A Query-based Pre-retrieval Model Selection Approach to Information Retrieval

Ben He and Iadh Ounis

Department of Computing Science
University of Glasgow
Glasgow G12 8QQ
United Kingdom
{ben, ounis}@dcs.gla.ac.uk

Abstract

In this paper, we propose a query-based pre-retrieval approach to the model selection problem, which automatically selects the best-performing retrieval model before the retrieval process takes place. In this approach, the queries are clustered according to their statistics and the best-performing retrieval model is associated to each cluster. For a given new query, we assign the closest cluster to the query, and then we apply the model associated to the cluster. We evaluate the model selection approach on the disk1&2 of the TREC collections. The results show that our model selection approach achieves stable performance, which could outperform the use of the optimal retrieval model indifferently for each query. The results also show that, interestingly, a retrieval model provides consistent performance for queries belonging to the same cluster.

1 Introduction

An information retrieval (IR) system receives a query from the user and returns the supposedly relevant documents. The set of relevant documents are determined by a proper retrieval model. Generally, the documents are ranked by their relevance scores, which are given by a weighting scheme [vR79]. Therefore, the retrieval model is usually heavily correlated with the notion of the weighting scheme. As a consequence, the effectiveness of a weighting scheme to discriminate the informative terms from the common terms, in a given document, has a strong impact on the quality of the retrieved set of documents.

In the recent decades, many retrieval models for IR have been developed from various perspectives (e.g. the models based on the $tf \cdot idf$ weighting scheme [Sal71], the BM25 formula [RWB98] and the language modelling approach [PC98,ZL02]). For a given collection and a given query, it is an interesting and challenging problem to automatically select the best retrieval model, which would provide the best retrieval performance. This problem is referred to as the *model selection* problem.

The key issue of the model selection problem is to assess the retrieval models. If we could accurately estimate the quality of each candidate retrieval model, then we can select the retrieval model(s) accordingly. Recently, there have been some efforts to tackle the assessment of the retrieval models. In [JFH01], Jin et al. proposed an automatic retrieval model evaluation method by computing the eigenvalues of the document-document matrix. In their work, each document is represented by a vector of terms, where each term weight is determined using a specific weighting model. Then, for a given collection, the models are evaluated according to the eigenvalues of the eigenvectors of the document-document matrix. In Manmatha et al's work [MRF01], the quality of a retrieval model is given by the probability of relevance. They combine the outputs of different retrieval models by modelling the score (i.e. the relevance score of the documents) distributions. The score distribution is modelled as a normal distribution for the relevant documents and an exponential distribution for the non-relevant documents. For a query where there is no relevance information available, the posterior probability $p(Rel|score)$, the probability of relevance of the document given the score of a retrieval model, is approximated by a Bayesian formula. Moreover, in a distributed information retrieval environment, Luo & Callan [LC02] assess the retrieval models by merging the results of different retrieval models using a regression model.

The three approaches described above are based on the analysis of the relevance scores given by the retrieval models. Therefore, using these approaches, the system cannot select the optimal model prior to the retrieval process.

On the contrary, our approach to the model selection is a pre-retrieval mechanism. For a given query, it automatically selects a retrieval model without the need to wait for the system's relevance scores.

Our work for the model selection problem is based on Amati & van Rijsbergen's probabilistic modular framework [Ama03,AvR02a]. The framework deploys more than 50 Divergence From Randomness (DFR) models for term weighting, including the widely tested I(n_exp)C2 and PL2 [ACR02,POAvR03] retrieval models. However, for a given retrieval task, the framework does not have a strategy to single out a model that would provide the best performance. Tables 1 and 2 list the mean average precision (MAP) obtained by different models on the TREC-7, 8 ad-hoc tasks[1] respectively. Here we just list the results given by the most stable and effective models in Amati & van Rijsbergen's framework. We can see that even on the same collection, the optimal model for each task could be different. For both tasks, the best model achieves approximately 5% higher MAP than the poorest one. If we empirically apply the optimal model on the TREC-7 ad-hoc task, i.e. I(n_exp)C2, on TREC-8 ad-hoc task, we will not achieve the optimal performance on this task (see Table 2). Indeed, it is usually not efficient to use a unique retrieval model across different retrieval tasks [POAvR03].

The purpose of this paper is to propose a query-based pre-retrieval approach to the selection of the most appropriate retrieval model. For a given query and a given collection, we aim to automatically select the best-performing retrieval model.

The remainder of this paper is organised as follows. Section 2 describes in details the proposed query-based pre-retrieval model selection approach. Section 3 introduces the experimental setting we use to evaluate the proposed approach. In Section 4, the evaluation results are listed and analysed. Finally, Section 5 concludes our work and provides directions for future research.

[1] The two retrieval tasks are based on the same TREC collection. Related information about the TREC collections can be found on the following web page: http://trec.nist.gov/data/intro_eng.html

Table 1. The mean average precision (MAP) on the TREC-7 ad-hoc task using the different retrieval models.

Model	MAP	Model	MAP
I(F)B2	0.1985	PL2	0.1894
I(n_exp)L2	0.1937	**I(n_exp)C2**	**0.2001**
PB2	0.1906	BB2	0.1985
I(n_exp)B2	0.1986	I(n)B2	0.1987
I(n)L2	0.1958	BL2	0.1932
I(F)L2	0.1933		

Table 2. The mean average precision (MAP) on the TREC-8 ad-hoc task using different retrieval models.

Model	MAP	Model	MAP
I(F)B2	0.2623	PL2	0.2571
I(n_exp)L2	0.2611	*I(n_exp)C2*	*0.2637*
PB2	0.2526	BB2	0.2632
I(n_exp)B2	0.2630	**I(n)B2**	**0.2649**
I(n)L2	0.2626	BL2	0.2608
I(F)L2	0.2607		

2 Query-based Model Selection

As stressed in the previous section, the current approaches assess the retrieval models, once the different involved models are compared based on the analysis of their relevance scores. This requires running the candidate retrieval models on each query before detecting the most appropriate one(s). As a consequence, such approaches are not very practical, especially in an interactive retrieval setting.

In this paper, we propose a query-based pre-retrieval model selection mechanism that assesses/selects the retrieval model(s) without knowing the set of retrieved documents, nor their associated scores.

To introduce our model selection approach in details, we start by motivating the approach in Section 2.1 and describing the involved query clustering process in Section 2.2. In Section 2.3, we provide a summary of the proposed model selection methodology.

2.1 Motivation: Outputs vs Intrinsic Features

The underlying idea of our approach is that the choice of the optimal retrieval model depends on the statistical characteristics of the query rather than on its output. Therefore, we assume that the best retrieval performance for queries having similar statistical features, would be achieved by the same retrieval model. In other words, the statistical features of a query could constitute a good indication for the model selection decision mechanism, and we might be able to discriminate various types of queries according to their statistical features.

If the queries could be categorised into groups, where the best-performing retrieval model could be identified for each group, then, selecting the best-retrieval model for a new query would mean identifying the closest group that shares its statistical features. This idea assumes that a retrieval model provides consistent performance for queries in the same cluster. In the experimental section of this paper, we will show that this assumption holds (see Section 4).

Therefore, our approach involves a training process where the queries are clustered according to their statistical characteristics, and the best retrieval model for each cluster of queries is obtained by taking previous relevance judgements into consideration. After the training process, for a given new query, we assign a cluster to the query according to its statistics, and then trigger the best-performing model associated to the assigned cluster.

A possible approach to clustering the queries is to take the users' feedback into account and cluster together the queries for which the users have visited similar documents [WNZ02]. However, since the retrieved documents are only known after the retrieval process, this approach is not appropriate for our pre-retrieval model selection mechanism.

Therefore, in the following section, we propose a query clustering method that is independent of the retrieval procedure. We provide the query features that could be used in the clustering process, and motivate their use.

2.2 Query Clustering

In order to have a query clustering method that is independent of the use of the users' feedback and/or the relevance scores, we need to find those features that could describe the natural characteristics of a query, and that are not affected by the retrieval process. Then, a query clustering process based on these features will be proposed.

For this purpose, we propose to study the statistics of the queries. For each query, we construct a feature vector, and then cluster the queries according to the similarity of each vector pair. The underlying problem of this approach is the right choice of the features required to faithfully represent a query.

Following the works that have been done in the language modelling approach to information retrieval [ZL01,CTZC02], and the previous experiments with various retrieval models and length normalisation approaches in Amati & van Rijsbergen's framework [Ama03,HO03a], we propose the following three features, on which the queries are clustered:

- *The query length*
  According to Zhai & Lafferty's work [ZL01], in the language modelling approach, the query length has a strong effect on the smoothing methods. In our previous work, we also found that the query length heavily affects the length normalisation methods of the probabilistic models [HO03a].
  For example, the optimal setting for the length normalisation 2 in Amati & van Rijsbergen's probabilistic framework is query-dependent [AvR02a]. Indeed, the empirically obtained setting of its parameter c is $c = 7$ for short queries and $c = 1$ for long queries, suggesting that the optimal setting depends on the query length. Therefore, the query

length could be an important characteristic of the queries. In this paper, we define the query length ql as the number of non-stop words in the query.

- *The relative informative amount carried in each query term*

  In general, each term is associated with an inverse document frequency ($idf(t)$) describing the informative amount that a term t carries. The $idf(t)$ factor is a decreasing function of the number of the documents containing the given query term t. It is widely used in IR (e.g. the $tf \cdot idf$ formula in the vector space model [Sal71]). In this work, we propose to use the distribution of the informative amount in the query terms as a factor characterising a query.

  In our approach, we use the quotient of the minimum idf among the query terms divided by the maximum idf among the query terms as an important statistical factor (γ) intrinsic to a given query:

  $$\gamma = \frac{\log(n_{t,max}/N)}{\log(n_{t,min}/N)}$$

  where n_t is the number of documents containing a particular query term t. $n_{t,max}$ and $n_{t,min}$ are the maximum and minimum n_t among the query terms respectively. N is the number of documents in the whole collection.

  For example, assume that a query is "information and retrieval", and the idf of the terms are ranked as $idf(retrieval) > idf(information) > idf(and)$. Then, the difference of the informative amount among the terms is extracted as

  $$\gamma = \frac{idf(and)}{idf(retrieval)}$$

  Other factors, including the mean, the standard deviation or the variance of the informative amount in the query terms etc., could be used to model the difference of the informative amount in the query terms. However, for this initial study, we believe that the proposed definition is a very good starting point.

- *The clarity/ambiguity of a query*

  When a user is searching on an IR system, the scope of his/her query is an important factor in the retrieval process. For example, a query like "information retrieval glasgow university" may require information about the Glasgow IR group, including staffs, students, publications and so on. Whilst for a query like "homepage of glasgow information retrieval group", the query seems to specifically require the web site of the Glasgow IR group. Therefore, we consider that the latter query is of a higher clarity than the former one. According to the work by Cronen-Townsend et al. [CTZC02], the clarity (or on the contrary, the ambiguity) is an intrinsic feature of a query, which has an important impact on the system performance. Therefore, it could be a factor in our query clustering process. Cronen-Townsend et al. proposed the clarity score of a query to measure the coherence of the language usage in documents, whose models are likely to generate the query. In their definition, the clarity of a query is the sum of the Kullback-Leibler divergence between the probability of generating each term in the vocabulary from the query and from the whole collection.

  Cronen-Townsend et al.'s definition comes from the language modelling approach to IR. A more general indication of the clarity of a query is the size of the document set

containing (at least one of) the query terms. As stressed in [POAvR03], the size of this document set is an important property of the query.
In this work, and following [POAvR03], we use the factor

$$\omega = -\frac{\log(n_Q/N)}{\log N}$$

to represent the clarity of a query, or its scope; where n_Q is the number of documents containing (at least one of) the query terms.
When n_Q is small, we will obtain a large ω value, which implies that the query is very specific.

Taking the above three features into account, each query will be represented by the feature vector $\overline{qf}$ given as follows:

$$\overline{qf} = (\rho \cdot ql, \gamma, \omega)$$

where

- ρ is a parameter. We experimentally set it to 0.2;
- ql is the query length;
- ω can be seen as the normalised *idf* factor for the whole query.

The three proposed features are not exhaustive. However, we believe that they should be able to faithfully characterise a query.

Finally, following the motivation of our approach described in Section 2.1, the feature vectors have to be clustered. In this work, we adopt the CURE algorithm [GRS98] to cluster the feature vectors in the above three-dimensional space. In the CURE algorithm, initially, each vector is an independent cluster. The similarity between two clusters is measured by the cosine similarity of the two closest vectors (having the highest cosine similarity), where the two vectors come from each cluster respectively. If we have n vectors to be processed, we start with n clusters. Then, we merge the closest pair of clusters (according to the cosine similarity measure) as a single cluster. The merging process is repeated until it results in k clusters. Here the number k of clusters is the halting criterion of the algorithm.

2.3 The Model Selection Mechanism

Having introduced a pre-retrieval query clustering method in the previous section, our model selection mechanism, as motivated in Section 2.1, can be summarised as follows:

- We cluster a set of training queries according to their intrinsic features.
- For each cluster, we select the best-performing model in terms of the precision/recall measures.

- Then for a new query, we assign the closest cluster to it and trigger the best-performing model associated to the assigned cluster.

Note that the query clustering procedure is done at the training stage. For a new query, we simply assign a cluster to the query without the need to wait for the relevance scores. Therefore, our approach is quite practical and efficient in terms of computational complexity.

In the following sections, we show how our model selection approach has been evaluated. We describe our experimental setting in Section 3, and provide the evaluation results and the related analysis in Section 4.

3 Experimental Setup

The purpose of our experiments is not only to evaluate our model selection approach, but also to check whether a retrieval model provides consistent performance for queries belonging to the same cluster. The latter will allow us to prove the underlying assumption of our approach (see Section 2.1). Thus, our experiments include two steps, i.e. the training process required by the model selection mechanism, and the evaluation part, which evaluates the model selection approach and verifies its underlying assumption.

In the experiments, our model selection mechanism involves 11 retrieval models developed within Amati & van Rijsbergen's Divergence From Randomness (DFR) probabilistic modular framework [AvR02a]. The models are listed in Table 3, where

Table 3. The retrieval models involved in our experiments.

Model	Formula
BB2	$w(t,d) = \frac{F+1}{n_t \cdot (tfn+1)}(-\log_2(N-1) - \log_2(e) + f(N+F-1, N+F-tfn-2) - f(F, F-tfn))$
BL2	$w(t,d) = \frac{1}{tfn+1}(-\log_2(N-1) - \log_2(e) + f(N+F-1, N+F-tfn-2) - f(F, F-tfn))$
PB2	$w(t,d) = \frac{F+1}{n_t \cdot (tfn+1)}(tfn \cdot \log_2 \frac{tfn}{\lambda} + (\lambda + \frac{1}{12 \cdot tfn} - tfn) \cdot \log_2 e + 0.5 \cdot \log_2(2\pi \cdot tfn))$
PL2	$w(t,d) = \frac{1}{tfn+1}(tfn \cdot \log_2 \frac{tfn}{\lambda} + (\lambda + \frac{1}{12 \cdot tfn} - tfn) \cdot \log_2 e + 0.5 \cdot \log_2(2\pi \cdot tfn))$
I(n)B2	$w(t,d) = \frac{F+1}{n_t \cdot (tfn+1)}(tfn \cdot \log_2 \frac{N+1}{n_t+0.5})$
I(n)L2	$w(t,d) = \frac{1}{tfn+1}(tfn \cdot \log_2 \frac{N+1}{n_t+0.5})$
I(F)B2	$w(t,d) = \frac{F+1}{n_t \cdot (tfn+1)}(tfn \cdot \log_2 \frac{N+1}{F+0.5})$
I(F)L2	$w(t,d) = \frac{1}{tfn+1}(tfn \cdot \log_2 \frac{N+1}{F+0.5})$
I(n_exp)B2	$w(t,d) = \frac{F+1}{n_t \cdot (tfn+1)}(tfn \cdot \log_2 \frac{N+1}{n_e+0.5})$
I(n_exp)L2	$w(t,d) = \frac{1}{tfn+1}(tfn \cdot \log_2 \frac{N+1}{n_e+0.5})$
I(n_exp)C2	$w(t,d) = \frac{F+1}{n_t \cdot (tfn_e+1)}(tfn_e \cdot \log_2 \frac{N+1}{n_e+0.5})$

- $w(t,d)$ is the within-document term weight of the term t in the document d.
- tf is the within-document frequency of the term t in the document d.
- F is the term frequency of the term t in the whole collection.
- N is the number of documents in the collection.
- n_t is the document frequency of the term t.
- n_e is given by:
$$N \cdot (1 - (1 - \frac{n_t}{N})^F)$$
- λ is given by $\frac{F}{N}$ and $F \ll N$.
- The relation f is given by the Stirling formula:
$$f(n,m) = (m + 0.5) \cdot \log_2(\frac{n}{m}) + (n - m) \cdot \log_2 n$$
- tfn is the normalised term frequency. It is given by the *normalisation 2* [AvR02a]:
$$tfn = tf \cdot \log_2(1 + c \cdot \frac{avg\_l}{l}) \tag{1}$$
where c is a parameter. l and avg_l are the document length of the document d and the average document length in the collection respectively.
- tfn_e is also the normalised term frequency. It is given by the modified version of the normalisation 2 [AvR02b]:
$$tfn_e = tf \cdot \log_e(1 + c \cdot \frac{avg\_l}{l})$$

An effective and stable document length normalisation method, i.e. the normalisation 2 (see Equation (1)), is applied in these models. The details of these models and the normalisation 2, can be found in [Ama03,AvR02a].

The only parameter c of the length normalisation method is automatically estimated by our tuning approach, which measures the normalisation effect on the term frequency distribution with respect to the document length distribution [HO03a]. The approach assumes a constant optimal normalisation effect with respect to the change of the within document frequency of the query terms. It assigns the parameter value such that it gives this constant. This tuning approach is applied in all the experiments of this paper.

For our experiments, we use the disk1&2 of the TREC collections as the test collection. This is due to the fact that there are more queries available on this collection (i.e. queries of the TREC-1, 2, and 3 ad-hoc tasks), which allows us to have a larger training query set and a better analysis. Thus, we use 100 queries, i.e. the queries of the TREC1, 2 ad-hoc tasks, as the training query set, and we use 50 queries, i.e. the queries of the TREC-3 ad-hoc task, as the evaluation query set.

Each query consists of 3 fields: title, description and narrative. In all our experiments, we use only the title field. Also, in our experiments, both documents and queries are stemmed, tokens from a standard stop-words list are removed and no query expansion mechanism is applied.

We start by extracting the statistical features of each training query as described in the previous section, such that a feature vector is constructed for each query. The CURE algorithm is then applied in order to group the feature vectors into clusters. In the experiments, we test the clustering process for different threshold settings from $k = 2$ to $k = 10$.

Next, in order to find the optimal retrieval model for each cluster, we run experiments on the training query set to obtain the average precision for each query using the 11 chosen retrieval models. For each cluster of queries, we consider the model providing the highest mean average precision over the cluster as the best-performing model associated to the cluster.

Then, we test our model selection mechanism on the evaluation query set. For each query in the evaluation query set, we assign the closest cluster to it and run the retrieval process using the best-performing model associated to the cluster.

Thus, if the clustering process results in k clusters $\{c_1, \ldots, c_k\}$, for each cluster c_i, we denote the queries in the training set and in the evaluation set belonging to c_i as Q_{T,c_i} and Q_{E,c_i} respectively. We also denote the best-performing model on Q_{T,c_i} as $M_{T,best,c_i}$. If we find that in most clusters, the model $M_{T,best,c_i}$ achieves also the best performance on Q_{E,c_i}, we can conclude that our assumption (see Section 2.1) that a retrieval model provides consistent performance for queries belonging to the same cluster holds.

Moreover, on the evaluation query set, we compare the performance of our model selection approach with the use of a unique optimal model indifferently for each query. Our baseline is the model that provides the highest mean average precision on the training query set. Therefore, the baseline is the strongest model obtained empirically on a large set of queries, i.e. the training query set, which could be seen as a robust baseline.

4 Experimental Results

As introduced in the above section, our experiments start with a training process, in which the training queries are clustered and the average precision on each training query is obtained by using different retrieval models.

Tables 4 lists the MAP (mean average precision) on the training query set using each single retrieval model uniformly for all the queries. PL2 is the best-performing model among the 11 chosen models. Therefore, it is considered as the baseline for the model selection mechanism. Moreover, on the training query set, the best model achieves a clearly higher MAP (8.50%) than the poorest one.

As shown in Table 5, the query clustering results vary with the threshold value k ($k = 2$ to $k = 10$). When the threshold is getting larger, the number of queries in each cluster is getting smaller, since the vectors are distributed in more groups. For all the applied threshold settings, the best-performing models on the clusters generated by the training process include four models, i.e. PL2, I(n_exp)C2 ,PB2 and I(n)B2. Note that as shown in Table 4, according

Table 4. The mean average precision (MAP) achieved by each single candidate model for the *training* query set T. From these results, we select PL2 as our baseline.

Model	MAP_T	Model	MAP_T
PL2	**0.2120**	I(n_exp)C2	0.2111
PB2	0.2084	I(F)B2	0.2061
I(n_exp)B2	0.2053	BB2	0.2037
I(n)B2	0.2022	I(n)L2	0.1991
I(F)L2	0.1970	I(n_exp)L2	0.1967
BL2	0.1954		

to their performance, the first three models are top ranked, while I(n)B2 is the 7th most efficient candidate model.

To check our assumption in Section 2.1, we run a set of experiments on the evaluation query set using the 11 chosen retrieval models and obtain the average precision for each query. For the queries in a cluster c_i, we compare $M_{T,best,c_i}$, the best-performing model for the training query set, to $M_{E,best,c_i}$, the best-performing model for the evaluation query set (see Table 6). If we can find that the two models are the same, then our assumption in Section 2.1 holds. For space reason[2], we just list the data obtained by the thresholds $k = 3$, $k = 5$ and $k = 7$. As shown in Table 6, for example, when the threshold setting is $k = 7$, the clustering process generates 7 clusters of queries. Out of the 7 clusters, for 4 clusters (i.e. the clusters 7.1, 7.5, 7.6 and 7.7), $M_{T,best,c_i}$ and $M_{E,best,c_i}$ are the same, which means that $M_{T,best,c_i}$ achieves also the best performance among the 11 models on Q_{E,c_i}, the queries belonging to c_i in the evaluation query set. For the cluster 7.2, although $M_{T,best,c_i}$ (i.e. PL2) and $M_{E,best,c_i}$ (i.e. I(n_exp)C2) are not exactly the same, $M_{T,best,c_i}$ provides good performance as well. The difference between the selected model PL2 and the optimal model I(n_exp)C2 is not significant in terms of performance for this cluster. The result of the cluster 7.4 is similar to the cluster 7.2. However, for the cluster 7.3, $M_{E,best,c_i}$ (i.e. PB2) is clearly better than $M_{T,best,c_i}$ (i.e. PL2), which is contradictory to our assumption in Section 2.1. Looking into this cluster, Table 5 shows that in the evaluation query set, there is only one query belonging to cluster 7.3. In this case, a single query may not provide enough evidence for our analysis. Indeed, the selected model $M_{T,best,c_i}$ achieves effective performance in almost all the cases for the three listed threshold settings (see Table 6). Therefore, our assumption that a retrieval model provides consistent performance for queries belonging to the same cluster holds.

Finally, we evaluate the proposed model selection mechanism on the evaluation query set. The results on the whole evaluation query set obtained by our model selection mechanism and by using the 11 chosen retrieval models are listed in Table 7. According to the evaluation results, using proper threshold settings, the model selection mechanism outperforms the strongest baseline on the evaluation query set. The best threshold setting is $k = 7$. Also, it is encouraging to see that for all the threshold settings, the model selection mechanism achieves very stable performance. Moreover, setting the threshold to $k = 6$, $k = 7$, $k = 9$ and $k = 10$, our model selection mechanism outperforms the use of any single model (see Table 7).

[2] We have also checked the data for other threshold settings, the results are quite compatible with those of the listed three threshold settings.

Table 5. Statistics of the model selection mechanism using different threshold settings. $M_{T,best,c_i}$ denotes the associated best-performing model of a cluster. $\#T$ and $\#E$ are the numbers of queries belonging to the cluster c_i in the training set T and the evaluation query set E respectively. For each threshold setting, we associate an ID to each cluster.

Threshold	ID	$M_{T,best,c_i}$	$(\#T, \#E)$	ID	$M_{T,best,c_i}$	$(\#T, \#E)$	ID	$M_{T,best,c_i}$	$(\#T, \#E)$
$k = 2$	2.1	I(n_exp)C2	(32, 25)	2.2	PL2	(68, 25)			
$k = 3$	3.1	I(n_exp)C2	(32, 25)	3.2	PL2	(48, 19)	3.3	I(n)B2	(20, 6)
$k = 4$	4.1	I(n_exp)C2	(32, 25)	4.2	PL2	(38, 14)	4.3	I(n)B2	(20, 6)
	4.4	I(n_exp)C2	(10, 5)						
$k = 5$	5.1	I(n_exp)C2	(32, 25)	5.2	PL2	(13, 9)	5.3	I(n)B2	(20, 6)
	5.4	I(n_exp)C2	(10, 5)	5.5	PL2	(25, 5)			
$k = 6$	6.1	I(n_exp)C2	(5, 1)	6.2	PL2	(13, 9)	6.3	I(n)B2	(20, 6)
	6.4	I(n_exp)C2	(10, 5)	6.5	PL2	(25, 5)	6.6	PL2	(27, 24)
$k = 7$	7.1	I(n_exp)C2	(5, 1)	7.2	PL2	(13, 9)	7.3	PL2	(7, 1)
	7.4	I(n_exp)C2	(10, 5)	7.5	PL2	(25, 5)	7.6	PL2	(27, 24)
	7.7	PB2	(13, 5)						
$k = 8$	8.1	I(n_exp)C2	(7, 8)	8.2	PL2	(7, 1)	8.3	PL2	(13, 9)
	8.4	I(n_exp)C2	(5, 1)	8.5	PL2	(25, 5)	8.6	PL2	(20, 16)
	8.7	PB2	(13, 5)	8.8	I(n_exp)C2	(10, 5)			
$k = 9$	9.1	I(n_exp)C2	(7, 8)	9.2	PB2	(8, 8)	9.3	PL2	(7, 1)
	9.4	I(n_exp)C2	(5, 1)	9.5	PL2	(13, 9)	9.6	PL2	(25, 5)
	9.7	PB2	(13, 5)	9.8	I(n_exp)C2	(10, 5)	9.9	PL2	(12, 8)
$k = 10$	10.1	I(n_exp)C2	(7, 8)	10.2	PB2	(8, 8)	10.3	PL2	(4, 4)
	10.4	I(n_exp)C2	(8, 4)	10.5	PL2	(7, 1)	10.6	I(n_exp)C2	(5, 1)
	10.7	PL2	(13, 9)	10.8	PB2	(13, 5)	10.9	PL2	(25, 5)
	10.10	I(n_exp)C2	(10, 5)						

5 Conclusions and future work

In this paper, we have proposed a methodology selecting the optimal retrieval model for a given query prior to the retrieval process. The evaluation results show that for various threshold settings, our model selection mechanism provides stable performance that is clearly as good as the results given by the strongest baseline, which is the best-performing retrieval model on a large training query set. Moreover, if appropriate threshold settings are used, the model selection mechanism outperforms the baseline.

We have also shown that, interestingly, a retrieval model provides consistent performance for queries belonging to the same cluster. Therefore, queries belonging to the same cluster favours some particular retrieval models.

We have obtained similar results for the disk4&5 (No CR) of the TREC collections, i.e. the collection of the TREC-7, 8 ad-hoc tasks [HO03b]. To avoid redundancy, in this paper, we just provide the results for the disk1&2 of the TREC collections.

Table 6. The performance of $M_{T,best,c_i}$ and $M_{E,best,c_i}$ on the clusters for the evaluation query set. For each cluster c_i, $M_{T,best,c_i}$ and $M_{E,best,c_i}$ are the best-performing model for c_i in the training query set T and the evaluation query set E respectively. $M_{E,best,c_i}$ is the mean average precision (MAP) on the cluster c_i. For each threshold setting, we associate an ID to each cluster. Δ is the gap between the $MAP_{Q_{E,c_i}}$ given by $M_{T,best,c_i}$ and $M_{E,best,c_i}$ respectively.

ID	$M_{T,best,c_i}$	$MAP_{Q_{E,c_i}}$	$M_{E,best,c_i}$	$MAP_{Q_{E,c_i}}$	$\Delta(\%)$
	$k = 3$				
3.1	I(n_exp)C2	0.2219	PB2	0.2239	0.89
3.2	PL2	0.3086	PL2	0.3086	0
3.3	I(n)B2	0.3928	PB2	0.4029	2.51
	$k = 5$				
5.1	I(n_exp)C2	0.2219	PB2	0.2239	0.89
5.2	PL2	0.3048	I(n_exp)C2	0.3065	0.55
5.3	I(n)B2	0.3928	PB2	0.4029	2.51
5.4	I(n_exp)C2	0.4583	I(F)B2	0.4627	0.95
5.5	PL2	0.1668	PL2	0.1668	0
	$k = 7$				
7.1	I(n_exp)C2	0.1505	I(n_exp)C2	0.1505	0
7.2	PL2	0.3048	I(n_exp)C2	0.3065	0.55
7.3	PL2	0.1968	PB2	0.2203	10.67
7.4	I(n_exp)C2	0.4583	I(F)B2	0.4627	0.95
7.5	PL2	0.1668	PL2	0.1668	0
7.6	PL2	0.2286	PL2	0.2286	0
7.7	PB2	0.4394	PB2	0.4394	0

We have provided a particular implementation of the proposed model selection approach. The involved components, including the features characterising a query, the definition of each feature, and the clustering algorithm, can be replaced with other possible candidates. Therefore, the performance of our approach could be improved if better replacements for the involved components are provided.

For example, in this paper, the queries are clustered in a three-dimensional space, where each vector consists of three features describing the statistics of a query. As stressed in Section 2.2, the three proposed features are not exhaustive. Moreover, the definition of each feature has various candidates. In the future, we will investigate other possible query features, and study possible alternative definitions for the proposed ones.

Finally, the study in this paper involved only 11 models of Amati & van Rijsbergen's DFR framework. In the future, we will also extend the approach to other possible choices, including other retrieval models in Amati & van Rijsbergen's DFR framework, and some of the recently proposed language models.

Table 7. The mean average precision obtained by using the model selection mechanism and using a fixed retrieval model indifferently for the *evaluation* query set. Selection(k) denotes the model selection mechanism by setting the threshold to k. The baseline is the optimal retrieval model for the training query set.

Model	Mean average precision
Different Retrieval Models	
PL2 (baseline)	**0.2766**
I(n_exp)C2	0.2747
PB2	0.2699
IFB2	0.2683
I(n_exp)B2	0.2681
BB2	0.2665
InB2	0.2664
InL2	0.2589
I(n_exp)L2	0.2554
IFL2	0.2552
BL2	0.2546
Model Selection	
Selection($k = 2$)	0.2757
Selection($k = 3$)	0.2753
Selection($k = 4$)	0.2755
Selection($k = 5$)	0.2755
Selection($k = 6$)	**0.2773**
Selection($k = 7$)	**0.2780**
Selection($k = 8$)	0.2762
Selection($k = 9$)	**0.2773**
Selection($k = 10$)	**0.2767**

6 Acknowledgments

This work is funded by a UK Engineering and Physical Sciences Research Council (EPSRC) project grant, number GR/R90543/01. The project funds the development of the Terrier Information Retrieval framework (URL: http://ir.dcs.gla.ac.uk/terrier). Also, we would like to thank Gianni Amati for his helpful comments on this work.

References

[ACR02] G. Amati, C. Carpineto, and G. Romano. FUB at TREC 10 web track: a probabilistic framework for topic relevance term weighting. In E.M. Voorhees and D.K. Harman, editors, *In Proceedings of the 10th Text Retrieval Conference TREC 2001*, pages 182–191, Gaithersburg, MD, 2002. NIST Special Pubblication 500-250.

[Ama03] G. Amati. *Probabilistic Models for Information Retrieval based on Divergence from Randomness.* PhD thesis, Department of Computing Science, University of Glasgow, 2003.

[AvR02a] G. Amati and C. J. van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. In *ACM Transactions on Information Systems (TOIS)*, volume 20(4), pages 357 – 389, October 2002.

[AvR02b] G. Amati and C. J. van Rijsbergen. Term frequency normalization via pareto distributions. In *Advances in Information Retrieval, 24th BCS-IRSG European Colloquium on IR Research Glasgow, UK, March 25-27, 2002 Proceedings.*, volume 2291 of *Lecture Notes in Computer Science*, pages 183 – 192. Springer, 2002.

[CTZC02] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 299 – 306, Tampere, Finland, 2002.

[GRS98] S. Guha, R. Rastogi, and K. Shim. CURE: an efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD Conference, Seatltle, WA*, pages 73–84, 1998.

[HO03a] B. He and I. Ounis. A study of parameter tuning for term frequency normalization. In *Proceedings of the Twelveth ACM CIKM International Conference on Information and Knowledge Management*, pages 10 – 16, New Orleans, LA, November 2003.

[HO03b] B. He and I. Ounis. University of gGlasgow at the Robust Track – a query-based model selection approach for the poorly-performing topics. In *Proceedings of the Twelth Text REtrieval Conference (TREC 2003)*, Gaithersburg, MD, 2003.

[JFH01] R. Jin, C. Falusos, and A. G. Hauptmann. Meta-scoring: Automatically evaluating term weighting schemes in ir without precision-recall. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 83–89, New Orleans, LA, 2001.

[LC02] S. Luo and J. Callan. Using sampled data and regression to merge search engine results. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 19–26, Tampere, Finland, 2002.

[MRF01] R. Manmatha, T. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–275, New Orleans, LA, 2001.

[PC98] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275 – 281, Melbourne, Australia, 1998.

[POAvR03] V. Plachouras, I. Ounis, G. Amati, and C. J. van Rijsbergen. University of Glasgow at the Web Track: Dynamic application of hyperlink analysis using the query scope. In *Proceedings of the Twelth Text REtrieval Conference (TREC 2003)*, Gaithersburg, MD, 2003.

[RWB98] S. Robertson, S. Walker, and M. Beaulieu. Okapi at trec-7: automatic ad hoc, filtering, vlc and interactive. In *NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC 7)*, pages 253 – 264, Gaithersburg, MD, 1998.

[Sal71] G. Salton. *The SMART Retrieval System.* Prentice Hall, New Jersey, 1971.

[vR79] C. J. van Rijsbergen. *Information Retrieval, 2nd edition.* Department of Computer Science, University of Glasgow, 1979.

[WNZ02] J. Wen, J.Y. Nie, and H. Zhang. Query clustering using user logs. *ACM Transactions on Information Systems (TOIS)*, 20(1)(1046-8188):59 – 81, 2002.

[ZL01] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334 – 342, New Orleans, LA, 2001.

[ZL02] C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49 – 56, Tampere, Finland, 2002.