IRIT at TREC 2011: Evaluation of query reformulation techniques for retrieving medical records

Duy Dinh, Lynda Tamine

IRIT laboratory - Paul Sabatier University,
118 route de Narbonne, 31062 Toulouse, France
{Duy.Dinh,Lynda.Tamine}@irit.fr

Abstract. In TREC 2011, we are motivated to participate in the medical record retrieval task, namely TRECMed. Our research focused on the evaluation of term weighting models and query expansion techniques within the medical record retrieval task. We compared the performance of different state-of-the-art *term weighting models* for retrieving *patient records* that might best suit the clinical information need. Afterwards, we evaluate different state-of-the-art *query expansion* (QE) techniques within an IR framework. We describe the IR system architecture and how we carried out the TREC experiments, and we present effectiveness results.

Key words: Term Weighting Models, Query Expansion, Query Removal, Medical Record Retrieval

1 Introduction

In TREC 2011, we focused on two main features of information retrieval (IR), especially for biomedical IR: (1) *term weighting*, and (2) *query reformulation*. We first investigate the effectiveness of three different state-of-the-art term weighting models that have been shown to work well in the past: the well-established BM25 model [1], the Divergence From Randomness model namely In_expB2 (Inverse Expected Document Frequency model with the Bernoulli ratio normalisation) [2] and the log-logistic model namely LGD (a log logistic model) [3]. Next, we experiment with three different state-of-the-art query expansion algorithms implemented in the Terrier IR platform [4].

The remainder of this paper is organized as follows: Section 2 describes our indexing and retrieval framework. Experimental results will be presented and discussed in section 3. Section 4 gives a conclusion of our participation in TRECMed 2011.

2 Indexing and retrieval framework

Our indexing and retrieval framework is based on an open source search engine, which has been widely used for research in IR. More specifically, we used the Terrier IR platform [4] for indexing and retrieving documents in the collection of patients' visits. Each visit contains a set of reports related to a particular patient. In the pre-processing stage, we combined all reports of each patient into a single TREC-like document to obtain a single patient record as the unit of the retrieval.

The indexing aims to organize, structure and store statistical and/or linguistic information about terms and documents in the collection allowing a rapid and efficient search. During the indexing stage, stop-words are removed from documents before stemming using the Porter algorithm [5].

The document retrieval aims to match the user query and document representations in order to retrieve a list of results that may satisfy the user information need. In our work, a document D containing terms used for formulating query Q is weighted by summing the score of each term figuring in document D:

$$RSV(D,Q) = \sum_{t \in Q} score(t \in D) \tag{1}$$

where $score(t \in D)$ is the query term weight calculated using a particular term weighting model. For evaluating the performance of current state-of-the-art weighting models, we chose three different term weighting models used in our experiments, namely BM25 [1], In_expB2 [2] and LGD [6]. We then applied several state-of-the-art pseudo-relevance feedback techniques using statistical measures such as the Bose-Einstein (Bo) statistics [2] and the Kullback-Leibler divergence [7] in order to select most related terms for enriching the original query.

2.1 The BM25 model

In the BM25 weighting model, the RSV of a document D for a query Q is:

$$RSV_{BM25}(D,Q) = \sum_{t \in Q} \frac{(k_1+1)*tfn}{K+tfn} * \frac{(k_3+1)*qtf}{k_3+qtf} * w^{(1)} \tag{2}$$

where

- tfn is the normalized within-document term frequency given by:

$$tfn = \frac{tf}{(1+b)+b*\frac{dl}{avg\_dl}}, \tag{3}$$

  where tf is the within-document term frequency, dl and avg_dl are respectively the document length and average document length,
- k_1, k_3 and b are tuning parameters, for which the default values are $k_1 = 1.2, k_3 = 8.0, b = 0.75$,

- K is $k_1 * ((1-b) + b * dl/avg\_dl)$,
- qtf is the within-query term frequency,
- $w^{(1)}$ is the idf (inverse document frequency) factor computed as:

$$w^{(1)} = log_2 \frac{N - N_t + 0.5}{N_t + 0.5} \tag{4}$$

where N is the total number of documents (or cardinality) in the collection, and N_t is the number of documents containing term t (also called as document frequency).

In the BM25 model, if $N < 2N_t$, i.e., term t is quite frequent in the collection, then t is assigned with a negative score in a particular document D because $w^{(1)} < 0$. In order not to penalize such terms, we just ignored them by giving a zero score when calculating the RSV(D, Q).

2.2 The In_expB2 model

For the In_expB2 model, query terms are weighted using the Inverse Expected Document Frequency model with Bernoulli after-effect and term frequency normalisation [2]. Formally, the RSV of a document D for a query Q is:

$$RSV_{In\_expB2}(D,Q) = \sum_{t \in D} qtf \times \frac{(tf+1) \times tfn_2}{N_t \times (tfn_2+1) \times ln2} \times \log_2 \frac{N+1}{N \times (1-e^{\frac{-tf}{N}}) + 0.5} \tag{5}$$

where

- t is a query term occurring in document D,
- N_t is the document frequency,
- N is the total number of documents in the collection,
- qtf is the query term frequency,
- tf is the within-document term frequency,
- tfn_2 is the normalised within-document term frequency, given by:

$$tfn_2 = \frac{tf}{ln2} \times log_2 \left[1 + c \times \frac{avg\_dl}{dl}\right] \tag{6}$$

2.3 The LGD model

In the LGD model, query terms are weighted using the log logistic distribution [6]. Formally, the RSV of a document D for a query Q is:

$$RSV_{LGD}(D,Q) = \sum_{t \in D} qtf \times \left[log_2(\frac{N_t}{N} + tf_n) - log_2(\frac{N_t}{N})\right] \tag{7}$$

where

- t is a query term occurring in document D,

- N_t is the document frequency (i.e., number of documents containing term t),
- N is the total number of documents in the collection,
- qtf is the query term frequency,
- tf_n is the normalised within-document term frequency, given by:

$$tf_n = tf \times log_2(1 + c \times \frac{avg\_dl}{dl}) \tag{8}$$

where avg_dl is the average document length (in tokens), dl is the document length (in tokens) and c is a multiplying factor or tuning parameter.

2.4 Query expansion

The DFR framework employs a query expansion (QE) mechanism that is a generalisation of Rocchio's method [8]: terms in the top-ranked documents retrieved in the first stage are weighted using a particular DFR term weighting model. In general, the weight of a term of the expanded query q^* derived from the original query q is obtained as follows:

$$weight(t \in q^*) = qtfn + \beta * \frac{Info_{DFR}}{MaxInfo} \tag{9}$$

where

- $qtfn$ is the normalised within-query term frequency,
- $MaxInfo = \arg_{t \in q^*} \max Info_{DFR}$,
- $Info_{DFR}$ is the term frequency in the expanded query induced by using a DFR model, that is:

$$Info_{DFR} = -\log_2 Prob(Freq(w|K)|Freq(w|C)) \tag{10}$$

where Prob is the probability of obtaining a given within-query term frequency from the top-ranked documents retrieved in the first stage. In the DFR framework, several measures are used to compute this probability such as: Bose-Einstein (Bo) statistics and Kullback-Leibler (KL) measure [2]. The former gives the following term frequency normalisation:

$$\begin{aligned} \text{Info}_{\text{Bo}} &= -\log_2 \text{Prob}(\text{Freq(w|K)}|\text{Freq(w|C)}) \\ &= -\log_2(\tfrac{1}{1+\lambda}) - \text{Freq(w|K)} * \log_2(\tfrac{\lambda}{1+\lambda}) \end{aligned} \tag{11}$$

where

- $Freq(w|K)$ (resp. $Freq(w|C)$) is the the term frequency within the top ranked documents (resp. the collection)
- $\lambda_{\text{Bo1}} = \frac{\text{Freq(w|C)}}{\text{N}}$ and $\lambda_{\text{Bo2}} = \frac{\text{TotalFreq(K)} * \text{Freq(w|C)}}{\text{TotalFreq(C)}}$, where TotalFreq(X) is the total frequency of term t in X documents (e.g., top-ranked documents, or the whole collection).

- $\beta = 0.4$ is the Rocchio's parameter.

while the latter gives the following term frequency normalisation:

$$\text{Info}_{\text{KL}} = \frac{Freq(w|K)}{TotalFreq(K)} * \log_2 \frac{Freq(w|K)*TotalFreq(C)}{Freq(w|C)*TotalFreq(K)} \quad (12)$$

We used the default settings in Terrier for indexing and retrieval, for example the parameter $b = 0.75$ for term weighting using the BM25 model, $c = 1.0$ for the In_expB2 [2] and LGD [6] models. For query expansion, we extracted 20 most representative terms from the top 20 ranked documents returned by the system from the first retrieval stage.

3 TRECMed submissions

3.1 Run description

We submitted four official runs to the TREC medical retrieval track. Our submitted runs are divided into two groups: the first one (2 runs) includes *automatic runs* and the second one (2 runs) includes *manual runs*. For each group of runs, we aim to evaluate the performance of state-of-the-art indexing and retrieval approaches compared to the performance of those enhanced with query expansion techniques. The description of the four submitted runs are as follows:

- $IRITa1$ – This run scored documents w.r.t a query using the In_expB2 model [2], with c set to 5.0. The purpose of this run was to evaluate the performance of a state-of-the-art IR approach
- $IRITa1QE1$ – This run scored documents w.r.t a query using the In_expB2 model [2] ($c = 5.0$) and enhanced with a Rocchio's query expansion technique using the Bo1 model [2]. The purpose of this run was to evaluate the utility of query expansion for IR.
- $IRITm1$ – This run was similar to the first run with the exception that some "redundant terms" such as "patient", "who" (which of course are not present in the stop-word list) are manually removed from the query. The purpose of this run was to evaluate the impact of redundant terms in the query on the IR performance.
- $IRITm1QE1$ – The last run was similar to the second run but with the exception that some "redundant terms" are removed from the query. In addition, we employed the Bo1 model [2] for query expansion. The purpose of this run was to evaluate the impact of query expansion and query removal on the IR performance.

For automatic runs, we use the default term processing pipeline in Terrier: stop-words are removed from documents and queries before stemming using the Porter algorithm [5]. For manual runs, we further removed query terms that are not present in the stop-word list but that we believe are not quite informative. For example, in the query "Patients with hearing loss", the term "with" is recognized

as a stop-word and is therefore automatically removed from documents/queries. However, the term "patients" is not a stop-word term but is not quite informative because the main subject matter of the query is "hearing loss" which includes "patients" as the retrieval unit. Therefore, the term "patients" is considered as redundant information, which can be a reason of the query drift problem in IR. Therefore, it should be removed from the query. In this section, we will demonstrate the reason why such redundant terms should be removed from the query while representative terms should be used to enrich the semantics of the query.

In what follows, we present the results of our official runs submitted to TRECMed 2011. Afterwards, we present the results obtained by experimenting with several state-of-the-art weighting models across different query expansion algorithms in the Terrier IR platform.

3.2 Official results

Table 1 shows the official results of our runs submitted to TRECMed 2011. According to the results, we observe that no significant performance gain is achieved in terms of MAP and P@X(x=10, 20) for both the automatic and manual runs with/without query expansion. Such obtained results may be influenced by the performance of the In_expB2 model and the Bo1 query expansion model implemented in Terrier on the collection of medical records. Each of the record is composed of a set of single reports related to a particular patient. Using the same scenarios like runs submitted to TRECMed 2011 with the exception that the default configuration in Terrier is used (e.g., b=0.75, c=1.0), we further carried out the experiments with the two other state-of-the-art models namely BM25 [1] and LGD [6] across three query expansion models namely Bo1, Bo2 [2] and KL [7].

Table 1. IR effectiveness obtained by each run on the TRECMed 2011 collection.

Measure / Run	bpref	MAP	P@10	P@20
$IRITa1$	0.4283	0.3323	0.4824	**0.4132**
$IRITa1QE1$	0.4283	**0.3344**	**0.4882**	0.3912
$IRITm1$	**0.4619**	0.3323	0.4824	**0.4132**
$IRITm1QE1$	**0.4619**	**0.3344**	**0.4882**	0.3912

3.3 Unofficial results

First of all, we study the impact of removing non-informative terms from the query. Then, we aim to show the effectiveness of three query expansion models (Bo1, Bo2 and KL) on the TRECMed 2011 collection. Finally, we aim to show the utility of combining query removal and query expansion for IR.

Effectiveness of query removal for IR. As can be seen in table 2, the MAP results of the LGD model are slightly different from the BM25 model (0.3058 *vs.* 0.3182) without query removal ($\overline{QR}$). However, the results in terms of P@x(x=10, 20) of the LGD model are dramatically better than the BM25 model (0.5118 *vs.* 0.4324 and 0.4103 *vs.* 0.3721). For query removal (QR), the conclusion is similar to the In_expB2 model: no improvement is achieved when using the BM25 model. However, when applying the LGD model along with QR, we obtained an improvement of +10.76 % in terms of MAP, +1.72 % in terms of P@10 and +5.39 % in terms of P@20 over the baseline LGD ($\overline{QR}$). The results in terms of MAP and P@x(x=10, 20) are even better than those obtained by the BM25 model with or without QR. Note that the LGD model is an information-based model based on the log-logistic and smoothed power law [3]. This method has shown high performances by focusing on modeling the notion of "burtiness" in IR [6]. This notion describes the behavior of words which tend to appear in bursts, i.e., once they appear in a document, they are more likely to appear again. For this reason, we retained the LGD model for the next experiments.

Table 2. IR effectiveness obtained by using the BM25 and LGD models on TRECMed 2011 collection with/without query removal (QR)

	bpref		**MAP**		**P@10**		**P@20**	
	$\overline{QR}$	QR	$\overline{QR}$	QR	$\overline{QR}$	QR	$\overline{QR}$	QR
BM25	0.4301	0.4301	0.3182	0.3182	0.4324	0.4324	0.3721	0.3721
LGD	0.4329	**0.4607**	0.3058	**0.3387**	0.5118	**0.5206**	0.4103	**0.4324**

Utility of combining query removal and query expansion for IR. Table 3 depicts the results obtained by the LGD model with and without query removal across three query expansion models on the TRECMed 2011. As we can see, the LGD model performs well when solely applying query removal (see table 2, run $QR + LGD$) or solely applying query expansion (*cf.* table 3, run $\overline{QR} + LGD + QE$, where QE stands for Bo1, or Bo2 or KL). When combining QR and QE, the performance in terms of MAP and P@x(x=10, 20) are even better than solely applying QR as well as than solely applying QE. Indeed, the best MAP obtained by run $QR+LGD+Bo1$ is achieved at **0.3944**. The best value of P@10 (resp. P@20) is obtained at **0.4794** (resp. **0.4912**). Therefore, we conclude that combining QR and QE within an appropriate weighting model allows to improve the IR performance. QR aims at focusing on the main subject matters of the query while QE aims at enriching the modified query with more related terms which better describe the semantics of the query. The intuition underlying QR is that the more the subject matters of the query are determined (by removing non-informative terms), the more the returned documents are close to the query.

In addition, the combination of QR and QE aims at retrieving more relevant documents w.r.t a given query without losing the recall and the precision.

Table 3. bpref, MAP and P@10 results obtained by the LGD model across 3 query expansion (QE) models (Bo1, Bo2, KL) with/without query removal (QR) on TRECMed 2011 collection

	bpref			**MAP**			**P@10**		
	Bo1	Bo2	KL	Bo1	Bo2	KL	Bo1	Bo2	KL
$\overline{QR} + LGD$	0.4954	0.3902	0.4779	0.3534	0.2323	0.3432	0.5441	0.4265	0.5265
$QR + LGD$	**0.5311**	0.4058	0.5086	**0.3944**	0.2496	0.3852	**0.5794**	0.4441	0.5676
TREC-best	0.5520			N/A			0.6560		
TREC-second	0.5520			N/A			0.6030		
TREC-third	0.5450			N/A			0.6030		
TREC-fourth	0.5220			N/A			0.5440		
TREC-median	0.4120			N/A			0.4760		

4 Conclusion

In TREC 2011, we participated in the TRECMed track, which is a medical record retrieval *adhoc* task. The underlying IR platform of our experiments is the Terrier search system. Our participation focused on the use of several IR models for term weighting as well as state-of-the-art query expansion models. The best results of our runs attest the effectiveness of combining query removal and query expansion for IR using a particular term weighing model, especially the most recent IR information-based model namely LGD.

References

1. Robertson, S.E., Walker, S., Hancock-Beaulieu, M.: Okapi at trec-7: Automatic ad hoc, filtering, vlc and interactive. In: TREC. (1998) 199–210
2. Amati, G.: Probabilistic models for Information Retrieval based on Divergence from Randomness. PhD thesis, University of Glasgow (2003)
3. Clinchant, S., Gaussier, É.: Information-based models for ad hoc IR. In: SIGIR. (2010) 234–241
4. Ounis, I.;Lioma, C.C.V.: Research directions in terrier. Novatica/UPGRADE Special Issue on Web Information Access, Ricardo Baeza-Yates et al. (Eds), Invited Paper (2007)
5. Porter, M.F. In: An algorithm for suffix stripping. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997) 313–316
6. Clinchant, S., Gaussier, É.: Retrieval constraints and word frequency distributions a log-logistic model for ir. Information Retrieval **14**(1) (2011) 5–25
7. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
8. Rocchio, J. In: Relevance Feedback in Information Retrieval. (1971) 313–323