Chapter 4

Text Retrieval

4.1 Introduction

Making large quantities of information available electronically is of limited use if it is not possible to obtain specific items easily. For this reason, this chapter considers text retrieval as a specific application of probabilistic language modelling. The primary focus is on *ad hoc* text retrieval. In this task, a collection of documents is provided, each of which consists of a string of text, possibly with some meta-information attached such as a title or a list of authors. Given a query, also consisting of a short string, the goal is to return an ordered set of documents such that documents which are most relevant to the query are returned first. Retrieval of richer documents, for example with layout information, or containing multimedia data, is considered to be outside of the scope of the problem. The notion of relevance is left vague, although it is usually compared against human judgements. Material in this section was presented in [22].

In this chapter, the symbol $\mathcal{C}$ will be used to represent the whole collection, which consists of documents $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots\}$. The symbol $\boldsymbol{q}$ will be used to represent the vector of terms present in the query.

4.2 Review of previous work

4.2.1 Vector space models

Vector space models [71] [57] were one of the earliest approaches to information retrieval. These models do not treat the problem probabilistically, but instead assign a vector to each document whose dimensionality is the size of the vocabulary,

$$\boldsymbol{d}_i = \sum_j d_{ij} \hat{\boldsymbol{t}}_j \tag{4.1}$$

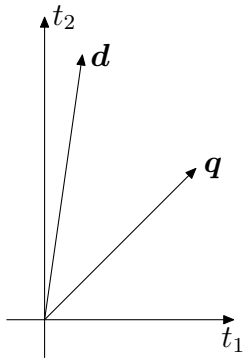

Figure 4.1: An illustration of the representation of a document, $\boldsymbol{d}$, and a query, $\boldsymbol{q}$, in the vector space model. In this example the vocabulary consists of just two terms, t_1 and t_2, which correspond to the to dimensions of the space. The components of each vector are the weights of these terms.

where $\left\{\hat{\boldsymbol{t}}_j\right\}$ is a set of basis vectors and d_{ij} is a *term weight* given to term j in document i. A corresponding vector is defined for the query,

$$\boldsymbol{q} = \sum_j q_j \hat{\boldsymbol{t}}_j \tag{4.2}$$

with q_j being the query term weight. This representation is illustrated in Figure 4.1.

The relevance is often defined in terms of the inner product of the normalised vectors,

$$R\left(\boldsymbol{d}_i, \boldsymbol{q}\right) = \frac{\boldsymbol{d}_i \cdot \boldsymbol{q}}{\left|\boldsymbol{d}_i\right| \left|\boldsymbol{q}\right|} \tag{4.3}$$

$$= \cos\theta_{\boldsymbol{qd}} \tag{4.4}$$

where $\theta_{\boldsymbol{qd}}$ is the angle between the vectors. It is common to use a form of *tf.idf* weighting, where *tf* stands for Term Frequency and *idf* stands for Inverse Document Frequency. In its simplest form this approach gives weights which are proportional to the number of times that each term appears in the corresponding document or query, and are inversely proportional to the number of documents in the collection containing the term. This is intuitively appealing, as a term which appears in many documents is less likely to be discriminative than one which only appears in a few. In practice a variety of non-linear transformations are used on the raw weights [57] [81].

4.2.2 Binary independence retrieval

Binary Independence Retrieval (BIR) express information retrieval in terms of probabilistic inference [72]. For each query there is assumed to be a set of relevant documents, with relevance being expressed in terms of the probability that a document is a member of this set. The original setup made use of a weight for each term appearing both in the document

and the query given by

$$w_i = \log \frac{p_i\,(1-q_i)}{q_i\,(1-p_i)} \tag{4.5}$$

where p_i and q_i are probabilities that a document chosen at random from the set of relevant and not relevant documents respectively contains term i. This expression is based on ratio of the odds of the terms appearing in the relevant and non-relevant sets respectively. Use of this model with estimates of these probabilities taken from the collection gives rise to the Robertson-Sparck Jones weight,

$$w_i^{RSJ} = \log \frac{(r_i + 0.5)\,(N - R - n_i + r_i + 0.5)}{(R - r_i + 0.5)\,(n_i - r_i + 0.5)} \tag{4.6}$$

where n_i is the number of documents containing term i, N is the total number of documents, r_i is the number of relevant documents containing the term, and R is the number of relevant documents. The additional values of 0.5 can be viewed as playing a role similar to smoothing using a Dirichlet prior as described in Section 2.3.2. Note that in order to make use of this weighting formula it is necessary to have estimates of the number of relevant documents and the associated word frequencies — a situation which is very rarely the case. At best we will have information concerning the relevance of a small number of documents based on user feedback, but in many cases none at all will be available. If this information is not known then a further simplification is often made,

$$w_i = \log \frac{N - n_i + 0.5}{n_i + 0.5} \tag{4.7}$$

which assumes that the relevant set is of negligible size compared to the whole collection. In this case, n_i can be approximated by the overall collection frequencies of terms. This function decreases monotonically with n_i/N. In other words, higher scores are assigned to terms which appear in few documents and are therefore highly discriminative, essentially embodying the principle of *idf* weighting.

Note that there is no term frequency component in (4.6). The weighting scheme is typically used in conjunction with other terms which reflect other aspects of the retrieval process. The most successful of these approaches is the Okapi BM-25 scheme [76]. The full score function in this case assigns a score of

$$s_i = \frac{(k_3 + 1)\,q_i}{k_3 + q_i} \cdot \frac{(k_1 + 1)\,f_i}{K + f_i} \cdot w_i^{RSJ} \tag{4.8}$$

to each term in the query, where q_i and f_i are the frequencies of term i in the query and document respectively, and

$$K = k_1\,(bL + (1 - b)) \tag{4.9}$$

with L being the overall length of the document. The relevance is the sum of these scores for

all terms in the query. Optionally an additional score of

$$k_2 n_q \frac{1-L}{1+L} \tag{4.10}$$

can be added for each document, providing normalisation of the document length, which is necessary to compensate for the fact that large documents are more likely to contain query terms 'by chance'. The quantities k_1, k_2, k_3 and b are parameters of the score function and must be set in advance, for example by optimisation over a training set of queries. A good summary of the development of these methods is provided in [75].

4.2.3 Language modelling approaches

A more recent approach to probabilistic retrieval makes use of language modelling [66]. In its original form, each of the documents in the collection is treated as being drawn from a separate language model. It is assumed that there is one relevant document, and that the query was generated from the same language model as that document. Bayes' rule can then be used to find the probability that each document is relevant. An alternative approach is to construct a language model from the query, and to rank documents according to the probability that they were generated by that language model. Further variations have considered the task to be equivalent to machine translation, where the 'query language' must be translated into the 'document language' or vice versa [10], or have used a measure of similarity such as the KL divergence [49].

As is the case with the language models described in Chapter 2, maximum likelihood estimates of the term distribution result in very poor performance, and it is generally found that smoothing is needed. Typically, smoothing is performed against a fixed base distribution, P^{base}, with the following two methods being common:

- **Linear Interpolation:** In this case, the maximum likelihood predictions for the document are simply mixed linearly with the base distribution

$$P(t) = \lambda P^{ML}(t) + (1-\lambda) P^{base}(t) \tag{4.11}$$

- **Dirichlet Smoothing:** This approach uses the predictive distribution under a Dirichlet prior where αP^{base}, for some positive constant α, is used as the base measure. The predictive distribution is therefore

$$P(t) = \frac{N}{N+\alpha} P^{ML}(t) + \frac{\alpha}{N+\alpha} P^{base}(t) \tag{4.12}$$

  where N is the total number of terms in the document.

The choice of the distribution with which to smooth is generally treated as external to the derivation of the model, and is often the maximum likelihood estimate of the distribution

for the collection as a whole. A notable exception is that used by the Twenty-One Consortium [31], which uses the normalised vector of document frequencies, although this is very much a heuristic choice. It is worth remarking that in the Dirichlet smoothing model, the choice of a prior distribution based on the data being modelled is not theoretically valid, so the model cannot be considered to be truly Bayesian.

4.2.4 Relevant sets

The methods described above differ in their approach to the relevant set. The BIR family of models are based on the assumption that there is set of relevant documents, the size of which is not specified in advance. The language modelling approach however often works under the assumption that there is precisely one document which is relevant to each query (although alternative interpretations are possible within this framework, for example in [47]).

Whether the assumption of a single relevant document is appropriate is a matter of debate, and quite probably depends on the nature of the application. Creation of language modelling strategies which do not make this assumption is certainly an area for future work in the field. It should also be noted that the evaluation method, to be discussed below, does assume multiple relevant documents, and therefore there is to some extent a conceptual gap between theory and practice for these approaches.

4.2.5 The probability ranking principle

How should information retrieval results be presented to the user? Assuming that relevance is treated as a binary property, so that a document is either relevant to a particular query or it isn't, the *probability ranking principle* [73] asserts that the optimum ordering is to present documents in decreasing order of probability of relevance. This result makes the assumption that relevances are independent. In other words, the probability of one document being relevant is not affected by the relevance of any other document. In practice the assumption of independence does not strictly hold, with the extreme case being duplicate documents whose relevance is fully correlated.

For the language modelling approach, the probabilistic interpretation of the relevance may be the probability of that document being the single item relevant to the query. In this case, it is possible to consider a cost function, c_i, which increases monotonically with the position of the relevant document in the returned list. The expected cost of a ranking is therefore

$$\langle c \rangle = \sum_i p_i \; c_{r_i} \tag{4.13}$$

where p_i is the probability that the i^{th} document is relevant and r_i is the position of document i in the returned list. This is minimised by returning the most probable documents first, with less probable documents being placed in positions which incur a higher cost.

4.2.6 Relevance feedback

Suppose that a retrieval has been performed using a query supplied by the user and resulting a set of documents which have been returned. Out of the returned documents, the user may indicate that some are relevant and some are not. This additional information can then be used to re-rank the collection, with the aim of improving the retrieval results. This approach is known as *relevance feedback*. A variation on this idea is *pseudorelevance feedback*, where the top few results of the initial query are assumed to be relevant when performing the re-ranking, avoiding the need for explicit user intervention.

The details of the approach vary from one retrieval algorithm to another, but in general the technique involves some combination of introducing additional query terms, and altering the weighting of exiting terms in the query. For example, in the vector space retrieval model, one approach is to move the query vector towards the centre of the relevance documents and away from the centre of those which are irrelevant:

$$\boldsymbol{q}_1 = a\boldsymbol{q} + b\sum_{i\in\mathcal{R}}\frac{\boldsymbol{d}_i}{|\mathcal{R}|} - c\sum_{j\in\mathcal{N}}\frac{\boldsymbol{d}_j}{|\mathcal{N}|} \tag{4.14}$$

where $\mathcal{R}$ and $\mathcal{N}$ are the sets of relevant and irrelevant documents respectively, and a, b and c are parameters. This approach is known as the Rocchio method [77]. Similar methods also exist for the BIR method [74].

When relevance feedback is used to expand the query one interpretation of the effect is that terms which are relevant to topics expressed in the query, but which are not contained in the query itself, are identified and included in the search. This approach is similar to Latent Dirichlet Allocation and other topic models described in Section 2.3.10 — one could imagine that both the query and the documents could be converted into a representation based on topics, and those representations could be used in place of the document terms themselves during information retrieval. Applications in information retrieval were indeed a significant motivation behind the development of topic models, and while they are not considered further in this thesis, it is likely that such models will play an important role in the future development of language modelling retrieval methods.

4.3 Whole collection models

The approaches described in Section 4.2.3 all treat the language models learned for each document as being independent. In this section an approach is considered which relaxes this requirement, permitting sharing of information between different documents.

Each document in the collection consists of a sequence of terms, denoted by $(t_1, t_2 \ldots)$. By assigning a label, y_i, to each term indicating the document from which it was taken, the whole collection can be viewed as a single set of (t, y) pairs. It is possible to define a probabilistic model over such pairs. To perform information retrieval, a label is associated with each term

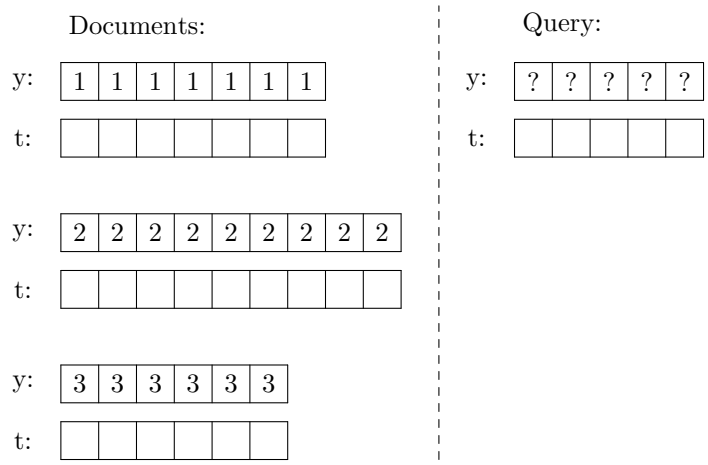

Figure 4.2: Illustration of the whole collection model. Associated with each term in the collection is a label indicating to which document it belongs. A label is also associated with each term in the query, which is viewed as an unobserved random variable. Relevance is defined as the probability that these labels will take the same value as those for a particular document.

in the query as well. It is assumed that the query (t, y) pairs were generated from the same model as those in the collection, with the constraint that the query labels must all take the same value, y_q. The relevance of document $\boldsymbol{d}$ to $\boldsymbol{q}$, $R(\boldsymbol{d}, \boldsymbol{q})$ can now be defined as the log probability that $y_q = y_d$, the label corresponding to $\boldsymbol{d}$. See Figure 4.2 for an illustration.

The symbols $\boldsymbol{t}_\mathcal{C}$ and $\boldsymbol{y}_\mathcal{C}$ will be used to denote vectors of the t and y values for all terms in the collection, and $\boldsymbol{t}_q$ denotes the vector of terms in the query. Predictions for y_q can be made using Bayes' rule,

$$\Pr(y_q \mid \boldsymbol{t}_q, \boldsymbol{y}_\mathcal{C}, \boldsymbol{t}_\mathcal{C}) \propto \Pr(\boldsymbol{t}_q \mid y_q, \boldsymbol{y}_\mathcal{C}, \boldsymbol{t}_\mathcal{C}) \cdot \Pr(y_q \mid \boldsymbol{y}_\mathcal{C}, \boldsymbol{t}_\mathcal{C}) \tag{4.15}$$

The prior, $\Pr(y_q \mid \boldsymbol{y}_\mathcal{C}, \boldsymbol{t}_\mathcal{C})$, is the probability that a given document is relevant in the absence of a query, and can be used to incorporate additional information. For example, an analysis of link structure in a hypertext collection using an algorithm such as PageRank [14] or hubs and authorities [42] can be used to estimate this probability. In this chapter however a uniform prior will be used.

It is possible to express the language modelling approach to information retrieval within this framework by using

$$\Pr\left(t_q^{(i)} \mid y_q, \boldsymbol{y}_\mathcal{C}, \boldsymbol{t}_\mathcal{C}\right) = P_{y_q}\left(t_q^{(i)}\right) \tag{4.16}$$

where $P_{y_d}(t)$ is the language model corresponding to document $\boldsymbol{d}$. The approach presented here can therefore be viewed as a generalisation of existing methods.

4.3.1 The model

The proposed model consists of a separate discrete distribution over terms for each document in the collection. The distribution corresponding to document j has parameter $\boldsymbol{q}_j$. Documents are constructed by making i.i.d. draws of terms from the corresponding distribution. The set of parameters for each of the discrete distributions is drawn from the hierarchical Dirichlet model as described in Section 2.5.1,

$$\begin{aligned} \boldsymbol{p} &\sim \text{Dirichlet}\left(\alpha_1 \boldsymbol{u}\right) && (4.17)\\ \boldsymbol{q}_j &\sim \text{Dirichlet}\left(\alpha_2 \boldsymbol{p}\right) && (4.18)\\ t_k &\sim \text{Categorical}\left(\boldsymbol{q}_{d_k}\right) && (4.19) \end{aligned}$$

This model is equivalent to that illustrated in Figure 2.9. The same approximation is used as in generalised PPM-A. Namely that when making predictions it is assumed that the oracle was asked precisely once for each term observed in each document. This means that the oracle counts will be exactly equal to the number of documents containing the corresponding term, or in other words the document frequency of the term. The model is further approximated by ignoring the effect on the predictive distribution of query terms that have already been seen when making predictions. This gives

$$\Pr\left(t \mid y\right) = \frac{1}{N_{y_d} + \alpha_2}\left(n_t\left(t, y\right) + \alpha_2 \,\hat{\mathrm{p}}\left(t\right)\right) \tag{4.20}$$

where

$$\hat{\mathrm{p}}\left(t\right) = \frac{n_d\left(t\right) + \alpha_1 / \left|\mathcal{V}\right|}{\sum_{t'} n_d\left(t'\right) + \alpha_1} \tag{4.21}$$

which can be interpreted as a modified document frequency term. $n_t\left(t, y\right)$ is the term frequency, or the number of times that term t appears in the document with label y and $n_d\left(t\right)$ is the document frequency, i.e. the number of documents containing t at least once. N_{y_d} is the length of document $\boldsymbol{d}$, and $\left|\mathcal{V}\right|$ is the total number of different terms in the collection.

The log probability of the whole query after a little rearrangement is

$$\begin{aligned} \log \Pr\left(\boldsymbol{t}_q \mid y_q\right) = &\sum_i \log\left(\frac{1}{N_{y_q} + \alpha_2}\right)\\ &+ \sum_i \log\left(1 + \frac{n_t\left(t_q^{(i)}, y_q\right)}{\alpha_2 \,\hat{\mathrm{p}}\left(t_q^{(i)}\right)}\right) + \sum_i \log\left(\alpha_2 \,\hat{\mathrm{p}}\left(t_q^{(i)}\right)\right) \end{aligned} \tag{4.22}$$

Note that the last term in this expression is not dependent on the query, and therefore may

be ignored for the purposes of ranking. The relevance can therefore be written as

$$R(\boldsymbol{d},\boldsymbol{q}) = \sum_i \log\left(1 + \frac{n_t\left(t_q^{(i)}, y_d\right)}{\alpha_2\,\hat{\mathrm{p}}\left(t_q^{(i)}\right)}\right) + N_q \log\left(\frac{1}{N_{y_d}+\alpha_2}\right) \tag{4.23}$$

where N_q is the length of the query and the sum runs over terms appearing in the query, such that terms appearing multiple times in the query contributing multiple times to the sum. The first term in this expression provides *tf.idf*-like query term weighting. The second term may be interpreted as providing global document length normalisation.

The hierarchical model is in many respects similar to Dirichlet smoothing, and therefore recovers the same document length normalisation term. The most important difference however is in the form of $\hat{\mathrm{p}}$ which arises *internally*. The only arbitrary choice is the top level prior, which is uniform. Conceptually this is a significant improvement over existing models, which as mentioned before either neglect document frequency information entirely, or introduce it in an *ad hoc* fashion.

The form of $\hat{\mathrm{p}}$ also solves a minor technical issue, namely the fact that both the maximum likelihood document and collection models assign zero probability to terms which have not been observed anywhere in the collection. Formally this results in a divergent score function, making it impossible to differentiate between documents for any query containing such a term. Of course, this can be solved by simply ignoring any terms which are absent in the collection, as they are not able to affect the results, but it is appealing to be able to produce a model which inherently avoids this problem.

4.3.2 Experimental evaluation

We tested the model on the *ad hoc* tasks from TREC-7 and TREC-8, as well as the Cranfield data set (see Appendix A). The TREC tasks each consist of a set of 50 queries complete with ground-truth relevance judgements against a standard set of documents. As specified for the respective TREC conferences, these tasks used all data on discs 4 and 5 except for the CR texts and with queries 351-400 and 401-450 respectively. The full query text consisting of the title, description and narrative was used in all cases (see Section 4.4.4 below).

The Cranfield data set is much smaller, and much more specialised, containing abstracts from technical papers on aeronautical engineering. In this case 225 queries are provided, again with ground-truth relevance judgements.

The only pre-processing that we did prior to indexing was basic stop-word removal (see Appendix B) and stemming using a Porter stemmer [67]. We performed the experiments using the LEMUR language modelling toolkit [64], which provides amongst other things a set of indexing routines as well as implementations of standard methods such as BM-25.

Precision and recall

Two metrics are commonly used to describe the effectiveness of information retrieval algorithms: precision and recall. Precision is defined by

$$\text{Precision} = \frac{\text{Number of relevant documents returned}}{\text{Number of documents returned}} \tag{4.24}$$

and recall by

$$\text{Recall} = \frac{\text{Number of relevant documents returned}}{\text{Total number of relevant documents}} \tag{4.25}$$

These quantities will clearly vary depending on the number of documents returned. Returning just one document will result in a high precision (assuming that document is relevant), but at a low recall. The converse will be true in the limit that the whole collection is returned. To indicate this trade off, precision–recall curves are used, showing how the two quantities vary with the number of documents returned. The performance can be further summarised by computing the average precision after each relevant document is returned

$$\text{Average precision} = \frac{1}{N} \sum_{n \in \mathcal{R}} \text{Precision}_n \tag{4.26}$$

Alternatively, one can consider the precision after a fixed number of documents have been returned. Many conventional search engines display ten documents per page, and it is likely that the user will not ask for a second page to be displayed, but will rather refine their search if a relevant document has not been found. Precision after ten documents is therefore the measure which was chosen in this work.

4.4 Results

4.4.1 Precision experiments

Average precision results for the various methods are shown in Figure 4.3, and precisions at 10 documents are shown in Figure 4.4. In all cases, α_1 was held fixed at 1,000 while α_2 was allowed to vary. Results for BM-25 (see Section 4.2.2) are shown too as horizontal dashed lines. The default values were used for the BM-25 parameters: $k_1 = 1.2$, $k_3 = 7$ and $b = 0.75$. No document length normalisation was performed, so the k_2 parameter is not required.

In all measures with the exception of average precision using the Cranfield data set, the hierarchical method gives the highest performance. In all of the experiments using the TREC data set the hierarchical method was able to out-perform BM-25, both in terms of the average precision and the precision at ten documents.

Cranfield:
1400 docs
225 queries

TREC 7:
528,155 docs
50 queries

TREC 8:
528,155 docs
50 queries

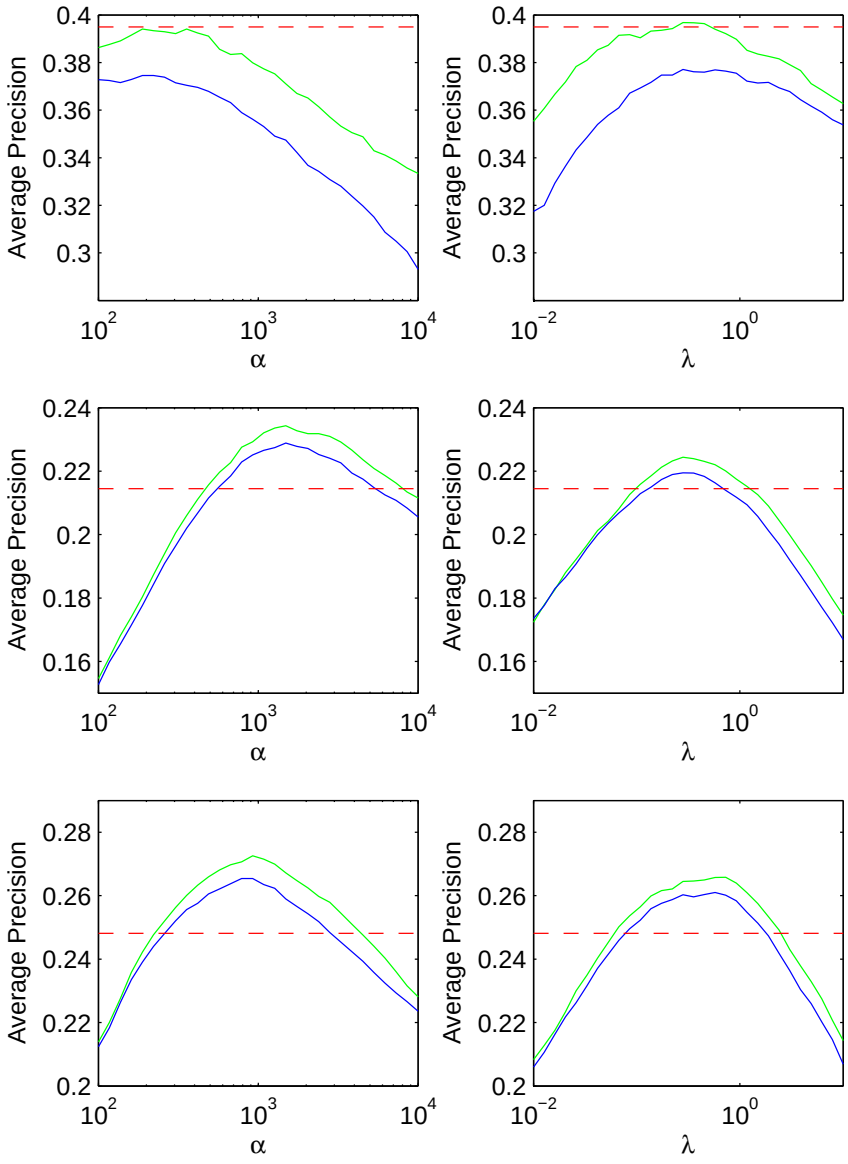

Figure 4.3: Average precision results. Left: the hierarchical Dirichlet model described in Section 4.3.1 (green) and Dirichlet smoothing with the maximum likelihood collection model (blue) as a function of the Dirichlet parameter α. Right: mixture models with the document frequency distribution (green) and the maximum likelihood collection model (blue) as a function of the mixing ratio λ. Results for BM-25 are shown as dashed horizontal lines.

Cranfield:
1400 docs
225 queries

TREC 7:
528,155 docs
50 queries

TREC 8:
528,155 docs
50 queries

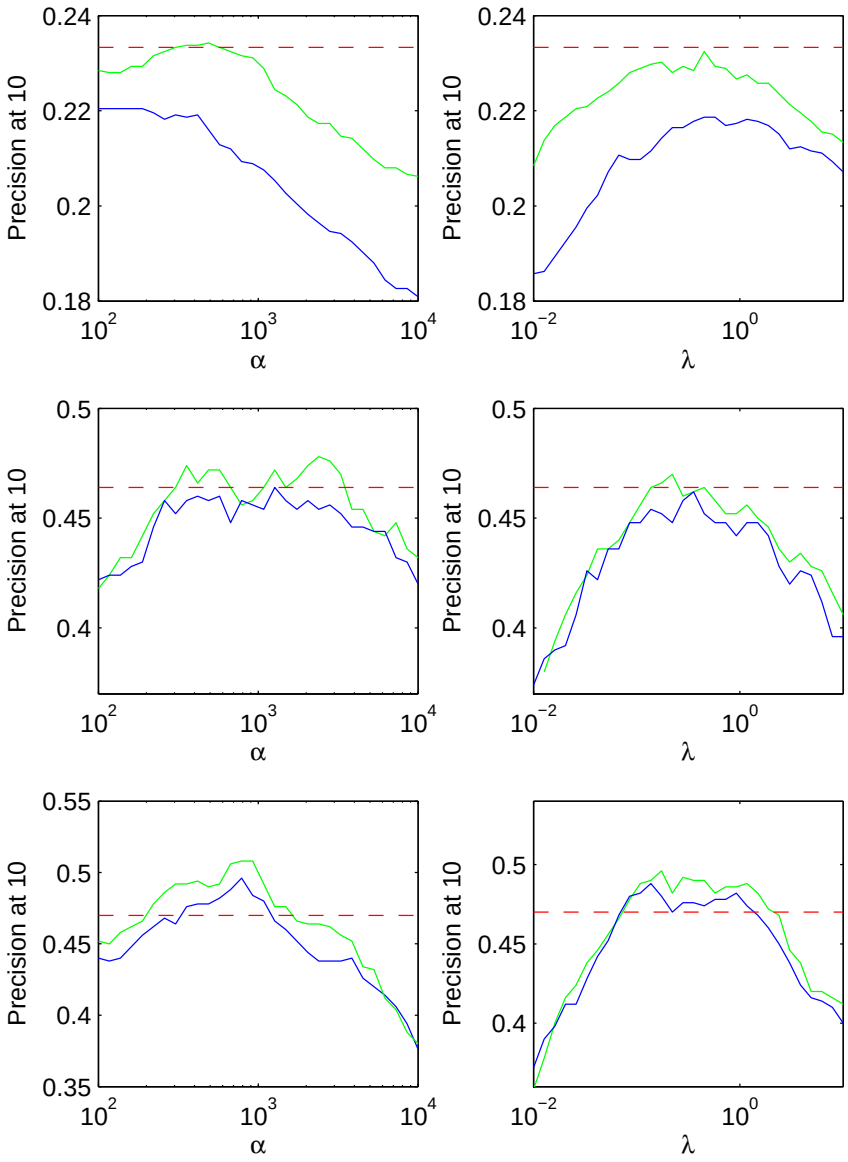

Figure 4.4: Mean precision results at 10 documents. Left: the hierarchical Dirichlet model described in Section 4.3.1 (green) and Dirichlet smoothing with the maximum likelihood collection model (blue) as a function of the Dirichlet parameter α. Right: mixture models with the document frequency distribution (green) and the maximum likelihood collection model (blue) as a function of the mixing ratio λ. Results for BM-25 are shown as dashed horizontal lines.

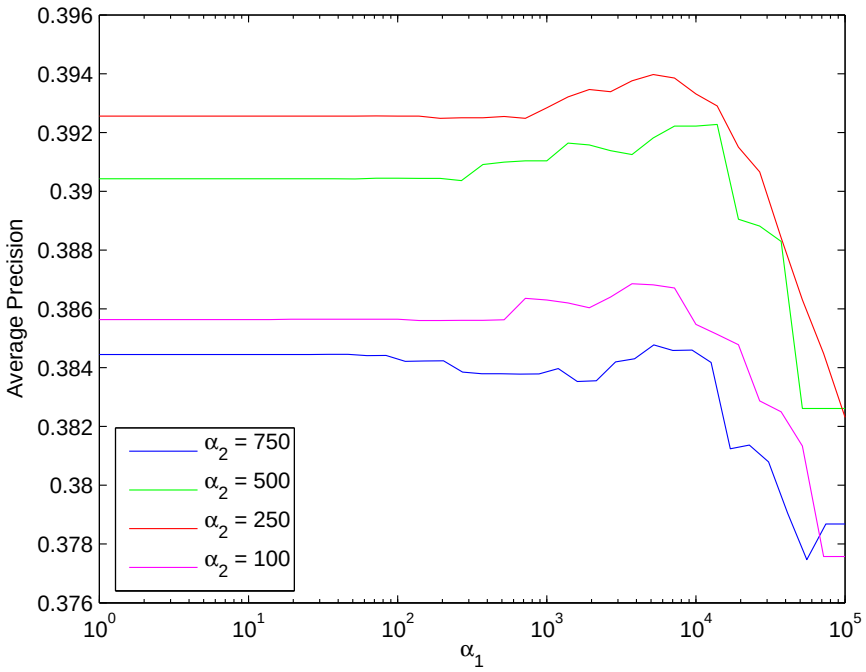

Figure 4.5: Variation of the average precision for the Cranfield data set as a function of α_1. Curves are shown for a variety of values of α_2.

4.4.2 Effect of varying α_1

The experiments described in the previous section kept α_1 fixed whilst varying α_2. As α_1, which describes the probability of escaping from the oracle to the top level prior, is expected to have less of an effect on the predictions, this restriction is reasonable, but it is of course valuable to see what happens if α_1 is varied. To evaluate the effects of this parameter, we calculated the average precision for the Cranfield data set as both α_1 and α_2 were varied. The results are shown in Figure 4.5. Over a wide range of values, between approximately 1 and 10^3, no change is seen due to changes in α_1. Between 10^3 and 10^4 a small improvement in performance is observed, and above this range the performance drops off sharply.

4.4.3 Precision–recall curves

To get a fuller picture of the performance of the new method, precision–recall curves were plotted. Curves for the TREC-7 and -8 query sets are shown in Figures 4.6 and 4.7, using $\alpha_1 = 750$ and $\alpha_2 = 1250$, with BM-25 (using the default parameters as described above) and the Twenty One model (a mixture model using document frequencies as the base distribution and $\lambda = 17/3$) shown for comparison. Although the curves are very close, careful examination shows that BM-25 performs well at low recall, while the Twenty One model gives slightly better performance towards the tail end of the results, where recall is high. The hierarchical

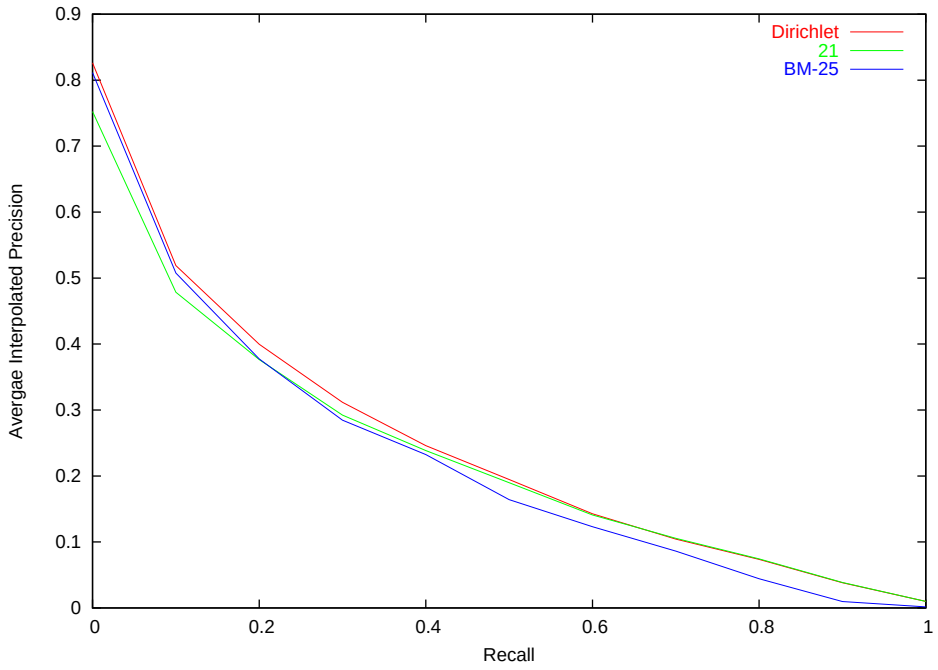

Figure 4.6: Precision–recall curve for the TREC 7 query set.

Sections	Abbreviation	Mean Length	
		TREC7	TREC8
Title	T	2.44	2.44
Title + Description	TD	16.5	16.2
Title + Description + Narrative	TDN	57.0	51.3

Table 4.1: The three query sets used to investigate the effect of query length, with mean number of terms per query.

model however fits the envelopes of the two curves, matching the better performance at both extremes.

4.4.4 Variation of query length

The TREC query sets include three sections for each query: title, description and narrative. By combining these sections it is possible to create queries of varying length and thus investigate the effect on retrieval performance (see Table 4.1). We investigated this effect for the three models described in the previous section. For this experiment the parameters of the hierarchical Dirichlet model were held fixed at $\alpha_2 = 1250$ and $\alpha_1 = 750$. Results for experiments varying the query length are shown in Figure 4.8, with numerical results for the TDN query set given in Table 4.2. As would be expected, the results show that retrieval performance increases with the length of the query. The hierarchical Dirichlet model gives the highest performance in all cases.

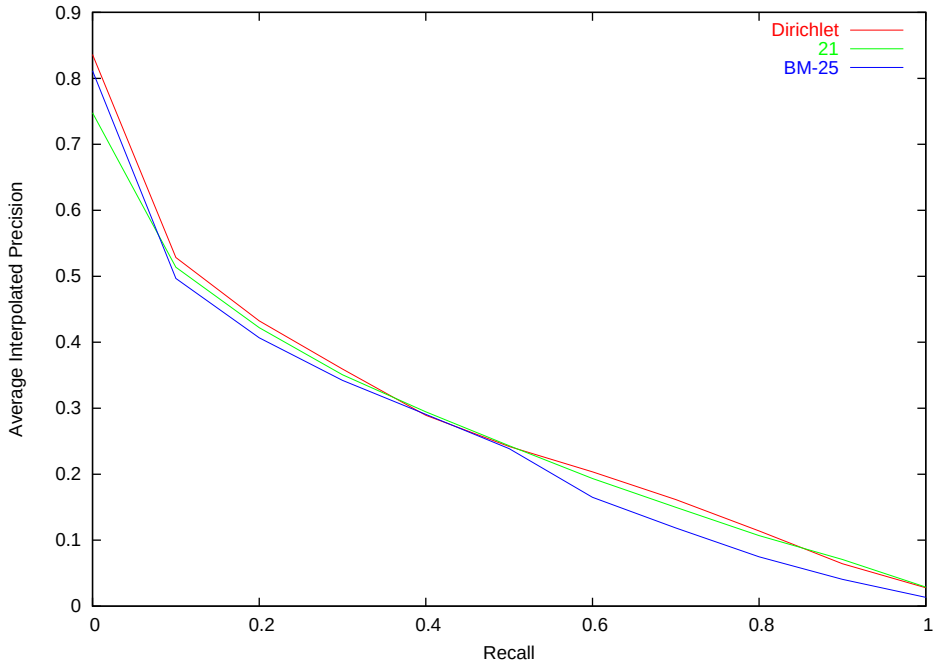

Figure 4.7: Precision–recall curve for the TREC 8 query set.

	TREC-7	**TREC-8**
BM-25	0.21	0.25
Twenty-One	0.22	0.26
Dirichlet	*0.23*	*0.27*

Table 4.2: Average non-interpolated precision scores over top 1000 documents for TREC-7 and TREC-8 *ad hoc* tasks. The Dirichlet results used $\alpha_2 = 1250$ and $\alpha_1 = 750$. BM-25 used $k_1 = 1.2$, $k_3 = 7$ and $b = 0.75$ and the Twenty-One model used $\lambda = 17/3$.

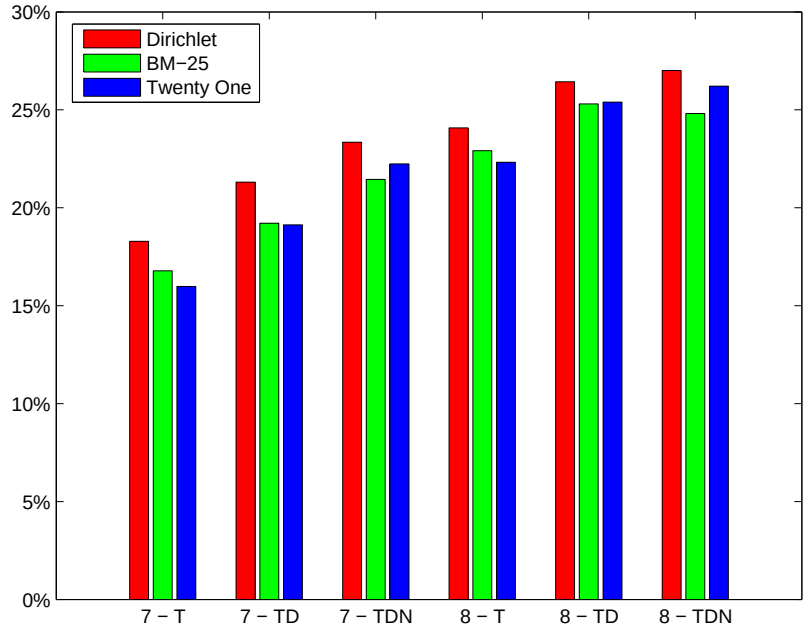

Figure 4.8: Summary of results for the TREC 7 and 8 query sets.

4.5 Deeper hierarchies and passage retrieval

This section extends the retrieval framework developed above to passage retrieval. The goal in passage retrieval is to extract particular passages from those documents which are especially relevant. At the most basic level, this is of use in identifying sections of the document to present to the user when displaying the results of the query. Furthermore, there is evidence that this approach can improve search results [39]. As many documents span multiple topics or contain large amounts of 'boilerplate' text such as menus in hypertext documents, legal disclaimers and so on, it is understandable why passage retrieval is worth pursuing.

The notion of a passage is not well defined, and several approaches have been used in the past. Typically passages are either author-specified, corresponding to sentences, paragraphs, marked sections in the text and so on, or are generated automatically, for example with a fixed length. Passages may or may not overlap. Approaches have also been tried which aim to consider all regions of the text as potential passages. For further background in this area, see [53], [39] and [16], as well as references therein.

Previous evaluations of language model based passage retrieval [53] have treated each passage as a separate document. In this section, the fact that passages belong to documents is made explicit by extending the Dirichlet model to include another level in the hierarchy. In the extended model a distribution over terms, $\boldsymbol{q}_{jk}$, is maintained for passage k in document j. These distributions are generated from a Dirichlet distribution which is shared by all passages

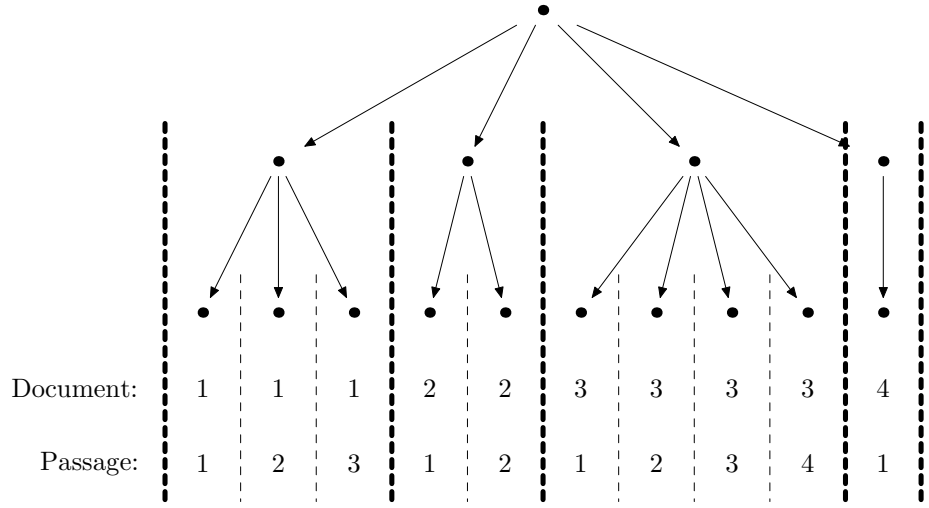

Figure 4.9: Schematic representation of the three-level model used for passage retrieval, showing how distributions in the hierarchy are allocated to passages. Each circle represents a distribution over terms. The distributions for each of the passages in a document share a common parent distribution. These distributions, of which there is one per document, share a single global distribution for the whole collection.

in a given document, and the parameter for that distribution, $\alpha \boldsymbol{q}_j$ is based on a Dirichlet shared by the whole collection as was previously the case.

$$\begin{aligned} \boldsymbol{p} &\sim \text{Dirichlet}\left(\alpha_1 \boldsymbol{u}\right) && (4.27)\\ \boldsymbol{q}_j &\sim \text{Dirichlet}\left(\alpha_2 \boldsymbol{p}\right) && (4.28)\\ \boldsymbol{q}_{jk} &\sim \text{Dirichlet}\left(\alpha_3 \boldsymbol{q}_j\right) && (4.29)\\ t_l &\sim \text{Categorical}\left(\boldsymbol{q}_{d_l p_l}\right) && (4.30) \end{aligned}$$

The model is equivalent to that shown in Figure 2.10 as a Bayesian network, with J indexing documents, K indexing passages within the documents and L indexing terms within each passage. See Figure 4.9 for a more schematic illustration of the model.

Using the minimal oracle approximation and ignoring any dependency of the predictive distribution on previously observed query terms as before, the probability of a query term t is given by

$$\Pr\left(t \mid d, p\right) = \frac{1}{\alpha_3 + N_p}\left(n_t\left(t,p\right) + \alpha_3 \frac{1}{\alpha_2 + N_d}\left(n_p\left(t,d\right) + \alpha_2 \frac{1}{N + \alpha_1}\left(n_d\left(t\right) + \frac{\alpha_1}{|\mathcal{V}|}\right)\right)\right) \quad (4.31)$$

where $n_t\left(t,p\right)$ is now the term frequency *within the passage*, $n_p\left(t,d\right)$ is the *passage frequency*, defined as the number of passages in document d containing the term at least once and $n_d\left(t\right)$

is the document frequency as before. N_p, N_d and N are the sums of these quantities over all terms in the vocabulary. Defining $\hat{p}(t)$ in a similar way to before,

$$\hat{\mathrm{p}}(t) = \frac{n_d(t) + \frac{\alpha_1}{|\mathcal{V}|}}{N + \alpha_1} \tag{4.32}$$

(4.31) can be re-arranged to give

$$\Pr(t \mid \boldsymbol{d}, \boldsymbol{p}) = \frac{1}{\alpha_3 + N_p} \frac{1}{\alpha_2 + N_d} \left(\frac{n_t(t,p)(N_d + \alpha_2) + \alpha_3 n_p(t,d)}{\alpha_2 \alpha_3 \hat{p}(t)} + 1 \right) \alpha_2 \alpha_3 \hat{p}(t) \tag{4.33}$$

Taking logarithms and ignoring terms which do not depend on the document, we obtain

$$\log \Pr(t \mid \boldsymbol{d}, \boldsymbol{p}) = \log \frac{1}{\alpha_3 + N_p} + \log \frac{1}{\alpha_2 + N_d} + \log \left(\frac{n_t(t,p)(N_d + \alpha_2) + \alpha_3 n_p(t,d)}{\alpha_2 \alpha_3 \hat{p}(t)} + 1 \right) \tag{4.34}$$

Finally, this can be summed over all terms in the query to give

$$R(\boldsymbol{p}, \boldsymbol{q}) = N_q \left(\log \frac{1}{\alpha_3 + N_p} + \log \frac{1}{\alpha_2 + N_d} \right) + \sum_i \log \left(\frac{n_t(t,p_i)(N_d + \alpha_2) + \alpha_3 n_p(t,d_i)}{\alpha_2 \alpha_3 \hat{p}(t_i)} + 1 \right) \tag{4.35}$$

As before, the relevance score can be interpreted as the sum of a document length normalisation term and a set of weights for each term in the query. However, to take into account the additional layer, the forms of these terms have changed.

4.5.1 Query term weighting

The most significant change in the form of the query term weights is that they now longer depend not only on the frequency of occurrence within the passage, but also on the number of times which the term occurs in the whole document. Consider a query for information on the Jaguar make of sports car (as opposed to the animal). A typical document matching this query might consist of reviews of a number of different cars from various manufacturers. The fact that the document is about cars rather than animals would most probably be clear from the introduction to the page, and would not necessarily be reiterated in each passage. This contextual information would be missed if the passage retrieval algorithm considered only terms in the passage itself, but is captured here using references to the whole document.

4.5.2 Document length normalisation

The change to this term is simply that it now takes into account the length of the whole document, not just the passage. As the score function now takes into account terms appearing

anywhere in the whole document, it is arguably appropriate that its length should be included too.

4.5.3 Implementation

The quantities required for this score function are not fundamentally different to those required for the document level algorithm. In essence, two indices are required, one storing counts at the document level and one at the passage level. Under the new score function documents which do not contain a given query term are still assigned zero weight, so an inverted index can be used to rapidly score a collection. Passages within a document containing a given term will still be assigned a non-zero weight even if that term does not appear in the passage itself. We must therefore use the document-level inverted index to find all documents containing each term in the query, then to calculate the score for each passage in that document. As the number of passages in a document is usually relatively small it is still possible to score the whole collection in a reasonable time.

4.5.4 Whole document retrieval

One way of measuring the performance of a passage retrieval method is to perform full document retrieval. While this does not represent the whole story in terms of the abilities of these methods, it does provide a qualitative way to compare different methods. There are a number of ways in which a passage retrieval method can be used to score full documents. We evaluated the following two methods:

- **Most Probable Passage:** Retrieval scores are calculated for all passages in a document, the largest of which is used as the score for the whole document. This approach is similar to that used in [16].
- **Overall Probability:** As the retrieval scores have an interpretation as log probability that the selected passage is relevant (up to a constant offset), and the method fundamentally assumes that precisely one passage is relevant, it is possible to define a score

$$R(d,q) = \log\left(\sum_{p\in d}\exp\left(R\left(p,q\right)\right)\right) \tag{4.36}$$

  which evaluates the probability that any one of the passages in a given document is relevant.

Results for these two approaches on the Cranfield corpus, with sentences being used for the passages, are shown in Figures 4.10 and 4.11. The performance of these two methods is comparable (the 'most probable passage approach' gave 23.6% precision at ten document, compared with 23.3% for the 'overall probability' approach). However, the 'overall probability' approach shows a sharper drop-off as α_3 and α_2 are increased.

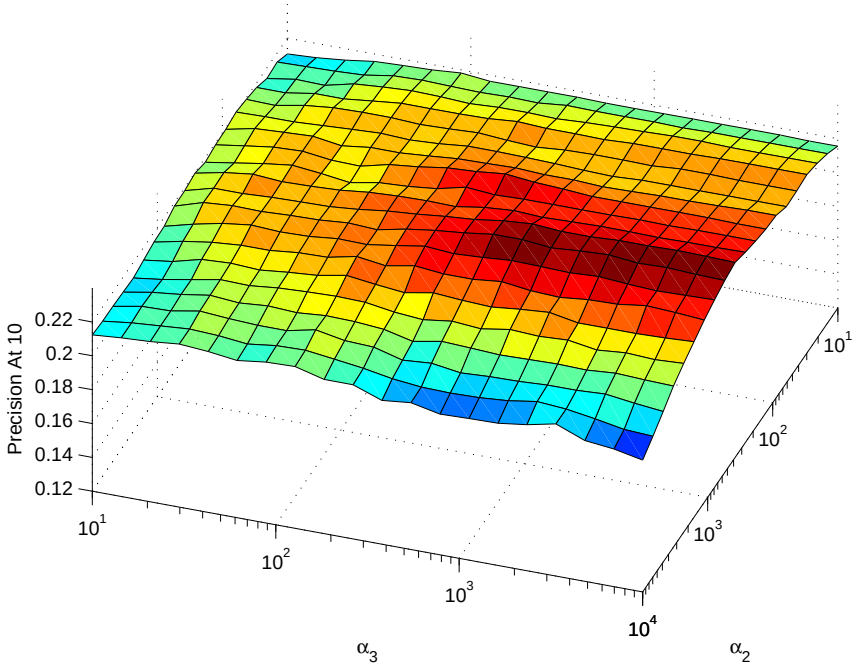

Figure 4.10: Passage retrieval scores for the 'most probable passage' approach.

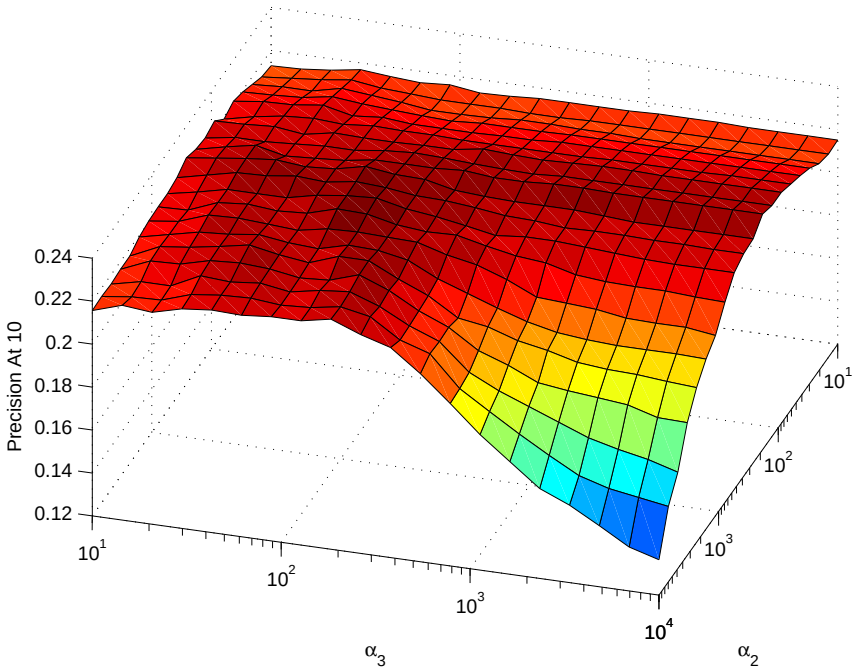

Figure 4.11: Passage retrieval scores for the 'overall probability' approach.

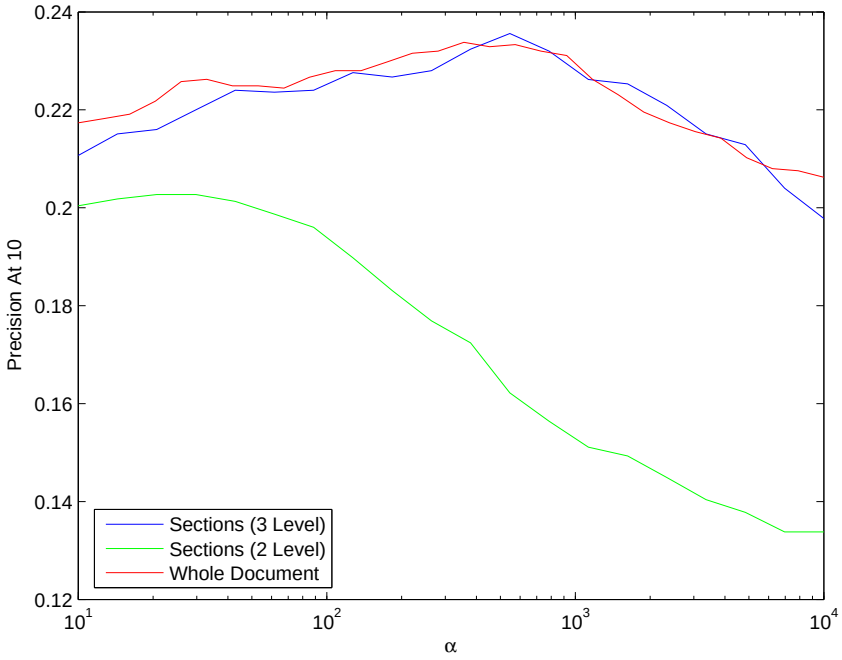

Figure 4.12: Comparison of passage retrieval scores using various methods

Figure 4.12 shows a comparison of three techniques: document retrieval using the method described above with 'most probable passage' scoring, a similar approach using the 2 level model (i.e. treating each passage independently, as if they were separate documents), and whole document retrieval using the approach described in Section 4.3. The results show that the three level approach clearly outperforms the two level approach at the passage level. The whole document approach does give a higher performance overall, but it should be noted that this may depend on the choice of passage types, and that previous studies have found that better performance can be achieved using overlapping passages [16].

4.5.5 Passage selection

A qualitative insight into the performance of the retrieval method can be obtained by looking at the passages which are assigned the highest scores within each of the returned documents. This information may be used to provide a summary of the documents returned, aiding the user in deciding which documents to pursue further. Figures 4.13 and 4.14 show the top ten documents returned for two selected queries from the Cranfield data set, together with summaries generated from the most highly ranked passages within those documents.

In many cases the summaries returned by this method are more informative than the titles of the documents themselves. For example, documents 484 and 578, returned in response to query 3 as shown in Figure 4.13. Neither mention 'composite slabs' in their titles, but do in the returned passages. The ability to return this sort of information is likely to aid the user

Query 3: *What problems of heat conduction in composite slabs have been solved so far.*

Document: 4 (*)
Title: one-dimensional transient heat conduction into a double-layer slab subjected to a linear heat input for a small time internal
Summary: analytic solutions are presented for the transient heat conduction in composite slabs exposed at one surface to a triangular heat rate

Document: 143 (*)
Title: heat flow in composite slabs
Summary:this paper presents the solution of the heat flow problem in composite walls under heat transfer conditions which are typical of uncooled rocket engine walls

Document: 484 (*)
Title: similarity laws for aerothermoelastic testing
Summary: the temperature is determined as a function of position and time in the case of linear heat conduction in a composite slab of ture throughout and the two external surface temperatures are considered to be prescribed functions

Document: 398 (*)
Title: conduction of heat in composite slabs
Summary: conduction of heat in composite slabs

Document: 90 (*)
Title: periodic temperature distribution in a two-layer composite slab
Summary: periodic temperature distribution in a two-layer composite slab

Document: 578
Title: new thermo-mechanical reciprocity relations with application to thermal stress analysis
Summary: this is illustrated by application to one dimensional problems of heating of a homogeneous or composite slab and directly verified by classical methods in the appendix

Document: 89 (*)
Title: periodic temperature distributions in a two layer composite slab
Summary: periodic temperature distributions in a two layer composite slab

Document: 541
Title: the stacking of compressor stage characteristics to give an overall compressor performance map
Summary: radiation cooling due to fourth power radiation from semi infinite solids and finite slabs together with radiation according to newton s law of cooling is then treated

Document: 180 (*)
Title: some problems on heat conduction in stratiform bodies
Summary: some problems on heat conduction in stratiform bodies

Document: 5 (*)
Title: one-dimensional transient heat flow in a multilayer slab .
Summary: one-dimensional transient heat flow in a multilayer slab .

Figure 4.13: Titles and summaries of returned documents for query 3 in the Cranfield data set. Documents marked (*) were marked as relevant in the data set.

Query 156: *what qualitative and quantitative material is available on ablation materials research*

Document: 1095 (*)
Title: an experimental investigation of ablating material at low and high enthalpy potentials .
Summary: qualitative measurements of the effective heats of ablation of several materials in supersonic air jets at stagnation temperature up to 11 000 f...

Document: 1096 (*)
Title: a theoretical study of stagnation point ablation .
Summary: comparisons of the results for several materials tested at the higher heating rates showed graphite to have the lowest ablation rate of all materials tested

Document: 552 (*)
Title: ablation of glassy materials around blunt bodies of revolution .
Summary: ablation of glassy materials around blunt bodies of revolution .

Document: 1064 (*)
Title: plastic stability theory of geometrically orthotropic plates and cylindrical shells .
Summary: the analytical predictions upon inclusion of the pertinent material property values should be applicable to other materials as well as teflon

Document: 1097 (*)
Title: an analytical investigation of ablation .
Summary: an experimental investigation of ablating material at low and high enthalpy potentials

Document: 1098 (*)
Title: a sensor for obtaining ablation rates .
Summary: the most significant result of the analysis is that the effective heat capacity of the ablation material increases linearly with stream enthalpy

Document: 1099 (*)
Title: a five-stage solid fuel sounding rocket system .
Summary: the predicted equilibrium surface temperatures on nonablating surfaces behind an ablating material were in agreement with the values derived from tests conducted with inconel cylinders having teflon hemispherical nose pieces

Document: 81 (*)
Title: theoretical investigation of the ablation of a glass-type heat protection shield of varied material properties at the stagnation point of a re-entering irbm .
Summary: theoretical investigation of the ablation of a glass type heat protection shield of varied material properties at the stagnation point of a re entering irbm

Document: 1278
Title: turbulent heat transfer on blunt-nosed bodies in two-dimensional and general three-dimensional hypersonic flow .
Summary: in this study of hypersonic ablation the pertinent conservation equations are derived and the simultaneous processes of diffusion convection...

Document: 1116
Title: stability of orthotropic cylindrical shells under combined loading .
Summary: the increasing use of fiber and whisker reinforced materials makes necessary the availability of methods of analyzing cylinders and cones composed of an orthotropic material

Figure 4.14: Titles and summaries of returned documents for query 156 in the Cranfield data set. Documents marked (*) were marked as relevant in the data set.

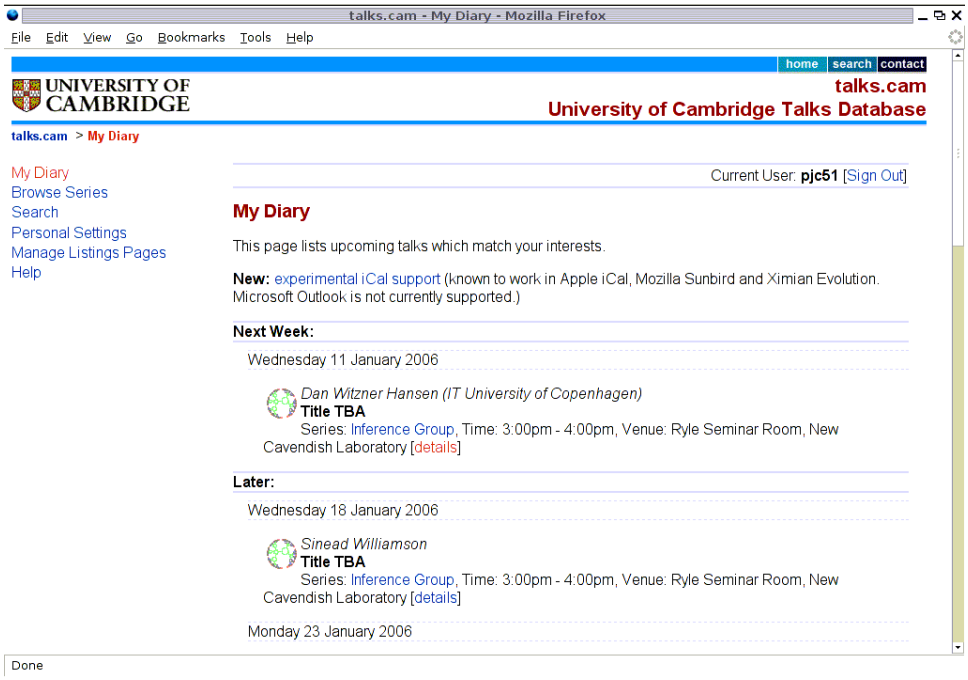

Figure 4.15: Screen capture of the 'My Diary' section of *talks.cam*. This page shows a personalised listing of future events, based on series to which the user has subscribed. Alternatively, the user may browse a directory of all events, or search using the facility described below.

in determining the relevance of the document without having to read it in full.

4.6 Application to a seminar database

Large academic institutions such as the University of Cambridge typically host hundreds of seminars and similar events each year. Currently, these are usually advertised through a combination of departmental web sites, postings on notice boards and email lists. The lack of a unified mechanism for advertising events results in an information overload for many people and sets up a barrier to inter-disciplinary interaction.

To address these problems, we created the *talks.cam* system [111], a web-based database of talks and seminars taking place in the University of Cambridge. The aim of the service is to provide a central location for listing such events, allowing users of the service to construct their own diaries containing events which are of interest to them (see Figure 4.15). In approximately 1 year, details of 350 talks were entered covering 35 seminar series.

In order to allow users to find events, a search facility was implemented using the information retrieval method described in Section 4.3. Each document in the index represents a single event, and consists of the title and abstract of the seminar as well as the name of the speaker. Porter stemming and stop-word removal were used as in the examples described above. The search facility allows searches to be restricted to talks occurring in the future or in the past.

4.7 Conclusions

This chapter has developed a text retrieval framework based on the hierarchical Dirichlet model introduced in Chapter 2. The model is different in a number of ways to existing techniques. Most importantly, the approach provides justification for the use of inverse document frequency statistics, which until now have been either absent or introduced as a heuristic in the language modelling approach to information retrieval. The suggested method was shown to perform well, giving better retrieval performance than existing methods on a number of test tasks.

The hierarchical model was further extended to the task of passage retrieval. This results in a score function for passages which not only considers the appearance of terms from the query in the passage itself, but also their appearance in the wider context of the whole document. The new passage retrieval method shows promising performance when compared to other approaches to this problem.