University of Glasgow at the Robust Track - A Query-based Model Selection Approach for the Poorly-performing Queries

Ben He
Department of Computing Science
University of Glasgow
ben@dcs.gla.ac.uk

Iadh Ounis
Department of Computing Science
University of Glasgow
ounis@dcs.gla.ac.uk

ABSTRACT

In this newly introduced Robust Track, we mainly tested a novel query-based approach for the selection of the most appropriate term-weighting model. In our approach, we cluster the queries according to their statistics and associate the best-performing term-weighting model to each cluster. For a given new query, we assign a cluster to the query according to its statistical features, then apply the model associated to the cluster. As shown by the experimental results, our query-based model selection approach does improve the poorly-performing queries compared to a baseline where a unique retrieval model is applied indifferently to all queries. Moreover, it seems that query expansion has detrimental effect on the poorly-performing queries, although it significantly achieves a higher mean average precision over all the 100 queries.

1. INTRODUCTION

Many term-weighting models have been proposed for information retrieval. For a given collection and a given query, it is an interesting and challenging problem to automatically select the term-weighting model, which would provide the best retrieval performance. It is referred to as the *model selection* problem.

Previous works on the model selection problem, including [6, 8, 7], are based on the analysis of the term-weighting models relevance scores. 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 strategy. For a given query, it automatically selects a term-weighting 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 Divergence From Randomness (DFR) probabilistic framework [2]. Their framework deploys more than 50 models for term weighting. However, for a given retrieval task/query, the framework does not have a strategy to single out a model that would provide a reliable performance. Table 1 and Table 2 list the mean average precision (MAP) obtained by different models on the TREC-7 and TREC-8 ad-hoc tasks respectively. Here we just list the results given by the most stable and effective models in Amati & van Rijsbergen's framework. As shown by the results, even on the same collection, the optimal model could be different for each task. Also, for each task, the best model could achieve up to 5.65% higher MAP than the poorest one.

Table 1: The mean average precision (MAP) for the TREC-7 ad-hoc task on the disk4, 5 (No CR) of the TREC collections using 11 different DFR models.

Model	MAP	Model	MAP
I(F)B2	.1985	PL2	.1894
I(n_exp)L2	.1937	PB2	.1906
I(n_exp)C2	.2001	BB2	.1985
I(n_exp)B2	.1986	I(n)B2	.1987
I(n)L2	.1958	BL2	.1932
I(F)L2	.1933		

Table 2: The mean average precision (MAP) for the TREC-8 ad-hoc task on the disk4, 5 (No CR) of the TREC collections using 11 different DFR models.

Model	MAP	Model	MAP
I(F)B2	.2623	PL2	.2571
I(n_exp)L2	.2611	PB2	.2526
I(n_exp)C2	.2637	BB2	.2632
I(n_exp)B2	.2630	I(n)B2	.2649
I(n)L2	.2626	BL2	.2608
I(F)L2	.2607		

The aim of this study is to test a query-based approach for the selection of the most appropriate term-weighting model. For a given query and a given collection, we propose to automatically select the best performing term-weighting model. For the Robust Track, our assumption is that the proposed pre-retrieval model selection mechanism would improve the poorly-performing queries, by applying a term-weighting model that maximises the average precision of each query.

The remainder of the paper is organized as follows. In Section 2 we introduce our query-based model selection approach. In Sections 3 and 4, we describe our experimental setup for the Robust Track, and provide the evaluation results and the related analysis. In the experiments, the performance of the proposed model selection approach is particularly assessed. Finally, we conclude our work in Section 5.

2. QUERY-BASED MODEL SELECTION

Our query-based model selection mechanism assumes that the performance of a term-weighting model depends on the statistics of the query. Therefore, the statistical features of a query could constitute a good indication for the model selection mechanism.

Our mechanism involves a training process where the queries are clustered according to their statistical characteristics, and the best term-weighting model for each cluster of queries is obtained by taking previous relevance judgements into consideration. For a given new query, we assign a cluster to the query according to its statistical features, and then apply the term-weighting model associated to the cluster.

A possible approach to clustering the queries is to take the users' feedback into account, and cluster together the queries for which users have visited similar documents [10]. However, since the retrieved document are only known after the retrieval process, this approach is not appropriate for our pre-retrieval model selection mechanism.

2.1 Query Clustering

We propose a query clustering method that is independent of the retrieval procedure. For each query, we construct a feature vector, 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 represent a query. In this paper, we propose the following three factors for the feature vector of a query:

- Query length

  According to Zhai & Lafferty's work [12], query length has a strong effect on the smoothing methods of language models. In our previous work, we also found that query length heavily affects the length normalisation methods of the probabilistic models [5]. Therefore, it can be an important feature in the clustering process.

  In this work, we use $\rho \cdot ql$ to represent this feature, where:

  - ρ is a parameter. We experimentally set it to 0.2.
  - ql is the query length, i.e. the number of unique terms in the query.

- The difference of the informative amount in the query terms

  To describe the informative amount that a term carries, we usually associate an inverse document frequency (*idf*) to the term. The *idf* factor is a decreasing function of the number of documents containing the term.

  Since the informative amount of a query term is correlated with the utility of a term in the retrieval process, the most informative term in a query, which has the highest *idf*, is supposed to be the most useful term in discriminating the relevant documents from the others in a collection. On the contrary, the least informative term in a query would add little weight to the relevant documents. Indeed, the least informative term in a query tends to be a "stop-word", which could have a detrimental effect on the retrieval. Hence, the difference of the informative amount among the query terms can be an important feature of a query.

  In this work, this difference is defined as the quotient of the minimum *idf* divided by the maximum *idf* among the query terms:

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

  where:

  - γ is the factor representing the distribution of the informative amount in the query terms.
  - idf_{max} and idf_{max} are the minimum and maximum *idf* among the query terms respectively.
  - 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.

- The clarity/ambiguity of a query

  In [4], 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.

  Another clue for the clarity of a query is the size of the document set containing (at least one of) the query terms. In [9], Plachouras et. al. suggested that it is an important property of a query.

  In this work, we follow the idea in [9], and use

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

  to represent the clarity a query, where:

  - N is again the number of documents in the whole collection.
  - n_q is the number of documents containing (at least one of) the query terms.

  Thus, when n_q is small, we will obtain a large ω value, which implies that the query is very specific, and therefore is of high clarity.

As a consequence, the feature vector $\overline{qf}$ for a query is given as:

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

Finally, the feature vectors have to be clustered. A specific similarity measure, and a clustering algorithm need to be specified. In this work, we use the cosine of the angle between two feature vectors in the above three-dimensional space and the agglomerating hierarchical clustering (AHC) algorithm [11] for the clustering process. In the AHC algorithm, initially, each vector is an independent cluster. The similarity between two clusters is measured by the cosine similarity of their centroids. 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.

2.2 The Model Selection Mechanism

Based on the query clustering method introduced in the previous section, our model selection mechanism can be summarised as follows:

- We cluster a set of training queries according to their intrinsic features, as proposed in the previous section.
- For each cluster, we select the best-performing model in terms of the precision/recall measures.
- Then for a new query, we assign its closest cluster and trigger the best-performing model associated to the assigned cluster.

The proposed mechanism aims to optimise the average precision for each query, which leads to a maximised overall performance for all the queries.

In Sections 3 and 4, we describe our experimental setting for the Robust Track, and provide the evaluation results and the related analysis. The experiments aim to evaluate the proposed model selection mechanism, especially its performance on the poorly-performing queries.

3. EXPERIMENTAL SETUP

The document collection used in the Robust Track is the disk4 and disk5 (no CR database) of the TREC collections. The 100 topics given by TREC for the Robust Track have two parts. The first part consists of the 50 poorly-performing queries of the TREC-6, 7, 8 ad-hoc tasks, namely 50 old queries. The remaining part consists of 50 new queries introduced by the Robust Track.

Each query consists of 3 fields: title, description and narrative. As required, we use only the description field.

Two different query sets are used as the training set of the model selection mechanism, i.e. the 50 old queries and the 100 queries of the TREC-7, 8 ad-hoc tasks. The latter includes 35 queries of the former.

For the experiments, our model selection mechanism involves 11 term-weighting models developed within Amati & van Rijsbergen's DFR probabilistic framework. These models are: I(n_exp)C2, I(n)L2, I(n)B2, I(n)B2, I(n_exp)B2, BL2 , I(F)B2, I(n_exp)L2, BB2, PL2 and PB2. An effective and stable length normalisation method, i.e. the *normalisation 2*, is applied in these models. The details of these models and the normalisation 2 can be found in [1].

In our experiments for the Robust Track, the model selection mechanism was evaluated through 6 runs (5 official runs and 1 additional run). Also, its performance with or without the use of query expansion was tested. The runs are designed as follows (Table 3 lists the IDs of the runs, Sel78 was the additional run):

- InexpC2

  This is the baseline. It applies a single model, i.e. the I(n_exp)C2 model [1], for all the queries. The I(n_exp)C2 model is developed within Amati & van Rijsbergen's DFR framework and is considered as an effective and robust model on the collection used in this Robust Track. Its formula is:

  $$w(t,d) = \frac{F+1}{n_t \cdot (tfn_e + 1)} \left(tfn_e \cdot \log_2 \frac{N+1}{n_e + 0.5}\right)$$

  where:

  - $w(t,d)$ is the weight of the term t in the document d.
  - F is the term frequency of the term t in the whole collection.
  - n_t is the document frequency of the term t.
  - N is the number of documents in the collection.
  - n_e is given by $N \cdot \left(1 - (1 - \frac{n_t}{N})^F\right)$.
  - tfn_e is the normalised term frequency. It is given by the modified version of the normalisation 2 [1]:

    $$tfn = tf \cdot \log_e(1 + c \cdot \frac{avg\_l}{l}) \qquad (1)$$

    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; tf is the raw term frequency.

- Sel50 and Sel78

  In order to compare our query-based model selection mechanism to the baseline, we proposed two runs. The two runs use different training queries sets. Sel50 uses the 50 poorly-performing queries (50 old queries), and Sel78 uses the 100 queries of the TREC-7, 8 ad-hoc tasks. Thus, the effect of the training set on the model selection mechanism performance could be tested. We experimentally set the halting criterion of our query clustering method to $k = 4$ for Sel50, and $k = 3$ for Sel78.

- InexpC2QE

  We also tested the model selection mechanism with the use of a query expansion methodology. This run constitutes our baseline for the runs applying the query expansion methodology. The run InexpC2QE applies I(n_exp)C2 and a query expansion methodology for all the queries. The query expansion methodology follows

Table 3: The IDs of the 6 involved runs. +QE and -QE indicate that query expansion is applied or not respectively. Sel78 is an additional run.

	-QE	+QE
Run ID	InexpC2	InexpC2QE
	Sel50	Sel50QE
	Sel78	Sel78QE

the idea of measuring divergence from randomness [1]. The approach can be seen as a generalisation of the approach used by Carpineto and Romano in which they applied the Kullback-Leibler divergence to the un-expanded version of BM25 [3]. For each query, we extract the 40 most informative terms from the top 10 retrieved documents as the expanded terms.

- Sel50QE and Sel78QE

  Both the runs Sel50QE and Sel78QE use the same setting as the runs Sel50 and Sel78 for model selection. However, they also apply the query expansion mechanism for each query. Moreover, we experimentally set the halting criterion of our query clustering method to $k = 4$ for Sel50QE, and $k = 2$ for Sel78QE.

The parameter c of the normalisation 2 (see Equation (1)) was estimated by our new tuning approach, which measures the normalisation effect on the term frequency distribution [5]. Using this tuning approach, we automatically set the parameter to $c = 1.96$.

4. EXPERIMENTAL RESULTS

There are mainly three quantitatives measures that could be used to evaluate the experiments in the Robust Track:

$\#\overline{Rel}$: The number of queries with no retrieved relevant documents in the top 10 ranks, computed over the complete set of queries.

MAP(X): The area under the curve when MAP (mean average precision) of the worst X queries is plotted against X.

MAP: The mean average precision over the complete set of queries.

Tables 4 and 5 list the results of our runs. Tables 6 and 7 list the involved models in the model selection process and the performance of each single model over all the 100 queries. From the tables, we can see that MAP(X) is quite low in all the cases. Therefore, we mainly compare the MAP and $\#\overline{Rel}$ measures of the runs.

As shown by the results, for the poorly-performing queries, using the 50 old queries as the training query set, Sel50 and Sel50QE achieve better MAP and $\#\overline{Rel}$ than InexpC2 and InexpC2QE respectively (see Tables 4 and 5).

Using more training queries, for the poorly-performing queries, the performance of Sel78 is nearly the same as InexpC2 (see Table 4). Moreover, with query expansion, Sel78QE outperforms InexpC2QE (see Table 5).

Compared to all the 6 runs, we can see that Sel78QE achieves the highest MAP over all the 100 queries (see Tables 4 and 5).

Table 4: Results of the runs without query expansion. Sel78 is an unofficial run.

Queries	Run ID	$\#\overline{Rel}$	MAP(X)	MAP
50 old	InexpC2	10	.0056	.1019
	Sel50	*7*	.0055	*.1054*
	Sel78	10	.0049	.1015
50 new	InexpC2	4	.0246	.3478
	Sel50	4	.0162	.3327
	Sel78	3	.0219	.3446
all 100	InexpC2	14	.0094	.2249
	Sel50	11	.0071	.2190
	Sel78	13	.0081	.2231

Table 5: Results of the runs with query expansion.

Queries	Run ID	$\#\overline{Rel}$	MAP(X)	MAP
50 old	InexpC2QE	17	.0011	.1169
	Sel50QE	*13*	.0045	*.1295*
	Sel78QE	16	.0032	.1238
50 new	InexpC2QE	7	.0191	.3600
	Sel50QE	8	.0053	.3480
	Sel78QE	9	.0071	.3626
all 100	InexpC2QE	24	.0039	.2384
	Sel50QE	21	.0037	.2387
	Sel78QE	25	.0030	*.2432*

The run Sel78QE results into the constitution of two query clusters, to which the models PL2 and I(n_exp)B2 are associated respectively (see Table 6). It is encouraging to see that although I(n_exp)B2 has a better performance in terms of MAP than PL2 does (see Table 7), using PL2 for most of the queries (i.e. 86 out of 100), and I(n_exp)B2 for the rest, our model selection mechanism achieves even higher MAP. This observation suggests that the performance of a term-weighting model is dependent on the statistics of the query. Indeed, with the use of query expansion, this run (i.e. Sel78QE) outperforms the use of each single model indifferently for all the 100 queries (see Table 7).

Moreover, it seems that the query expansion methodology significantly improves the MAP, but has detrimental effect on the poorly-performing queries in terms of $\#\overline{Rel}$.

The performance of the query expansion methodology could be explained by its underlying assumption. It assumes that the top 10 ranked documents are highly relevant, then extracts the 40 most informative terms from them as the expanded query terms. Therefore, for the poorly-performing queries, the top 10 returned documents are likely to give false relevance information. As a consequence, the poorly-performing queries lead to the failure of the query expansion methodology.

5. CONCLUSIONS

In the Robust Track, we mainly evaluated our query-based model selection mechanism based on query clustering. The performance of the model selection mechanism was tested with two different training query sets and, with or without query expansion.

According to the evaluation results, if a proper training query set is used, our query-based term-weighting model selection does improve the performance of the poorly-performing queries compared to the baseline, where a unique term-

Table 6: Statistics of the model selection mechanism. M_Cluster is the term-weighting model associated to a cluster of queries. #Queries is the number of queries (in the whole 100 queries) belonging to the cluster.)

Run ID	M_Cluster	#Queries
Sel50 ($k = 4$)	PB2	18
	I(n_exp)C2	6
	I(F)B2	7
	I(F)L2	69
Sel78 ($k = 3$)	I(n_exp)B2	86
	I(n_exp)C2	14
	PL2	0
Sel50QE ($k = 4$)	I(F)B2	7
	PB2	6
	PL2	18
	I(F)L2	69
Sel78QE ($k = 2$)	PL2	86
	I(n_exp)B2	14

Table 7: Single model Vs. Model selection over all the 100 queries.

Without QE			With QE		
Model	#$\overline{Rel}$	MAP	Model	#$\overline{Rel}$	MAP
BB2	13	.2203	BB2	24	.2367
BL2	19	.2115	BL2	26	.2338
PB2	15	.2102	PB2	23	.2240
PL2	14	.2098	PL2	*18*	.2329
I(F)B2	14	.2230	I(F)B2	22	.2396
I(F)L2	17	.2156	I(F)L2	24	.2368
I(n)B2	*12*	.2181	I(n)B2	26	.2375
I(n)L2	15	.2160	I(n)L2	21	.2381
I(n_exp)B2	13	.2234	I(n_exp)B2	22	.2404
I(n_exp)C2	14	*.2250*	**I(n_exp)C2**	24	.2396
I(n_exp)L2	17	.2148	I(n_exp)L2	26	.2386
Sel50	14	.2190	Sel50QE	21	.2387
Sel78	13	.2231	Sel78QE	25	*.2432*

weighting model has been applied uniformly to all queries.

Moreover, it seems that query expansion has detrimental effect on the poorly-performing queries, although it achieves a significantly higher average precision measure over all the 100 queries. This observation can be explained by the underlying mechanism of the query expansion methodology.

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).

7. REFERENCES

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

[2] 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.

[3] C. Carpineto, R. de Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1 – 27, 2001.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] J. R. Wen and H. J. Zhang. *Information Retrieval and Clustering*, chapter Query Clustering in the Web Context, pages 1 – 30. Kluwer Academic Publishers, 2002.

[12] 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.