aim to retrieve documents that 'distill' or capture a list of key resources about a particular topic. The target documents would be like a user's bookmark file that captures key Web entry-pages on a topic. These requests are called 'topic distillation' requests.

The topic sets we use with the .GOV collection are:

- TREC-2002 Named-Page queries NP1–150
- TREC-2002 Topic Distillation queries 551–600

3.5 Query Length

TREC topic statements for *ad hoc* search consist of an identifier, title, description, and narrative describing the user's information need. The title is a very short description of the information need, usually consisting of a few keywords. The description and narrative add progressive levels of detail about the topic, specifying conditions that might contribute to or detract from the relevance of a page. A sample TREC topic is shown in Figure 1.

Title only searches are representative of Web searches, because the title field in the TREC topic is only several keywords in length. The more verbose 'title + description + narrative' form of query representation assumes that the user is willing to give greater detail about their information need.

To study the effects on retrieval performance using both short and long queries, we combine parts of the topic statement as follows: title only; title + description; and, title + description + narrative. Where we combine topic fields, we simply form a bag of words with the words from each part of the topic. We weight the contribution of each component of the query identically.

3.6 Pre-Processing

We stem both queries and collections using the Porter stemmer [Porter, 1980], and eliminate terms from queries and collections that are in the SMART system stop list [Salton, 1971, 1999].

We note that there are many different stemming algorithms and lists of stop words. We choose to use the Porter stemmer and SMART retrieval system stop list because these are widely used in information retrieval experiments and at TREC.

We index the full text (natural language textual content) of the documents in the test collections. Evaluations of *ad hoc* retrieval at the Text Retrieval Conferences have found that anchor text, document structure and link structure (i.e. in-links and outlinks) can be useful sources of evidence for the relevance of documents, and there are many successful experiments using these additional features. However, as we are interested in conducting a comparative analysis of retrieval models, we do not make use of such additional features to avoid the introduction of additional confounding factors into the analysis. Using only the core retrieval model should not favour any individual IR modelling approach.

3.7 IR System Evaluation

The field of information retrieval has a strong history of experimental evaluation. The relative performance of systems is generally compared by measuring how many relevant answers the search has identified, and how early in the ranked list these answers occur. A variety of metrics have been proposed, emphasising different aspects of system performance, and different search tasks. For our topic finding and topic distillation experiments, we use three different ways of measuring precision of the ranked list that an IR system produces:

- the mean of the precision scores obtained after the system retrieves each relevant document (mean average precision or MAP)
- precision at R documents retrieved, where R is the number of known relevant documents in the collection (R-precision)
- the number of relevant documents in the top ten documents retrieved (Precision@10).

Buckley and Voorhees [2005] report that mean average precision (MAP), the average of precision scores at each relevant document retrieved, is the single measure most often used in IR research to represent the overall effectiveness performance of a system.

For named-page finding, it is generally assumed that the user is interested in retrieving a single specific resource. The above precision measures are therefore not appropriate for the named-page finding task. Instead, the mean reciprocal rank (MRR) is used; this is the inverse of the rank position at which the relevant resource was returned, averaged over a set of queries.

To investigate whether a true effect has occurred, and that variations in results are not due to chance alone, statistical significance testing is often employed to attach a level of confidence to observed outcomes. Sanderson and Zobel [2005] have demonstrated that the paired *t*-test is a suitable instrument for investigating differences in information retrieval system performance; we use this test for significance testing in our experiments.

4 Results

We structure our findings by type of task: topic finding, named-page finding, and topic distillation. For each task, we also identify the type of collection used (newswire and Web data).

To investigate the effects of query length, we work with different combinations of fields in each TREC topic to form the query that is submitted to Lemur. Title-only queries are only several words long. The description adds more definition to the few words in the title. The narrative contains a concise description of what aspects of the topic make the document relevant. To represent a combination of topic fields, we use the following codes:

- 'T' = title only
- 'TD' = title + description
- 'TDN' = title + description + narrative

Table 2: Topic Finding Task, Queries 401–450, TREC-8 Newswire. The highest score for each collection/measurement pair is shown in bold.

Collection	Method	Parameter	MAP	R-Prec.	Prec@10
Trec8 T	Okapi BM25	Okapi	0.2292	0.2820	0.4380
	JM	$\lambda = 0.7$	0.2310 (p=0.8181)	0.2889 (p=0.3495)	0.4220 (p=0.3824)
	Dir	$\mu = 2,000$	**0.2470** (p=0.0757)	0.2911 (p=0.3739)	**0.4560** (p=0.3710)
	Dis	$\delta = 0.7$	0.2384 (p=0.0686)	0.2935 (p=0.0776)	0.4440 (p=0.6727)
	Two-Stage	auto	0.2406 (p=0.0650)	**0.2953** (p=0.0369)	0.4260 (p=0.4282)
Trec8 TD	Okapi BM25	Okapi	0.2528	0.2908	0.4640
	JM	$\lambda = 0.7$	0.2582 (p=0.5226)	0.3038 (p=0.1886)	0.4600 (p=0.8372)
	Dir	$\mu = 2,000$	**0.2621** (p=0.3308)	0.3043 (p=0.1587)	0.4460 (p=0.3034)
	Dis	$\delta = 0.7$	0.2599 (p=0.1737)	**0.3105** (p=0.0203)	**0.4880** (p=0.1534)
	Two-Stage	auto	0.2445 (p=0.2455)	0.2933 (p=0.7698)	0.4400 (p=0.1351)
Trec8 TDN	Okapi BM25	Okapi	0.2454	0.3012	0.4560
	JM	$\lambda = 0.7$	**0.2608** (p=0.0379)	**0.3090** (p=0.3733)	0.4880 (p=0.1725)
	Dir	$\mu = 2,000$	0.2597 (p=0.1334)	0.3026 (p=0.8805)	0.4660 (p=0.4616)
	Dis	$\delta = 0.7$	0.2459 (p=0.9540)	0.2983 (p=0.7660)	**0.4920** (p=0.1723)
	Two-Stage	auto	0.2093 (p=0.0004)	0.2631 (p=0.0018)	0.4520 (p=0.8875)

In our result tables, the highest scores for each collection/measurement pair are shown in bold font. The p-values for a paired t-test are shown beneath the Mean Average Precision, R-Precision, Precision@10 and Mean Reciprocal Rank scores in the tables. In the discussion of results, where a p-value is less than 0.05, we take this as an indication of statistical significance. However, reporting actual p-values allows for the adjustment of the significance threshold.

4.1 Topic Finding

Topic finding is the classic *ad hoc* search task. In response to the user's query, the IR system searches a collection of indexed documents, and returns an ordered list of answer resources. The more effective the IR system's search algorithm is, the better the quality of the ranked list of retrieved results.

Newswire Data: Table 2 shows the results for queries 401–450 on the TREC-8 newswire data. For the newswire collection, language models perform well. However, there is variation between the smoothing approaches: each scores highest on at least one of the three metrics for a variant of the topic.

Based on MAP scores, the ordering of methods for the newswire collection is as follows:

- T (title-only): $Dirichlet > Two - Stage > AbsDiscounting > Okapi > Jelinek - Mercer$
- TDN (all topic fields): $Jelinek - Mercer > Dirichlet > AbsDiscounting = Okapi > Two - Stage$

The mean average precision score for Jelinek-Mercer using longer versions of queries is significantly better than Okapi BM25 (paired t-test, $p < 0.05$).

Web Data (WT10g): Table 3 shows the results for queries 451–500 for the TREC-9 *ad hoc* search on the WT10g Web data. The results on the same collection using TREC-10 (2001) queries 501–550 are given in Table 4.

Only Dirichlet consistently outperforms the Okapi BM25 baseline on the Web data. The difference between mean average precision scores for Dirichlet and Okapi BM25 is statistically significant for title-only queries (paired t-test, $p < 0.05$), and Dirichlet smoothing is numerically superior for longer queries.

Using MAP scores, an ordering of methods for the Web data collections and queries is:

- T (title-only): $Dirichlet > Okapi = Two - Stage > AbsDiscounting > Jelinek - Mercer$
- TDN (all topic fields): $Dirichlet > Okapi > AbsDiscounting > Jelinek - Mercer > Two - Stage$

Overall, for the topic finding task, we find that Dirichlet smoothing consistently performs better than Okapi BM25 on both newswire and Web data. While Jelinek-Mercer performs well on the smaller newswire collection, particularly for longer queries for which our parameter choice is close to optimal, this finding does not hold for the Web data. Two-stage smoothing performs well for title-only queries, but its performance deteriorates as the query becomes more verbose.

Regarding query length, both Okapi BM25 and the language model approaches performed better with the longer, verbose versions of the topics than with the title field only.

Because Jelinek-Mercer smoothing does not perform as well as expected for longer queries on the Web data, and since the optimal value of λ for Jelinek-Mercer smoothing is dependent on the collection and query set, we conducted separate sensitivity tests using the TREC-9 Web data. However, the results of these (not reported here for space reasons) suggest

Table 3: Topic Finding Task, Queries 451–500, TREC-9 Web data.

Collection	Method	Parameter	MAP	R-Prec.	Prec@10
Trec9 T	Okapi BM25	Okapi	0.1602	0.1969	0.2458
	JM	$\lambda = 0.7$	0.1212 (p=0.0340)	0.1338 (p=0.0140)	0.1604 (p=0.0007)
	Dir	$\mu = 2,000$	**0.1864** (p=0.0171)	**0.2229** (p=0.0457)	**0.2771** (p=0.1789)
	Dis	$\delta = 0.7$	0.1397 (p=0.2009)	0.1641 (p=0.1648)	0.1875 (p=0.0058)
	Two-Stage	auto	0.1722 (p=0.1268)	0.2085 (p=0.1302)	0.2479 (p=0.8497)
Trec9 TD	Okapi BM25	Okapi	0.1950	0.2399	0.3060
	JM	$\lambda = 0.7$	0.1799 (p=0.1098)	0.2181 (p=0.0526)	0.2680 (p=0.0735)
	Dir	$\mu = 2,000$	**0.2302** (p=0.0111)	**0.2754** (p=0.0039)	**0.3460** (p=0.0792)
	Dis	$\delta = 0.7$	0.1833 (p=0.4258)	0.2233 (p=0.4629)	0.3020 (p=0.8355)
	Two-Stage	auto	0.1681 (p=0.0767)	0.2063 (p=0.1543)	0.2920 (p=0.4306)
Trec9 TDN	Okapi BM25	Okapi	0.2053	0.2528	0.3300
	JM	$\lambda = 0.7$	0.1788 (p=0.1152)	0.2082 (p=0.1035)	0.2840 (p=0.0219)
	Dir	$\mu = 2,000$	**0.2164** (p=0.5235)	**0.2566** (p=0.8702)	**0.3440** (p=0.4916)
	Dis	$\delta = 0.7$	0.1919 (p=0.3787)	0.2332 (p=0.4166)	0.3240 (p=0.7362)
	Two-Stage	auto	0.1602 (p=0.0256)	0.2056 (p=0.0515)	0.2820 (p=0.0100)

that even tuning this parameter does not lift absolute Jelinek-Mercer MAP scores above Okapi BM25.

4.2 Named-Page Finding Task

For the named-page finding task, the system should return one document — the specific named resource. The task reflects the desire of the searcher to find a resource that is known or suspected to exist; it is assumed that only this resource (or a duplicate of it) is relevant to the request. The resource is a single significant document, and may not be a site-entry page. A named-page topic in the TREC framework is specified by a short number of keywords only, without a description or narrative section. Two sample named-page finding topics from TREC-2001 are 'Hotel Grand, Thailand' and 'Quicken Support'.

Table 5 shows the results of our evaluation for the named-page finding task using WT10g Web data and topics NP1–145. Results of the named-page finding task using .GOV domain Web data and topics NP1–150 are given in Table 6.

With a total of nearly 300 named-page queries, and using two distinct Web data collections, we found that Okapi BM25 achieved best performance for the named-page finding task. The most successful method for *ad hoc* search, Dirichlet smoothing, came last in our comparisons, performing significantly worse than Okapi. Two-stage smoothing showed similar performance to Okapi, and the difference in mean reciprocal rank scores is not significant ($p > 0.05$).

Ranking by MRR score, the ordering of methods for named-page finding is:

- $Okapi > Two - Stage > AbsDiscounting > Jelinek - Mercer > Dirichlet$

One reason for the success of Okapi BM25 at this task may be that it behaves like coordination level ranking [Hiemstra, 2001]. Coordination level ranking partially ranks documents with n query terms above documents with $n - 1$ terms, and this property is especially useful for very short queries. The language modelling approaches do not benefit from adjusting term probabilities using the frequency of terms in the collection, or assigning probability mass to unseen terms. In fact, the more aggressive the smoothing, the smaller the level of coordination between query terms [Kraaij, 2004]. A simpler query term matching strategy like Okapi BM25 gives strong results for short queries.

To perform a truly effective search for named Web resources, an IR system should consider the page's position in a site hierarchy and the page's importance, rather than only the page's text. Employing evidence such as hyperlink measures, anchor text, and URL structure is very useful for the named-page finding task. By using content terms only, the mean reciprocal rank scores, and other metrics given in our Results section, may not reflect the best possible performance for each language modelling approach. However, our focus is on a comparative analysis between approaches. We therefore restrict our attention to content-only evidence for all retrieval models that we consider.

4.3 Topic Distillation Task

Topic distillation requests are similar to named-page finding requests; the objective is to return high-level, authoritative Web pages that 'facilitate navigation into a site' [Hersh, 2004]. The best system approaches for such requests use a combination of Web page features. These include anchor text for links between and on HTML Web pages (link text is likely to describe what the target page is about), URL length, and measures of page importance.

For example, for a request 'obesity in the U.S.', there are many United States government Web pages that mention these keywords in their content. The correct answer for the topic distillation task is `http://www.surgeongeneral.gov/topics/`

Table 4: Topic Finding Task, Queries 501–550, TREC-2001 Web data.

Collection	Method	Parameter	MAP	R-Prec.	Prec@10
TREC-2001 T	Okapi BM25	Okapi	0.1522	0.2056	0.2918
	JM	$\lambda = 0.7$	0.1113 (p=0.0003)	0.1505 (p=0.0037)	0.2122 (p=0.0003)
	Dir	$\mu = 2,000$	**0.1774** (p=0.0307)	**0.2238** (p=0.3236)	**0.3184** (p=0.3165)
	Dis	$\delta = 0.7$	0.1370 (p=0.0511)	0.1906 (p=0.053)	0.2653 (p=0.1348)
	Two-Stage	auto	0.1441 (p=0.2963)	0.1934 (p=0.3992)	0.2898 (p=0.8962)
TREC-2001 TD	Okapi BM25	Okapi	0.1786	0.2234	0.3520
	JM	$\lambda = 0.7$	0.1425 (p=0.0004)	0.1855 (p=0.0054)	0.2920 (p=0.0003)
	Dir	$\mu = 2,000$	**0.1984** (p=0.1150)	**0.2385** (p=0.4409)	**0.3760** (p=0.2857)
	Dis	$\delta = 0.7$	0.1653 (p=0.2528)	0.2072 (p=0.3014)	0.3480 (p=0.8024)
	Two-Stage	auto	0.1542 (p=0.0716)	0.2014 (p=0.1753)	0.3360 (p=0.4293)
TREC-2001 TDN	Okapi BM25	Okapi	0.1942	0.2356	**0.3860**
	JM	$\lambda = 0.7$	0.1657 (p=0.0025)	0.2085 (p=0.0523)	0.3140 (p=0.0011)
	Dir	$\mu = 2,000$	**0.2051** (p=0.1868)	**0.2400** (p=0.6025)	0.3500 (p=0.0685)
	Dis	$\delta = 0.7$	0.1774 (p=0.0403)	0.2300 (p=0.4898)	0.3540 (p=0.0167)
	Two-Stage	auto	0.1505 (p=0.0003)	0.2024 (p=0.0011)	0.3220 (p=0.0028)

Table 5: Named-Page Finding Task, Queries NP1–145, TREC 2001 Data.

Collection	Method	Parameter	Mean Reciprocal Rank	Success@10
WT10G	Okapi BM25	Okapi	**0.2947**	**0.517**
	JM	$\lambda = 0.7$	0.2237 (p=0.0025)	0.400
	Dir	$\mu = 2,000$	0.1642 (p<0.0001)	0.297
	Dis	$\delta = 0.7$	0.2808 (p=0.4689)	0.476
	Two-Stage	auto	0.2734 (p=0.0912)	0.503

`obesity/`, which is an authoritative page on the topic. The page is judged the correct answer as it introduces the topic and links to other useful and relevant resources. The page above is also an entry page at the most appropriate level in the Surgeon General website's hierarchy.

Topic distillation as a search task is different to *ad hoc* topical searches. For *ad hoc* search, the user wants the most topically relevant documents returned to them. For topic distillation, the system should return to the user a Web page that is an entry page to other high-quality resources on the topic.

We show results for queries 551–600, which were part of the TREC-2002 Topic Distillation task, in Table 7. It can be seen that Jelinek-Mercer and absolute discounting smoothing approaches work well for longer versions of the topics. For queries formed from joining the 'title' and 'description' fields of each topic, absolute discounting produces the best results, but the difference is not statistically significant ($p > 0.05$). For title-only queries, two-stage smoothing performs best, but not significantly better than Okapi BM25 ($p > 0.05$).

Using MAP, which Craswell and Hawking [2004] indicate as an appropriate measure for the topic distillation task, the ordering of methods for topic distillation is:

- T (title-only): $Two - Stage = Okapi > AbsDiscounting > Dirichlet > Jelinek - Mercer$
- TDN (all topic fields): $Jelinek - Mercer = AbsDiscounting > Okapi > Two - Stage > Dirichlet$

Our results indicate that absolute discounting and Jelinek-Mercer smoothing (with optimal parameters) can outperform the strong Okapi BM25 baseline for longer queries. Jelinek-Mercer smoothing performs poorly for short queries; using a smaller value for λ improves retrieval accuracy, but the revised approach does not improve significantly on Okapi BM25. Dirichlet smoothing, which was the strongest overall performer for topic finding search, performed badly at named-page finding and topic distillation tasks.

An overall trend is that the Okapi BM25 function performs best with shorter queries for topic distillation requests. This finding is not repeated in our results for topic finding search, however.

It may be possible that the longer topic descriptions contain a lot of 'noise' words, the presence of which degrade precision for Okapi BM25. The language modelling approaches appear to handle the verbosity of the queries more effectively than Okapi-BM25 as the form of request gets longer. This is one

Table 6: Named-Page Task, Queries NP1–150, TREC 2002 Data.

Collection	Method	Parameter	Mean Reciprocal Rank	Success@10
.GOV	Okapi BM25	Okapi	**0.5993**	0.760
	JM	$\lambda = 0.7$	0.4833 (p<0.0001)	0.693
	Dir	$\mu = 2,000$	0.4459 (p<0.0001)	0.647
	Dis	$\delta = 0.7$	0.5366 (p=0.0004)	0.733
	Two-Stage	auto	0.5790 (p=0.2178)	**0.773**

Table 7: Topic Distillation Task, Queries 551–600, TREC 2001 Data.

Collection	Method	Parameter	MAP	R-Prec.	Prec@10
Queries 551–600 T	Okapi BM25	Okapi	0.1883	0.2185	0.2420
	JM	$\lambda = 0.7$	0.1348 (p=0.0137)	0.1643 (p=0.0290)	0.1960 (p=0.0290)
	Dir	$\mu = 2,000$	0.1486 (p=0.0042)	0.1759 (p=0.0120)	0.1840 (p=0.0447)
	Dis	$\delta = 0.7$	0.1840 (p=0.6481)	0.2078 (p=0.4187)	0.2440 (p=0.9012)
	Two-Stage	auto	**0.1936** (p=0.5326)	**0.2259** (p=0.5044)	**0.2620** (p=0.2617)
Queries 551–600 TD	Okapi BM25	Okapi	0.1476	0.1803	0.1840
	JM	$\lambda = 0.7$	0.1438 (p=0.8649)	0.1734 (p=0.7860)	0.2000 (p=0.4045)
	Dir	$\mu = 2,000$	0.1349 (p=0.3059)	0.1622 (p=0.4224)	0.1940 (p=0.5789)
	Dis	$\delta = 0.7$	**0.1533** (p=0.7364)	**0.1816** (p=0.9572)	**0.2080** (p=0.2290)
	Two-Stage	auto	0.1337 (p=0.4953)	0.1660 (p=0.5400)	0.2060 (p=0.2193)
Queries 551–600 TDN	Okapi BM25	Okapi	0.1287	0.1462	0.1800
	JM	$\lambda = 0.7$	**0.1536** (p=0.0441)	0.1800 (p=0.0120)	**0.2120** (p=0.2062)
	Dir	$\mu = 2,000$	0.1186 (p=0.1760)	0.1471 (p=0.9316)	0.1720 (p=0.6273)
	Dis	$\delta = 0.7$	0.1506 (p=0.0443)	**0.1825** (p=0.0112)	0.2100 (p=0.1603)
	Two-Stage	auto	0.1277 (p=0.9336)	0.1550 (p=0.4848)	0.1980 (p=0.4695)

function of the role of smoothing, i.e. that of 'explaining' common words in the query [Zhai and Lafferty, 2004].

5 Conclusion and Future Work

The aim of this study was to investigate the relative performance of the probabilistic and language model approaches to information retrieval. To our knowledge, it is the first large-scale study that has examined the performance of these core retrieval technologies, using consistent settings, across a range of scenarios including different types of data, different search tasks, and different query lengths. Our results indicate that there is no single retrieval approach that can offer optimal performance under all conditions, but that each can offer different advantages in different situations.

For the topic finding task using short keyword queries, Dirichlet prior smoothing significantly outperformed the probabilistic Okapi BM25 method (two-tailed t-test, $p < 0.05$). Two-stage smoothing was the next best language modelling method for title-only queries. For longer queries, Dirichlet smoothing also out-performed all other methods. We did not find that the other smoothing methods (absolute discounting, Jelinek-Mercer and two-stage smoothing) could outperform Okapi BM25 over the two collections we used.

The named-page task is very different to topic finding; an exact match on query terms frequently is not sufficient to find the specific named resource. However, Okapi BM25 performed well. As the named-page queries are short descriptions, Okapi BM25 behaved like coordination level ranking, a ranking strategy which is appropriate for very short queries. Dirichlet smoothing, which performed well at topic finding tasks, gave less emphasis to the relative weights of the query terms, and was the worst performer at named-page finding.

For the topic distillation task, absolute discounting and Jelinek-Mercer worked well with the longer versions of the TREC topics, and significantly outperformed Okapi BM25 (two-tailed t-test, $p < 0.05$). Okapi BM25 performed better with shorter queries for this task; we suspect that the presence of 'noise' words in the longer queries degraded precision for Okapi BM25. The language modelling approaches are able to 'explain' such non-informative words in the query more effectively.

All language modelling approaches required smoothing parameters which must be determined empirically for optimal performance. Sensitivity analysis showed little change to the relative ordering of language model smoothing methods when we chose different values for parameters.

The research undertaken here provided information about the types of task and forms of query for which language models are best suited. In IR, performance is dependent to a degree on the type of collection and the set of queries. In our study, we restricted our attention to representing the content of a document only, not its link structure, anchor text or other markup — features that are important for retrieval on web data. A wider study with multiple sources of evidence like link structure would broaden the applicability of these findings. Stemming and stopping are generally applied in most IR experiments; however it is not clear that they bring benefits to the retrieval tasks used here. The interaction between combinations of stemming, stopping, and query length would be worthwhile to study more closely. Our broad-based study was able to provide useful insight into the relative performance of core retrieval models. Future work investigating the impact of these other factors would further enhance the understanding of the relative performance of retrieval techniques.

References

P. Bailey, N. Craswell, and D. Hawking. Engineering a multi-purpose test collection for Web retrieval experiments. *Information Processing and Management*, 39(6):853–871, 2003.

A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2): 3–10, 2002.

C. Buckley and E. M. Voorhees. *TREC: Experiment and Evaluation in Information Retrieval*, chapter 3, pages 53–75. The MIT Press, 2005.

N. Craswell and D. Hawking. Overview of the TREC-2002 Web Track. In *Proceedings of the Eleventh Text REtrieval Conference (TREC-2002)*, Gaithersburg, Maryland USA, November 2002. NIST Special Publication.

N. Craswell and D. Hawking. Overview of the TREC-2004 Web Track. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC-2004)*, Gaithersburg, Maryland USA, November 2004. NIST Special Publication.

W. Hersh. Information Retrieval: A Health & Biomedical Perspective (Second Edition) - Update - Chapter 8, 2004. `http://medir.ohsu.edu/~hersh/irbook/updates/update8.html` (accessed 30 September 2006).

D. Hiemstra. *Using language models for information retrieval.* PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2001.

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 174–185. NIST Special Publication, 1998.

F. Jelinek and R. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Workshop on Pattern Recognition in Practice*, 1980.

W. Kraaij. *Variations on Language Modeling for Information Retrieval.* PhD thesis, University of Twente, June 2004.

V. Lavrenko. SIGIR 2003 Tutorial: Language Modeling in Information Retrieval, 2003. `http://ciir.cs.umass.edu/~lavrenko/SIGIR2003-Tutorial.pdf` (accessed 30 September 2006).

V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 120–127, New Orleans, LA, 2001.

H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.

J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 275–281, Melbourne, Australia, 1998.

M. Porter. An algorithm for suffix stripping. *Program*, 14: 130–137, 1980.

S. E. Robertson and S. Walker. Okapi/Keenbow at TREC-8. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC-8).* NIST Special Publication, 1999.

S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 232–241, Dublin, Ireland, 1994.

G. Salton. *The SMART Retrieval System - Experiments in Automatic Document Processing.* Prentice-Hall, 1971.

G. Salton. SMART Version 11.0 Stop word list, 1999. `ftp://ftp.cs.cornell.edu/pub/smart/english.stop` (accessed 30 September 2006).

M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 162–169, Salvador, Brazil, Aug. 2005.

F. Song and W. B. Croft. A general language model for information retrieval. In *CIKM '99: Proceedings of the Eighth International Conference on Information and Knowledge Management*, pages 316–321, New York, NY, USA, 1999.

K. Sparck Jones and S. E. Robertson. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - Part 1. *Information Processing and Management*, 36(6):779–808, 2000.

C. Zhai. Statistical language models for information retrieval, 2006. `http://sifaka.cs.uiuc.edu/lmir/sigir06-tutorial-lmir.pdf` (accessed 30 September 2006).

C. Zhai. *Risk Minimization and Language Modeling in Text Retrieval.* PhD thesis, Carnegie Mellon University, 2002.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 334–342, New Orleans, Louisiana, United States, 2001a. ACM Press.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions On Information Systems*, 22(2):179–214, 2004.

C. Zhai and J. D. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the Tenth International Conference on Information and Knowledge Management*, pages 403–410, 2001b.