RICOH at TREC-10 : Web Track Ad-hoc Task

Hideo Itoh, Hiroko Mano and Yasushi Ogawa
Software Research Center, RICOH Co., Ltd.
1-1-17 Koishikawa, Bunkyo-ku, Tokyo 112-0002, JAPAN
{hideo,mano,yogawa}@src.ricoh.co.jp

1 Introduction

This year we participated in the Web track and submitted four title-only runs {`ricMM, ricAP, ricMS, ricST`} which were automatically produced for ad-hoc task. This is our third participation in TREC. Last year in the TREC-9 main web track, our system achieved the best performance in automatic results. However the following problems could be pointed out at the same time.

- Our system uses many parameters in term-weighting and document-scoring. The value of each parameter should be tuned to improve retrieval effectiveness using test collections. However we cannot use enough relevance information in most of practical situations.
- Automatic query expansion using pseudo-relevance feedback occasionally produces a negative effect. For example in TREC-9, the performance for title-only query was hurt by query expansion using pseudo-relevance feedback.

In TREC-10, we tackled the above problems in addition to taking account of practical retrieval efficiency.

2 System description

Before describing our approach, we give the system description as background. For the TREC-10 experiments, we revised query processing although the framework is the same as that of TREC-9 [4]. The basic features of the system are as follows :

- Effective document ranking based on Okapi's approach [10] with some modifications.
- Scalable and efficient indexing and search based on the inverted file system which can be used for both Japanese and English text [6]
- Originally developed English tokenizer and stemmer for word indexing and query processing.

In what follows, we describe the full automatic process of document ranking retrieval for the TREC-10 web track ad-hoc task.

2.1 Query term extraction

We used only the "title" field of each topic. Input topic string is transformed into a sequence of stemmed tokens using the English tokenizer and stemmer. Query terms are extracted by matching some patterns against the sequence. We can flexibly control term extraction using the patterns which are described in regular expression on each token's string or tag assigned by the tokenizer. The tag indicates the character type (alphabetical, numeral, and so on) of the token string. Stop words are eliminated using the Fox's [3] word list.

For initial retrieval, both "single term" and "phrasal term" are used. A phrasal term consists of two adjacent terms and represented by a proximity operator to match only adjacent occurrences in target documents.

In order to moderate the effect of the phrasal term, a "combination value" is assigned to each phrasal term and multiplied by the weight in the process of term weighting. The value is estimated by ψ/b, where ψ is a tuning parameter and b is the number of extracted phrasal terms.

2.2 Initial retrieval

Each query term is submitted one by one to the ranking search system, which assigns a weight to the term and scores documents including it. Retrieved documents are merged and sorted on the score in descent order. The specification of term weighting and document scoring is explained in the section 3.1.

2.3 Seed document selection

Top ranked documents are assumed to be pseudo-relevant to the topic and selected as a "seed" of query expansion. The maximum number of seed documents is ten.

To avoid duplication in the seed documents, the document which has the same length and score as that of the document previously selected is skipped [4].

We also skipped any document the length of which exceeds the average in the collection, since the lengthy document may include many heterogeneous topics and may not be effective as a seed.

2.4 Query expansion

Candidates of expansion terms are extracted from the seed documents by pattern matching as in query term extraction mentioned above. Each candidate is pooled with its seed document frequency and term frequency in each seed document.

We do not use phrasal terms for query expansion because phrasal terms may be less effective to improve recall and risky in case of pseudo-relevance feedback.

The weight of initial query term is re-calculated with the Robertson/Spark-Jones formula [7] if the term is found in the candidate pool.

Before selecting expansion terms among the candidates, we assign to each candidate the Robertson's Selection Value [8] and Robertson/Spark-Jones relevance weight. In this step, candidates the document frequency of which is less than one thousand are eliminated because these term may be less effective for query expansion.

We did not mix the weight based on the seed documents with the prior weight based on the document collection as in [10][4]. Instead, the rank of a document is mixed between two document rankings in the data fusion manner (see the section 4).

The candidates are ranked on the RSV and top-ranked terms are selected as expansion terms. In consideration of practical retrieval time, we used only ten expansion terms.

As in the case of phrasal query terms, a "combination value" is assigned to each expansion term. The value is estimated by ξ/e, where ξ is a tuning parameter and e is the number of expansion terms.

2.5 Final retrieval

Each query and expansion term is submitted one by one to the ranking search system as in initial retrieval. Since the submitted term is already weighted in the previous phase, the system simply multiplies the weight by the document score to give a final score to the document including the term.

3 Automatic parameter estimation

Many parameters have been used in our retrieval process and the system performance heavily depends on whether the set of the values fits with the target query and document collection. Therefore parameter tuning is inevitable to maintain retrieval effectiveness. However in most practical settings such as in commercial use, we cannot use enough information for the parameter tuning. In the TREC-10 experiments, we tried to automatically adapt some parameter values to any given query and document collection.

3.1 Term weighting

The ranking system uses the following term-weighting formula (1).

$$w_t = \log\left(k_4' \cdot \frac{N}{n_t} + 1\right), \tag{1}$$

where w_t is the weight of the term t, N is the number of documents in the collection, n_t is the number of the documents in which t occurs and k_4' is the parameter of the formula.

The formula (1) is based on the Robertson/Spark-Jones formula (2).

$$w_t = \log \frac{p_t\ (1 - q_t)}{q_t\ (1 - p_t)} \tag{2}$$

where $p_t = \mathrm{P}(t \text{ occurs} \mid \text{relevant document})$ and $q_t = \mathrm{P}(t \text{ occurs} \mid \text{non-relevant document})$. While q_t is estimated by n_t/N as usual, p_t is estimated in our formula (1) as follows:

$$p_t = p0 + (1 - p0)q_t \tag{3}$$

where $p0$ is an unknown constant called a "base probability" in this report [1]. Using the formulas (1), (2) and (3), we get the relation between k_4' and $p0$ as :

$$k_4' = \frac{p0}{1 - p0} \tag{4}$$

Instead of conventional parameter k_4', we adapt the base probability $p0$ to a given query using following the heuristics :

> The average of p_t may monotonically decrease as the number of query terms u increases. Especially, if u equals 1 then the p_t nearly equals to 1.

To implement this heuristics, we give the following estimation :

$$p0_i = \frac{\rho}{u_i} \tag{5}$$

where $p0_i$ is a base probability for topic i, u_i is the number of single terms in topic i and ρ is a constant the value of which nearly equals to 1 (the actual value is 0.9). As a result, we estimate the weight for term t in topic i with the following formula (6)

$$w_{t,i} = \log(\frac{\rho}{\rho - u_i} \cdot \frac{N}{n_t} + 1) \tag{6}$$

Fig.1 shows the relation between the average of p_t and the number of query terms in test collections of the TREC-7, TREC-8 and TREC-9 ad-hoc retrieval task. For each "desc" field which includes n terms, we got the average of p_t of the term set using relevance data. We think the data has supported the heuristics mentioned above because in most desc fields (about 87%), the number of query terms is less than eight.

3.2 Document scoring

We also tried to adapt parameters used for document scoring. Because the effect was negative in the result of the official run (`ricST`), we describe the method briefly.

[1] If $0 \leq p0 \leq 1$ then the weight never gets negative. It enables us to treat term-weights simply and consistently in any case [4].

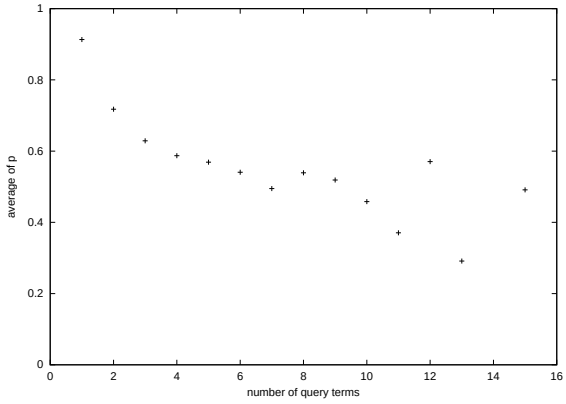

Figure 1: Average of p and number of query terms

In the process of conventional document ranking, each document d is given the score $s_{d,t}$ for the term t with the following formula (7) :

$$s_{d,t} = \frac{f_{d,t}}{f_{d,t} + \kappa \ ((1-\lambda) + \lambda \frac{l_d}{L})} \cdot w_t \tag{7}$$

where $f_{d,t}$ is within-document frequency of t in d, l_d is the length of d, L is the average length of documents. κ and λ are tuning parameters for document scoring and the actual values used in the official runs (`ricMM, ricMS, ricAP`) are 0.5 and 0.2 respectively.

For a query term t, the value of λ is estimated using the median m_t and the variance v_t of the lengths of retrieved documents as follows :

$$\lambda_t = \frac{m_t}{m_t + v_t} \tag{8}$$

In addition to the above estimation, we used m_t instead of L in the formula (7).

The value of κ is fixed (the actual value is 1) and we applied the following non-linear score transformation :

$$newscore(d,t) = \frac{w_t}{max^\mu - min^\mu} \left(score(d,t)^\mu - min^\mu\right) + \epsilon \tag{9}$$

where ϵ is a constant with a very small value. The value of μ is given by

$$\mu = \frac{min + ave}{2} \tag{10}$$

where max, ave and min are those of document scores given by the formula (7). Using the transformation and fixing the value of κ, we tried to normalize the document score distribution.

4 Rank merging

Automatic query expansion using pseudo-relevance feedback occasionally produces a negative effect [12][2]. In Fig.2 we draw a comparison of the average precision between initial retrieval (unexpanded) and final retrieval (expanded) on the TREC-9 title-only ad-hoc retrieval data. Each dot represents one of the 50 topics. If the dot is above (below) the bisecting line, then the performance is improved (hurt) by query expansion.

In case of a short query, the information need is often under-specified and the effect of pseudo-relevance feedback becomes very unstable.

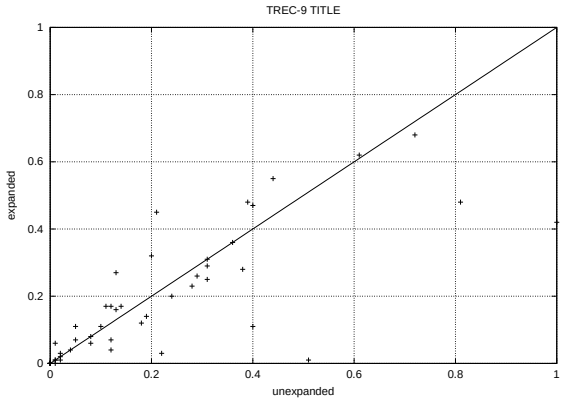

Figure 2: Inconsistency of query expansion

To remedy the inconsistent effect of pseudo-relevance feedback, we construct a final ranking by merging first ranking with query terms and second ranking with both query and expansion terms. We call the method "rank merging" in this report. This is a kind of the data fusion problem [1][11].

Specifically we give scores in a final ranking by the following formula :

$$score(d) = \frac{1}{(1-\beta) \cdot first\_rank(d) + \beta \cdot second\_rank(d)} \tag{11}$$

where β is a tuning parameter with the actual value 0.6. $first\_rank(d)$ ($second\_rank$) is the rank of the document d in first (second) ranking. The target of this merging is restricted to only documents the rank of which in first ranking is smaller than a threshold $maxRank$. The actual value of the threshold is 20.

5 Results

Results of our submitted runs are summarized in Table 1. These runs were automatically produced using only title field. No link information in a HTML document was used. Query

expansion by pseudo-relevance feedback was applied to all runs.
Three results {`ricMS, ricMM, ricAP`} are in the the third-ranked group among all of automatic title-only official runs. In our analysis of the results, however, we got the following findings.

- Skipping duplications in seed documents (section 2.3) produced a slightly positive effect.
- Skipping lengthy seed documents (section 2.3) clearly produced a negative effect.
- Eliminating an expansion term the document frequency of which is small (section 2.4) slightly hurt the performance.
- The loss expected from dropping the collection-wide weight (section 2.4) could not be compensated for by the rank merging.

We found that if the problematic procedures mentioned above had not been taken, the average precision in the almost same setting as `ricMS` would have been 0.2247.

Rank merging slightly improved the average precision in comparison with the baseline. Since the threshold $maxRank$ is 20, the rank merging influences only the top part of the ranking. However it improved P@10 which is crucial to ad-hoc retrieval users.

The automatic estimation of parameter ρ exerted a little but positive influence on retrieval performance. We think this is due to the narrow range of the number of query terms in a title topic. RET100 of this run is the best one among all official runs. This partially comes from the effectiveness of our stemmer and the scalability of the system which enable us to index all documents in WT10G.

The score transformation clearly hurt the retrieval performance. Since parameters κ and $lambda$ influence on retrieval performance more than the other parameters, automatic tuning of these parameters is very attractive. After the submission, we continued to develop a more effective and theoretically grounded method for the automatic tuning, taking account of Robertson's approximations to the two-Poisson model [9]. However, it has still been under way.

RUN	AveP	P@10	RET100	Experiment (what was different from baseline)
ricMS	0.2068	0.3360	16.84	baseline
ricMM	0.2084	0.3420	16.84	rank merging
ricAP	0.2077	0.3380	17.62	automatic estimation of parameter ρ
ricST	0.1933	0.3260	16.20	non-linear score transformation

Table 1: Results of Web track ad-hoc task official runs

References

[1] N. Belkin et al., "Combining the evidence of multiple query representations for information retrieval", Information Processing and Maragement, 31, 3, pp.431–448, 1995.

[2] C. Carpineto et. al., "An information-theoretic approach to automatic query Expansion", ACM Transactions on Information Systems, 19, 1, pp.1–27, 2001

[3] C. Fox, "A stop list for general text", *ACM SIGIR Forum*, 24, 2, pp. 19–35, 1991.

[4] Y. Ogawa, H. Mano, M. Narita, and S. Honma, "Structuring and expanding queries in the probabilistic model", In *The Eighth Text REtrieval Conference (TREC-9)*, pp.427–435, 2001.

[5] Y. Ogawa, H. Mano, M. Narita, and S. Honma, "Structuring and expanding queries in the probabilistic model", In *The Eighth Text REtrieval Conference (TREC-8)*, pp.541–548, 2000.

[6] Y. Ogawa, T. Matsuda, "An efficient document retrieval method using n-gram indexing" (in Japanese), *Transactions of IEICE*, J82-D-I, 1, pp. 121–129, 1999.

[7] S. E. Robertson, K. Spark-Jones, "Relevance weighting of search terms", Journal of ASIS, 27, pp.129-146, 1976

[8] S. E. Robertson, "On term selection for query Expansion", Journal of Documentation, 46, 4, pp.359–364, 1990

[9] S. E. Robertson, S. Walker, "Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval", In *Proc. of 17 th ACM SIGIR Conf.*, pp.232–241, 1994.

[10] S. E. Robertson, S. Walker, "On relevance weights with little relevance information", In *Proc. of 20th ACM SIGIR Conf.*, pp.16–24, 1997.

[11] H. Schutze, J. O. Pedersen, "A cooccurrence-based thesaurus and two applications to information retrieval", Information Processing and Management, 33, 3, pp. 307-318, 1997

[12] J. Xu, W. B. Croft, "Improving the effectiveness of information retrieval with local context analysis", ACM Transactions on Information Systems, 18, 1, pp.79–112, 2000