Evaluating a novel kind of retrieval models based on relevance decision making in a relevance feedback environment

H.C. Wu, K.F. Dang, R.W.P. Luk, J. Allan[1], K.L. Kwok[2], K.F. Wong[3], G. Ngai and Y. Li

Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[1]Department of Computer Science, University of Massachusetts, USA
[2]Department of Computer Science, City University of New York, USA
[3]Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong

Abstract

This paper presents the results of our participation in the relevance feedback track using our novel retrieval models. These models simulate human relevance decision-making. For each document location of a query term, information from its document-context at that location determines the relevance decision outcomes there. The relevance values for all documents locations of all query terms in the same document are combined to form the final relevance value for that document. Two probabilistic models are developed, and one of them is directly related to the TF-IDF term weights. Our initial retrieval is a passage-based retrieval. Passage scores of the same document are combined by the Dombi fuzzy disjunction operator. Later, we found that the Markov random field (MRF) model produces better results than our initial retrieval system (without relevance information). If we apply our novel retrieval models using the initial retrieval list of the MRF model, the retrieval effectiveness of our models will be improved. These informal run results using the MRF model used in conjunction with our novel models are also presented.

1. Introduction

In this participation, we are interested in an alternative RF approach. Previously, we developed various document-context dependent retrieval models [1] that operate in a RF environment. In here, we further developed and used a fully probabilistic retrieval model. Our approach uses the relevance judgment of the feedback documents to estimate the parameters of our probabilistic models. These models are descriptive ones because they do not adjust their parameters according to their performance, unlike normative models (e.g., support vector machine [2]) that optimize their performance using training data. In addition, we extended the common TF-IDF term weights so that they become document-context dependent.

The rest of this paper is organized as follows. Section 2 presents our models. Section 3 describes the set up of our experiments. Section 4 analyze and discusses our results of our formal runs. Section 5 looks at the results of the informal runs. Finally, Section 6 concludes.

2. Our Approach

Our models are based on the premise that local relevance at a particular document location is determined by the information in the context of that location. The general equation of making relevance decision [3] may be captured by the following formula:

$$\nabla(d,q) := C(\{\partial_{d,k}(c(d,k),q) : 1 \leq k \leq |d|\}) \quad (1)$$

where $\nabla(\bullet)$ is the document-wide relevance decision, $\partial_{d,k}(\bullet)$ is the local relevance decision at location k for document d, $c(d, k)$ is the document context, which is a sequence of a fixed number of terms, at location k in document d, and $C(\bullet)$ is a function that combines the local relevance values into a document-wide relevance value.

2.1 Model 1

In this model, we use the document-context based model which is similar to [3]. $c(d, k)$ starts at position k-n to k+n such that it has $2n$+1 terms (i.e., the context size is $2n$+1 and n=25 in the experiments). In the document-context based model, we give scores to the contexts and the scores are summed together to produce the document score. By the query centric assumption [3], we only consider the contexts of query terms. In the experiments of the relevance feedback track, besides the contexts of query terms, we also consider the contexts of expansion terms. The set of expansion terms is denoted by Q_e. We will discuss how expansion terms are found after presenting the ranking formula.

Let R be the binary random variable for relevance where $R = r$ means relevant and $R = \bar{r}$ means irrelevant. The score of a document d is calculated by summing the scores of the contexts $c(d, k)$ where $d[k]$ is a term in $Q \cup Q_e$. The score of a context is the log of the ratio between the probability of seeing the context in the relevance model of $d[k]$ (i.e., $p(c(d,k) \mid d[k], Q, R = r)$) and the probability of seeing the context in the irrelevance model of $d[k]$ (i.e., $p(c(d,k) \mid d[k], Q, R = \bar{r})$).

$$Score(d) = \sum_{k \ni d[k] \in Q \wedge d[k] \in Q_e} \log \frac{p(c(d,k) \mid d[k], Q, R = r)}{p(c(d,k) \mid d[k], Q, R = \bar{r})}$$

The probabilities of seeing the context $c(d, k)$ in the relevance model and irrelevance model of $d[k]$ are given by multiplying the probabilities by seeing the terms in $c(d, k)$ in the relevance model and irrelevance model of $d[k]$ respectively:

$$\frac{p(c(d,k) \mid d[k], Q, R = r)}{p(c(d,k) \mid d[k], Q, R = \bar{r})} = \frac{\prod_{p=-n}^{n} p(d[k+p] \mid d[k], Q, R = r)}{\prod_{p=-n}^{n} p(d[k+p] \mid d[k], Q, R = \bar{r})}$$

The probability of seeing the term t in the relevance model of q_i is given by the ratio between the occurrence frequency of the term t in those contexts and the total occurrence frequencies of all the terms in those contexts. Similarly for the irrelevance model of q_i which considers the contexts of q_i in the irrelevant documents. For the collection model of q_i, it considers all the contexts of q_i in the documents from the initial retrieval list which is the result of Set A. The probabilities from the relevance model and the irrelevance model are smoothed by the probability from the collection model using linear interpolation with the weight of the relevance/irrelevance model set to 0.1. This is because some of the terms may not occur in the relevance/irrelevance model and they receive zero probabilities which will give undefined result in calculation of the context score. After smoothing with the collection model, all probabilities are larger then zero.

In finding the expansion terms (in Q_e), a relevance model and an irrelevance model for the query Q are constructed similar to those constructed for each query term q_i described in the previous paragraph. The difference is that instead of considering the contexts of a particular query term, we consider the contexts of all query terms (i.e., combining the relevance/irrelevance models of individual query terms to form a relevance/irrelevance model for the query). The terms in the relevance model are ranked by the difference between the probability given by the relevance model and the probability given by the irrelevance model. Those terms with the difference smaller than zero are discarded. In our experiments, top 500 terms from the ranked term list are considered as expansion terms. For each expansion, similar to each query term, a relevance model, an irrelevance model and a collection model are constructed.

2.2 Model 2

Based on the judged N documents (N=1 for RF08.B and N=6 for RF08.C) we apply (1) query expansion (QE) followed by (2) boost and discount, as described below.

(a) QE Stage: We select query expansion terms from document-contexts within each judged documents. Document-contexts are text windows centered on query terms. The context size is fixed to be 41. We then obtain the vectors $\overrightarrow{q_{rel}}$ and $\overrightarrow{q_{irr}}$ whose elements represent the terms contained in the judged relevant and judged irrelevant documents respectively. The value of a term t is given by

$$Score(t) = \frac{Freq(t)}{1+Freq(t)} \times NoDoc(t) \times idf(t) \times \left(1+\frac{tmprf(t)}{1+tmprf(t)}\right)$$

where $Freq(t)$ = total term frequency of t in the judged relevant / irrelevant document-contexts, $NoDoc(t)$ = number of judged relevant / irrelevant document-contexts that contain t, $idf(t)$ = inverse document frequency of t in the whole collection, and $tmprf(t)$=$df(t)$-$NoDoc(t)$+1, with $df(t)$= document frequency of t in the collection. For each of $\overrightarrow{q_{rel}}$ and $\overrightarrow{q_{irr}}$, we include a fixed number of terms (N_{QE}) with the highest scores. We set N_{QE}=80. The relative weights of the judged relevant and judged irrelevant documents may be specified by a parameter β, so that the query expansion vector is

$$\overrightarrow{q_{QE}} = \beta \frac{\overrightarrow{q_{rel}}}{|q_{rel}|} - (1-\beta)\frac{\overrightarrow{q_{irr}}}{|q_{irr}|}.$$

Finally, we obtain an expanded query by mixing the original query $\vec{q}$ and $\overrightarrow{q_{QE}}$:

$$\overrightarrow{q_{RF}} = \alpha \frac{\vec{q}}{|q|} + (1-\alpha)\frac{\overrightarrow{q_{QE}}}{|q_{QE}|}.$$

We have used the values $\alpha = 0.3$ and $\beta = 0.6$.

(b) Boost and discount (B&D) stage: We perform a second retrieval with the expanded query $\overrightarrow{q_{RF}}$ based on a vector space model using BM25 term weights. For the terms that appear in the original query $\vec{q}$, we directly modify the *tf* component of the BM25 term weight, utilizing evidence from the judged documents. Generally the specific usage of a query term can be deduced by examining the words in its vicinity. Hence the ‘collocation terms’, defined as the terms that appear within a document-context centred on a query term in a judged relevant / irrelevant document can provide evidence for / against the relevance of a non-judged document. Hence, in a non-judged document, if the words within a document-context of a query term are similar to the collocation terms found form judged relevant documents, this would support the document as likely to be relevant too. In our algorithm, we implement this effect by giving a 'boost' to the *tf* component of the BM25 term-weight of the query term. Similarly, if the words lying in the document-context of a query term match the collocation terms extracted from judged irrelevant documents, we 'discount' a certain amount of the *tf* value.

Suppose $\vec{q}=\{q_1, q_2, \ldots, q_n\}$. In the B&D algorithm, we adjust $tf(q_i)$ according to the matching of words appearing in the context-windows centred on q_i with 'boost' or 'discount' collocation terms. Let B_B and B_D denote the sets of 'boost' and 'discount' collocation terms respectively. These are the context terms extracted from judged relevant and irrelevant documents. In general, the size of the context windows for extracting B_B and B_D terms may be denoted by $consize_B$ and $consize_D$ respectively. We introduce the variables c_b and c_d as matching counts of the 'boost' and 'discount' collocation terms, defined as follows.

$$c_b(q_i) = \sum\nolimits_w increment_B(w)$$

where the sum is over all terms occurring in a document-context centred on q_i, and

$$increment_B(w) = \begin{cases} idf(w)/idf_0 & \text{if } w \in B_B \\ 0 & \text{otherwise} \end{cases}.$$

In the above equation, $idf(w) = \log_{10}(N + 0.5/df(w) + 0.5)$ where N is the total number of documents in the collection, $df(w)$ is the document frequency of w, and $idf_0 = \log_{10}((N + 0.5)/0.5)$. The discount values $c_d(q)$ are defined similarly by matching the words with those in the set of 'discount collocation terms', B_D.

We directly adjust the *tf* factor of query term q:

$$tf(q_i) \leftarrow tf(q_i) + \sum_k tf_{BD}(q_i, k),$$

where the sum is over all locations k of the occurrences of q_i in the document, and

$$tf_{BD}(q_i, k) = BFactor * BCnt(\gamma \cdot c_b - c_d).$$

In the above, $BCnt(x)$ is a linear function with $BCnt(0)=0$ and saturates at $x = \pm consize_M$, where $consize_M$ is the context size for matching collocation terms. Specifically, $BCnt(-consize_M) = -1$ and $BCnt(consize_M) = 1$. *BFactor* is a constant that controls the effect of B&D, and γ is a constant that adjusts the relative weighting of boost and discount. We have chosen the following set of parameters: $consize_B = 21$, $consize_D = 11$, $consize_M = 11$, $BFactor = 6.0$ and $\gamma = 2$.

3. Set Up and Calibration

We used a PC-cluster (called MATRIX) to perform the indexing and retrieval. The GOV2 document collection is distributed to 40 nodes. Each node holds about 10+G bytes documents which are indexed in about 10 hours when there are other jobs running at the time. Note that each node has one CPU that has only one core. On average, the size of each index plus other auxiliary files (e.g., dictionary) in a node is about 500M bytes.

For the initial retrieval without any RF (i.e., Set A), we calibrated our retrieval system using the 2005-2007 Terabyte track collections. Passages of at most 300 terms are used as the basic unit of retrieval. The passage scores are based on our version of the BM25 term weights [4]. These passage scores are normalised between zero and one, and their normalised values are treated as membership values that are fed into the fuzzy disjunction Dombi operator [5]. Pseudo-relevance feedback (PRF) is used. It reads the top 15 passages in the initial retrieval list in order to expand the original title query by adding 80 expansion terms. The mean average precision (MAP) of the initial retrieval was around 23%. The top 3000 documents of this retrieval list are re-ranked by our model 1 and 2 for feedback sets B-E, which contain different number of relevance documents and nonrelevant documents in different sets. Set B contains one relevant feedback document for each query. Set C contains 3 relevant and 3 non-relevant feedback documents for each query. Set D contains 10 judged feedback documents for each query. Finally, Set E contains many judged relevant and nonrelevant feedback documents.

For our calibration, we compared the performance of our model 1 with SVM that is trained using the feedback documents. Our model 1 and SVM re-ranked the same initial retrieval list produced by our retrieval system with PRF. Results are shown in Table 1. The MAPs of our model 1 are better than the corresponding MAPs of the SVM for feedback sets C and D. The statistical significance of their MAP differences is at the 95% confidence level.

Table 1: TREC RF Track: Comparison with Support Vector Machine (Model 1 is not calibrated by any RF track queries)

Set	P@10		P@30		MAP		R-Precision	
	Model 1	SVM	Model 1	SVM	Model 1	SVM	Model 1	SVM
B	.3144	.3330	.1977	.2090	.2601	.2503	.2643	.2585
C	.4367	.4455	.2345	.2535	.3719*	.3573	.3684	.3630
D	.5205	.5064	.2684	.2818	.4225*	.4041	.4222	.4042
E	.6394	.6428	.4129	.4138	.6195	.6189	.6045	.6011

Key: * means statistical significance at 95% confidence level (or p-value < 0.05). Sets B, C, D and E have different numbers of relevant documents (i.e., 1 relevant, 3 relevant and 3 nonrelevant, 10 judged documents, many judged documents, respectively).

4. Formal Runs

The formal runs report results for two subsets of queries. One subset is the terabyte queries and the other subset is the million query track queries. We obtain the relevance feedback results for feedback set B-E for model 1 but only set B-C for model 2 due to lack of time.

4.1 Terabyte Track Subset

Table 2: Formal runs for (31) Terabyte track queries.

Initial Retrieval	P@10	MAP	R-Prec			
Set A	.3419	.1670	.1892			
Ours	Model 1			Model 2		
Sets	P@10	MAP	R-Prec	P@10	MAP	R-Prec
B	.2387	.1224	.1504	.2645	.1239	.1649
C	.2839	.1423	.1666	.3032	.1314	.1629
D	.2548	.1356	.1610			
E	.2839	.1599	.1857			
TREC	Best			Median		
Measures	P@10	MAP	R-Prec	P@10	MAP	R-Prec
Average	.7129	.4215	.4530	.2839	.1427	.1801

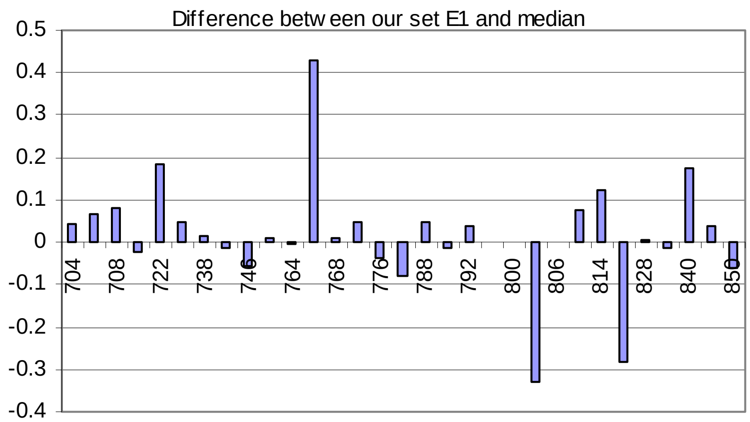

Figure 1: MAP difference between our model 1 and the median performance of 31 Terabyte track queries.

Table 2 shows the performance of our models, and the averages of best and of the median performance of all participants for 31 Terabyte track queries. Interestingly, the performance of our ad hoc retrieval without any RF is better than any of our own models with RF. The performance of our models is similar to the average of the median performance of each query. Figure 1 shows the MAP difference between our model 1 and the median performance of each query.

4.2 Million Query Track Subset

Table 3 shows the performance of our formal runs for million query track queries used in this RF track. For this subset of queries, the StatAP and MTC AP estimates are reported. The performance of our model 1 using feedback set E is similar to the median performance of the participants. Figures 2 and 3 show the MTC AP estimate and StateAP differences between our model 1 and median performance, respectively.

Table 3: Performance of formal runs for million query track queries.

Initial Retrieval	StatAP		MTC AP Estimate	
Set A	.2413		.0443	
Ours	Model 1	Model 2	Model 1	Model 2
B	.2077	.2170	.0440	.0441
C	.2348	.1754	.0502	.0443
D	.2453		.0510	
E	.2414		.0536	
TREC	Best	Median	Best	Median
Average	.8109	.1946	.0868	.0564

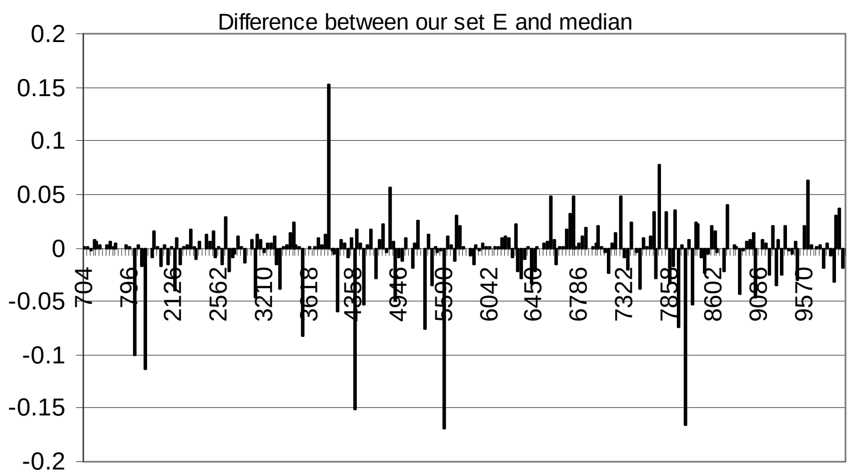

Figure 2: MTC AP estimate difference between our model 1 and the median performance of million query track queries.

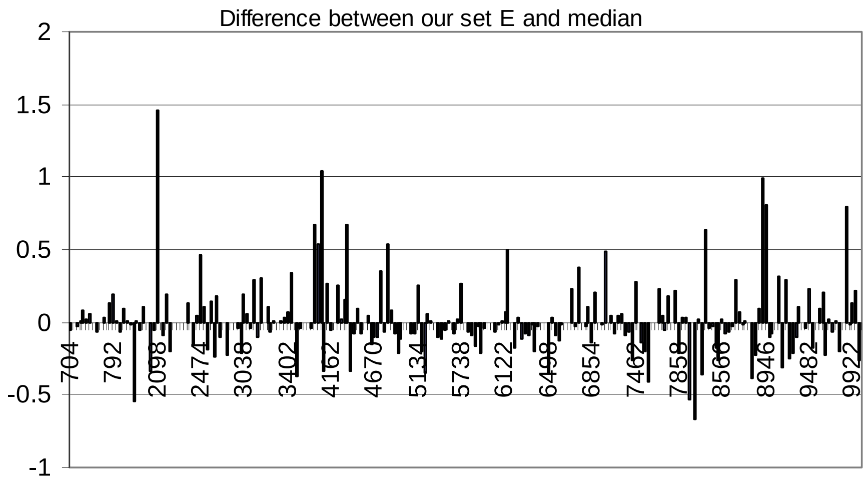

Figure 3: StatAP difference between our model 1 and the median performance of million query track queries.

5. Informal Runs

For the informal runs, we tested whether better performance may be obtained when the initial retrieval containing more relevant documents is used for re-ranking. In this case, we used the MRF [6] model provided by the Lemur package [7] to generate the initial retrieval list. Stop word removal is not used. The porter stemmer [8] is used. The setting of the μ parameter for individual terms is 1500 and 4000 for the windows of the MRF model. The size of the index is 210G bytes. This index is created by a dedicated machine for 19 hours. The retrieval time is 2-3 hours for all title queries of this RF track. No PRF is used because the results using PRF are worst than those without PRF. Table 4 shows that the retrieval performance improves (c.f. Table 1) when the performance of the initial retrieval list used for re-ranking is improved apart from feedback set B which contains only one relevant document for each query.

Table 4: Informal runs using the initial retrieval list generated by Lemur and re-ranked by our model 1 for feedback sets B-E.

Set	P@10	P@30	MAP	R-precision
A	.3977	.2750	.3339	.3389
B	.3654	.2317	.3148	.3042
C	.4859	.2728	.4380	.4234
D	.5837	.3048	.4970	.4834
E	.7175	.4677	.7309	.6962

5. Conclusion

We developed two novel models for RF. The performance of this model is similar to the median performance of all the runs by all participants. If better performing initial retrieval is used, then we expect that the retrieval using the judged documents in the feedback set performs better than the original retrieval using the same set of feedback documents.

Acknowledgement

This work is supported by HKPU Project # G-YG29. Robert thanks the Center for Intelligent Information Retrieval, University of Massachusetts (UMASS) for facilitating him to develop in part the basic IR system, when he was on leave at UMASS.

References

1. Wu, C.H., R.W.P. Luk, K.F. Wong, K.L. Kwok and W.J. Li (2005) A retrospective study of probabilistic context-based retrieval. In *Proceedings of the ACM SIGIR '05*, pp. 663-664.
2. Drucker, H., Shahrary, B. and Gibbon, D.C. (2002) Support vector machines: relevance feedback and information retrieval. *Information Processing & Management* 38(3): 305-323.
3. Wu, H.C., R.W.P. Luk, K.F. Wong and K.L. Kwok (2008) Interpreting TF-IDF term weights as making relevance decisions. *ACM Transactions on Information Systems* 26(3).
4. Robertson, S.E. and Walker, S. (1994) Some simple approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the ACM SIGIR '94*, pp. 232-241.
5. Dombi, J. (1982) A general class of fuzzy operators, the DeMorgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. *Fuzzy Sets and Systems* 8: 149-163.
6. Metzler, D. and Croft, W.B. (2005) A Markov random field model for term dependencies. In *Proceedings of the ACM SIGIR '05*, pp. 472-479.
7. Ogilvie, P. and Callan, J. (2002.) Experiments using the Lemur toolkit. In *Proceedings of the 2001 Text REtrieval Conference (TREC 2001)*, pp. 103-108.
8. Porter, M. (1980) An algorithm for suffix stripping. *Program*, 14, 3, 130-137.